J Vis (2023) 26:611–629
https://doi.org/10.1007/s12650-022-00896-x

REGULAR PAPER

Zihao Zhu · Yue Shen · Sujia Zhu · Gefei Zhang · Ronghua Liang · Guodao Sun

Towards better pattern enhancement in temporal evolving set visualization

Received: 29 July 2022 / Revised: 20 September 2022 / Accepted: 11 October 2022 / Published online: 27 November 2022

Abstract Temporal evolving set data are time-varying and growing ubiquitous in person re-identification, parameter choice, and streaming data analysis. We construct a workflow to analyze and explore the inconspicuous pattern between multiple nonadjacent sets in temporal evolving set data. We propose a progressive timeline layout algorithm based on a mathematical optimization model to place the set element after data update, our layout algorithm can calculate the coordinates of elements in a short time and preserve the distance ratio between elements. To relax the visual clutter when visualizing sets' relationships, we design two types of pattern enhancement strategies and their combinations: optimization-based pattern enhancement strategy and design-based pattern enhancement strategy. We conduct a comprehensive evaluation to verify and compare our pattern enhancement strategies including a quantitative experiment, two case studies, and an informal user study. The results show that our pattern enhancement strategies can effectively help users identify inconspicuous patterns. Our workflow and strategies show broad application prospects and we hope it could be a fundamental component in data projection pattern mining and streaming data analysis.

Keywords Temporal evolving set data · Pattern enhancement strategy · Progressive timeline layout algorithm

1 Introduction

Data projection is a widely used technique in data visualization for reducing data dimension, refining data feature (Chang et al. (2022)), and summarizing data information (Sun et al. (2013); Nonato and Aupetit (2018)). The most commonly used visualization form to represent data distribution after dimension

Z. Zhu · Y. Shen · S. Zhu · G. Zhang · R. Liang · G. Sun (✉)
College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China
E-mail: guodao@zjut.edu.cn

Z. Zhu
E-mail: iszhuzh@outlook.com

Y. Shen
E-mail: shenyueee98@gmail.com

S. Zhu
E-mail: zhusujiashiny@gmail.com

G. Zhang
E-mail: gopherzhang@163.com

R. Liang
E-mail: rhliang@zjut.edu.cn

reduction is scatter plot (Espadoto et al. (2019)). Since large-scale data bring visual overlapping to scatter plots, many papers aim to highlight the pattern or structure of the scatter plot to explore data distribution. However, in temporal data, the pattern of projection scatter plot always changes over time. In order to track this dynamic pattern, many papers aim to track the evolution of structure, trends, and outliers of scatter plots (Xia et al. (2018)). Outliers exploration, as an important part of projection mining, could help users better understand and enhance temporal evolving set including increasing or decreasing the number of items.

Most researchers implement studies and track the position change of points in fixed sets. However, most outliers do not appear or disappear at the same time. In other words, the elements in the outlier set tend to be changed with time varying, which conforms with the real data characteristics in actual scenarios, such as streaming data analysis (Keim et al. (2006)) and person re-identification (Yaghoubi et al. (2021)). For example, person re-identification model would provide a person ranking list that conforms to the user's query. Limited by scalability, users only focus on Top K data points. Furthermore, due to the change of data and user's query, the set of points would change drastically over time. Another example is streaming data analysis (Krstajić and Keim (2013); Dasgupta et al. (2018)), due to the dynamic characteristics of streaming data, the data element of dataset is usually time-varying or unfixed. It is difficult to analyze the inconspicuous pattern between data in two nonadjacent times. It is necessary to research visualization techniques of this temporal evolving set to help users realize its evolution and pattern.

Noted that a temporal evolving set has two basic characteristics. The first is time. Since the temporal evolving set changes over time, the set is evolving means that the elements in set increase or decrease frequently. It is necessary to display the information of the time dimension. We utilize the layout based on the timeline to display the set evolution from the horizontal direction. The second is the relationship between sets such as intersection set. The intersection set contains points in both two sets which could be important for users. For example, researchers in *evolutionary set theory* focus on the evolution of dynamic sets and their relationships (Tarnita et al. (2009)). One of the similar data structures is dynamic graph (Sizemore and Bassett (2018); Beck et al. (2017); Burch et al. (2021)). However, the temporal evolving set is different from the dynamic graph, although dynamic set can be seen as a special type of dynamic graph, the relationships in dynamic graph are those links between nodes at the same time. The dynamic set does not have links between elements in set at the same time but the intersection of sets can be seen as the relationships over time in temporal evolving set.

The typical example of the temporal evolving set is time series data or streaming data because these data contain time information and could map them as a temporal evolving set. As for the non-temporal data, temporal evolving set data could be extracted along with the interactive analysis process. Suppose that we have such a scene about data projection. In multi-dimensional data analysis, we usually need to try different methods and parameters. Then we may pay more attention to outliers, clusters, or points of interests. After several attempts, the change of outliers generates a temporal evolving set. Noted that the original data does not have temporal information, but the analysis process brings temporal information.

We utilize timeline-based visualization to represent the temporal evolving set because timeline-based visualization could display the time-varying information of the set in a horizontal direction efficiently. We propose a progressive layout optimization algorithm based on a mathematical programming model to layout the elements in set after new data are generated. Meanwhile, we utilize curves to link the elements in the intersection set of adjacent time. We do not link the elements by curves in the intersection set of two timestamps which are not adjacent. First, if we link two elements in the intersection set of any two timestamps, it requires $O(N^2)$ number of linked set pairs, where N is the number of timestamps. Second, it will bring visual confusion and burden for users to explore relationships between sets. In order to make up for this deficiency, we adopt two types of pattern enhancement strategies to highlight the links between the selected two sets.

Pattern enhancement is a visualization technique to enhance or highlight the hiding and implicit pattern. In this paper, we propose several strategies to enhance patterns between two selected sets including the optimization-based and design-based pattern enhancement strategies. The former enhances implicit pattern by relayouting the elements and the design-based strategy enhances implicit pattern by visual design based on perception. We also discuss the combined strategies which utilize both two or more pattern enhancement strategies. In order to verify our pattern enhancement strategies, we conduct an experiment to examine and compare these strategies.

To summarize, the major contributions of our paper contain the followings:

- We design and implement a progressive timeline layout algorithm based on a mathematical optimization model to position the element in set.

- We propose optimization-based and design-based pattern enhancement strategies to highlight the relationship between two selected sets which could be a fundamental component in timeline-based visualization.
- We construct a workflow for temporal evolving set data visualization and conduct a comprehensive evaluation to verify the core pattern enhancement part.

In the remaining part of this paper, we first describe the related work of our research in four parts in Sect. 2. Second, we construct our workflow and analyze the tasks in Sect. 3. We then describe and implement our progressive timeline layout algorithm and summarize our two pattern enhancement strategies in Sect. 4. To verify our work, we conduct a quantitative experiment, two case studies, and an informal user study in Sect. 5. In Sect. 6, we discuss our results, limitation, and future work. Finally, we summarize a conclusion about our work in Sect. 7.

2 Related work

In this section, we discuss the related work from the following four aspects, including data projection, dynamic set visualization, pattern enhancement, and timeline-based layout.

2.1 Data projection

Data projection is a process of data transformation from one space to another space. It could also be called dimension reduction when project data from a high dimension to a low dimension. Data projection help extract and refine the information from original data. There are many projection methods which are widely used in data visualization such as MDS (Mead (1992)), PCA (Abdi and Williams (2010)), t-SNE (Van der Maaten and Hinton (2008)), UMAP (McInnes et al. (2018)), and their variants. Espadoto et al. (2019) survey and examine these dimension techniques from a quantitative aspect.

One of the most commonly used visualization after data projection is scatter plot, due to the massive volume of data, scatter plot will cause confusion and ambiguity. However, many researchers aim to visualize the global structure or trend of the scatter plot such as sampling the dense points in the scatter plot (Mayorga and Gleicher (2013)), comparing differences between clusters (Eckelt et al. (2022)), or comparing the different dimension reduction results by subspace (Sun et al. (2021)). For example, Mayorga and Gleicher (2013) sample the dense points and group them into contours to visualize the closed shapes. Sun et al. (2021) track and compare the sensitivity and difference of dimension reduction results in different subspaces.

