

# RASIPAM: Interactive Pattern Mining of Multivariate Event Sequences in Racket Sports

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

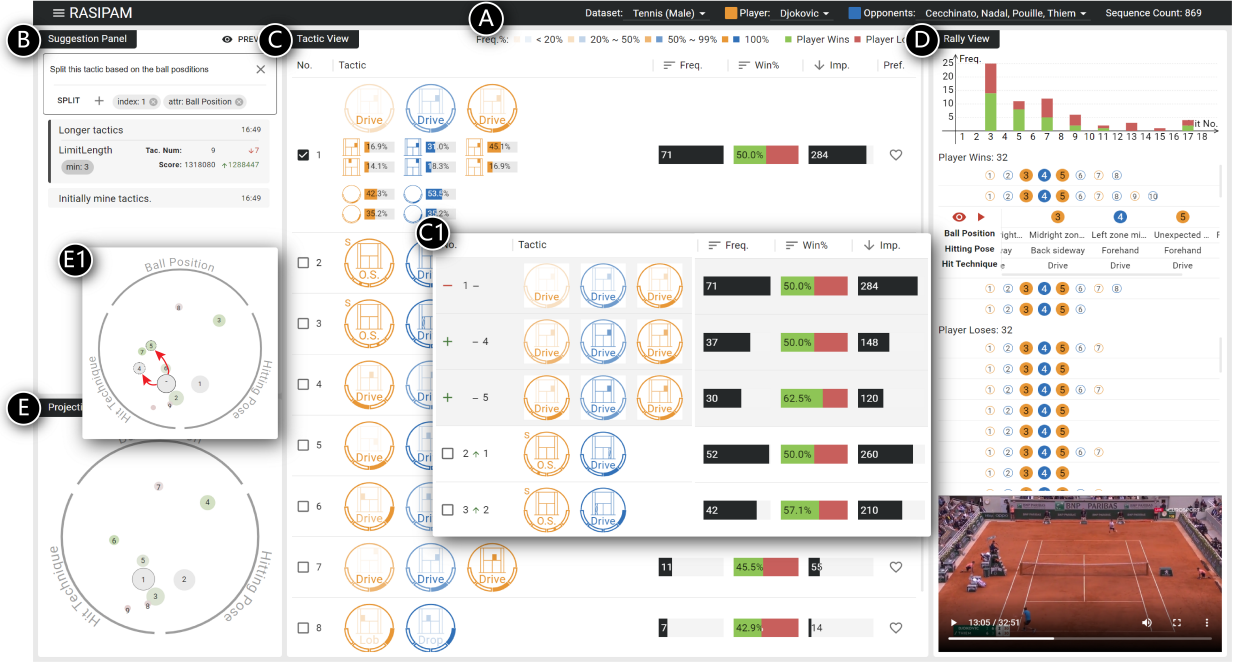


Fig. 1. A screenshot of our system for interactive tactic mining. Domain experts first need to select a player of interest on the control bar (A) to analyze his/her tactics. Then the system mines an initial set of tactics and visualizes them through intuitive glyphs in the Tactic View (C). Meanwhile, the Project View (E) provides an overview of the tactics through a semantic projection method, allowing domain experts to quickly understand the tactics. Experts are then allowed to give suggestions to improve the tactics based on their domain knowledge in the Suggestion Panel (B). The system will refine the tactic set and compare the new tactics with the original ones (C1 and E1). For detailed exploration, the Rally View (D) shows more details about a chosen tactic, including a bar chart displaying where the tactic is used, a list of all rallies involving the tactic, and a video player for the video segment of each rally.

**Abstract**— Experts in racket sports like tennis and badminton use tactical analysis to gain insight into competitors’ playing styles. Many data-driven methods apply pattern mining to racket sports data — which is often recorded as multivariate event sequences — to uncover sports tactics. However, tactics obtained in this way are often inconsistent with those deduced by experts through their domain knowledge, which can be confusing to those experts. This work introduces RASIPAM, a **R**acket-**S**ports **I**nteractive **P**attern **M**ining system, which allows experts to incorporate their knowledge into data mining algorithms to discover meaningful tactics interactively. RASIPAM consists of a constraint-based pattern mining algorithm that responds to the analysis demands of experts: Experts provide suggestions for finding tactics in intuitive written language, and these suggestions are translated into constraints to run the algorithm. RASIPAM further introduces a tailored visual interface that allows experts to compare the new tactics with the original ones and decide whether to apply a given adjustment. This interactive workflow iteratively progresses until experts are satisfied with all tactics. We conduct a quantitative experiment to show that our algorithm supports real-time interaction. Two case studies in tennis and in badminton respectively, each involving two domain experts, are conducted to show the effectiveness and usefulness of the system.

**Index Terms**—Sports Analytics, Multivariate Event Sequence, Interactive Pattern Mining, Comparative Visual Design.

## 1 INTRODUCTION

- J. Wu, Z. Guo, and Y. Wu are with the State Key Lab of CAD&CG, Zhejiang University. Y. Wu is also with the Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies. E-mail: {wujiang5521, ziyanguo27, ycwu}@zju.edu.cn. Y. Wu is the corresponding author.
- D. Liu is with MIT. E-mail: dongyu@mit.edu.

