

# TacticFlow: Visual Analytics of Ever-Changing Tactics in Racket Sports

Jiang Wu, Dongyu Liu, Ziyang Guo, Qingyang Xu, and Yingcai Wu

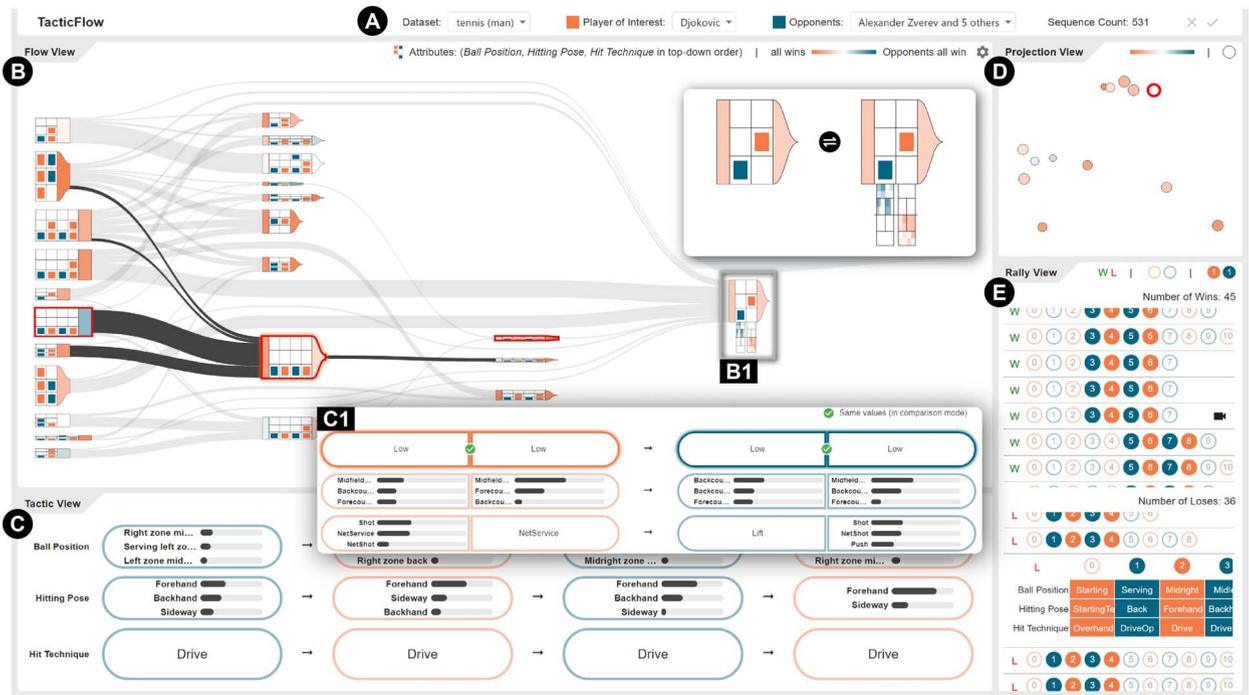


Fig. 1. A screenshot of the user interface for visually analyzing the tactic progression in racket sports. There exist five main views: the Control Bar (A), the Flow View (B), the Tactic View (C), the Projection View (D), and the Rally View (E). An analysis process may be that users first filter the sequences to be analyzed in the Control Bar. The system mines tactics from these sequences, and the Projection View projects the tactics into a 2-D plane based on the similarity among these tactics. After users clicking on a point of interest, they can view where the tactic is used and the related tactic progression in the Flow View. We propose an intuitive glyph design (B1) for each node to visualize the usage of a tactic. Users can also view the tactic in detail in the Tactic View. A comparison mode of this view (C1) further supports comparing two tactics. Finally, in the Rally View, users can view the raw sequences that apply the selected tactic.

**Abstract**— Event sequence mining is often used to summarize patterns from hundreds of sequences but faces special challenges when handling racket sports data. In racket sports (e.g., tennis and badminton), a player hitting the ball is considered a multivariate event consisting of multiple attributes (e.g., hit technique and ball position). A rally (i.e., a series of consecutive hits beginning with one player serving the ball and ending with one player winning a point) thereby can be viewed as a multivariate event sequence. Mining frequent patterns and depicting how patterns change over time is instructive and meaningful to players who want to learn more short-term competitive strategies (i.e., tactics) that encompass multiple hits. However, players in racket sports usually change their tactics rapidly according to the opponent's reaction, resulting in ever-changing tactic progression. In this work, we introduce a tailored visualization system built on a novel multivariate sequence pattern mining algorithm to facilitate explorative identification and analysis of various tactics and tactic progression. The algorithm can mine multiple non-overlapping multivariate patterns from hundreds of sequences effectively. Based on the mined results, we propose a glyph-based Sankey diagram to visualize the ever-changing tactic progression and support interactive data exploration. Through two case studies with four domain experts in tennis and badminton, we demonstrate that our system can effectively obtain insights about tactic progression in most racket sports. We further discuss the strengths and the limitations of our system based on domain experts' feedback.

**Index Terms**—Sports Analytics, Multivariate Event Sequence, Sequential Pattern Mining, Progression Analysis.

## 1 INTRODUCTION

- Jiang Wu, Ziyang Guo, Qingyang Xu, and Yingcai Wu are with State Key Lab of CAD&CG, Zhejiang University. E-mail: wujiang5521, ziyangguo27, qingyangxu17, ycwu@zju.edu.cn. Yingcai Wu is the corresponding author.
- Dongyu Liu is with Massachusetts Institute of Technology. E-mail: dongyu@mit.edu.

Manuscript received 21 Mar. 2021; revised 13 June 2021; accepted 8 Aug. 2021.

Date of publication 29 Sept. 2021; date of current version 22 Dec. 2021.

Digital Object Identifier no. 10.1109/TVCG.2021.3114832

Racket sports, including tennis, badminton, and table tennis, are popular worldwide. In a singles *match* of a racket sport, two players alternate *hitting* the ball, beginning with one player's serve and ending with a player scoring a point. This series of alternate hits is called a *rally*. A player may consider multiple, detailed *characteristics* when hitting the

ball, such as hit technique and ball position. In this sense, each hit can be considered a multivariate event – with each of these characteristics comprising an attribute – and each rally can thus be viewed as an event sequence. To win a rally, a player may adopt several *tactics*: short-term competitive strategies that utilize various characteristics over the course of the next few hits. In tennis, for example, one tactic may consist of three hits: 1) Player 1 hits the ball to the left side of the tennis court (ball position); 2) Player 2 runs to this position to return the ball (player position predicted by Player 1); 3) Player 1 uses a powerful hitting technique, a *drive*, to hit the ball to the right side of the tennis court so that the opponent cannot easily return the ball (hit technique; ball position). Because tactics often determine the outcome of a match, domain experts hope to gain insights into how they are used.

However, analyzing tactics is complicated because they are versatile and ever-changing over the course of a high-level match, due to the following dynamics: First, **players change tactics in response to the opponents in a rally**. As two players alternate hitting the ball, each player changes tactics rapidly according to how his/her opponent reacts. Second, **players prefer different tactics in the same situation**. A player may choose a tactic for a specific situation, such as when serving or if the opponent uses a certain move. A player may also “mix it up” over the course of a game, making different choices in the same situation so that the opponent cannot predict his/her tactics and adopt counter-tactics. Third, **a player using the same tactic in different situations can yield changing results**. A player may be good at a particular tactic and adopt it in a variety of situations. The tactic may work well in some situations but not others, leading to different results.

Such features of tactics introduce two challenges. First, mining particular tactics is challenging. Tactics are non-overlapping multivariate patterns, and there may be more than one in a sequence. To the best of our knowledge, no existing algorithm can be easily applied to mine tactics. Second, even if we have identified the tactics, domain experts need to analyze tactic progression. Many studies [29] have proven that visualizations are beneficial for progression analysis. However, the ever-changing progression of multivariate tactics can bring heavy visual clutter, preventing domain experts from exploring tactic progression.

To address the aforementioned challenges, we introduce TacticFlow, a visual analytics system for analyzing ever-changing tactics in racket sports. To tackle the first challenge, we propose a data-driven definition of tactics, describing a tactic by a multivariate pattern, and a multivariate pattern mining algorithm. To tackle the second challenge, we introduce a tailored user interface to visualize ever-changing tactics with multiple levels of detail [50]. The core view is a glyph-based Sankey diagram to demonstrate the progression of tactics, where we propose a glyph design to help domain experts find interesting tactics quickly. We further enable rich data exploration interactions, such as comparing two tactics and tracking the progression.

Our system is evaluated through two case studies on two real-world datasets — a tennis dataset and a badminton dataset – in consultation with four domain experts, two from each sport. The case studies show that our system can help domain experts gain insights into how players changed tactics. After the case studies concluded, we conducted interviews with the experts and gathered feedback from them on the usability and effectiveness of our system. All domain experts praised our system for helping players improve their competitive performance.

In summary, the main contributions of this work include:

- An effective pattern mining pipeline for identifying various tactics and discovering ever-changing tactic progressions.
- A tailored user interface for visually and interactively analyzing ever-changing tactics in racket sports.
- Two case studies on real-world racket sports datasets along with expert interviews, demonstrating the usability of our system.

## 2 RELATED WORK

### 2.1 Multivariate Pattern Mining

Pattern mining algorithms are widely used for analyzing event sequences [29]. Some traditional sequential pattern mining (SPM) algorithms search all subsequences for which the frequency is higher than a certain *threshold* [19, 20]. MSDD [44] first introduced a multivariate SPM algorithm, inspiring many other multivariate SPM al-

gorithms [9, 42, 52, 59]. However, SPM methods suffer from *pattern explosion* [40] – they return few patterns when the *threshold* is high, but a great number of patterns when the *threshold* is slightly lower. Domain experts may waste considerable time examining numerous meaningless (more precisely, similar and repetitive) patterns.

Recently, the Minimum Description Length (MDL) principle [2, 4, 25, 53] has been introduced to summarize patterns. Unlike SPM methods that search all frequent subsequences, MDL-based methods discover a set of patterns that summarize the dataset well. The pattern set is usually small, and the patterns are usually meaningful, avoiding the problem of *pattern explosion*. Wu et al. [60] introduced a steerable MDL-based multivariate pattern mining algorithm for interactive pattern exploration. However, this algorithm can only extract one pattern for a sequence and thus cannot reveal the progression of tactics. Ditto [2] can mine multiple multivariate patterns in a sequence, but these patterns can overlap, making it difficult to analyze the progression of tactics. In this paper, we propose a new pattern mining pipeline for identifying tactics and detecting tactic progression in racket sports.