The other concern in projected scatter plot is outliers. Outliers usually represent those data that do not obey the overall distribution of data. Visualization of outliers in scatter plot will help users find and realize the meaning of outliers. For example, Sohns et al. (2021) utilize an augmentation strategy of projected data and set visualization techniques to help users find outliers in scatter plots. Xia et al. (2018) propose three metrics to measure the importance of points for outliers, clusterings, and trends.

2.2 Dynamic set visualization

Dynamic set visualization aims to visualize and present the dynamic evolution of set data and links between elements in set temporally. On the one hand, dynamic set can be seen as a simplified dynamic graph without edges between elements in set. Beck et al. (2017) bring an overall taxonomy and summarization of dynamic graphs. Dynamic set visualization can be applied to dynamic graph visualization. On the other hand, graphs and other structures such as hypergraphs (Fischer et al. (2021)), also can be seen as the combination and integration of multiple sets including the set of nodes and edges.

Set visualization utilizes visualization techniques to represent the structure and relationship such as difference between set (Alsallakh et al. (2016); Sadana et al. (2014)). Alsallakh et al. (2016) summarize an overall survey of set visualization. In order to visualize the intersection set, UpSet (Lex et al. (2014)) utilizes matrix layout and aggregation techniques to visualize intersection set. But they do not extend their techniques to dynamic set visualization.

In dynamic set visualization, many researchers aim at tracking the evolution by visualizing set relationships such as Bubble Sets (Collins et al. (2009), TimeSets (Nguyen et al. (2016); Xu et al. (2020), timeline-based visualization (von Landesberger et al. (2012), or optimizing animation between two timestamps (Mizuno et al. (2019)). For example, Bubble Sets use bubbles to represent the overlapping relationship of sets and show the evolution of sets over time in the horizontal direction. TimeSets Nguyen et al. (2016) employ groups to represent the same event between sets dynamically. von Landesberger et al. (2012) visualize dynamic categorical data by timeline-based technique but lack of visualization of set relationships. On the other hand, many researchers explore the relationship between sets and their elements such as TimeSets (Xu et al. (2020)) and Set Streams (Agarwal and Beck (2020)). For example, Set Streams represent the branching and merging process of set as streams to help compare element evolution in set.

2.3 Pattern enhancement

Pattern enhancement is a visualization technique that aims to magnify and highlight the implicit or inconspicuous pattern in visualization. Although it may cause local distortion and deformation, it is necessary to enhance the local or global hiding structure for users to realize and explore the visualization.

Pattern enhancement has many applications such as magnifying the area of landmine (Jayatilaka et al. (2010)), reflecting cooperation and competition of social media (Sun et al. (2014)), embedding temporal information in map (Sun et al. (2017)), or enhancing the local and global visual pattern (Cao et al. (2010)). For example, Sun et al. (2014) add constraints in a layout model to enlarge the wiggle of the flow map. Ankerst et al. (1998) formulate a linear assignment problem and reorder the dimension of multi-dimension data to enhance and magnify the similarity pattern.

In order to enhance and magnify the hiding pattern in data, many techniques are employed such as sorting by user-defined mission (Tatu et al. (2009)), integrating scatter plots into parallel coordinate (Yuan et al. (2009)), visual deformation (Wang et al. (2019)), and machine learning methods (Yuan et al. (2021)). For example, many researchers utilize focus+context technique and visual deformation to visualize the structure more clearly, which can be seen as a type of pattern enhancement. Wang et al. (2019) employ the fisheye technique in a large node-link layout to visualize the structure more clearly for pattern exploration. Haunert and Sering Haunert and Sering (2011) draw the road map under local radial deformation which could help users find routes in dense road map quickly. Similarly, Wang and Chi (2011) utilize focus+-context technique to enlarge the focus region of the metro map. Other works (such as Fink et al. (2012)) enhance visual region by removing label overlapping.

2.4 Timeline-based layout

Timeline is a visualization form to represent the dynamic data in a vertical or horizontal direction. Many researchers utilize timeline-based layouts to visualize the progression of events (Guo et al. (2019, 2018)) and storyline layouts (Tanahashi and Ma (2012)).

Storyline layout, as a typical timeline-based layout, attracts much attention for many researchers. Most of their works optimize the storyline layout from an esthetic aspect. The basic metrics in storyline layout are line crossings, line wiggles, distance, and space. Tanahashi and Ma (2012) layout storyline visualization by minimizing line crossings and swings. Furthermore, they improve their methods and propose an efficient framework to layout the storyline (Tanahashi et al. (2015)). Liu et al. (2013) propose an optimization approach to generate an esthetically appealing storyline visualization.

One of the other timeline-based visualization forms is based on flow map. Flow map is similar to storyline, it is often used to visualize temporal data and has many applications in social media data (Sun et al. (2014)) and dynamic graphs visualization (Cui et al. (2014)). In the esthetic aspect, line crossings and wiggles are also important metrics (Di Bartolomeo and Hu (2016); Bu et al. (2021)). Furthermore, Sine-Stream (Bu et al. (2021)) improves the esthetic and readability of flow graph layout by minimizing sine illusion.

Besides, other researchers modify timeline-based layouts and propose their variants such as SchemaLine (Nguyen et al. (2014)) and timeline trees (Burch et al. (2008)). For example, timeline trees propose a hierarchical timeline tree to visualize the progression of transactions in hierarchical data. Timeline-based layouts usually employ esthetic metrics as their optimization objective and utilize mathematical optimization to result in a fine layout.

3 Task analysis

In this section, we introduce the basic workflow of our algorithm and refine the task analysis.

3.1 Workflow

As we mentioned in Sect. 1, the temporal data could fit our workflow and the non-temporal data could be transformed to temporal data by interactive analysis process. As for the temporal data such as traffic trajectory data or streaming data, we define different objects in data as elements in set. For example, different cars in a zone each day could be defined as a temporal evolving set. As for the non-temporal data such as multi-dimensional data, we could try different projection methods to analyze data and collect points of our interests such as outliers. Then we regard this temporal point set as a temporal evolving set.

After the data preparation, we utilize timeline-based techniques to visualize the temporal evolving set because the timeline-based visualization could adapt to the change and update of data. Therefore, we employ a progressive layout algorithm to determine the node positions in timeline visualization. The algorithm will be introduced in Sect. 4.

However, the timeline visualization changes constantly while the data are updating. To compare the set in a previous time stamp and a current time stamp, the pattern enhancement strategies could bring a significant visual effect and enhance the implicit pattern in a large number of nodes and lines in timeline visualization. We summarize our workflow in Fig. 1.

3.2 Task

Alsallakh et al. (2016) summarize the basic tasks in set data visualization. Based on our workflow and the temporal characteristics of temporal evolving set, we extend and summarize our tasks as followings:

T1. **Analyze the process of generating and changing new elements in the temporal evolving set.** Due to the non-fixed characteristics of the dataset, we hope to display the new data points in the visualization. In this way, users can analyze the process of generating new elements in sets while the temporal evolving set is changing.

T2. **Compare the differences between elements in the original and new set.** Each element belongs to a temporal evolving set. In two adjacent time stamps, visualizing the intersection between two adjacent sets can help users to observe and compare the difference and changing process between them.

T3. **Help comparison of two different sets with pattern enhancement strategies.** When users select and compare two nonadjacent sets, they could be disturbed by other set elements. To help users find and compare the different two sets quickly. Our visualization should enhance and highlight the implicit pattern in the timeline layout to help explore the difference between them.

4 Algorithm

We describe the algorithm of timeline layout in detail from two aspects, including progressive timeline layout and pattern enhancement with user interaction.