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxx/TVCG.201x.xxxxxx

Coaches and data analysts working in racket sports like tennis and badminton use tactical analysis to gain high-level insight into these sports, such as players’ preferred playing styles and competitive weaknesses [40]. In racket sports, a *rally* starts with one player serving the ball, progresses as two players *hit* the ball in alternation, and ends with one player scoring a point. For each hit, a player must consider multiple hit features such as the hitting technique and the ball position. By skillfully combining different hit features over several consecutive hits, a player may adopt a *tactic* for winning the game. Fig. 2 demonstrates an example in tennis – Player 1 (P1) may play a three-hit “serve-and-volley” tactic against Player 2 (P2). *Hit 1*: P1 serves the ball to the

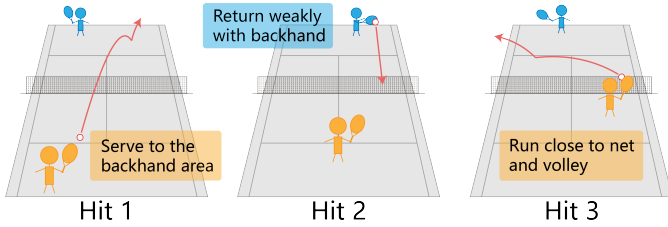


Fig. 2. An example of a tennis tactic with three hits — “serve-and-volley.”

backhand area (*ball position*) of P2; *Hit 2*: P2 hits a weak return via his/her backhand (*hitting pose predicted by P1*); *Hit 3*: P1 runs close to the net (*player position*), volleys the ball back (*hitting technique*), and win a point. To help experts quickly find such tactics, some data-driven methods [40, 46] model each hit as a multivariate event and each rally as an event sequence. By applying pattern mining algorithms, such methods can discover recurring patterns and analyze them as tactics.

However, domain experts’ knowledge can lead them to interpret the results of data-driven methods in specific ways. For instance, if an algorithm finds that P1 frequently performs the aforementioned “serve-and-volley” tactic, experts may interpret this finding differently based on what they know about P1. If P1 prefers net tactics, experts may want to focus on the third hit, where P1 runs close to the net to volley the ball. Otherwise, experts may want to merge tactics with a similar first hit, where P1 hits the ball to P2’s backhand, preventing P2 from returning the ball effectively. Experts use the domain knowledge gathered over the years to perform such fine-grained analysis. Lacking this domain knowledge, data-driven algorithms cannot support such further adjustments, such as merging tactics with specific similar hits. Thus, the mined tactics are inconsistent with what experts know, confusing and difficult to use [22, 42]. An experiment presented in Sect. 7.1.1 also suggested that few tactics expected by experts could be directly and precisely mined by data-driven algorithms. Experts would prefer to work interactively with algorithms, giving their knowledge-based suggestions so that the algorithm’s findings better meet their needs.

However, although interactive pattern mining is well-studied [3, 4, 23, 28], it is still challenging to apply it to multivariate tactics. One issue is **how to handle experts’ fine-grained suggestions on multivariate tactics**. Experts may give meticulous feedback about the algorithm’s treatment of a single hit feature because small details may determine the outcome of a highly competitive game. To the best of our knowledge, no work has studied interactive pattern mining that includes such fine-grained adjustments. Another issue is **how to ensure experts can intuitively evaluate the adjusted tactics**. After the algorithm adjusts the tactics, experts need to explore the adjusted tactics and judge whether the adjustment is reasonable. However, because multivariate tactics involve so many hit features of many hits, understanding an adjustment that may involve changes among many tactics (e.g., merging several similar tactics) is extremely difficult and time-consuming. Intuitive methods are required to help experts quickly verify the results.

To solve these issues, we propose RASIPAM, a **R**acket-Sports **I**nteractive **P**attern **M**ining system. To tackle the first challenge, we **propose a constraint-based pattern mining algorithm**. We conduct a pre-study to collect experts’ suggestions and delimit the space where suggestions occur. We further define a set of natural language templates to transfer experts’ suggestions into math representations, i.e., constraints. The algorithm can generate candidate patterns based on these constraints and then use a data-driven metric to select the best candidates for experts, combining domain knowledge with a data-driven method. To address the second challenge, we **propose an interactive visual interface** with a comparative visual design. A semantic-based projection view and a glyph design for visualizing multivariate hit are proposed to allow experts to evaluate the adjustment by comparing the new tactics with the original ones, with multiple levels of detail [34].

Finally, we present two quantitative experiments evaluating the quality of the adjusted tactics and the speed of the algorithm, which demonstrate our algorithm’s effectiveness, efficiency, and necessity. We also evaluate the usability of our system through two case studies, performed with two experts each from tennis and badminton, which prove that our system can effectively incorporate expert knowledge to interactively

mine meaningful tactics for domain experts.

In summary, this work has the following contributions:

- A design study in close collaboration with experts in racket sports, which helps to carve the domain requirements on interactive tactic mining and summarize nine potential adjustments to tactics.
- A human steerable multivariate pattern mining algorithm that allows fine-grained adjustment of patterns based on experts’ suggestions.
- A user interface that contains a natural language interface to help experts quickly adjust tactics and a comparative visual design to help experts evaluate the adjusted tactics.
- Two quantitative experiments that evaluate the effectiveness and efficiency of the algorithm, and two case studies on tennis and badminton that demonstrate our system’s usefulness.

## 2 RELATED WORK

### 2.1 Interactive Pattern Mining for Event Sequences

Methods based on Sequential Pattern Mining (SPM) find all subsequences whose frequency is higher than a threshold [12]. Due to the well-known problem of *pattern explosion* [27], users usually need to adjust the threshold manually and constantly to obtain a suitable number of patterns, where interactive mining algorithms are needed [1]. Many SPM-based algorithms [21, 30, 31] cached mined patterns while incrementally mining new patterns. Wojciechowski et al. [43] extended the adjustment to the number of values/events in each pattern. However, SPM-based methods may generate similar and repetitive patterns that are frequent but meaningless for tactical analysis [27].

Interactive machine learning (IML) has been much-discussed recently because it incorporates user feedback into model computations, enhancing the interpretability of the results [10]. ProtoSteer [28] proposed a tailored interactive pattern mining system based on a deep sequence model [29]. However, training an effective model for mining racket sports tactics is difficult, because players have unique tactics and continually change them as needed to win against particular opponents.

Over the past two decades, algorithms based on the Minimum Description Length (MDL) principle have emerged [16]. Unlike SPM-based methods, MDL-based methods find a small set of optimized patterns instead of surfacing all the patterns that appear frequently. In addition, unlike IML-based methods, MDL-based methods do not rely on training data. Meanwhile, many MDL-based methods have investigated how to incorporate domain knowledge into pattern mining. Tatti et al. [36] allowed users to define constraints to support interactive analysis. One-Click Mining [4] could generate new patterns based on users’ binary feedback (like/dislike) about current patterns. However, to the best of our knowledge, no MDL-based methods have studied interactive mining of multivariate event sequences. Our work proposes an MDL-based algorithm for mining multivariate patterns interactively.