### 2.2 Progression Analysis of Event Sequence

The comprehensive survey of event sequence [1] indicated that progression analysis is generally regarded as an essential task, which “*aims to uncover the evolution of one event during a period of time.*” Many previous studies [16, 33, 35, 47] adopted timeline-based visualizations to display each sequence, encoding events with visual metaphors, as progression analysis is mainly related to time. These studies visualized the progression of each individual sequence, but analyzing the progression of hundreds of sequences was time-consuming. To overcome this problem, Flow-based designs [10, 17, 24, 32, 41, 58, 66] simplified the visualizations by aggregating similar sequences and constructing flow charts. Each node represents a certain type of event occurring in multiple sequences at a similar time, and the links reveal how the sequences progress among these nodes over time. However, as the flow-based designs usually visualize the progression between two consecutive events, this flow may be extremely long when handling long sequences. DecisionFlow addressed this problem through milestone-based data representation and visualization. However, experts in racket sports cannot define and select the milestone events manually, because ever-changing tactics can lead to ever-changing milestones. EventThread [27, 28] extended the flow-based design to the stage-based design, where multiple events occurred in a particular period are aggregated into a node. However, EventThread focus on univariate event sequences and cannot be applied to multivariate tactics easily. Eventpad [5, 6] can handle multivariate sequences, but focuses on event-level exploration and searching rather than progression analysis. We extend the stage-based design by proposing a glyph-based Sankey diagram, where a tailored glyph for each node visualizes the usage of a multivariate tactic.

### 2.3 Visual Analytics of Event Sequences in Sports

Event sequence data is a general data model widely applied in various sports, and visual analytics of event sequence data is increasingly essential for sports [46]. In team sports, such as soccer [1, 45, 63], basketball [7], rugby [12, 31] and baseball [36], events include passing, dribbling, etc., and a sequence usually describes how a team organizes an attack. In racket sports, each stroke can be considered as an event and each rally as a sequence [37]. Wu et al. [60] proposed a generic framework for all racket sports to compare the tactical patterns of two players. CourtTime [48] visualized the position where the ball falls in the tennis court with a novel visual metaphor. ShuttleSpace [64] and TIVEE [11] analyzed badminton data immersively. In table tennis, iTTVis [61] revealed the relationship among multiple attributes with a matrix, and Tac-Simur [56] simulated the results of tactical changes based on the Markov Chain. Tac-Miner [55] proposed a knowledge-driven definition of “tactic” (i.e., three consecutive strokes) and discovered all tactics by simply traversing each sequence. In contrast to Tac-Miner, this paper proposes a data-driven definition of “tactic,” which facilitates the discovery of new insights beyond existing knowledge, and incorporates a pattern-mining algorithm in order to summarize tactics from all sequences. Moreover, the progression of tactics is an essential and understudied issue in sports. This paper proposes a tailored design for

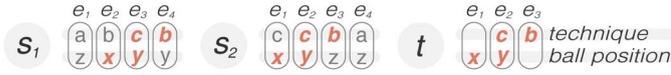


Fig. 2. Two event sequences and a tactic in racket sports. Sequences  $s_1$  and  $s_2$  both comprise four events over two attributes, namely technique and ball position. Tactic  $t$  has 3 events, where the blanks indicate the positions filled by different values in different sequences.

visual analysis of tactic progression in racket sports.

### 3 BACKGROUND

For the past five years, we have held weekly meetings with two teams of domain experts, one in tennis and the other in badminton. Each team consists of a professor and several Ph.D. students in sports science, and both provide data analysis for national teams. The results of our collaboration are as follows, including the data model, the analysis tasks, and a tailored visual analytics system.

#### 3.1 Data Model

**Raw data.** Because in racket sports players alternate hitting the ball or shuttle, we model this data as event sequences, where each hit is regarded as an event and a series of consecutive hits is regarded as a sequence. Formally, our dataset contains hundreds of *sequences*, denoted  $S = \{s_1, s_2, \dots, s_n\}$ , where  $n$  is the number of sequences. Each sequence  $s_i$  is a vector of ordered *events*, denoted  $s_i = \{e_1, e_2, \dots, e_m\}$ , indicating  $m$  alternative hits from an initial serve to a winning hit. Each hit is described as a multivariate event, denoted by  $e_i = \{a_1 = v_1, a_2 = v_2, \dots, a_k = v_k\}$ , where we consider  $k$  categorical attributes from  $a_1$  to  $a_k$  (e.g., the technique used to hit, the rotation direction) and the corresponding values from  $v_1$  to  $v_k$ , respectively. Although racket sports are similar, different sports have different attributes. For example, the height of the ball is essential in badminton but not in table tennis. Moreover, even if some attributes are involved in many sports, the values can be different. For example, the position of the ball is of concern in all racket sports, but a tennis court is much larger than a ping pong table, leading to more optional ball positions in tennis.

**Mined tactics.** In a data-driven way, we define a *tactic* as a frequent subsequence  $t$  that consists of two players' alternating hits, in two steps. First, given an event  $e_a = \{a_1 = v_1, a_2 = v_2, \dots, a_k = v_k\}$  and  $l$  integers satisfying  $1 \leq i_1 < i_2 < \dots < i_l \leq k$ , we define  $e_b = \{a_{i_1} = v_{i_1}, a_{i_2} = v_{i_2}, \dots, a_{i_l} = v_{i_l}\}$  as a partial event of  $e_a$ , which preserves  $l$  attributes of  $e_a$  and drops other attributes, denoted  $e_b \preceq e_a$ . For example, in Fig. 2, event  $e_3$  of tactic  $t$  is a partial event of event  $e_3$  of sequence  $s$ , where the technique is preserved and the ball position is dropped. Second, given sequence  $s_a = \{e_{a_1}^1, e_{a_2}^1, \dots, e_{a_m}^1\}$  and  $s_b = \{e_{b_1}^1, e_{b_2}^1, \dots, e_{b_l}^1\}$ , we define  $s_b$  as a subsequence of  $s_a$ , if there exist  $l$  integers  $1 \leq i_1 < i_2 < \dots < i_l \leq m$  satisfying  $e_{b_x}^1 \preceq e_{a_{i_x}}^1$  for each  $1 \leq x \leq l$ . For example, in Fig. 2, tactic  $t$  is a subsequence of sequence  $s_1$ .

To ensure that the tactics are meaningful, we further add two constraints. First, a **tactic should be compact**. Players usually consider only about three consecutive hits in a tactic because predicting the opponent's actions is difficult after several hits. Thus, the hits in a tactic must be consecutive, and the length is about three. Second, **two tactics should not overlap**. A player can perform only one tactic at a time. Thus, for each pair of tactics that occur in the same sequence, one tactic must end before the other tactic starts.

#### 3.2 Task Analysis

We collected the requirements for visually analyzing changeable tactics from domain experts and summarized the analysis tasks as follows.

**T1 Identify the tactics used by a player.** Given that each player has personal tactics, domain experts usually specify a player of interest and analyze his/her personal tactics. Moreover, domain experts usually select many similar opponents (e.g., all left-handed opponents) to know the player's tactics for competing against such opponents comprehensively. Our system should mine and visualize the tactics of a certain player.

**T2 Reveal the tactic progression.** When finding a tactic of interest, domain experts need to know what tactics it comes from and what

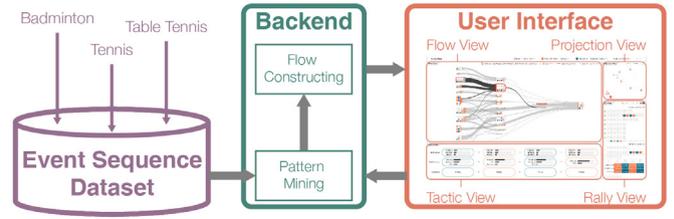


Fig. 3. The framework of our system, which consists of an event sequence dataset, a backend with core algorithms, and a user interface. The gray arrows indicate the data flow.

tactics it may progress into. By understanding the various changes, experts can know when a tactic is used and the results that the tactic can lead to. Our system should reveal the progressive relationships among versatile tactics.

#### T3 Point out the tactics worth analyzing across a full progression.

A player may adopt versatile tactics to compete against opponents. Instead of analyzing each tactic and exploring what comes before and after, domain experts prefer to analyze tactics that can affect the overall progression significantly (e.g., a tactic that can directly score a point without tactics coming after it). Our system should visualize the tactics with multiple levels of detail, where the visualizations at the highest level should help users quickly find the tactics worth analyzing.

#### T4 Compare two multivariate tactics.

Comparison is essential for data analysis of sequential data [22, 26, 38]. For two tactics, domain experts usually want to know which is better and why it is better than the other. Especially for two similar ones, experts need to know which characteristics make them perform differently. Our system should support comparison between two tactics.

#### T5 Display the raw sequences.

Experts need to examine a tactic in the context of a rally in order to know what happened in real games so that they can communicate each rally to the player. Our system should display the raw data for data exploration in detail.

### 3.3 System Framework

We introduce a generic visual analytics system to satisfy the aforementioned domain requirements (Fig. 3). The system is made up of a dataset, a backend, and a user interface. The dataset contains event sequences collected from different racket sports. The backend reads sequences from the dataset, mines tactics in these sequences, and constructs a flow chart that reveals how players change their tactics. The user interface comprises five main views to visualize the mined results and support steerable data exploration.

We implement the dataset through MongoDB. The backend is developed based on the Flask framework in Python 3, using RESTful API. We implement our algorithm in C++, which is called by the backend, to achieve high performance efficiently. We adopt React, a well-known JavaScript library, to build the user interface. We open all the source on OSF (<https://osf.io/9x4yp/>).

## 4 ALGORITHM

We introduce a new pattern mining pipeline to discover tactics in racket sports and how they change. In this section, we briefly introduce the Minimum Description Length (MDL) principle as the basis of our method and then demonstrate our implementation in detail.

### 4.1 MDL principle

The MDL principle was initially a theory used for data compression [25]. Given a dataset  $D$ , the MDL principle tries to find a model  $M$  that can summarize the dataset well, and describes the dataset based on the model. MDL assumes that the optimal model brings the shortest description length of the dataset. Formally, MDL optimizes the model  $M$  to obtain a minimum  $L(M) + L(D|M)$ , where  $L(M)$  represents the description length of the model and  $L(D|M)$  represents the description length of the datasets based on the model.