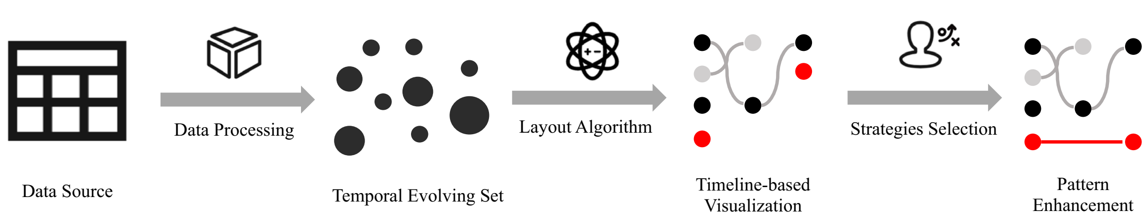


Fig. 1 Our workflow about pattern enhancement in temporal evolving set data visualization. First, process data from data source to extract temporal evolving set data. Second, utilize layout algorithm to generate timeline-based visualization. Third, select pattern enhancement strategies to highlight the implicit pattern with users' feedback for set comparison and exploration

4.1 Data description

We denote the temporal evolving set as S^t, where the t represents the time stamp. Noted that S^t is a dynamic set that changes with time and user operations, we need to distinguish the data subsets that change in S^t from the previous time $t-1$ and the data subsets that are fixed. We denote $S^t = S_o^t \cup S_n^t$, where S_o^t represents the set of original points in the point set, and S_n^t represents the set of new points, it is obvious that $S_o^t \subset S^{t-1}$.

In order to represent the coordinate of points in S^t in layout algorithm, we denote the coordinate of points in each S^t as y_i^t, where y_i^t is a continuous variable. We denote the order of points in each S^t as x_i^t, where x_i^t is a discrete variable which is larger than 0.

4.2 Progressive timeline layout

The main idea of progressive timeline layout is that we do not change the order and coordinate of the points in the previous time $S^0, S^1, S^2, \ldots, S^{t-1}$, when layout the positions of the points in S^t.

Our layout algorithm includes two steps: The first step is to determine the order of points in S^t, and the second step is to scale the distance and determine the continuous coordinate of each points in S^t.

Determine the order. In the discrete layout, our goal is to keep the order of fixed points in S_o^t and insert the new points in S_n^t. While inserting new points, we aim to make the original distance ratio of the new points as harmonious as possible. We formulate the layout requirements as the following model:

$$
\begin{aligned}
\min \quad & \sum_{i,j,k} \mathbb{1}_{\{(|x_i^t - x_j^t| - |x_i^t - x_k^t|)(d_{ij} - d_{ik}) < 0\}} \\
\text{s.t.} \quad & \begin{cases} x_i^t < x_j^t \quad \mathit{where} \quad y_i^{t-1} < y_j^{t-1} \quad \forall i,j \in S_o^t \\ \qquad\qquad\qquad x_i^t \in \{1, 2, \ldots, |S^t|\} \end{cases}
\end{aligned} \tag{1}
$$

In model 1, the objective function contains an indicator function $\mathbb{1}_{\{\cdot\}}$ which means the function takes 1 if and only if the condition of the indicator function is true. In the model, x_i^t takes a value between 1 and $|S^t|$, and d_{ij} is the distance between point i and point j on the projected image. Noted that when $d_{ij} - d_{ik}$ and $|x_i - x_j| - |x_i - x_k|$ have different signs. For example, if $d_{ij} - d_{ik} > 0$ and $|x_i - x_j| - |x_i - x_k| < 0$, the coordinate distance of i, j is greater than i, k and the feature distance of i, j is less than i, k, which leads to an imbalance between the coordinate distance between points and their feature distances, which is contrary to our goal. If the feature distance of i and j is larger than i and k, the model could make the difference of point order satisfy the feature distance order as much as possible.

The constraint in model 1 aims to keep the order of original points in S_o^t, it only works for points in the set S_o^t which ensures they have coordinate in time $t-1$.

The model 1 is a nonlinear programming model which has a nonlinear objective function and linear constraints. Considering that the objective function in model 1 is cumbersome, we utilize linearization technique and other decision variables to simplify the model, and solve it by Gurobi solver (https://www.gurobi.com). Linearization technique can help us deal with indicator function when we program by Gurobi interface. For example, in order to deal with the absolute value in the objective function, we define two 0-1 decision variables p_{ij} and q_{ijk}. p_{ij} is 1 if point i locates higher than j, otherwise, p_{ij} is 0. q_{ijk} is 1 if the condition in the indicator function is true, otherwise, q_{ijk} is 0. By utilizing p_{ij} and q_{ijk}, we could rewrite the objective function in Eq. 1 as followings:

$$
\min \quad \sum_{i,j,k} q_{ijk} \tag{2}
$$

where the Eq. 6 is a linear function which avoid the cumbersome form in Eq. 1. However, we need to add new constraints to describe the relationship between x_i^t and p_{ij}. The new constraints can be expressed as followings:

$$p_{ij} + p_{ji} = 1$$
$$x_i + 1 - x_j \leq M(1 - p_{ji}) \tag{3}$$

where Eq. 3 combines x_i^t and p_{ij}. Similarly, the constraint which combines x_i^t and q_{ijk} can be written as followings:

$$p_{ij}(x_j - x_i) + p_{ji}(x_i - x_j) - p_{ik}(x_k - x_i) - p_{ki}(x_i - x_k) \leq \frac{M(1 - q_{ijk})}{d_{ij} - d_{ik}} \tag{4}$$

Combining new constraints and new objective function, we obtain a new nonlinear programming model which could be accepted and solved in Groubi interface.

Scale the coordinate. After determining the order of points in S^t, we scale and compute the continuous coordinate of points. We consider these three optimization objective functions.

- *Ratio Objective* The ratio objective function is an extension of the objective function in discrete optimization. Given two points i and j, the ratio of the coordinate distance and the feature distance between them can be written as $\frac{|y_i - y_j|}{d_{ij}}$. We aim to make the ratios between the different points to be as close as possible. We use δ^t to denote the range of ratio and minimize it, the objective function can be written as followings:

$$f_1 = \max_{i,j \in S^t}\{\frac{|y_i^t - y_j^t|}{d_{ij}}\} - \min_{i,j \in S^t}\{\frac{|y_i^t - y_j^t|}{d_{ij}}\} \tag{5}$$

where $max\{\frac{|y_i^t - y_j^t|}{d_{ij}}\}$ is the maximum of the ratio for all point pairs in set and $min\{\frac{|y_i^t - y_j^t|}{d_{ij}}\}$ is the minimum of the ratio for all point pairs in set. The less f_1 could make the distance between point pair follow the distance more in projected coordinate plane.
- *Boundary Objective* The boundary objective function aims to avoid the range of point coordinate to become too large. It makes the range of point closer by these following objective function:

$$f_2 = \max_{i \in S^t}\{y_i\} - \min_{i \in S^t}\{y_i\} \tag{6}$$

where $max\{y_i\}$ is the maximum of the vertical coordinate of all points and $min\{y_i\}$ is the minimum of the vertical coordinate of all points. The less f_2 could make the distance of between point pair closer.
- *Wiggle Objective* The wiggle objective function aims to avoid the wiggle of line between original points in S_o^t. The wiggle objective is a classical esthetic metric and objective function in timeline-based layout Tanahashi and Ma (2012). We describe the wiggle objective function as followings:

$$f_3 = \sum_{i \in S_o^t} \omega_i (y_i^t - y_i^{t-1})^2 \tag{7}$$

where ω_i is the weight of original points to measure the importance of points. The less f_3 could make the distance between elements both in adjacent set closer.

Besides, we consider these two constraints including gap constraint, boundary constraint, and order constraint:

- *Gap Constraint* The boundary objective function makes the distance of points closer. In order to avoid visual clutter and interference caused by too close points overlapping, we establish gap constraints to limit the minimum distance between points, which can be written as followings:

$$|y_i^t - y_j^t| \geq D_{min} \quad \forall i,j \in S^t \tag{8}$$

where D_{min} is the minimum of the distance between two points.
- *Boundary Constraint* The boundary constraint aims to layout the points in the height of the rectangular canvas, which can be formulated as the following terms:

$$0 \leq y_i^t \leq H \quad \forall i \in S^t \tag{9}$$

- *Order Constraint* The order constraint utilizes the result from model 1. We utilize these constraints to preserve the order of original points by x_i^t, we write the order constraint into the followings:

$$y_i^t \geq y_j^t + D_{min} \quad where \quad x_i^t > x_j^t \quad \forall i,j \in S^t \tag{10}$$

where the x_i^t in the above constraints is already determined by the discrete optimization model.