### 2.2 Visual Analytics of Multivariate Event Sequences

A comprehensive survey by Guo et al. [17] summarizes a wealth of research on visual analytics of event sequences, which considers multivariate event sequence visualization as a challenging and promising research direction. Some visualizations directly displayed high-dimensional data in detail as matrices [24] and lines [19]. Pattern-Finder [11] and EventPad [6, 7] propose novel interfaces to query multivariate sequential patterns. Wang et al. [38, 39] applied causality analysis to multivariate event sequence data. Many researchers have applied visual analytics to sports data because it is a common type of multivariate event sequence data. Wu et al. [47] used a sequence of matrices to visualize intra-event relationships (i.e., among multiple attributes) as well as inter-event ones. Tac-Miner [40] and Tac-Simur [41] designed intuitive glyphs to encode multivariate events in table tennis, which inspired our work to design glyphs tailored for tennis and badminton. Based on the glyph design, our work mainly contributes to a comparative visual design for comparing the adjusted tactics with the original ones, allowing experts to evaluate their suggested adjustments.

### 2.3 Visual Analytics of Sports Tactics

Tactics are sets of high-level decisions made by players, which can significantly affect the outcome of games [18], leading to experts’ needs for visual analytics [9, 35, 44]. In team sports like soccer, tactics

are mainly related to the team members’ cooperation [5], such as the formation changes [48] and multiple players’ movement coordination [2, 49]. In racket sports, tactical analysis mainly focuses on correlations among consecutive hits [8, 20]. Wang et al. [40] and Wu et al. [45, 46] proposed data-driven algorithms for mining multivariate tactics. These algorithms analyze tactics in a data-driven way by proposing novel models to discover insights automatically for users. Our work further incorporates experts’ knowledge into the algorithm to mine meaningful tactics. We mainly contribute to a pre-study that collects hundreds of experts’ potential suggestions on tactics and a human steerable tactic mining algorithm that can fine-tune tactics based on these suggestions.

### 3 DESIGN STUDY

This section presents a design study for summarizing the analysis tasks involved with interactive tactical analysis, and proposes the analysis workflow and system architecture for performing these tasks.

#### 3.1 Setup and Study Process

Our study follows the design study methodology of Sedlmair et al. [32].

**Experts.** We collaborated with five domain experts in three racket sports — tennis (E1, E2), badminton (E3, E4), and table tennis (E1, E5) — *E1 focuses on table tennis but is also proficient in tennis*. All experts were professional athletes and are now sports science data analysts with more than five years of experience.

**Process.** Over the past year, we held weekly meetings with these five experts to discuss interactive tactical analysis in three steps.

*Step 1:* Given that each sport has domain-specific analysis methods, we interviewed experts separately according to their focal sports. For each sport, we first confirmed the need for interactive tactical analysis. We then gathered each expert’s concerns about and suggestions for interactive tactical analysis. Each interview lasted about one hour.

*Step 2:* By summarizing the experts’ feedback, we further formalized the analysis tasks for all racket sports (**T1-T5** in Sect. 3.2), which were further discussed and improved in three meetings with all the experts. Due to commonalities between all racket sports, most experts’ concerns could be covered by the same task across the different sports (**T1-T4**). However, experts from different sports cared about different details. For example, E5 preferred to know the frequency of tactics at the hit-by-hit level because rallies in table tennis are usually short, meaning every hit matters. But E4 did not care about this because of the long rallies in badminton. We covered all these detailed concerns with **T5**.

*Step 3:* We developed a prototype system for domain experts, along with real use cases, to further gather their feedback on the analysis tasks and the system design. After two months of polishing, we finalized our analysis tasks and obtained the current version of the system.

#### 3.2 Task Analysis

The main tasks our system needs to perform during interactive tactical analysis are as follows, where **T1-T4** are common to all racket sports.

- T1 Get an overview of an individual player’s tactics.** Experts prefer player-centric tactical analysis, because each player uses unique tactics. Rather than analyzing each tactic of a player directly, experts expect to start with an overview to learn things such as how many tactics result in more wins, and whether the player specializes in a few tactics or adopts more tactics to confuse opponents.
- T2 Adjust mined tactics according to experts’ suggestions.** Experts expect to give knowledge-based suggestions to the algorithm about how to better mine tactics. For example, the “seesaw battle” tactic (i.e., *both players hitting the ball with a drive technique*) is common in tennis games, and is easily detected by the algorithm. However, experts may argue that the ball position determines the outcome of a seesaw battle and require our algorithm to subdivide this tactic into several subtactics based on different ball position changes. Because of the low frequency of each subdivided tactic, a data-driven algorithm usually fails to directly discover them.
- T3 Evaluate the adjusted results.** Experts’ knowledge comes from past games, and may lose usefulness as a player’s tactics evolve over time. Just as the experts suggest adjustments, experts expect our system to evaluate their suggestions — *providing data-driven metrics and comparing adjusted tactics with the original ones* — to



Fig. 3. Our system architecture for interactive tactical analysis.

prevent them from making inappropriate adjustments. For example, when experts find a tactic not typical of a certain player and suggest removing it, our system could provide data indicating that this may be a newly developed tactic rather than anomaly, thus reminding experts to exercise caution before discounting it.

- T4 Discover similar tactics.** Players often achieve different tactical goals (e.g., confusing their opponents) by changing a few hit features of a previous tactic, resulting in a slightly different tactic. Finding these tactics is valuable for tactical analysis because it helps experts study which changes are effective. Meanwhile, merging similar tactics that actually lead to similar outcomes can help experts avoid spending their analysis time unnecessarily.

- T5 Display raw sequences in detail.** The raw sequences can also include details about the context of a tactic, i.e. the previous and subsequent hits, which indicate the impetus for and the results of using the tactic, respectively. Experts expressed a desire to explore the raw sequences by viewing statistics (E1, E5), observing the detailed hit features of each hit (E1 - E5), and watching videos (E1 - E4), in order to obtain a deep understanding of the tactic.