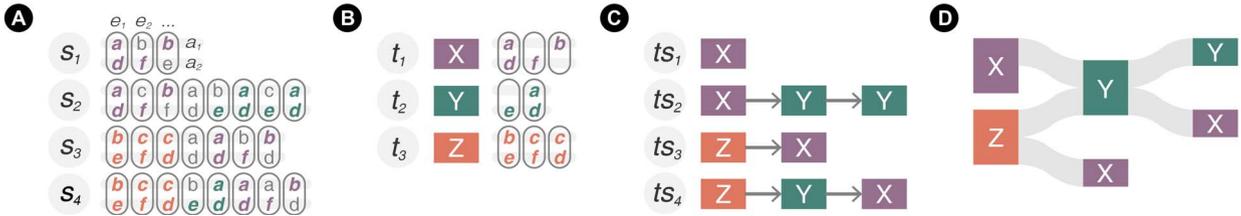


Fig. 4. The pipeline of our algorithm. (A) There exist four event sequences ( $s_1$  to  $s_4$ ) over two attributes ( $a_1$  and  $a_2$ ) in the dataset. (B) We mine three tactics ( $X, Y$ , and  $Z$ ) from the four sequences. (C) We transfer the original event sequences into tactic sequences ( $ts_1, ts_2, ts_3$  and  $ts_4$ ). (D) We summarize a flow diagram from the tactic sequences to reveal the tactic progression.

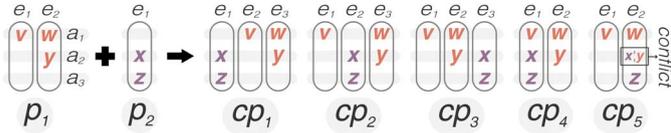


Fig. 5. Example of generating candidate patterns. There exist two patterns ( $p_1$  and  $p_2$ ) over three attributes ( $a_1$  to  $a_3$ ). Combining  $p_1$  and  $p_2$  in different alignments can obtain five candidate patterns ( $cp_1$  to  $cp_5$ ). But  $cp_5$  is invalid because there is a conflicting value.

Recently, MDL has been increasingly adopted in sequential pattern mining [2, 34, 51, 53] and related visual analytics systems [8, 15, 60]. Given a dataset of event sequences, MDL considers a set of patterns as the model to describe the original sequences and define a measure to calculate the description length. By minimizing the description length, the algorithm can discover an optimized set of patterns.

## 4.2 Algorithm Implementation

Our algorithm is highly correlated with our visualizations. Considering that there exist two core analysis targets: the tactics and the tactic progression, we implement our algorithm in two steps: **identifying the tactics** and **discovering the tactic progression**. Detailed implementation of these two steps is as follows.

### 4.2.1 Identifying Tactics

In this step, we introduce a multivariate pattern mining algorithm to summarize a small set of frequent tactics from hundreds of original sequences. For example, in Fig. 4, we mine three tactics ( $t_1$  to  $t_3$ ) from four sequences ( $s_1$  to  $s_4$ ). Inspired by Ditto [2], we propose a generic multivariate pattern mining algorithm based on MDL. The core idea is similar to the Generative Adversarial Network (GAN), where the algorithm consists of two parts: a pattern **generator** and a pattern **discriminator**. The generator continuously generates new candidate patterns based on the current pattern set. The discriminator provides a measure based on MDL to judge whether a pattern is beneficial to summarizing the dataset. The algorithm iteratively adds the good patterns to the pattern set and removes the bad ones until the pattern set no longer changes. We adopt this idea due to two reasons: 1) The generator can selectively generate tactics instead of searching the ample space of versatile tactics, reducing the running time. 2) The discriminator can define a measure flexibly based on domain requirements to filter the tactics, which are meaningful to domain analysis.

**Generator:** The generator first initializes the pattern set with all single values, i.e., a partial event with only one value. Then, at every iteration, the generator selects a pair of tactics in the pattern set and combines them at different alignments to generate candidate patterns. For example, in Fig 5, we can generate five candidate patterns ( $cp_1$  to  $cp_5$ ) by combining two patterns ( $p_1$  and  $p_2$ ) at different alignments. However, we regard  $cp_5$  as invalid because there exists a conflicting value. To ensure that the combination of tactics is reasonable, the generator prioritizes the combination of the two tactics that have the most times of simultaneous occurrences.

**Discriminator:** The discriminator adopts the MDL principle to determine whether a pattern  $p$  should be included in the pattern set  $P$ . Given a pattern set  $P$  and a candidate pattern  $cp$ , the discriminator

will regard  $cp$  as beneficial to summarizing dataset  $S$  if  $L(S, P + cp) < L(S, P)$ , where  $L(S, P)$  is the description length using a given pattern set  $P$  to describe the original dataset  $S$ . After  $cp$  is added to  $P$ , the discriminator will regard pattern  $p$ , which is not a single value and is used to generate  $cp$ , as redundant if  $L(S, P - p) < L(S, P)$ . We compute  $L(S, P)$  by considering three pieces of information that domain experts mainly concern about in visual analytics. We explain these three pieces of information as follows, taking Fig. 4 as an example.

- **The number of tactics.** There exist three tactics,  $X, Y$ , and  $Z$ .
- **The number of usages.** We describe the sequences by searching the usage of each tactic. In Fig. 4, we highlight each usage of these three tactics with the corresponding colors in the original sequences. Pattern  $X$  is used 4 times,  $Y$  is used 3 times, and  $Z$  is used 2 times.
- **The number of single values.** After describing the sequences by mined tactics, there may exist single values that cannot be described by tactics. In Fig. 4, we color these single values in grey. Sequence  $s_1$  has 2 single values,  $s_2$  has 4 ones,  $s_3$  has 1 one, and  $s_4$  has 3 ones.

We need to balance these three pieces of information that users obtain in our system. If there exist more tactics and more usages of tactics, the number of single values will decrease, leading to lower information loss. However, the tactic progression can be more complicated, leading to higher visual clutter and analysis burden. In contrast, if there exist fewer usages of tactics, the number of single values increases. Even if we obtain the low visual clutter, the information loss is high so that experts cannot obtain meaningful insights. To trade off between information loss and visual clutter [54], we compute  $L(S, P)$  as follows:

$$L(S, P) = \alpha|P| + \beta \sum_{p_i \in P} usage(p_i) + \gamma \sum_{s_i \in S} single\_values(s_i), \quad (1)$$

where  $|P|$  represents the number of tactics,  $usage(p_i)$  represents how many times pattern  $p_i$  is used,  $single\_values(s_i)$  represents the number of single values in sequence  $s_i$ , and parameters  $\alpha, \beta$ , and  $\gamma$  control the three terms. For convenience, we always set  $\alpha$  to 1. When  $\beta$  increases, the mined patterns will consist of more values, such that fewer patterns are used to summarize  $S$ . When  $\gamma$  increases, the algorithm will mine as many patterns as possible from a sequence so that fewer single values exist. By defining such measures for computing the description length, we can effectively find all tactics that meet the domain requirements.

### 4.2.2 Discovering Progression

This step is designed for highly abstracting the complex original sequences into a clear Sankey diagram that reveals the tactic progression, based on the mined tactics (Fig. 4(A and B  $\rightarrow$  D)). Given a pattern set  $P$  and original sequences  $S$ , the algorithm generates a directed acyclic graph  $G$  (i.e., the Sankey diagram). The abstraction mainly involves two simplifications as follows.

**Event sequences  $\rightarrow$  Tactic sequences:** We first simplify each original sequence into a tactic sequence in the order in which the tactics appear, so that we can omit the events that are not in tactics and reduce visual clutter. For example, as shown in Fig. 4(C), we obtain four tactic sequences,  $ts_1$  to  $ts_4$ , by simplifying the four original sequences,  $s_1$  to  $s_4$ , respectively. Each tactic sequence is thereby a vector  $ts = \{t_1, t_2, \dots, t_l\}$ , where  $l$  is the length, and  $t_i$  indicates a tactic.

**Tactic sequences  $\rightarrow$  Tactic flow:** Given a set of tactic sequences  $TS = \{ts_1, ts_2, \dots, ts_n\}$ , we construct a directed acyclic graph to summarize how tactics progress (Fig. 4(C  $\rightarrow$  D)). Each node represents a

tactic, and each link from node  $X$  to node  $Y$  represents the sequences that contain two consecutive tactics from tactic  $X$  to tactic  $Y$ . The algorithm consists of three steps — aligning, merging, and linking. First, the algorithm aligns all tactics based on their indexes in the original sequences, ensuring that aligned tactics occur almost simultaneously. For example, assuming that tactic  $t_a$  and  $t_b$  are adopted in the  $a_1$ -th hits to the  $a_2$ -th hits and the  $b_1$ -th hits to the  $b_2$ -th hits, respectively, we require that  $a_1 \leq b_2$  and  $b_1 \leq a_2$ . Second, for each group of aligned tactics, we merge the same tactics as a node. For example, in Fig. 4(D), we merge two  $X$ s aligned at the first column, two  $Z$ s aligned at the first column, and two  $Y$ s aligned at the second column. Third, the algorithm will link each set of two nodes that occur consecutively in several tactic sequences, as we show in Fig. 4(D).

## 5 SYSTEM DESIGN

In this section, we introduce our visual designs and interactions to satisfy the aforementioned analysis tasks.

### 5.1 User Interface

The user interface consists of five main views (Fig. 1), namely, the Control Bar (A), the Flow View (B), the Tactic View (C), the Projection View (D), and the Rally View (E). Users usually start from the Control Bar, selecting one player and his/her opponents as the analysis target. The system filters all the rallies played by the selected players, and then discovers the tactics and tactic progression, which usually takes about one minute. Users can turn to the Flow View to explore the tactic progression. They can click on a node of interest in the Flow View, and the Tactic View then visualizes the tactic in detail. In addition, the Rally View displays all the rallies containing the tactic of interest, providing access to the raw sequences. We also design the Projection View to allow users to find similar tactics, providing a Levenshtein distance-based measure to compute the similarity. Users can click on two similar tactics and compare them in the Flow View, which highlights where they are used, and the Tactic View, which enables a comparison mode for comparing two tactics. Further explanation is as follows.

#### 5.1.1 Control Bar

The Control Bar allows users to select a dataset from one of several racket sports, one player of interest (IP), and multiple opponents (OPs) in order to analyze the tactics of the IP against these OPs (T1). The system filters the rallies and displays the number of filtered rallies on the right. Throughout the user interface, we distinguish the IP from the OPs by *hue* because it is the most effective way to differentiate among a small number of categories [43]. We chose orange for the IP because it is an energetic and attention-getting color that domain experts preferred. We chose blue to stand for the OPs because it contrasts with orange.