We combine the objective function and constraints as a multi-objective programming model, due to the complexity of solving a multi-objective model, we use a weighted linear combination to simplify these three objective functions as these followings:

$$\min \quad \alpha_1 f_1 + \alpha_2 f_2 + \alpha_3 f_3 \tag{11}$$

where $\alpha_1, \alpha_2, \alpha_3$ are the positive weight to balance the three objective functions. After several examination, we choose 100, 0.01, 0.1 as their value. Because the ratio objective usually has a fewer range than the other two objective functions. Finally, we obtain a quadratic programming model with linear constraint. Similarly, in order to rewrite the objective function to fit the Gurobi interface. For example, we rewrite the Eq. 6 as followings:

$$f_2 = z_2 - z_1 \tag{12}$$

where z_1 and z_2 are two variables, and we use them to limit the maximum and minimum of y_i by the following constraints:

$$z_1 \leq y_i \leq z_2 \quad \forall i \in S^t \tag{13}$$

Similarly, we utilize the linear techniques to deal with the objective function of the model to fit the Gurobi programming interface, then we solve the model by Gurobi.

4.3 Pattern enhancement

Timeline-based layout optimization aims to minimize curve wiggle and height from an esthetic aspect. However, it may mask or hide some inconspicuous patterns from being discovered by users. One way is to highlight these patterns in other views, but in order to take full advantage of the pixel space of the timeline visualization, and consider the scalability of selecting multiple sets. We utilize pattern enhancement strategies.

Pattern enhancement aims to highlight the implicit patterns in the layout. For example, when users hope to compare two projection scatter plot in time 2 and 4, they find it is difficult. Because the original timeline layout is progressive layout which only links the elements in intersection set of two adjacent time intervals, which means elements in $S^2 \cap S^4$ are easily ignored. In order to enhance these implicit patterns, we need to enhance these patterns. Since we utilize a mathematical optimization model in timeline layout, we consider pattern enhancement from optimization-based and design-based aspects. The taxonomy of our pattern enhancement strategies is shown in Fig. 2. In this section, we suppose that we have already selected two nonadjacent projection scatter plot and obtained two PoI set S^k and S^t, where $t - k \geq 2$.

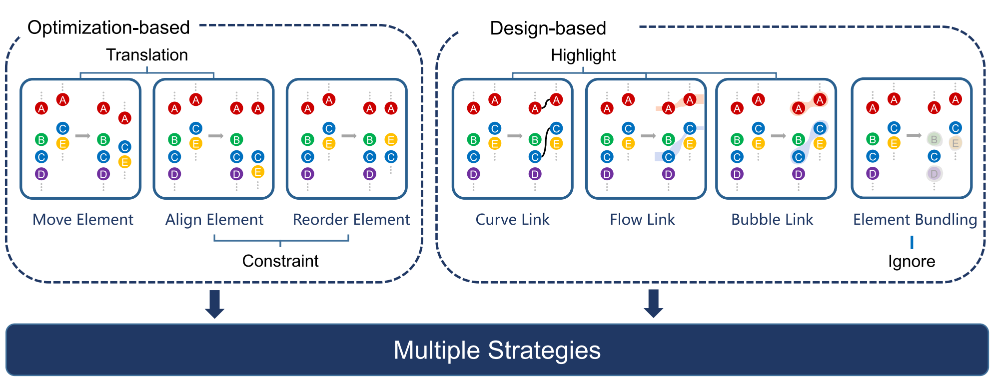
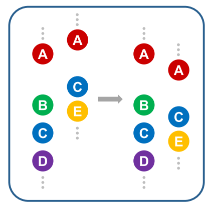


Fig. 2 Pattern enhancement strategies. In optimization-based pattern enhancement strategies, we classify these strategies as translation-based and constraint-based. In design-based pattern enhancement strategies, these strategies could be classified as highlighting the same element and ignoring the extra element. Both optimization-based and design-based strategies could be combined into multiple strategies

4.3.1 *Optimization-based pattern enhancement*

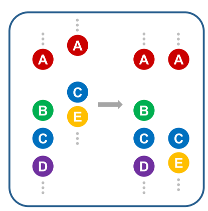
Optimization-based pattern enhancement strategies optimize and highlight the implicit pattern by improving and adjusting the constraint or objective function in mathematical optimization model. In order to optimize the layout to enhance the mode, Sun et al. (2014) utilize an optimization-based method to enhance the pattern. Inspired by their work, we summarize optimization-based pattern enhancement as followings:



Element Movement. In order to enhance the pattern between elements both in S^k and S^t, a feasible method is to move these elements as close as possible in vertical direction. As the left figure shows, all elements in set are moved down. However, the critical problem is how to align multiple same element in S^k and S^t and compute the movement length. Therefore, we consider moving the whole set column to a suitable position in vertical direction, the objective function of element movement optimization model can be written as followings:

$$\min \quad \max_{i \in S^k \cap S^t} \{|y_i^t - y_i^k|\} \tag{14}$$

where i is the element in the intersection set of S^k and S^t. The meaning of these objective functions is to minimize the largest horizontal height difference of elements in the intersection set. We also utilize the *Order Constraint* to preserve the original order of elements.



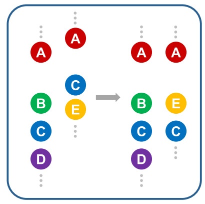
Element Alignment. Placing the same elements in the same horizontal direction could reduce the visual burden when they find the same element. As the left figure shows, we align all elements in the intersection set. Therefore, we need to utilize constraints to limit their coordinates. Considering the number of elements in the intersection set may be more than one, we define a new 0-1 decision variable s_i, s_i is 1 when the element is aligned, otherwise s_i is 0. We add the new constraints into the optimization model as followings:

$$-M(1 - s_i) \leq y_i^k - y_i^t \leq M(1 - s_i) \quad \forall i \in S^k \cap S^t \tag{15}$$

where the above constraints guarantee the alignment of elements when the decision variable is equal to 1, and we add an objective function as followings:

$$\max \sum_{i \in S^k \cap S^t} s_i \tag{16}$$

where the objective function aims to align elements as much as possible, we also utilize the *Order Constraint* to preserve the order and *Gap Constraint* to avoid too close node distance.



Element Reorder. Another way to enhance the pattern is to reorder the element for the late time t. Because t and k are not adjacent time, the mathematical optimization model in Sect. 4.2 do not consider and add constraints to limit their relationship. Therefore, we modify the model to relayout the elements in S^t by the order of S^k. The *Order Constraint* in Eq. 1 can be rewritten as followings:

$$x_i^t < x_j^t \quad where \quad y_i^k < y_j^k \quad \forall i, j \in S^t \cap S^k \tag{17}$$

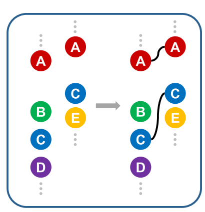
where the above constraint ensures that the elements in S^t follow the order of the elements in S^k. Then we utilize the optimization model in Sect. 4.2 to scale the coordinate and get the final results.

The above three strategies are based on optimization, where the element movement strategy enhances the pattern by modifying the objective function of the optimization model. The element reorder enhances the implicit pattern by modifying the constraints of the optimization model. In particular, the element alignment enhances the implicit pattern by modifying both objective function and constraints in the optimization model. On the other hand, the element movement and element alignment are based on the translation of nodes without changing the order of elements.

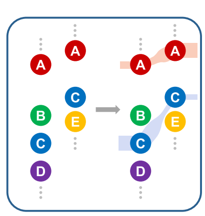
In fact, our optimization-based pattern enhancement strategies could extend to adapt to the case of multiple sets. For example, suppose that the users selected several sets $S^{n_1}, S^{n_2}, \ldots, S^{n_k}$, where $n_1, n_2, \ldots, n_k$ is an ordered index sequence, we can change the condition of the above three pattern enhancement strategies as $\bigcap_{i=1}^{k} S^{n_i}$.