#### 3.3 Analysis Workflow

We propose an analysis workflow based on a human-in-the-loop architecture (Fig. 3) to address the tasks delineated above, which are implemented through an open-source prototype system<sup>1</sup> consisting of a database, a backend, and a frontend. The **database** stores the raw sequences from different racket sports (Sect. 4.1). The **backend** runs a mining algorithm (Sect. 5) to discover a set of tactics that satisfy the constraints suggested by experts (Sect. 4.2). The **frontend** visualizes the tactics for experts and collects their knowledge-based suggestions in return (Sect. 6). Our workflow starts when the backend mines an initial tactic set from the dataset, which are then visualized by the frontend (**T1, T4, T5**). Next, experts explore the visualizations and suggest our algorithm to adjust these tactics (**T2**). Our system provides evaluation of the adjusted results (**T3**), helping experts preview the results and determine whether to apply the adjustment. Experts can give suggestions iteratively until they have no more suggestions.

### 4 DATA MODEL

In this section, we introduce the data structures of the raw sequences and tactics. We then delimit the space of domain-specific constraints, and describe the pre-study we performed to obtain those constraints.

#### 4.1 Data Structure

**Raw Sequences:** A dataset consists of  $n$  raw sequences, denoted  $S = \{s_1, s_2, \dots, s_n\}$ . Each sequence  $s = \{e_1, e_2, \dots, e_l\}$  represents a rally with  $l$  hits, with the player who served the ball and the player who won the point as metadata. Each event  $e = \{v_1, v_2, \dots, v_k\}$  represents a hit with  $k$  categorical values indicating multiple hit features.

**Tactics:** At the beginning of the process, and again after each adjustment, our algorithm outputs a tactic set  $T = \{t_1, t_2, \dots, t_m\}$  with  $m$  tactics. Each tactic  $t_i$  is defined as a multivariate subsequence shared by many rallies, which is *value-nullable* and *consecutive*. For example, in Fig. 5, tactic  $t$  is applied in both  $s_1$  and  $s_2$  (highlighted by blue). Tactic  $t$  consists of four *single values* from  $sv_1$  to  $sv_4$  (each of which is a non-null value) and two null values (meaning that the corresponding values are different in  $s_1$  and  $s_2$ ). Meanwhile, in  $s_1$  and  $s_2$ ,  $t$  is always used consecutively without any hits in between. In addition, each tactic records the reference to the rallies and the hits where the tactic was applied as the metadata.

<sup>1</sup><https://anonymous.4open.science/r/Rasipam-A695/>. Only synthetic data exists due to the confidentiality of real-world datasets.



Type	No.	Constraint	Suggestion Type	Suggestion Example
global	c1	$\text{idx\_con}(t_i) = \text{out-of-range freq. \%}$	set tactic index	Only show serving tactics
	c2	$\text{len\_con}(t_i) = 0 \text{ if len}(t_i) \text{ in range else } 1$	set tactic length	All tactics should be longer
	c3	$\text{imp}(v_i) = c, \text{ where } -1 \leq c \leq 1$	set variable importance	Pay more attention to the hitting technique
local	c4	$t_i \rightarrow \emptyset$	ignore a tactic	This tactic has no meaning
	c5	$t_i \rightarrow \text{sub}(t_i, \text{values})$	specify a tactic	Ball position is important for this seesaw battle
	c6	$t_i \rightarrow \text{sup}(t_i, \text{values})$	refine tactics	Merge these two similar tactics
	c7	$t_i \rightarrow \text{exp}(t_i, \text{direction})$	expand a tactic	The follow-up changes for this tactic are important
	c8	$t_i \rightarrow \text{trim}(t_i, \text{direction})$	trim a tactic	The first hit of this tactic does not matter
	c9	$t_i \rightarrow \text{rep}(t_i, \text{original value, target value})$	replace value	I expect the player to drive the ball at the third hit

Fig. 4. The constraint space with nine constraints, each of which supports one type of suggestion, demonstrated through an example given by experts. Three global constraints satisfy experts’ limitations for the overall tactic set, while six local ones help experts fine-tune specific tactics.

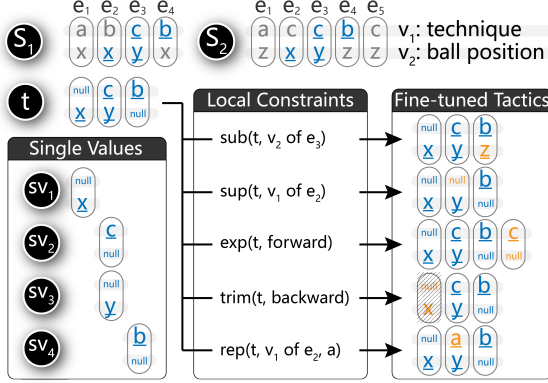


Fig. 5. Sequences  $s_1$  and  $s_2$  both involve two hit features, where tactic  $t$  is used (highlighted in blue). We further present how each local constraint affects the tactic (highlighted in orange).

## 4.2 Constraints

Many previous works have studied generic constraints for mining univariate patterns [36, 43]. However, for the multivariate tactics in racket sports, we need fine-grained and domain-specific constraints to satisfy experts’ suggestions. Thus, we conducted a pre-study with the five experts (E1-E5) to find all the constraints needed, i.e., the constraint space. The process of this study and its results are as follows.

### 4.2.1 Study Process

The study was divided into three stages.

**Preparation Stage:** We prepared a real tactic set for the experts to spur the suggestions that may arise in real analysis scenarios, which contains 107/102/95 tactics mined from 3,052/3,123/2,997 rallies played between top players in tennis/badminton/table tennis respectively, by an open-source tactic mining algorithm [46]. We also prepared an initial constraint space from the perspective of data mining, with several example suggestions for each constraint for experts reference.