#### 5.1.2 Flow View

The Flow View uses a Sankey diagram to show an ever-changing tactic progression (T1, T2). The diagram is in chronological order from left to right. Each node indicates a tactic along with certain contexts (i.e., specific pre-tactics). Multiple nodes may share the same tactic but have different contexts. We introduce a glyph-based design (Fig. 1(B1)) to encode the tactic and its context (T3) because glyphs can visualize multidimensional data with intuitive visual metaphors [3], preferred by domain experts. Each flow from one node to another indicates that IP first uses a tactic and then changes into another tactic without other tactics in between, where the *width* encodes the number of sequences.

**Glyph Design.** The glyph for each node consists of four parts, which follow a timeline-based design to maintain consistency with the chronological order of the Sankey diagram. Each part encodes one piece of key information about the tactic, helping users find tactics of interest quickly. Details are as follows:

*The “head” for contexts.* The leftmost rectangle (i.e., the head) encodes the context in which the tactic was used. The *height* encodes the number of rallies that used this tactic in that context. The *hue* encodes whether IP or OPs have a higher winning rate before using the tactic (i.e., the before-tactic winning rate). The *saturation* further encodes the difference in this winning rate. The before-tactic winning rate reveals whether a player is already at an advantage or a disadvantage before

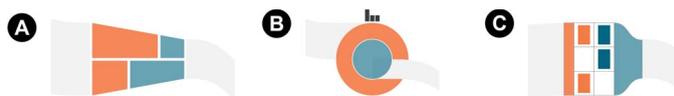


Fig. 6. Three potential designs for the glyph. We finally choose C because it follows the timeline-based design, thereby easy to understand.

choosing (or not choosing) the tactic at issue, allowing domain experts to better understand the effect of the tactic itself. For example, assume that there are three tactics, and both tactic A and tactic B can lead to tactic C. To compute the before-tactic winning rate of tactic C, we calculate the average winning rate after using tactic A and tactic B.

*The “body” for the multivariate tactic.* Inspired by MatrixWave [67], we propose a matrix-based design (i.e., the body) for encoding a multivariate tactic and revealing which attributes are most important to that tactic. Each column represents an event, and each row represents an attribute. Each cell thereby indicates a value, where the *solid* cells highlight the values of attributes in the tactic, and the *hue* encodes whether IP or OPs hit the ball.

*The “tail” for transition.* The shrinking flow (i.e., the tail) illustrates how rallies progress after the tactic is used. Some rallies end immediately, while others go on to include more tactics, which we call posterior tactics. For each tail, the *height* of the right side, which also indicates the number of rallies that have posterior tactics. The *difference in height* on the two sides of the flow thereby encodes the number of rallies that end after using the tactic. A node with a significant height difference indicates a tactic that often leads directly to scoring or losing a point. The tail has similar color encodings to the head, encoding the winning rate after the tactic is used. By comparing the color of the head with the color of the tail, users can determine the tactic’s effect and whether it is a good tactical choice.

*The expandable “belly” for spatial values.* Spatial values (e.g., ball position) play a role in all racket sports and are informative for domain experts. Inspired by the work of Liu et al. [39] and Perin et al. [45], which allow users to switch among different glyphs that visualize multiple levels of detail, we propose an expandable glyph design, toggled by double-clicking on the glyph, to encode the detailed values of a spatial attribute. Inspired by studies on spatial-temporal visualizations [14, 18, 30, 49, 57, 62], we superimpose a heatmap on a court-shaped diagram, below each column of the matrix, indicating where the ball bounced on the court before the corresponding hit. For each area in the heatmap, the hue encodes the player who receives the ball, and the opacity encodes the frequency.

We have considered other design alternatives for the glyph. We first consider the trapezoid-based design (Fig. 6(A)), where the upper part and the lower part encode the before-tactic and after-tactic winning rates, respectively, by the *area* of the left block (IP wins) and the right block (OPs win). The abstract design makes it difficult for users to remember the meaning of each block. We also consider the circle-based design (Fig. 6(B)), where the outer circle encodes the contexts, the inner circle encodes the results, and the bar chart on the top reveals the most related attribute. Although the design is a visual metaphor for a ball, the encodings are not intuitive enough to be recognized and memorized. Finally, we adopt the timeline-based design (Fig. 6(C)), which maintains consistency with the chronological order of the Sankey diagram, and is therefore easy to understand.

**Layout algorithm.** Multiple flows, especially cross-layer ones, may exist in racket sports because of the ever-changing tactics. Flow crossing and the flow-node overlap can be heavy, making it difficult for users to obtain insights. We address this problem in three ways. 1) We insert dummy nodes [65] for each cross-layer flow to avoid flow-node overlap. However, when many dummy nodes exist, the layout may be so sparse that users cannot view the whole flow chart. Thus, we allow users to choose whether to use the dummy nodes (Fig. 9) or not (Fig. 8). When dummy nodes are not used, we add a gap between the overlapping node-edge pair to prevent users from thinking they are connected. 2) We adopt a heuristic layout algorithm [21] to reduce flow crossing. An alternative is an ILP-based crossing reduction algorithm [65], which can lead to the fewest number of crossings. However, the ILP-based algorithm may take too long to run when many dummy nodes exist.

Thus, we choose the heuristic one to balance between run time and the effect of the layout. 3) We further allow users to hover over a node to highlight all the flows linked to it.

### 5.1.3 Projection View

The Projection View (Fig. 1(D)) reveals similarities among tactics (**T1**, **T3**). Each tactic is represented by a point, and the distance between two points encodes the similarity between the two tactics. For each point, the *area* encodes the frequency of the tactic. The *hue* and the *saturation* encode the winning rate after using the tactic, consistent with the tail of our glyph design in the Flow View. To reveal the similarity, we adopt t-SNE to project each tactic into the 2-D plane. We chose t-SNE for two reasons. First, t-SNE strives to preserve local neighborhoods, which satisfies the preferences of experts, who only care about similar tactics and not how different less similar ones are. Second, it can accept a distance matrix as input so that we can manually compute the similarity and encode it as distance. We compute the difference between two tactics based on Levenshtein distance, an edit-based similarity measure, because the tactics are short and of unequal length [13]. We consider five tailored edits – each with a cost of 1 – for transferring multivariate tactics: adding/deleting/replacing a value and inserting/deleting an event without any values. However, spatial attributes (e.g. ball position) differ from others (e.g. hitting technique) because we can intuitively perceive the distance between two spatial values. Thus, we calculate the Euclidean distance to represent the edit cost between two spatial values. We set the largest Euclidean distance for each attribute as 2. For example, when indicating where the ball bounces on a half-court, we define the diagonal length of the half-court as 2. This setting ensures that the average edit cost for changing one spatial value to another is about 1, so that spatial attributes are not overweighted compared to other attributes.

### 5.1.4 Tactic View

When users click on the glyph or point of a tactic, the Tactic View can visualize the multivariate tactic in tabular form (**T1**), where each row represents one attribute, and each column represents one event. Each obround indicates the value of an attribute within an event, while the *hue* of the background encodes who hits the ball. For each value within the tactic, we directly display this value as *text* because there are so many possibilities that it does not make sense to design different encodings or to expect users to remember them. For values where the player has multiple choices for applying the tactic, we visualize the frequency of each choice through a bar chart, one bar for one choice. The bar chart helps users understand how the tactic is used in detail by including values ignored by the pattern mining algorithm. Due to space limitations, we surface the three most frequently seen values because these are most important to domain experts. Users can also hover on the obround to view the frequency of all values.

When users then right click on another tactic, this view can enable a comparison mode (Fig. 1(C1)) to compare the two tactics (**T4**). To reduce users' learning costs, we simply divide each obround into two parts and juxtapose the corresponding values of the two tactics for a one-to-one comparison [23]. To ensure that similar hits are aligned, we also propose a greedy alignment strategy based on the relative indices and which player made each hit. For example, tactic A consists of hits 1, 2, and 3, and tactic B consists of hits 7, 8, and 9. If hit 1 and hit 7 are made by the same player, we will align them directly. Otherwise, we will either align hit 1 and hit 8 or hit 2 and hit 7, depending on which

### 5.1.5 Rally View

The Rally View (Fig. 1(E)) lists all the rallies that contain a tactic selected by users (**T5**), one per row. The view is divided into two parts according to the outcome of each rally so that users can compare rallies that IP won (the upper parts, indicated by a W on the left) and that he/she lost (the lower part, indicated by an L on the left). The top-right corner of each part displays the total number of wins and losses. For each row, circles on the right represent the hits in the rally, where the *hue* encodes whether IP or OPs hit the ball, and the number is the index of the hit. We highlight hits within the tactic with a solid circle so that users can know where the tactic is applied. When the user clicks on a

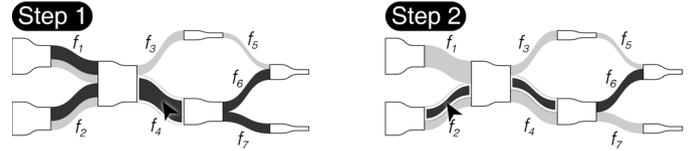


Fig. 7. Example of tracking rallies. Given a Sankey diagram containing seven flows ( $f_1$  to  $f_7$ ), we click on two flows ( $f_4$  and  $f_2$ ) in two steps. The clicked flows will be marked with a white stroke, and the related flows will be highlighted with a darker color.

rally, the exact values are expanded into a table. Users can also hover over a row to find a video button on the right, which triggers a video of the appropriate segment of the rally.

### 5.1.6 Interactions

**Navigating the flow chart.** Because there may exist numerous tactics, users can zoom in, zoom out and pan the Flow View by scrolling the wheel and dragging the canvas, respectively. To prevent users from accidentally moving the flow chart offscreen and losing track of it, users can reset the canvas to its original position by double-clicking.

**Tracking the progression of some rallies.** Users can click on any of the several flows in the Flow View to track rallies that contain these flows. After a flow is clicked on, the Flow View filters the rallies and highlights the flows that these rallies flow through with a darker color. The width of the darker flows encodes the number of filtered rallies. For example Fig. 7 shows a Sankey diagram containing seven flows ( $f_1$  to  $f_7$ ). In step 1, we click on flow  $f_4$ , which is then marked with a stroke. The previous flows ( $f_1$  and  $f_2$ ) and the posterior flows ( $f_6$  and  $f_7$ ) are also highlighted with a darker color. In step 2, we further click on flow  $f_2$  in the rallies. As a result, flow  $f_6$  is still highlighted, but flow  $f_7$  is no longer highlighted, indicating that all rallies that flow from  $f_2$  and  $f_4$  will flow to  $f_6$  but not  $f_7$ .