4.3.2 Design-based pattern enhancement

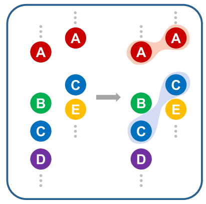
The design-based pattern enhancement highlight the implicit pattern by visual design. Visual design could encode more information in visualization and save the time of solving mathematical optimization model, which could spend a lot of time in the large-scale data. We summarize the designed-based pattern enhancement as followings:



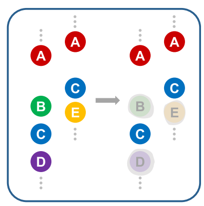
Curve Link. Drawing a smooth curve between two points is a direct and simple approach of design-based pattern enhancement, which is shown as the left figure. In timeline visualization, we utilize a quadratic Bézier curve to draw the links with elements in $S^{t-1} \cap S^t$ between two adjacent timestamps. In order to enhance pattern mined in dynamic sets, we draw a smooth curve in between elements in intersection sets $S^k \cap S^t$.



Flow Link. Instead of curve link, another method for enhancing implicit patterns is utilizing shadow flows to link the same element, which is shown as the left figure. We draw a flow for each element in the intersection set. Each flow link two same elements in the selected set. The advantage of this flow is that the width of each flow could encode the extra information such as the distance sum of element in different times.



Bubble Link. Inspired by Bubble Sets (Collins et al. (2009)), we design the bubble link for highlighting the element of intersection set, which is shown as the left figure. One bubble link the elements in the intersection set which highlight the same element, and we draw bubbles for all elements in the intersection set. The color and opacity of the bubble can encode the extra information of elements such as the importance metric of elements.



Element Bundling. In order to enhance and highlight the extra element, we combine and bind the elements in different set $S^t - S^k$, which means those elements in S^t but not in S^k. The grey bubble of extra elements aims to help users ignore these elements and highlight the same elements. The disadvantage of element bundling is that users may pay more attention to those bound elements.

In the above four pattern enhancement strategies, the former three strategies, curve link and flow link highlight the elements by adding new graphics, the element bundling ignores and hides the visual disturbance which is caused by other elements not in the intersection set.

4.3.3 Multiple pattern enhancement

Combining two or several pattern enhancement techniques is a feasible scheme. In this section, we discuss the combination of pattern enhancement strategies. Using multiple pattern enhancement strategies is not a simple addition of these strategies, it may not enhance the implicit pattern or not work well.

Multiple optimization-based pattern enhancement.

The advantage of our optimization model in pattern enhancement is that we could apply multiple optimization-based pattern enhancement strategies by integrating their objective function and constraints into a unified optimization model. We discuss the combination of pattern enhancement strategies as followings:

- *Element Movement & Alignment* Element movement moves the whole column in timeline visualization while element alignment moves the intersection element in two sets. To combine these two strategies, we need to balance their objective functions in Eq. 14 and Eq. 16 by an extra parameter to transform them into a single-objective optimization model.
- *Element Movement & Reorder* In this combination, the constraints in Eq. 17 can limit the order of elements in the intersection set. Eq. 14 is an objective function that is added to the optimization model when scaling the coordinate. This combination would integrate their advantages.

- *Element Alignment & Reorder* In element alignment, elements in two sets with inconsistent order would not align. Combining with the element reorder, the element in the intersection set could reorder and align fully.

Multiple mixed pattern enhancement. The design-based pattern enhancement strategies do not change the optimization model. Therefore, it does not lead to failures. Multiple design-based pattern enhancement strategies could highlight the implicit pattern more and encode more extra information. Combining optimization-based pattern enhancement strategies and design-based pattern enhancement strategies could integrate their advantages. For example, the curve link or flow link brings more line wiggles. Because there are original line links every adjacent time pair between time t and k. However, element reorder could decrease line wiggles by changing the order to avoid this visual burden.

5 Evaluation

In this section, we describe our evaluation about pattern enhancement strategies including quantitative experiment, two cases studies, and an informal user study.

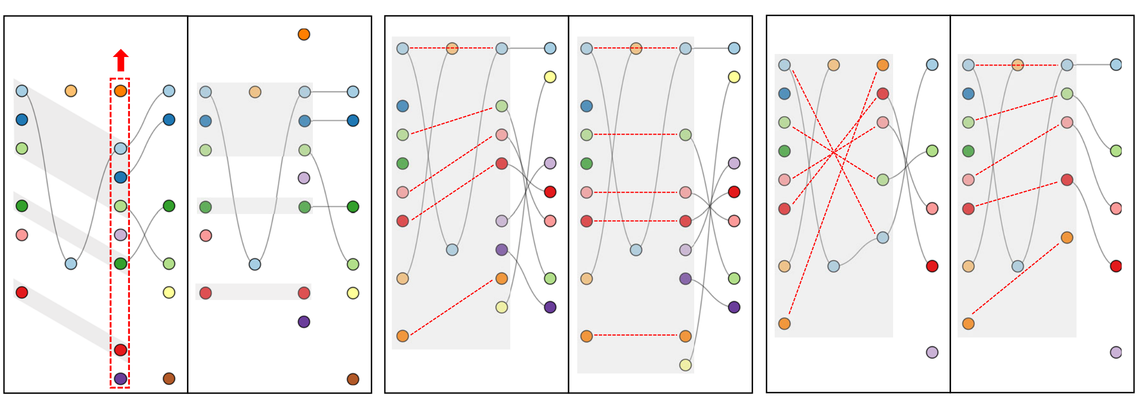
5.1 Experiment

To evaluate our layout algorithm quantitatively, we conduct a quantitative experiment. We use Gurobi Solver (https://www.gurobi.com) in a computer with an Intel Core i7 processor and 16GB memory. We compute the running time of the progressive timeline layout algorithm. The results are shown as followings:

In Table 1, the results of the quantitative experiment show that our progressive timeline layout algorithm would cost more time when the data scale increase. Meanwhile, the fewer intersection elements result in less running time. It fits our expectation because the positions of original nodes are not changed while layouting new nodes at a late moment, which could be also seen as a local optimal algorithm.

Meanwhile, we compute the running time of three optimization-based pattern enhancement strategies. We utilize these pattern enhancement strategies to adjust two nonadjacent time stamps. They both contain twenty nodes. The results are shown as followings:

In Table 2, three pattern enhancement strategies cost less time than the progressive timeline layout algorithm under the same data scale. The *Element Movement* needs the least running time because it only adds an objective function. The two strategies need more running time because they add extra constraints to the optimization model.



(a) Element Movement **(b)** Element Alignment **(c)** Element Reorder

Fig. 3 Results of optimization-based pattern enhancement strategies, there are three pattern enhancement strategies to highlight elements in the intersection set between the first set and third set. (a) Element movement minimizes the largest difference of horizontal height of each element in the intersection set. (b) Element alignment aligns elements in the same horizontal line as much as possible. (c) Element reorder rearranges the order of elements in the intersection set

Table 1 Experiment results of progressive timeline layout algorithm

Node number	15	20	25	50	100
Constraints number	555	990	1550	6225	24950
Running time (s)	0.29	0.59	1.28	9.05	74.25

Table 2 Experiment results of optimization-based pattern enhancement strategies

Strategies	Element movement	Element alignment	Element reorder
Constraints number	6	196	1435
Running time (s)	0.001	0.01	0.03

5.2 Case I: tracking outliers in projection data

In the first case, we utilize a person re-identification query dataset, and extract the feature information in the image ranking list after selecting a query. Then we utilize these feature vectors to project in different methods and parameters. The feature vector is a 128-dimension vector, and we utilize MDS, t-SNE, PCA, and UMAP. By examining different parameters and the top K feature vectors, we obtain the projected data. Then we compute the distance between points in the feature dimension and projected dimension. We choose the top 5% as outliers after ranking the points by the absolute difference of distance in feature and projected dimension.

In this case, the multi-dimensional data before projection does not have temporal information essentially. But the interactive analysis process, to track outliers in different projection, brings the temporal information which could apply the timeline-based layout. We try several methods and record outliers. Finally, we utilize our algorithm to layout the timeline visualization. The result is shown in Fig. 4.

In fact, different combinations of pattern enhancement strategies and different positions that applied strategies both cause the huge number of our experiments. Due to the time limit, we only try three different positions and several combinations of different pattern enhancement strategies. As for the single strategy, we examine all seven strategies. As for multiple pattern enhancement strategies, we choose one optimization-based pattern enhancement strategy and one design-based pattern enhancement strategy. There are twelve combinations of them. We compare a part of the results and show them in Fig. 4, see the complete results in our GitHub repository[1].