**Collection Phase:** The five experts were invited to give suggestions on our prepared tactics – each tactic was reviewed by at least one expert. We first showed them the example suggestions and encouraged them to give suggestions beyond the examples, as long as the suggestions fit their actual needs. Their suggestions and the tactics they expected were recorded for further analysis.

**Summarization Phase:** We filtered 297 different suggestions from the 349 ones collected and classified them into nine suggestion types (Fig. 4). We further expanded the constraint space to cover all these suggestions and held discussions with experts to ensure comprehensive support of possible suggestions from experts.

### 4.2.2 Results

We list the final constraints and the supported suggestion types in Fig. 4 and elaborate how we apply each constraint in Sect. 5.2. Due to space limitations, we will give a brief introduction here and present

more details in the appendix. We classify the constraints into two types according to their effects.

**Global constraints** affect the whole tactic set, leading to an adjusted tactic set that is completely different from the original one (thus, **T3** is not applicable). Taking c1 in Fig. 4 as an example, experts may only want to analyze serving tactics (*the expert’s suggestion*). Rather than just filtering a few serving tactics from the original tactic set, the system should mine a new set of serving tactics for detailed analysis (*limiting the index of the tactics’ first hit to 1*). To mine only serving tactics, our algorithm will add a loss term  $\text{idx\_con}(t_i)$  to an objective function, giving a high cost to other tactics (*the constraint, which is elaborated in Sect. 5.2*).

**Local constraints** require fine-tuning of a few tactics, without changing other tactics. Taking c6 in Fig. 4 as an example, experts may suggest merging two similar tactics. The system should then refine the original two tactics to a super-tactic that contains all common values in the tactics to be merged. In this case, our algorithm will add a constraint “ $t_i \rightarrow \text{sup}(t_i, \text{values})$ ” to each original tactic, which means replacing the original tactic  $t_i$  with the super-tactic  $\text{sup}(t_i, \text{values})$ , where *values* represent the common values in the two original tactics.

## 5 INTERACTIVE TACTIC MINING ALGORITHM

We propose an interactive multivariate pattern mining algorithm to meet experts’ tactic-mining needs in three scenarios as follows (Fig. 6).

- S1** Initially, the algorithm needs to mine an initial tactic set  $T$  from the rallies dataset  $S$ , saving experts’ time needed to discover tactics manually (indicated by purple arrows). Given the dataset  $S$  and the current tactic set  $T$ , experts may iteratively suggest constraints to adjust  $T$ , leading to the next two scenarios.
- S2** To satisfy the global constraints, our algorithm needs to re-mine a new tactic set  $T_{\text{new}}$  from dataset  $S$  (indicated by green arrows).
- S3** To satisfy experts’ local constraints, our algorithm needs to fine-tune several specific tactics in  $T$ , while leaving other tactics unchanged (indicated by red arrows).

To accomplish this, we extend the existing algorithm of TacticFlow [46], which can mine an initial tactic set (for **S1**). The main contributions of our algorithm lie in processing global and local constraints to enable interactive tactic mining (for **S2** and **S3**).

### 5.1 Initializing Tactic Set

The algorithm receives a raw sequence dataset  $S$  as input and tries to mine a set of initial tactics  $T$  from  $S$  as output. The mining process is based on the MDL principle, which regards  $T$  as a model to describe  $S$  and regards the model that results in the minimum description length (i.e., a metric for computing the information cost [16]) as the best option. The algorithm consists of a generator and an optimizer. The generator randomly combines two tactics in the tactic set at different alignments to generate new candidates, starting from combining single values, such as the *drive* technique, the *backcourt* ball position, and so on. Every time the generator generates a candidate, the optimizer uses a domain-specific metric to compute the description length  $L(S, T)$  (Eq. 1), which mainly considers the number of tactics  $|T|$ , the frequency of each tactic  $\text{freq}(t_i)$ , and the single values  $sv(s_i)$  that cannot be described by the

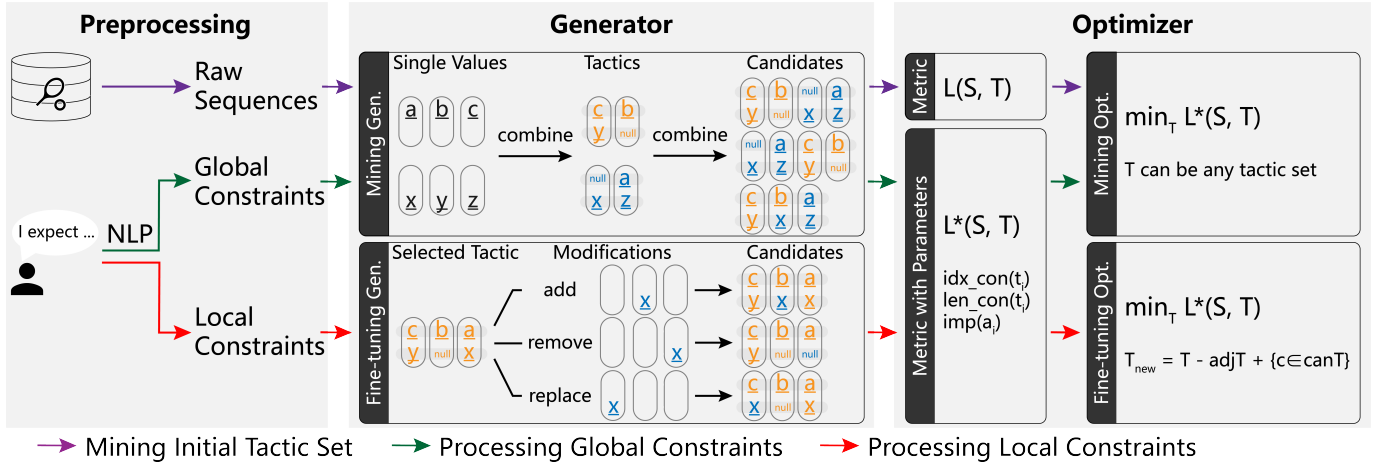


Fig. 6. Algorithm framework for mining tactics in three scenarios. **Purple arrows**: the algorithm can mine an initial tactic set from raw sequences. **Green arrows**: the algorithm proposes several parameters to process global constraints and re-mine the tactic set. **Red arrows**: the algorithm proposes a fine-tuning generator and a fine-tuning optimizer to fine-tune specific tactics based on local constraints.