**Link-highlighting the same tactics in different views.** We link all the visual elements that encode the same tactics together to allow users to explore a tactic of interest in different views. For each tactic, the linked visual elements include a point that represents the tactic in the Projection View and several nodes where the tactic is applied in the Flow View. When users click on one of these, the user interface highlights all linked visual elements with shadow. Simultaneously, the Tactic View displays the tactic, and the Rally View displays all the rallies in which the tactic is used.

**Comparing similar tactics.** We support the comparison of tactics in the Projection View, which reveals similarities directly. To compare two tactics – both represented by points – users click on one point and then right-click on a second. The Tactic View will enable the comparison mode, and the Flow View will highlight the nodes where the two tactics are used with two levels of shadow, in order to differentiate them.

**Displaying details on demand.** Since users may forget the meaning of encoding or need to know specific values, they can hover over any visual element to reveal a tooltip that shows the accurate values. For example, users can hover on a node in Flow View to view the before-tactic and after-tactic winning rate, the frequency, and the number of rallies with posterior progression. Users can also hover on a point in Projection View to view the frequency and the winning rate.

## 6 EVALUATION

This section first measures the efficiency and effectiveness of our algorithm via an algorithm evaluation. Two case studies with four domain experts, as well as interviews with these experts, further demonstrate the usability of our system.

### 6.1 Algorithm Evaluation

We quantitatively evaluated our multivariate pattern mining algorithm on 5 multi-scaled synthetic datasets and 2 real-world datasets. The average performance of our algorithm on each dataset is reported in Table 1. The 5 synthetic datasets vary in terms of the number of sequences, the average length of each sequence, the number of attributes, and the number of optional values of each attribute. The 2

Table 1. The results of the quantitative experiments. We generated 5 synthetic datasets ( $D_1$  to  $D_5$ ) and collected two real-world datasets ( $D_T$  and  $D_B$ ), varying in terms of the number of sequences ( $S$ ), the length of each sequence ( $|s_i|$ ), the number of attributes ( $|A|$ ), and the number of optional values of each attribute ( $|V_k|$ ). After running the algorithm, we recorded the runtime ( $t$ ), the number of mined tactics ( $|T|$ ), and the number of planted tactics detected ( $|PT_d|$ , only for synthetic datasets).

Datasets					Results		
$id$	$ S $	$ s_i $	$ A $	$ V_k $	$t(s)$	$ T $	$ PT_d $
$D_1$	500	10	3	10	28.9	35	5
$D_2$	<b>700</b>	10	3	10	56.1	26	5
$D_3$	500	<b>20</b>	3	10	53.1	26	5
$D_4$	500	10	<b>5</b>	10	84.3	58	5
$D_5$	500	10	3	<b>20</b>	25.9	30	5
$D_T$	531	8.43	3	26 / 12 / 14	35.6	35	/
$D_B$	338	12.82	3	4 / 6 / 15	56.2	25	/

real-world datasets were provided by domain experts and cannot be made public due to a confidentiality agreement. These were also used in our case studies (Sec. 6.2). Because no other tactic-mining algorithm of this type exists to serve as a baseline, we tested only our algorithm, using a computer with a 2 GHz CPU and 16 GB of memory.

**Synthetic Dataset Generation.** Each synthetic dataset was generated in three steps: 1) We randomly generated  $S$  multivariate event sequences over  $|A|$  attributes, each with  $|V_k|$  optional values. The length of each sequence was  $|s_i|$ . 2) We randomly generated five multivariate tactics. The length of each tactic was between 2 and 4. 3) We randomly planted the patterns into sequences without overlapping, ensuring that each pattern covered 10% of events in the datasets. All the parameters here are similar to those that would appear in a real-world dataset.

**Results.** We discuss the performance of our algorithm in two aspects. 1) *Runtime*  $t$  is about one minute. This runtime is acceptable for users because we only run the algorithm once and save all the data for interactive exploration in memory. 2) *Effectiveness.* All planted tactics in the synthetic datasets could be detected ( $|PT_d| = 5$ ), where we regard a planted tactic as detected if a mined tactic contains the planted tactic. The number of all mined tactics ( $|T|$ ) for each dataset is about 30, which is small enough for domain experts to analyze.

## 6.2 Case Studies

We obtained two datasets from domain experts for case studies — a tennis dataset  $D_T$  and a badminton dataset  $D_B$ . Both datasets are based on several matches during high-level events in 2019, all played between top players in the quarterfinals or later. We only considered rallies that contained five or more hits because athletes generally do not change tactics during short rallies. Dataset  $D_T$  was collected from six matches, where *Djokovic* served the ball and competed against six other top-10 players. The dataset recorded three attributes for each event, including the hitting position (26 values), the player’s pose (12 values), and the hitting technique (14 values). Dataset  $D_B$  was collected from five matches, where *Momota Kento* played against four other top-10 players. The dataset recorded three attributes, including the shuttle’s height (4 values), the hitting position (6 values), and the hitting technique (15 values). All the attributes and values are listed in the Appendix.

For each dataset, we invited two domain experts engaged in data analysis of racket sports to conduct a case study together. One expert is a professor, and the other is a Ph.D. candidate. They are both former professional players and have provided data analysis for a national team for more than five years. We deployed the system online and conducted case studies through online meetings, each lasting for about 20 minutes. We first introduced the system to the experts and explained the encodings and interactions. After they were familiar with the system, we allowed them to explore the data by themselves, where we recorded the workflow and their explanations for the findings.

### 6.2.1 Tennis

In this case, experts explored the ever-changing tactics of *Djokovic* and further discovered how to compete against him (Fig. 8). They

first filtered out all tactics with a frequency of less than 9, in order to simplify the visualizations. They were quickly attracted to the Flow View (without dummy nodes), where most nodes ended with an orange tail, indicating that *Djokovic* could score more points than his opponents while using most tactics. The experts explained that even when his opponents were all top athletes, *Djokovic* still had a significant advantage against them. They became particularly curious about how to win against *Djokovic*. Exploring the data with this in mind, they obtained several insights, the most impressive of which are as follows.

**Predicting *Djokovic*’s serving helps to win against him.** Because knowing *Djokovic*’s strengths can reduce the probability of losing to him, experts browsed the Sankey diagram to search for tactics that allowed him to score the most points, looking especially for serving tactics. They quickly found two at the start of the Sankey diagram (Fig. 8 (A1 and A2)), where the tails were orange and shrank significantly. Experts concluded that *Djokovic* uses high-quality serves to score points directly. One expert commented, “*the glyphs help me find tactics of interest quickly instead of examining the tactics one by one, saving my time.*” By double-clicking on the nodes to expand them, experts further found that *Djokovic* served in predictable ways: When he served at the right of the centerline, he always hit the ball to the right corner (Fig. 8 (A1a)). And when he served at the left of the centerline, he usually hit the ball to the outside corner of the left zone (Fig. 8 (A2a)). Experts suggested that while competing against *Djokovic*, an athlete could predict which area he would hit the ball to by where he stood. One expert commented, “*prediction is essential for athletes to return a high-quality ball and keep the competitive edge in the following hits.*”

**Grabbing the first offense at the second hit helps to win against *Djokovic*.** Experts continued to search for tactics that have effectively countered *Djokovic*’s serving. They focused on a tactic (Fig. 8 (A3a)) at the start of the Sankey diagram, which ended with a blue tail, and clicked on it. They first noticed that the system highlighted three nodes where the tactic was adopted, but that the other two nodes (Fig. 8 (A3b and A3c)) had orange tails, meaning that the tactic scored many points for *Djokovic* in the other two situations. There existed a wide flow from A3a to A3b, which indicated that the tactic could lose many points for *Djokovic* when used once but score many points when using it twice. Experts further turned to the Tactic View to explore the tactic in detail (Fig. 8(B3)). The tactic was most related to the hitting technique and the hitting pose. Two players used an offensive technique called *drive*, and tended to use *forehand* to hit the ball. Finally, experts viewed the Rally View (Fig. 8(C3)) and found that this tactic usually led to *Djokovic* losing points when used at the start of the rallies. The stacked bar chart made after the meeting (Fig. 8(D3)) also proved these findings, where the x-axis was the index of the hit where the tactic is used, and the y-axis was the number of rallies that *Djokovic* won (in green) or lost (in red). Experts concluded that one effective strategy for winning against *Djokovic* involves grabbing the first offense at the second hit. In addition, ending the rally as soon as possible is better than allowing a long-term counterattack, which tends to be beneficial to *Djokovic*. Both experts felt interested in this conclusion, with one commenting that “*only when we know the tactic progression can we comprehensively know the tactic in depth.*”

### 6.2.2 Badminton

In this case study, experts explored how *Momota* scored points. Unlike tennis, where rallies are usually short, rallies in badminton are much longer, meaning serving tactics have a smaller impact on the outcome. Thus, badminton experts preferred to analyze how athletes adopted tactics near the end of a rally instead of focusing on serving tactics.

Experts first found that the number of tactics was so large that the flow chart was complicated. They found it helpful to filter out tactics that occurred less than 9 times. One expert commented, “*I can choose to analyze one player’s main strategies in a short time or analyze him/her comprehensively in a long time.*” At first glance, they found that there were only 11 points in the Projection View (Fig. 9(A)), but numerous nodes in the Flow View (Fig. 9(B)). They thought this was correct and explained that defense is usually easier than offense in badminton because the badminton racket is long while the court is small. Thus, most badminton players prefer to *counterattack* — patiently defending