The Fig. 4 shows that the optimization-based pattern enhancement strategies perform as our expect, since the bubble link also highlight the elements in intersection sets.

5.3 Case II: Exploring location in spatiotemporal data

The second case utilizes the dataset of VAST challenge 2015 (2015), an example of temporal data. The dataset contains millions of tourist position information in an amusement park on the weekend. We choose the location information of tourists within ninety minutes on Sunday morning, and filter the twenty-five tourists with the most location information. Then we divide the time period by ten-minute stamps and count tourists in a fixed area in the park, where the area is one-quarter of the park area. As for the weight in progressive timeline layout algorithm, we utilize similarity of tourists, the longest common subsequence problem (LCSS) could be used to describe the similarity of location sequences in the temporal dimension (Zheng et al. (2014)), we utilize LCSS to compute the similarity of tourist location sequences.

After processing the dataset, we utilize our progressive timeline layout algorithm to determine the position of nodes. Each node represents a tourist in the park. Then we examine different pattern enhancement strategies and their combinations to highlight the elements of intersection set in the fourth set and sixth set. Four results of our strategies are shown in Fig. 5. In Fig. 5(d), we utilize moving distance in ten minutes to encode the width of the flow. Due to the space limit, we show the other results in our repository.

[1] GitHub: https://github.com/zjutvis/PE-TESV.

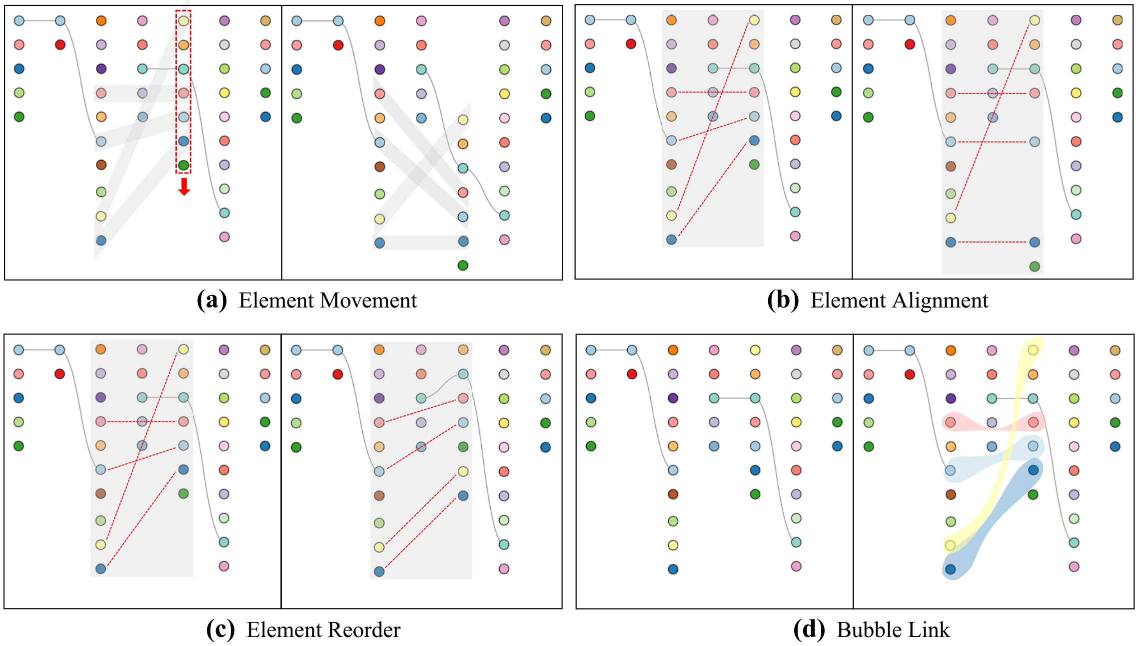


Fig. 4 Case I: Tracking outlier in projection data. We attempt different methods and parameters to project multi-dimensional data. Then we employ our pattern enhancement strategies to highlight the intersection element in the third column and fifth column. Four subfigures show the effect of different strategies from origin (left) to new (right)

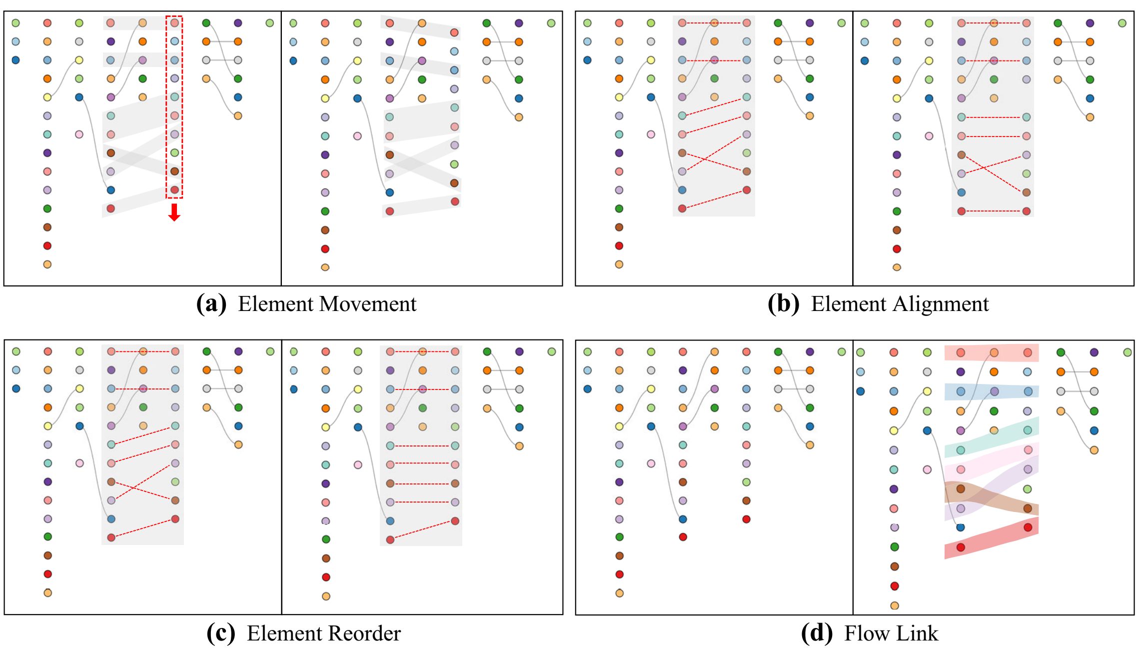


Fig. 5 Case II: Exploring location in spatiotemporal data. We extract temporal evolving set data from temporal data and utilize pattern enhancement strategies to enhance the pattern in the fourth and sixth columns in timeline-based visualization. In each strategy, the elements of the intersection set are highlighted from the origin (left) to the new (right). Each subfigure shows different pattern enhancement strategies. We utilize the tourist distance in each time stamp to encode the width of the flow

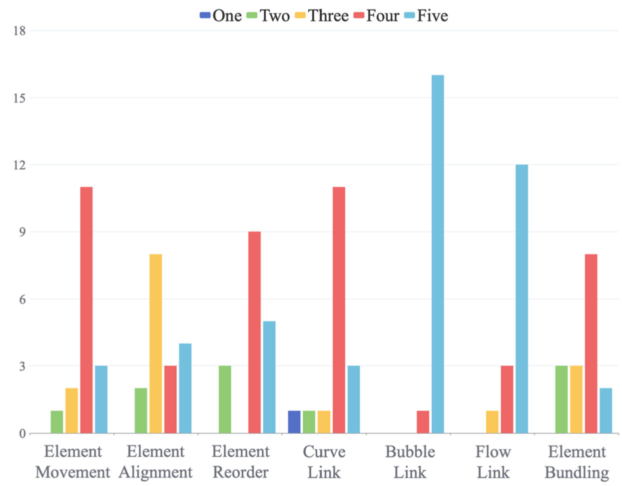


Fig. 6 Results of user study. The score from one to five represents the significance of each pattern enhancement strategy to highlight the intersection elements. Most participants think that our pattern enhancement strategies could enhance the implicit pattern partly

5.4 User study

To compare the effect of pattern enhancement strategies, we conduct an informal user study to verify our pattern enhancement strategies. In the informal user study, we choose timeline-based visualization and their result after applying each pattern enhancement strategy for users to compare the results.