tactics in each sequence. Constants  $\alpha$  and  $\beta$  control the weight of each term to ensure that each term matters, considering that experts generally prefer to analyze about 20 tactics (making  $|T|$  small), which are used many times to describe hundreds of sequences (making  $freq(t_i)$  large) and leave thousands of single values (making  $sv(s_i)$  very large). The optimizer further optimizes the tactic set by adding candidates that can reduce the description length into the tactic set and removing tactics that cannot benefit from optimizing the description length from the tactic set. The generator and optimizer work iteratively until the tactic set does not change in a certain iteration, whereupon the algorithm obtains a tactic set with the appropriate minimum description length.

$$L(S, T) = |T| + \alpha \sum_{t_i \in T} freq(t_i) + \beta \sum_{s_i \in S} sv(s_i), \quad (1)$$

## 5.2 Processing Constraints

Our algorithm contributes to two workflows — one each for processing global and local constraints. Global constraints require changing the entire tactic set, re-defining what kind of tactic set is the best (i.e., the one expected by experts). Following the MDL principle, which associates the best model with the minimum description length, we change the metric (“Metric with Parameters” in Fig. 6) to discover the new tactic set regarded as the best. In contrast, local constraints require the algorithm to fine-tune specific tactics, leaving others unchanged. We propose a new generator and optimizer for fine-tuning (“Fine-tuning Gen.” and “Fine-tuning Opt.” in Fig. 6).

### 5.2.1 Global Constraints

We introduce a new metric  $L^*(S, T)$  (Eq. 2) to fit the global constraints. The new metric adds three loss terms, giving a high cost to the tactics that do not fit the global constraints so that the optimizer prefers to remove them. This mainly involves three parameters as follows.

- **c1**:  $idx\_con(t_i)$ . For each tactic  $t_i$  in  $T$ , we find the usages at in-range indexes (e.g., 1-4 for serving tactics) and out-of-range ones and calculate the percentage of out-of-range usages as  $idx\_con(t_i)$ . We multiply the frequency of each tactic with  $idx\_con(t_i)$  to give a high cost for out-of-range usages.
- **c2**:  $len\_con(t_i)$ . For each tactic  $t_i$ ,  $len\_con(t_i)$  represents whether the length of  $t_i$  occurs in a range expected by experts (e.g.,  $> 3$  for tactics expected to include more than three hits) — 0 means in range, and 1 means out of range. We multiply the frequency of each tactic with  $len\_con(t_i)$  to give a high cost for tactics with out-of-range length.
- **c3**:  $imp(v_i)$ . We denote the importance of each hit feature value as  $imp(v_i)$ , ranging from -1 to 1 and defaulting to 0. We apply c3 by multiplying the number of single values with the importance of the corresponding hit feature, giving a high cost to any single values that cannot be described by tactics but are regarded important by experts.

To make our metric  $L^*(S, T)$  comparable with  $L(S, T)$ , we directly use the same parameters alpha and beta as TacticFlow to control the weight of each term. Note that, initially with no global constraints,  $idx\_con(t_i) = len\_con(t_i) = imp(v_i) = 0$ , leading to  $L^*(S, T) = L(S, T)$ .

$$L^*(S, T) = L(S, T) + \alpha \sum_{t_i \in T} freq(t_i) \times idx\_con(t_i) + \alpha \sum_{t_i \in T} freq(t_i) \times len\_con(t_i) + \beta \sum_{s_i \in S} sv(v_i, s_i) \times imp(v_i), \quad (2)$$

### 5.2.2 Local Constraints

We propose a modification-minimized generator to generate new candidates that satisfy the local constraints based on a tactic  $t$ . Meanwhile, we propose a local sensitive optimizer to fine-tune certain aspects of the tactic set while leaving other tactics unchanged.

**Generator.** Based on the tactic to adjust, the generator uses a BFS algorithm to search for all candidates that satisfy the local constraints (Fig. 5), applying a minimum number of local modifications. Each modification (shown as “Fine-tuning Gen.” in Fig. 6) consists of a target value to change and an action selected from *add*, *remove*, and *replace*. For example, to trim the last hit with two non-null values from a tactic (c8 in Fig. 4), the generator will generate one candidate by applying two modifications — removing the first value and the second value in the last hit, respectively. As another example, to expand a tactic with one more hit (c7 in Fig. 4), the generator will apply only one modification — adding one value to the back of the tactic. But there usually exist numerous values that can be added, which lead to many candidate tactics. In this way, the generator accurately generates candidates that meet experts’ expectations with the minimum modifications, making it easy for experts to compare the new tactics with the original ones.

**Optimizer.** The fine-tuning optimizer is local sensitive, which only replaces the tactics suggested to adjust with the generated candidates that can minimize the description length, keeping other tactics unchanged (“Fine-tuning Opt.” in Fig. 6). Formally, the optimizer tries to find the best tactic set  $T_{new} = T - adjT + \{c \in canT\}$ , where  $T$  is the tactic set to adjust,  $adjT$  is the tactics experts would like to adjust, and  $canT$  is all the candidates generated by the generator. The optimizer performs in three steps as follows (Alg. 1).

1. **Line 1**: The algorithm removes all tactics that experts have suggested adjusting from the original tactic set.
2. **Line 2-4**: The algorithm iterates all the candidates, with high-frequency ones first. If a candidate facilitates reducing the description length, it will be added to the tactic set. The optimizer keeps the same metric  $L^*(S, T)$  used when applying global constraints so that the description length is comparable.
3. **Line 5-7**: Every time the algorithm adds a candidate, it prunes the tactic set by removing redundant candidates (i.e., those candidates

whose inclusion leads to a larger description length), leaving other tactics not related to the constraints unchanged.