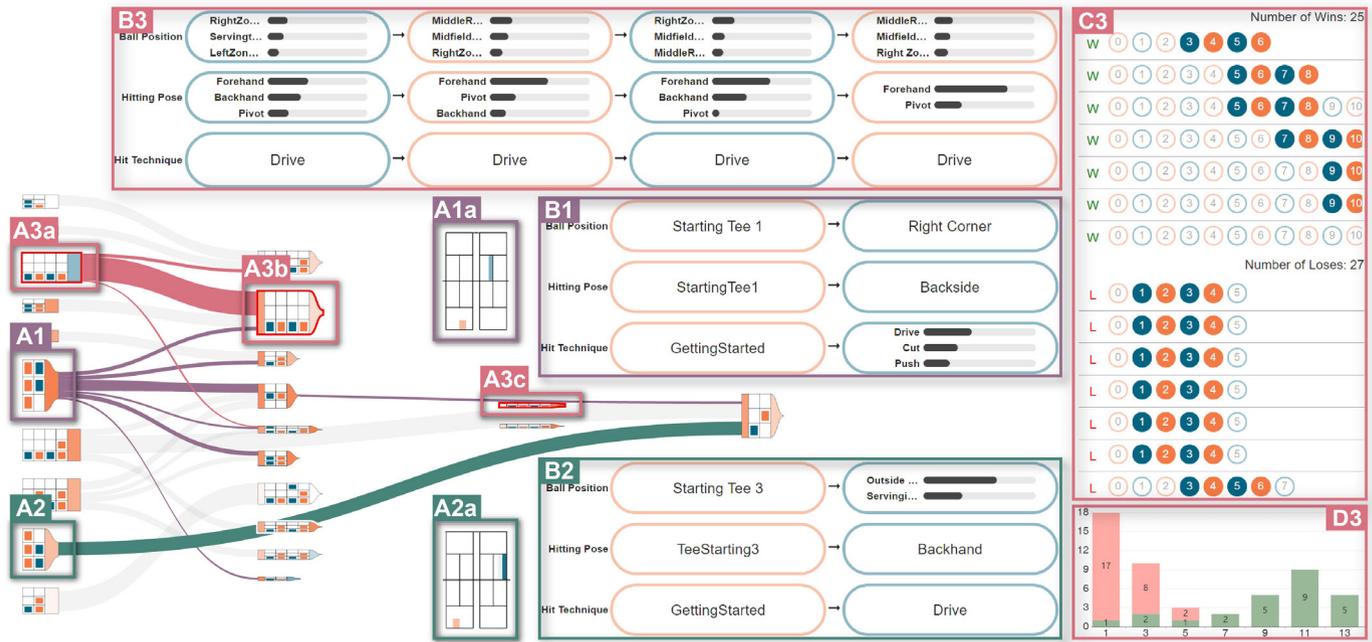


Fig. 8. Screenshots of our system visualizing tennis analysis (Sec. 6.2.1). Experts explored three tactics, and we highlight them in purple, green, and pink, respectively. A1 and B1 are related to the first tactic, visualizing where the tactic is used and the details of the tactic, respectively. So do A2 and B2, which are related to the second tactic. The third tactic (B3) is used in three situations (A3a, A3b, and A3c). C3 shows the rallies that adopt this tactic. D3 visualizes the relationship between the number of wins (green line) or losses (red line) and the index of the hit where the tactic is used.

with multiple hits, and fiercely attacking after they find their opponent’s flaws. The key to switching from defense to offense was forcing the opponent to run to one side of the court so that he/she could not return a shuttle hit to the other side. Although the limited area of the court limits the number of possible tactics, players can repetitively hit the shuttle to different sides, resulting in a complicated progression. This finding inspired them to start the exploration from the Projection View. Their two most impressive insights are as follows.

**Superb net tactics brought opportunities for Momota to attack in the backcourt** Experts quickly found the largest orange point in the Projection View (Fig. 9(A1)), which indicated that *Momota* used this tactic frequently and won lots of points. By clicking on it, experts explored the tactic in detail. They first turned to the Tactic View (Fig. 9(C), the left part)). They found that both players hit the shuttle at the low height, which was risky (the shuttle may be caught in the net), but could bring high rewards (it is difficult for the opponent to return). In all cases, *Momota*’s opponents used the technique *lift*, which sends the shuttle high and to the backcourt. Experts speculated that *Momota*’s superb net tactics forced the opponent to lift the shuttle, creating opportunities for him to attack powerfully in the backcourt. One expert commented, “I like the simple texts and bar charts. They are easier to understand than complicated design.” Experts next turned to the Flow View (Fig. 9(C), with dummy nodes) to further explore the use of this tactic. They found that *Momota* usually brought out this tactic at the start of a rally (Fig. 9(B1a)). They also found that the node B1b received flows from many previous nodes, indicating that *Momota* used this tactic in many different cases. Experts clicked on the flow to explore the progression. They noticed that flows existed connecting nodes B1a, B1b, and B1c, which meant that *Momota* might use this tactic as a serving tactic and then use it repeatedly to win against opponents. Both experts praised this interaction, with one saying that “the system guides me to focus on the tactic progression of my interest.”

**Momota should not serve the shuttle to the left of the mid-court when using this tactic.** Experts further discovered a small blue point (Fig. 9(A2)) near the largest orange point, which indicated that these two tactics were similar but led to different performances. In this description, we refer to the orange tactic as Tactic 1 and the blue one as Tactic 2. By right-clicking on the blue point, experts compared Tactic 2 with Tactic 1. One expert believed it useful to compare two

similar tactics with different winning rates, “athletes could quickly find a tactical weakness they had not seen before, and know how to correct them.” They first turned to the Flow View (Fig. 9(B)) and compared the contexts of these two tactics. In contrast to Tactic 1, which was used at many different times, Tactic 2 was usually used at the start of a rally (Fig. 9(B2a, B2b)), especially as a serving tactic. They further compared these two tactics in the Tactic View (Fig. 9(C)). They found that within these two tactics, only one value differed — the position where *Momota* hit the shuttle to on the second hit. *Momota* usually hit the shuttle to the right of the court in Tactic 1 but the left of the mid-court in Tactic 2. By viewing the posterior progression of Tactic 2, experts found that *Momota* was unable to attack consecutively. The experts speculated that presently, *Momota* might adopt Tactic 2 as an alternative to Tactic 1. However, serving the shuttle to the left of the mid-court might be a weak tactic because *Momota*’s opponents were all right-handed, who can return the shuttle in a suitable pose. One expert commented excitedly, “this finding may help our athletes win more points against Momota.”

### 6.3 Expert Interview

After the case studies, we conducted an in-depth interview with each expert about the platform’s usability and their suggestions. They praised our system highly for its professionalism and its ability to bring new insights. The Flow View impressed the experts mostly due to its innovative design and ability to provide experts with the overall impression of the ever-changing tactics. Another expert suggested that we could recommend the “optimal” flow path after a node, showing the best choices for both players because he believed that the Sankey diagram could be used as a decision tree. All experts also found the system interactions intuitive. One expert thought highly of the tooltips and the link-highlight, saying “the system is user-friendly. Want to know what it means? Hover on it. Want to know more? Click on it.” All experts agreed that our system provided a novel perspective for domain analysis in terms of tactic progression. They also believed that our system could significantly improve athletes’ tactical performance if employed by the national team. One expert summarized, “we only knew the single tactics of a player before but never knew when would the player use a tactic and which tactics he/she planned for subsequent hits. The system brings new insights about tactics to us so that we can

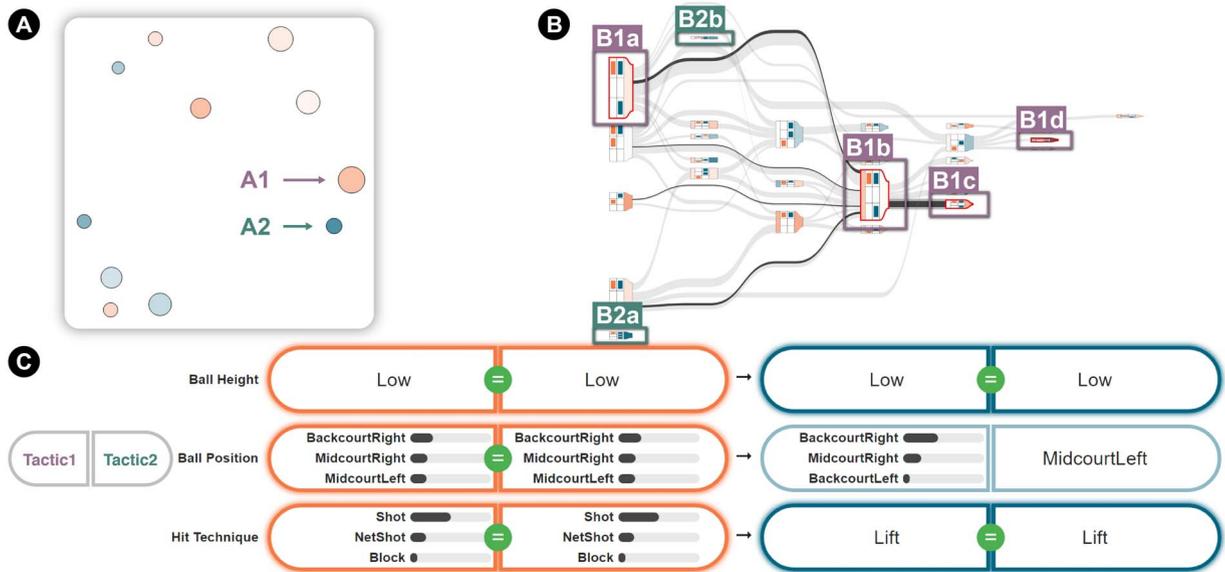


Fig. 9. Screenshots of our system visualizing badminton analysis (Sec. 6.2.2). (A) is the Projection View, where experts analyzed two tactics – Tactic1 (A1) and Tactic2 (A2). (B) is the Flow View, where B1a to B1d represent the usages of Tactic1, and B2a and B2b represent the usages of Tactic2. (C) is the Tactic View, comparing the two tactics, where Tactic1 is on the left, and Tactic2 is on the right.

*think more to play against some tactics.”*

## 7 DISCUSSION

In this work, we propose a novel visual analytics system for analyzing ever-changing tactic progression in racket sports to help domain experts discover how athletes change their tactics. In the field of visual analytics, this work introduces a novel multivariate pattern mining algorithm and tailored glyph-based visualizations to discover pattern progressions. In the field of domain analysis, we introduce a new analysis method to help experts and athletes understand tactic progression. Through case studies, domain experts discovered how top-tier athletes changed their tactics and learned how to compete against them effectively.

**Generalizability.** Although we have focused on racket sports in this paper, we argue that our work can inspire research in other sports that require progression analysis of multivariate sequential patterns. Taking soccer as an example, each individual pass or shot could be considered as an event, and a series of passes until the final shot could be considered as a sequence. For each pass, experts may be concerned with the players’ positions, each team’s formation, and so on. Progression analysis is essential because a team can adopt ever-changing passing tactics, which involve several passes, depending on whether the ball is in the back, middle, or front of the field. One of our experts, who also has knowledge about soccer, commented that our work could be used to analyze the progression of passing tactics. He said, “such analysis can help us understand how a team creates the opportunity for a shot by using passing tactics progressively.”

**Scalability.** We discuss the scalability of three main views in our system. 1) When we compare two long tactics in the Tactic View, the width of the Tactic View may diverge. We first use an adaptive layout, so that tactics can be displayed as completely as possible. We also allow users to scroll the Tactic View horizontally when the tactics are too long to be shown on one screen. 2) When many nodes and flows exist in the Flow View, the flow chart can become too complicated to understand (e.g. in our badminton case). Experts praised our interactions for navigating the Sankey diagram and filtering the nodes, which “*greatly simplified the flow chart with the most important information preserved.*” 3) When many points exist in the Projection View, these points may be overlapped. Our system can globally scale the radius of all points automatically according to the number of points, in order to lighten the overlap between them.