We invite 17 participants to conduct the informal user study, which all major in computer science. Fourteen participants have basic knowledge of visualization. Each participant is invited to compare the origin visualization and the one which applies a pattern enhancement strategy. Participants could choose pattern enhancement strategies they like and appraise the effect of strategies.

In our user study, participants are supposed to evaluate the significant degree of each pattern enhancement strategy and grade each strategy from one to five score. Then participants could choose the best one they thought in optimization-based pattern enhancement strategies, design-based pattern enhancement strategies, and all of them. The results of our informal user study are shown in Fig. 6.

The results of our user study show that most participants think optimization-based pattern enhancement strategies could enhance the intersection elements between two sets, since the element movement strategy performs well. The design-based pattern enhancement strategies could highlight the intersection elements, most participants think these strategies perform well, particularly bubble link. As for optimization-based pattern enhancement strategies, most participants like element movement and element reorder. As for design-based pattern enhancement strategies, most participants like the bubble link. In the process of user study, one participant think that he could not understand element bundling because element bundling make him pay more attention to those elements that are not in the intersection set. "*Overlay visual element enhancement can be used together with the reorder enhancement*", "*Adding visual elements should avoid interference with circle color*", as two participants suggest.

6 Discussion

In this section, we discuss the limitation and future work about our workflow, layout algorithm, and pattern enhancement strategy. Our work does not apply to real world data temporarily, but exploring the possibilities could aid future research.

6.1 Limitation

Our limitation contains three parts as followings:

Workflow. The limitation of our workflow contains three parts. First, our workflow lacks verification and validation in real-world scenarios, which makes our workflow difficult to apply at the moment. Second, while our workflow supports user exploration, it cannot recommend implicit patterns in the temporal and evolving set data to help users identify patterns promptly. Third, our workflow does not extend to other types of data, such as dynamic graphs, which limits the scalability of our workflow.

Layout algorithm. Although our layout algorithm performs well in time efficiency and esthetic metric, the limitation contains two parts. One is that our layout algorithm is a progressive algorithm. When we layout the elements of the set in time t, we will not change the timeline layout before time t. Essentially, it is a local optimal strategy, which may not produce a global fine layout resulting in esthetics. Another is that our layout algorithm contains two stages including determining the order and scaling the coordinate. The time complexity of our algorithm is higher than those one-stage timeline layout algorithms.

Pattern enhancement strategy. The limitation of our pattern enhancement strategy contains two parts. One is that the optimization-based pattern enhancement strategies highlight the elements by rearranging the position of elements, which could spend much time in solving the optimization model. Another is that the design-based pattern enhancement strategies may cause visual ambiguity and clutter in perception. For example, participants in our user study suggest that element bundling strategy shade those elements which are not in the intersection set which may confuse users.

6.2 Future work

After the analysis of limitations, we summarize our future works as follows.

Workflow extension. Our workflow aims to utilize pattern enhancement strategies to highlight the implicit pattern. The basic idea is employed by many researchers (Sun et al. (2014); Cao et al. (2010)) and could be a fundamental component in visualization. Apart from timeline-based visualization, we consider extending our workflow to other forms of visualization. Our future work also contains involving user interaction with the workflow better. With the human-in-the-loop idea, pattern enhancement strategies could perform well as a good supplement for temporal evolving set data visualization.

Data & visualization. Our workflow aims at temporal and evolving set data, but it is not only suitable for temporal and evolving set data. For example, our timeline-based visualization shows the extension feasibility of dynamic graphs. Dynamic graph data contains the relationship between entities in one timestamp, which is represented as a link between points in timeline-based visualization. On the other hand, there are many techniques to visualize set or graph data. Our pattern enhancement strategies need to be generalized to other visualization techniques. For example, if users do not employ timeline-based visualization, the optimization-based pattern strategies need to be considered and rewritten. As a result, improving the visual design and implementing more visualization techniques based on our strategies is an important work in the future.

Algorithm scalability. The scalability of our algorithm is a critical problem when the size of data grows rapidly. Our progressive timeline layout algorithm contains two parts. When we determine the order, we utilize an integer programming model to arrange the order of elements. This problem is that it is an NP-hard problem that could spend a lot of time to solve the large-scale data. It is necessary to find an efficient algorithm to deal with large-scale data in the future. In the part of scaling the coordinate, we employ a multi-objective nonlinear optimization model and utilize a weighted linear combination to simplify three objective functions. The future work will examine the effects of layouts under different weights.

Strategy extension. In the future, we will design and examine more optimization-based pattern enhancement strategies. We could integrate them into a unified optimization model. We also consider deformation as an optimization-based pattern enhancement strategy. However, the disadvantage of deformation may break the ratio of the distance between elements in our timeline-based visualization. In terms of design-based pattern enhancement strategy, we will consider human perception to help propose more significant designs to enhance the implicit pattern. Otherwise, it is necessary to examine the combination of multiple pattern enhancement strategies. Another future work about extending the pattern enhancement strategy is to integrate more interaction techniques, which could help users select set elements and browse element details.

System & application. Our work is not an application-driven research but could be a fundamental component for timeline-based visualization and pattern enhancement. In the future, we aim to find more application scenarios to verify the usefulness of our workflow in temporal and temporal evolving set data. Streaming data are a good choice due to the characteristics of time varying. Based on this data type, the pattern enhancement strategies could highlight the implicit pattern of streaming data between continuous timestamps. In addition, we could design and implement a visual analytic system to involve our workflow, which is helpful for applying our work in specific domains.

7 Conclusion

In this paper, we propose a workflow to analyze and explore the implicit pattern among nonadjacent sets in temporal evolving set data. As for the dynamic characteristics of data, we utilize the timeline-based visualization and design a progressive timeline layout algorithm to compute the position after data update. As for the challenge of the relationship visualization between two sets, we propose the optimization-based and design-based pattern enhancement strategies, and discuss the extension for multiple sets analysis. We also provide a quantitative experiment to examine the algorithm efficiency. To compare our strategies and illustrate the usability of our proposed strategies, we also conduct two case studies and an informal user study Finally, we discuss the limitation and feature work of the workflow. We hope the pattern enhancement strategies could be a fundamental component in temporal evolving set visualization.

Acknowledgements This work was supported in part by the National Key Research and Development Program of China (2020YFB1707700), National Natural Science Foundation of China (61972356, 62036009). The authors are grateful to all reviewers for valuable comments and participants in user study.

References

Abdi H, Williams LJ (2010) Principal component analysis. Wiley Interdiscip Rev Comput Stat 2(4):433–459

Agarwal S, Beck F (2020) Set streams: visual exploration of dynamic overlapping sets. Comput Gr Forum 39(3):383–391

Alsallakh B, Micallef L, Aigner W, Hauser H, Miksch S, Rodgers P (2016) The state-of-the-art of set visualization. Comput Gr Forum 35(1):234–260

Ankerst M, Berchtold S, Keim DA (1998) Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In: Proceedings of the IEEE symposium on information visualization, pp 52–60

Beck F, Burch M, Diehl S, Weiskopf D (2017) A taxonomy and survey of dynamic graph visualization. In: Computer graphics forum, Wiley Online Library

Bu C, Zhang Q, Wang Q, Zhang J, Sedlmair M, Deussen O, Wang Y (2021) SineStream: improving the readability of streamgraphs by minimizing sine illusion effects. IEEE Trans Vis Comput Gr 27(2):1634–1643

Burch M, Beck F, Diehl S (2008) Timeline trees: visualizing sequences of transactions in information hierarchies. In: Proceedings of the working conference on advanced visual interfaces, AVI '08, p 75–82

Burch M, Ten Brinke KB, Castella A, Peters GKS, Shteriyanov V, Vlasvinkel R (2021) Dynamic graph exploration by interactively linked node-link diagrams and matrix visualizations. Vis Comput Ind Biomed Art 4(1):1–14

Cao N, Sun J, Lin Y-R, Gotz D, Liu S, Qu H (2010) FacetAtlas: multifaceted visualization for rich text corpora. IEEE Trans Vis Comput Gr 16(6):1172–1181

Chang B, Sun G, Li T, Huang H, Liang R (2022) MUSE: visual analysis of musical semantic sequence. IEEE transact on visual comput graphic, pp 1–1