There exists a trade-off for interactive analysis — we ensure that at least one candidate will be added to the tactic set (line 3 and line 5), satisfying experts’ needs for adjustments, while the description length may be larger than the original tactic set. A small increase is acceptable for the trade-off, but a large one may indicate that experts have made an inappropriate adjustment, which should be prevented. Thus, we provide experts with data-driven metrics for scoring a tactic set and evaluating the tactical importance of each tactic. We score a tactic set by  $score(S, T) = L(S, \emptyset) - L(S, T)$  to provide experts with a more intuitive evaluation on a tactic set than  $L(S, T)$  because  $score(S, T)$  is usually much smaller than  $L(S, T)$  (hence easy to read and compare) and is positively correlated to the quality of the tactic set (i.e., a higher score means a better tactic set). Similarly, we evaluate the tactical importance of a tactic by  $tac\_imp(t) = L(S, T) - L(S, T - t)$ , where tactics with higher tactical importance are more valuable.

## 6 VISUAL INTERFACE

Our proposed visual interface consists of a control bar (Fig. 1A) and four main views (Fig. 1B-E). This section introduces the visualizations and interactions in the order our workflow progresses. Experts can start using the system by specifying the dataset, the **player** of interest, and the **opponents** on the control bar. The system spends seconds filtering the rallies played by the player against the opponents from the dataset and running the algorithm to obtain an initial tactic set. Experts can explore these tactics in Projection View (Fig. 1E), where we reveal the similarity among tactics (**T1**, **T4**), and Tactic View (Fig. 1C), where we list all tactics (**T1**). When they find tactics that should be adjusted, they can give suggestions in written language in the Suggestion Panel (Fig. 1B) and preview the results (**T2**). Projection View and Tactic View will then enable a preview mode to compare the new tactics with the original ones and present data-driven evaluations of the adjustment (**T3**). During the entire workflow, experts can view the rallies involving a tactic of interest in Rally View (Fig. 1D) for detailed exploration (**T5**). The visual designs in each view are as follows.

### 6.1 Projection View

Projection View (Fig. 1E) projects the tactics onto a 2-D plane, helping experts overview tactics and reveal their similarity (**T1**, **T4**).

**Projection.** Experts can get an overview of all tactics from the overall layout. To enable this broad understanding and get across the effects of adjustments, the projection algorithm is essential. After each adjustment, the system needs to re-run the projection algorithm to account for new tactics, where the position changes reveal the correlation between the new tactics and the original ones. Expert expect other tactics to keep stationary to avoid visual distraction. While t-SNE is a useful projection algorithm for visualizing data [14, 37], it is not suitable for our system, because the results are not stable after each run.

Instead, we propose a semantic projection method based on a polar coordinate system, which can generate a fixed position for each tactic based on semantics (Fig. 7). For each tactic, the polar angle encodes the top-two relevant attributes (i.e., the two attributes with the most

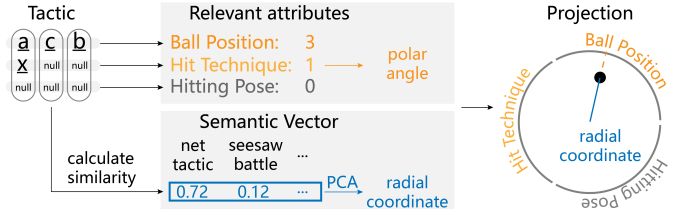


Fig. 7. An example that illustrates our proposed semantic-based projection, which mainly considers the two most relevant attributes and the basis tactics it resembles.

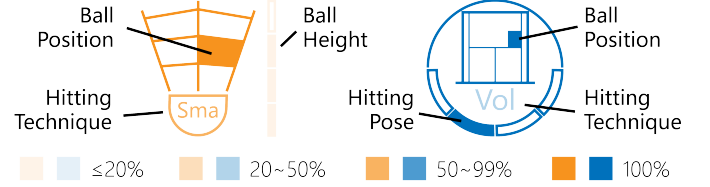


Fig. 8. The glyphs for badminton (left) and tennis (right), consisting of multiple components encoding detailed hit features. For each null value, we apply a multi-level opacity encoding for the frequency percentage of the value with highest frequency.

non-null values) so that experts can analyze a specific tactic type by exploring the corresponding direction. Meanwhile, we adopt PCA, which can generate fixed projection results, to project each tactic to a 1-D coordinate system as the radial coordinate. However, tactics vary in length and contain categorical values that are difficult to quantize (e.g., the hitting technique), which cannot be used as inputs for PCA. Thus, we characterize a tactic as a semantic vector based on several basis tactics, constructed in three steps: 1) We required experts to provide ten different typical tactics, such as seesaw battles and net tactics, to be used as basis tactics baked in our system. 2) We calculate the Levenshtein distance of the tactic at hand from each basis tactic to represent their similarity, following TacticFlow. 3) We construct the vector with the similarities in order and normalize it.

**Encodings of each point.** Experts prefer to quickly find the tactics worthy of analyzing (i.e. with high frequency or high tactical importance) and then evaluate the effectiveness of these tactics (i.e., the win rate). All three variables are important, but encoding them simultaneously can lead to severe visual clutter, especially when there exist many points. Thus, we encode either the frequency or the importance with the size of each point, allowing users to manually toggle in a settings panel. The color from green to red encodes the win rate from 100% to 0%. When experts find a tactic of interest, they can turn to Tactic View for further exploration in two ways. 1) We display the ranking numbers of the top-ten tactics in Tactic View on the corresponding points. 2) Hovering over a point can highlight the tactic in Tactic View.

### 6.2 Tactic View

Tactic View (Fig. 1C) lists all tactics (**T1**). Each row represents a tactic and contains six columns as follows.

**Ranking Columns** (Freq., Win%, Imp.). Experts may start by ranking the list to find the most valuable tactics. We use lineups [15] to provide multiple rankings based on the frequency (Freq.), the win rate (Win%), and the tactical importance (Imp.), visualized by the three bars. Experts can select one of the three to satisfy different analysis needs.