**Design lessons learned.** Our glyph design process taught us an important lesson — when a local design is complex, keeping it consistent

with global design can help users recognize and remember encodings. We introduce three potential glyph designs for encoding a wealth of essential information about multivariate tactics and the contexts of usages. The experts preferred the design where chronology is consistent with the overall Sankey diagram, saying they are able to infer the meaning of each part and thus understand the complex design.

**Limitations and future work.** Our current design cannot support numerical data well. Racket sports involve some important numerical attributes, such as ball speed. Currently, we first quantize attributes like these into multiple ranges, where each is a discrete category, and then apply our system. Although this method is workable for analyzing numerical data in practice, it interrupts numerical accuracy (two similar values may be divided into two ranges). In the future, we plan to propose an effective measure for computing description length that considers both numerical data and categorical data. We also plan to extend the current design by integrating charts, which more intuitively visualize numerical data.

## 8 CONCLUSION

In this study, we present a visual analytics system to analyze the tactic progression in racket sports. The system models the racket sports data as multivariate event sequences and discovers the tactics by mining multiple non-overlapping multivariate patterns. Considering that these tactics are versatile and ever-changing, we propose an effective pattern mining algorithm. Based on the mined tactics, we propose a user interface to visually analyze progressions from one tactic to the next. The core view is a glyph-based Sankey diagram, where each glyph encodes a wealth of important information about a tactic and can further help users focus on the tactics that may have the most significant effect on a progression. The user interface also supports interactions that help users explore tactics in multiple levels of detail. Through two case studies on badminton and tennis and interviews with four domain experts, we demonstrate the usability and effectiveness of our system.

## ACKNOWLEDGMENTS

The work was supported by NSFC (62072400) and Zhejiang Provincial Natural Science Foundation (LR18F020001). We thank the Collaborative Innovation Center of Artificial Intelligence by MOE, and Zhejiang Provincial Government (ZJU). We also thank the anonymous reviewers for their helpful comments.

## REFERENCES

- [1] G. Andrienko, N. Andrienko, G. Anzer, P. Bauer, G. Budziak, G. Fuchs, D. Hecker, H. Weber, and S. Wrobel. Constructing spaces and times for tactical analysis in football. *IEEE Transactions on Visualization and Computer Graphics*, 2019. doi: [10.1109/TVCG.2019.2952129](https://doi.org/10.1109/TVCG.2019.2952129).
- [2] R. Bertens, J. Vreeken, and A. Siebes. Keeping it short and simple: Summarising complex event sequences with multivariate patterns. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 735–744, 2016. doi: [10.1145/2939672.2939761](https://doi.org/10.1145/2939672.2939761).
- [3] R. Borgo, J. Kehler, D. H. Chung, E. Maguire, R. S. Laramée, H. Hauser, M. Ward, and M. Chen. Glyph-based visualization: Foundations, design guidelines, techniques and applications. In *Eurographics (STARs)*, pp. 39–63, 2013. doi: [10.2312/conf/EG2013/stars/039-063](https://doi.org/10.2312/conf/EG2013/stars/039-063).
- [4] T. Calders, C. W. Günther, M. Pechenizkiy, and A. Rozinat. Using minimum description length for process mining. In *Proceedings of the ACM symposium on Applied Computing*, pp. 1451–1455, 2009. doi: [10.1145/1529282.1529606](https://doi.org/10.1145/1529282.1529606).
- [5] B. C. Cappers, P. N. Meessen, S. Etalle, and J. J. Van Wijk. Eventpad: Rapid malware analysis and reverse engineering using visual analytics. In *2018 IEEE Symposium on Visualization for Cyber Security (VizSec)*, pp. 1–8. IEEE, 2018. doi: [10.1109/VIZSEC.2018.8709230](https://doi.org/10.1109/VIZSEC.2018.8709230).
- [6] B. C. Cappers and J. J. van Wijk. Exploring multivariate event sequences using rules, aggregations, and selections. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):532–541, 2017. doi: [10.1109/TVCG.2017.2745278](https://doi.org/10.1109/TVCG.2017.2745278).
- [7] D. Cervone, A. D’Amour, L. Bornn, and K. Goldsberry. POINTWISE: Predicting points and valuing decisions in real time with NBA optical tracking data. In *Proceedings of the 8th MIT Sloan Sports Analytics Conference, Boston, MA, USA*, vol. 28, p. 3, 2014.
- [8] Y. Chen, P. Xu, and L. Ren. Sequence synopsis: Optimize visual summary of temporal event data. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):45–55, 2017. doi: [10.1109/TVCG.2017.2745083](https://doi.org/10.1109/TVCG.2017.2745083).
- [9] Y.-C. Chen, J.-C. Jiang, W.-C. Peng, and S.-Y. Lee. An efficient algorithm for mining time interval-based patterns in large database. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pp. 49–58, 2010. doi: [10.1145/1871437.1871448](https://doi.org/10.1145/1871437.1871448).
- [10] N. Chotisarn, L. Merino, X. Zheng, S. Lonapalawong, T. Zhang, M. Xu, and W. Chen. A systematic literature review of modern software visualization. *Journal of Visualization*, 23(4):539–558, 2020. doi: [10.1007/s12650-020-00647-w](https://doi.org/10.1007/s12650-020-00647-w).
- [11] X. Chu, X. Xie, S. Ye, H. Lu, H. Xiao, Z. Yuan, Z. Chen, H. Zhang, and Y. Wu. TIVEE: Visual exploration and explanation of badminton tactics in immersive visualizations. *IEEE Transactions on Visualization and Computer Graphics*, p. To appear, 2022.
- [12] D. H. Chung, P. Legg, M. Parry, I. Griffiths, R. Brown, R. Laramée, and M. Chen. Visual analytics for multivariate sorting of sport event data. In *Workshop on Sports Data Visualization*, vol. 3, 2013.
- [13] W. W. Cohen, P. Ravikumar, S. E. Fienberg, et al. A comparison of string distance metrics for name-matching tasks. In *IWeb*, vol. 3, pp. 73–78, 2003.
- [14] H. Dai, Y. Tao, and H. Lin. Visual analytics of urban transportation from a bike-sharing and taxi perspective. *Journal of Visualization*, 23(6):1053–1070, 2020. doi: [10.1007/s12650-020-00673-8](https://doi.org/10.1007/s12650-020-00673-8).
- [15] Z. Deng, D. Weng, J. Chen, R. Liu, Z. Wang, J. Bao, Y. Zheng, and Y. Wu. AirVis: Visual analytics of air pollution propagation. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):800–810, 2019. doi: [10.1109/TVCG.2019.2934670](https://doi.org/10.1109/TVCG.2019.2934670).
- [16] Z. Deng, D. Weng, Y. Liang, J. Bao, Y. Zheng, T. Schreck, M. Xu, and Y. Wu. Visual cascade analytics of large-scale spatiotemporal data. *IEEE Transactions on Visualization and Computer Graphics*, 2021. doi: [10.1109/TVCG.2021.3071387](https://doi.org/10.1109/TVCG.2021.3071387).
- [17] S. Di Bartolomeo, Y. Zhang, F. Sheng, and C. Dunne. Sequence braiding: Visual overviews of temporal event sequences and attributes. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1353–1363, 2020. doi: [10.1109/TVCG.2020.3030442](https://doi.org/10.1109/TVCG.2020.3030442).
- [18] V. Filipov, V. Schetinger, K. Raminger, N. Soursos, S. Zapke, and S. Miksch. Gone full circle: A radial approach to visualize event-based networks in digital humanities. *Visual Informatics*, 5(1):45–60, 2021. doi: [10.1016/j.visinf.2021.01.001](https://doi.org/10.1016/j.visinf.2021.01.001).
- [19] P. Fournier-Viger, J. C.-W. Lin, A. Gomariz, T. Gueniche, A. Soltani, Z. Deng, and H. T. Lam. The SPMF open-source data mining library version 2. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 36–40. Springer, 2016. doi: [10.1007/978-3-319-46131-1\\_8](https://doi.org/10.1007/978-3-319-46131-1_8).
- [20] P. Fournier-Viger, J. C.-W. Lin, R. U. Kiran, Y. S. Koh, and R. Thomas. A survey of sequential pattern mining. *Data Science and Pattern Recognition*, 1(1):54–77, 2017.
- [21] E. R. Gansner, E. Koutsofos, S. C. North, and K.-P. Vo. A technique for drawing directed graphs. *IEEE Transactions on Software Engineering*, 19(3):214–230, 1993. doi: [10.1109/32.221135](https://doi.org/10.1109/32.221135).
- [22] L. Giovannangeli, R. Bourqui, R. Giot, and D. Auber. Toward automatic comparison of visualization techniques: Application to graph visualization. *Visual Informatics*, 4(2):86–98, 2020. doi: [10.1016/j.visinf.2020.04.002](https://doi.org/10.1016/j.visinf.2020.04.002).
- [23] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–309, 2011. doi: [10.1177/1473871611416549](https://doi.org/10.1177/1473871611416549).
- [24] D. Gotz and H. Stavropoulos. DecisionFlow: Visual analytics for high-dimensional temporal event sequence data. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1783–1792, 2014. doi: [10.1109/TVCG.2014.2346682](https://doi.org/10.1109/TVCG.2014.2346682).
- [25] P. D. Grünwald and A. Grunwald. *The minimum description length principle*. MIT press, 2007.
- [26] 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*, 2020. doi: [10.1016/j.visinf.2020.04.001](https://doi.org/10.1016/j.visinf.2020.04.001).
- [27] S. Guo, Z. Jin, D. Gotz, F. Du, H. Zha, and N. Cao. Visual progression analysis of event sequence data. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):417–426, 2018. doi: [10.1109/TVCG.2018.2864885](https://doi.org/10.1109/TVCG.2018.2864885).
- [28] 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*, 24(1):56–65, 2017. doi: [10.1109/TVCG.2017.2745320](https://doi.org/10.1109/TVCG.2017.2745320).
- [29] Y. Guo, S. Guo, Z. Jin, S. Kaul, D. Gotz, and N. Cao. Survey on visual analysis of event sequence data. *arXiv preprint arXiv:2006.14291*, 2020.
- [30] T. Hu, H. Zheng, C. Liang, S. Zhu, N. Imirzian, Y. Zhang, C. Wang, D. P. Hughes, and D. Z. Chen. Antvis: a web-based visual analytics tool for exploring ant movement data. *Visual Informatics*, 4(1):58–70, 2020. doi: [10.1016/j.visinf.2020.02.001](https://doi.org/10.1016/j.visinf.2020.02.001).
- [31] Y. Ishikawa et al. TideGrapher: Visual analytics of tactical situations for rugby matches. *Visual Informatics*, 2(1):60–70, 2018. doi: [10.1016/j.visinf.2018.04.007](https://doi.org/10.1016/j.visinf.2018.04.007).
- [32] Z. Jin, N. Cao, Y. Shi, W. Wu, and Y. Wu. EcoLens: visual analysis of ecological regions in urban contexts using traffic data. *Journal of Visualization*, 24(2):349–364, 2021. doi: [10.1007/s12650-020-00707-1](https://doi.org/10.1007/s12650-020-00707-1).
- [33] Z. Jin, S. Guo, N. Chen, D. Weiskopf, D. Gotz, and N. Cao. Visual causality analysis of event sequence data. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1343–1352, 2020. doi: [10.1109/TVCG.2020.3030465](https://doi.org/10.1109/TVCG.2020.3030465).
- [34] K. Kawabata, Y. Matsubara, and Y. Sakurai. StreamScope: Automatic pattern discovery over data streams. In *Proceedings of the First International Workshop on Exploiting Artificial Intelligence Techniques for Data Management*, pp. 1–8, 2018. doi: [10.1145/3211954.3211959](https://doi.org/10.1145/3211954.3211959).
- [35] M. Krstajic, E. Bertini, and D. Keim. CloudLines: Compact display of event episodes in multiple time-series. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2432–2439, 2011. doi: [10.1109/TVCG.2011.179](https://doi.org/10.1109/TVCG.2011.179).
- [36] M. Lage, J. P. Ono, D. Cervone, J. Chiang, C. Dietrich, and C. T. Silva. StatCast dashboard: Exploration of spatiotemporal baseball data. *IEEE Computer Graphics and Applications*, 36(5):28–37, 2016. doi: [10.1109/MCG.2016.101](https://doi.org/10.1109/MCG.2016.101).
- [37] J. Lan, J. Wang, X. Shu, Z. Zhou, H. Zhang, and Y. Wu. RallyComparator: Visual comparison of the multivariate and spatial stroke sequence in a table tennis rally. *To appear in Journal of Visualization*, 2021.
- [38] Y. Li, T. Fujiwara, Y. K. Choi, K. K. Kim, and K.-L. Ma. A visual analytics system for multi-model comparison on clinical data predictions. *Visual Informatics*, 4(2):122–131, 2020. doi: [10.1016/j.visinf.2020.04.005](https://doi.org/10.1016/j.visinf.2020.04.005).
- [39] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu. Towards better analysis of deep convolutional neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):91–100, 2016. doi: [10.1109/TVCG.2016.2598831](https://doi.org/10.1109/TVCG.2016.2598831).
- [40] V. Menger. An experimental analysis of the pattern explosion. Master’s thesis, 2015. doi: [10.13140/RG.2.1.1324.4968](https://doi.org/10.13140/RG.2.1.1324.4968).
- [41] M. Monroe, R. Lan, H. Lee, C. Plaisant, and B. Shneiderman. Temporal event sequence simplification. *IEEE Transactions on Visu-*