Collins C, Penn G, Carpendale S (2009) Bubble sets: revealing set relations with isocontours over existing visualizations. IEEE Trans Vis Comput Gr 15(6):929–936

Cui W, Wang X, Liu S, Riche NH, Madhyastha TM, Ma KL, Guo B (2014) Let it flow: a static method for exploring dynamic graphs. In: Proceedings of IEEE pacific visual symposium, pp 121–128

Dasgupta A, Arendt DL, Franklin LR, Wong PC, Cook KA (2018) Human factors in streaming data analysis: challenges and opportunities for information visualization. Comput Gr Forum 37(1):254–272

Di Bartolomeo M, Hu Y (2016) There is more to streamgraphs than movies: better aesthetics via ordering and lassoing. In: Computer graphics forum, Wiley Online Library

Eckelt K, Hinterreiter A, Adelberger P, Walchshofer C, Dhanoa V, Humer C, Heckmann M, Steinparz C, Streit M (2022) Visual exploration of relationships and structure in low-dimensional embeddings. IEEE transaction on visualization and computer graphics, pp 1–1

Espadoto M, Martins RM, Kerren A, Hirata NS, Telea AC (2019) Toward a quantitative survey of dimension reduction techniques. IEEE Trans Vis Comput Gr 27(3):2153–2173

Fink M, Haunert JH, Schulz A, Spoerhase J, Wolff A (2012) Algorithms for labeling focus regions. IEEE Trans Vis Comput Gr 18(12):2583–2592
Fischer MT, Frings A, Keim DA, Seebacher D (2021) Towards a survey on static and dynamic hypergraph visualizations. In: 2021 IEEE visualization conference (VIS), pp 81–85
Guo S, Jin Z, Gotz D, Du F, Zha H, Cao N (2019) Visual progression analysis of event sequence data. IEEE Trans Vis Comput Gr 25(1):417–426
Guo S, Xu K, Zhao R, Gotz D, Zha H, Cao N (2018) EventThread: visual summarization and stage analysis of event sequence data. IEEE Trans Vis Comput Gr 24(1):56–65
Haunert JH, Sering L (2011) Drawing road networks with focus regions. IEEE Trans Vis Comput Gr 17(12):2555–2562
Jayatilaka LG, Bertuccelli LF, Staszewski J, Gajos KZ (2010) PETALS: a visual interface for landmine detection. In: Adjunct proceedings of the 23nd annual ACM symposium on user interface software and technology, UIST '10, p 427-428
Keim D, Mansmann F, Schneidewind J, Ziegler H (2006) Challenges in visual data analysis. In: Tenth international conference on information visualisation (IV'06), pp 9–16
Krstajić M, Keim DA (2013) Visualization of streaming data: observing change and context in information visualization techniques. In: Proceedings of the international conference on big data, pp 41–47
Lex A, Gehlenborg N, Strobelt H, Vuillemot R, Pfister H (2014) Upset: visualization of intersecting sets. IEEE Trans Vis Comput Gr 20(12):1983–1992
Liu S, Wu Y, Wei E, Liu M, Liu Y (2013) StoryFlow: tracking the evolution of stories. IEEE Trans Vis Comput Gr 19(12):2436–2445
Mayorga A, Gleicher M (2013) Splatterplots: overcoming overdraw in scatter plots. IEEE Trans Vis Comput Gr 19(9):1526–1538
McInnes L, Healy J, Melville J (2018) Umap: uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426
Mead A (1992) Review of the development of multidimensional scaling methods. J Royal Stat Soc: Ser D (Stat) 41(1):27–39
Mizuno K, Wu H-Y, Takahashi S, Igarashi T (2019) Optimizing stepwise animation in dynamic set diagrams. Comput Gr Forum 38(3):13–24
Nguyen PH, Xu K, Walker R, Wong BW (2014) SchemaLine: timeline visualization for sensemaking. In: 2014 18th international conference on information Visualisation, pp 225–233
Nguyen PH, Xu K, Walker R, Wong BW (2016) TimeSets: timeline visualization with set relations. Inf Vis 15(3):253–269
Nonato LG, Aupetit M (2018) Multidimensional projection for visual analytics: linking techniques with distortions, tasks, and layout enrichment. IEEE Trans Vis Comput Gr 25(8):2650–2673
Sadana R, Major T, Dove A, Stasko J (2014) OnSet: a visualization technique for Large-scale binary set data. IEEE Trans Vis Comput Gr 20(12):1993–2002
Sizemore AE, Bassett DS (2018) Dynamic graph metrics: tutorial, toolbox, and tale. NeuroImage 180:417–427
Sohns J-T, Schmitt M, Jirasek F, Hasse H, Leitte H (2021) Attribute-based explanation of non-linear embeddings of High-dimensional data. IEEE Trans Vis Comput Gr 28(1):540–550
Sun G, Liang R, Qu H, Wu Y (2017) Embedding Spatio-temporal information into maps by Route-zooming. IEEE Trans Vis Comput Gr 23(5):1506–1519
Sun G, Wu Y, Liu S, Peng TQ, Zhu JJ, Liang R (2014) EvoRiver: visual analysis of topic coopetition on social media. IEEE Trans Vis Comput Gr 20(12):1753–1762
Sun G, Zhu S, Jiang Q, Xia W, Liang R (2021) EvoSets: tracking the sensitivity of dimensionality reduction results across subspaces. IEEE transaction on big data, pp 1–1
Sun G-D, Wu Y-C, Liang R-H, Liu S-X (2013) A survey of visual analytics techniques and applications: state-of-the-art research and future challenges. J Comput Sci Technol 28(5):852–867
Tanahashi Y, Hsueh CH, Ma KL (2015) An efficient framework for generating storyline visualizations from streaming data. IEEE Trans Vis Comput Gr 21(6):730–742
Tanahashi Y, Ma KL (2012) Design considerations for optimizing storyline visualizations. IEEE Trans Vis Comput Gr 18(12):2679–2688
Tarnita CE, Antal T, Ohtsuki H, Nowak MA (2009) Evolutionary dynamics in set structured populations. Proc Natl Acad Sci 106(21):8601–8604
Tatu A, Albuquerque G, Eisemann M, Schneidewind J, Theisel H, Magnork M, Keim D (2009) Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. In: 2009 IEEE symposium on visual analytics science and technology, pp 59–66
Van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9(11)
von Landesberger T, Bremm S, Andrienko N, Andrienko G, Tekušová M (2012) Visual analytics methods for categoric spatio-temporal data. In: 2012 IEEE conference on visual analytics science and technology (VAST), pp 183–192
VAST Challenge 2015. http://vacommunity.org/VAST+Challenge+2015
Wang Y, Wang Y, Zhang H, Sun Y, Fu CW, Sedlmair M, Chen B, Deussen O (2019) Structure-aware fisheye views for efficient large graph exploration. IEEE Trans Vis Comput Gr 25(1):566–575
Wang YS, Chi MT (2011) Focus+context metro maps. IEEE Trans Vis Comput Gr 17(12):2528–2535
Xia J, Gao L, Kong K, Zhao Y, Chen Y, Kui X, Liang Y (2018) Exploring linear projections for revealing clusters, outliers, and trends in subsets of multi-dimensional datasets. J Vis Lang Comput 48:52–60
Xu K, Salisu S, Nguyen PH, Walker R, Wong BLW, Wagstaff A, Phillips G, Biggs M (2020) TimeSets: temporal sensemaking in intelligence analysis. IEEE Comput Gr Appl 40(3):83–93
Yaghoubi E, Kumar A, Proença H (2021) Sss-pr: a short survey of surveys in person re-identification. Pattern Recogn Lett 143:50–57
Yuan J, Chen C, Yang W, Liu M, Xia J, Liu S (2021) A survey of visual analytics techniques for machine learning. Comput Vis Med 7(1):3–36

Yuan X, Guo P, Xiao H, Zhou H, Qu H (2009) Scattering points in parallel coordinates. IEEE Trans Vis Comput Gr 15(6):1001–1008

Zheng Y, Capra L, Wolfson O, Yang H (2014) Urban computing: concepts, methodologies, and applications. ACM Trans Intell Syst Technol 5(3):1–55