- alization and Computer Graphics, 19(12):2227–2236, 2013. doi: [10.1109/TVCG.2013.200](https://doi.org/10.1109/TVCG.2013.200).
- [42] F. Mörchen and A. Ultsch. Efficient mining of understandable patterns from multivariate interval time series. *Data Mining and Knowledge Discovery*, 15(2):181–215, 2007. doi: [10.1007/s10618-007-0070-1](https://doi.org/10.1007/s10618-007-0070-1).
- [43] T. Munzner. *Visualization analysis and design*. CRC press, 2014. doi: [10.1201/b17511](https://doi.org/10.1201/b17511).
- [44] T. Oates and P. R. Cohen. Searching for structure in multiple streams of data. In *International Conference on Machine Learning*, vol. 96, pp. 346–354. Citeseer, 1996.
- [45] C. Perin, R. Vuillemot, and J.-D. Fekete. SoccerStories: A kick-off for visual soccer analysis. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2506–2515, 2013. doi: [10.1109/TVCG.2013.192](https://doi.org/10.1109/TVCG.2013.192).
- [46] C. Perin, R. Vuillemot, C. D. Stolper, J. T. Stasko, J. Wood, and S. Carpendale. State of the art of sports data visualization. In *Computer Graphics Forum*, vol. 37, pp. 663–686. Wiley Online Library, 2018. doi: [10.1111/cgf.13447](https://doi.org/10.1111/cgf.13447).
- [47] 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*, pp. 221–227, 1996. doi: [10.1145/238386.238493](https://doi.org/10.1145/238386.238493).
- [48] T. Polk, D. Jäckle, J. Häußler, and J. Yang. CourtTime: Generating actionable insights into tennis matches using visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):397–406, 2019. doi: [10.1109/TVCG.2019.2934243](https://doi.org/10.1109/TVCG.2019.2934243).
- [49] L. Shi, C. Huang, M. Liu, J. Yan, T. Jiang, Z. Tan, Y. Hu, W. Chen, and X. Zhang. Urbanmotion: Visual analysis of metropolitan-scale sparse trajectories. *IEEE Transactions on Visualization and Computer Graphics*, 2020. doi: [10.1109/TVCG.2020.2992200](https://doi.org/10.1109/TVCG.2020.2992200).
- [50] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of IEEE symposium on visual languages*, pp. 336–343, 1996. doi: [10.1109/VL.1996.545307](https://doi.org/10.1109/VL.1996.545307).
- [51] A. Siebes, J. Vreeken, and M. v. Leeuwen. Item sets that compress. In *Proceedings of the SIAM International Conference on Data Mining*, pp. 395–406, 2006. doi: [10.1137/1.9781611972764.35](https://doi.org/10.1137/1.9781611972764.35).
- [52] N. Tatti and B. Cule. Mining closed episodes with simultaneous events. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1172–1180, 2011. doi: [10.1145/2020408.2020589](https://doi.org/10.1145/2020408.2020589).
- [53] N. Tatti and J. Vreeken. The long and the short of it: summarising event sequences with serial episodes. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 462–470, 2012. doi: [10.1145/2339530.2339606](https://doi.org/10.1145/2339530.2339606).
- [54] R. Veras and C. Collins. Optimizing hierarchical visualizations with the minimum description length principle. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):631–640, 2016. doi: [10.1109/TVCG.2016.2598591](https://doi.org/10.1109/TVCG.2016.2598591).
- [55] J. Wang, J. Wu, A. Cao, Z. Zhou, H. Zhang, and Y. Wu. Tac-miner: Visual tactic mining for multiple table tennis matches. *IEEE Transactions on Visualization and Computer Graphics*, 27(6):2770–2782, 2021. doi: [10.1109/TVCG.2021.3074576](https://doi.org/10.1109/TVCG.2021.3074576).
- [56] J. Wang, K. Zhao, D. Deng, A. Cao, X. Xie, Z. Zhou, H. Zhang, and Y. Wu. Tac-Simur: Tactic-based simulative visual analytics of table tennis. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):407–417, 2019. doi: [10.1109/TVCG.2019.2934630](https://doi.org/10.1109/TVCG.2019.2934630).
- [57] D. Weng, C. Zheng, Z. Deng, M. Ma, J. Bao, Y. Zheng, M. Xu, and Y. Wu. Towards better bus networks: A visual analytics approach. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):817–827, 2021. doi: [10.1109/TVCG.2020.3030458](https://doi.org/10.1109/TVCG.2020.3030458).
- [58] 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*, pp. 1747–1756, 2011. doi: [10.1145/1978942.1979196](https://doi.org/10.1145/1978942.1979196).
- [59] C.-W. Wu, Y.-F. Lin, P. S. Yu, and V. S. Tseng. Mining high utility episodes in complex event sequences. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 536–544, 2013. doi: [10.1145/2487575.2487654](https://doi.org/10.1145/2487575.2487654).
- [60] J. Wu, Z. Guo, Z. Wang, Q. Xu, and Y. Wu. Visual analytics of multivariate event sequence data in racquet sports. In *2020 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 36–47. IEEE, 2020. doi: [10.1109/VAST50239.2020.00009](https://doi.org/10.1109/VAST50239.2020.00009).
- [61] Y. Wu, J. Lan, X. Shu, C. Ji, K. Zhao, J. Wang, and H. Zhang. iTTVis: Interactive visualization of table tennis data. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):709–718, 2017. doi: [10.1109/TVCG.2017.2744218](https://doi.org/10.1109/TVCG.2017.2744218).
- [62] Y. Wu, D. Weng, Z. Deng, J. Bao, M. Xu, Z. Wang, Y. Zheng, Z. Ding, and W. Chen. Towards better detection and analysis of massive spatiotemporal co-occurrence patterns. *IEEE Transactions on Intelligent Transportation Systems*, 22(6):3387–3402, 2021. doi: [10.1109/TITS.2020.2983226](https://doi.org/10.1109/TITS.2020.2983226).
- [63] X. Xie, J. Wang, H. Liang, D. Deng, S. Cheng, H. Zhang, W. Chen, and Y. Wu. PassVizor: Toward better understanding of the dynamics of soccer passes. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1322–1331, 2021. doi: [10.1109/TVCG.2020.3030359](https://doi.org/10.1109/TVCG.2020.3030359).
- [64] S. Ye, Z. Chen, X. Chu, Y. Wang, S. Fu, L. Shen, K. Zhou, and W. Yingcai. ShuttleSpace: Exploring and analyzing movement trajectory in immersive visualization. *IEEE transactions on visualization and computer graphics*, pp. 860–869, 2021. doi: [10.1109/TVCG.2020.3030392](https://doi.org/10.1109/TVCG.2020.3030392).
- [65] D. C. Zarate, P. Le Bodic, T. Dwyer, G. Gange, and P. Stuckey. Optimal sankey diagrams via integer programming. In *2018 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 135–139. IEEE. doi: [10.1109/PacificVis.2018.00025](https://doi.org/10.1109/PacificVis.2018.00025).
- [66] T. Zhang, Z. Chen, Z. Zhao, X. Luo, W. Zheng, and W. Chen. Fault-tracer: interactive visual exploration of fault propagation patterns in power grid simulation data. *Journal of Visualization*, pp. 1–14, 2021. doi: [10.1007/s12650-020-00741-z](https://doi.org/10.1007/s12650-020-00741-z).
- [67] J. Zhao, Z. Liu, M. Dontcheva, A. Hertzmann, and A. Wilson. MatrixWave: Visual comparison of event sequence data. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 259–268, 2015. doi: [10.1145/2702123.2702419](https://doi.org/10.1145/2702123.2702419).