**Tactic Column** (Tactic). After selecting a suitable ranking, experts may scan the tactic list top-down and explore the tactics one by one. Each hit in each tactic is encoded by a tailored glyph, which aggregates the corresponding hit in the raw rallies where this tactic was used, with the color encoding the player who hits the ball. The glyph is based on a metaphorical icon, such as the shuttlecock and the tennis ball (Fig. 8), familiar and intuitive to experts [26, 40, 45, 47, 50, 51]. Each glyph encodes a multivariate hit with multiple non-overlapping components, each encoding a hit feature. For example, in the tennis glyph in Fig. 8, the blue block on the tennis court encodes the position where the ball bounces on the ground. All detailed designs are in the appendix.

---

#### Algorithm 1: Fine-tuning Optimizer

---

**Input:** Sequence set  $S$ , current tactic set  $T$ ,  
tactics to adjust  $adjT$ , candidate tactics  $canT$

**Output:** New tactic set  $T_{new}$

```

1  $T_{new} = T - adjT$ 
2 for  $ct \leftarrow canT$  in order of frequency do
3   if  $L^*(S, T_{new} + ct) < L^*(S, T_{new})$  or  $canT \cap T_{new} \neq \emptyset$  then
4      $T_{new} = T_{new} + ct$ 
5   for  $t \leftarrow canT \cap T_{new}$  and  $t \neq ct$  do
6     if  $L^*(S, T_{new} - t) \leq L^*(S, T_{new})$  then
7        $T_{new} = T_{new} - t$ 
8 return  $T_{new}$ 

```

---



The main novelty of our glyphs lies in encoding null values with uncertainty, each of which indicates that there exists multiple possible values every time using the tactic. For each null value, we first count the frequency of each possible value. Then, we display the possible value with the most frequency in our glyph, where the opacity of the corresponding component varies on multiple levels based on the frequency percentage. Such a design can help experts see how the tactic most often progresses, as well as how uncertain each value is. Furthermore, experts can expand a tactic to find the two most possible values for each null, where the bars visualize the frequency percentage.

**Operation Columns** (No., Pref.). When experts are satisfied with a tactic, they can click the favorite icon (Pref.) to fix it, avoiding future adjustments. When experts find some tactics to adjust, they can select the tactics (No.) and give suggestions in the Suggestion Panel.

### 6.3 Suggestion Panel

Experts can give new suggestions in the top input box (T2) and view past suggestions in the list below (Fig. 1B), where we allow experts to express their suggestions in natural language. Natural language is one of the best methods to enable experts to interact with complex algorithms and models [13, 25, 33, 52]. Considering that experts are generally not familiar with math-based constraints, there is a steep learning curve for them to learn the constraint space, let alone to express their suggestions through constraints. Thus, we implement a template-based NLP algorithm by NLTK in Python. The templates are built based on the suggestions collected in the pre-study described in Sect. 4.2.1, such as “(hit features) (is/are) important for (tactics)” (c5 in Fig. 4). After experts input a sentence in the top input box, the algorithm will try to match the sentence with each template, thereby finding the constraint type and extracting the parameters. To ensure the correct mapping, we display the mapped constraint and the parsed parameters and allow experts to modify them directly. In addition, experts can undo the most recent adjustment and click on a suggestion in the history list below to view the corresponding historical data.

### 6.4 Preview Mode

Experts can preview the results of an adjustment after giving a suggestion and then determine whether to apply the adjustment (T3), through the preview mode in Projection View (Fig. 1E1) and Tactic View (Fig. 1C1). C1 and E1 demonstrate one example of splitting one tactic (with solid border) into two (with dashed borders). The two new tactics have different colors, indicating different win rates. Experts may regard this as a meaningful adjustment because the original combination of these two tactics had hidden their differences. They may turn to Tactic View for detailed exploration. Tactic View moves the three tactics to the top for comparison, with icons  $-$  and  $+$  representing the original tactic and the new tactics, respectively. To reveal their differences in detail, experts can compare the glyphs and even click on each tactic to observe the related rallies in Rally View.

### 6.5 Rally View

Rally View (Fig. 1D) provides the most detailed type of information — the rallies related to a tactic (T5). This view consists of three sub-views that may be useful for tactical analysis, explained below from top to bottom: 1) The stacked bar chart visualizes **the hit indexes of the tactic**, helping experts know where the tactic is and should be used. The x-axis and y-axis represent the hit index and the frequency, respectively. The green and red bars represent the number of wins and losses, respectively, when the tactic starts with the corresponding hit. 2) The two rally lists display **the detailed hit features of each rally**. One list shows winning rallies and the other shows losing ones. Each row shows one rally, and each circle represents a hit, where solid circles encode the hits involving the tactic being explored. Experts can further explore the detailed hit features of each hit by clicking the rally. 3) Experts prefer to watch **the video fragment of each rally** to find details not recorded in the data, such as players’ facial expressions.

## 7 EVALUATION

We evaluate our algorithm’s performance and our system’s usefulness through two quantitative experiments and two case studies, respectively.

Table 1. The results of the quantitative experiments. We generated 6 synthetic datasets ( $D_1$  to  $D_6$ ), varying in the number of sequences ( $S$ ), the length of each sequence ( $|s_i|$ ), the number of features in one hit ( $k$ ), the number of tactics ( $|T|$ ), and the number of options for each hit feature ( $|V|$ ). For each dataset, we present the average runtime of the initial mining ( $t_i$ ) and applying global ( $avg. t_g$ ) and local constraints ( $avg. t_l$ ).

Datasets						Results		
$id$	$ S $	$ s_i $	$k$	$ T $	$ V $	$t_i(s)$	$avg. t_g(s)$	$avg. t_l(s)$
$D_1$	500	10	3	25	10	51.4	50.6	0.02
$D_2$	<b>700</b>	10	3	25	10	97.3	97.1	0.03
$D_3$	500	<b>20</b>	3	25	10	73.0	73.4	0.05
$D_4$	500	10	<b>5</b>	25	10	80.2	80.1	0.03
$D_5$	500	10	3	<b>50</b>	10	74.5	73.3	0.01
$D_6$	500	10	3	25	<b>20</b>	28.7	28.9	0.02

### 7.1 Algorithm Evaluation

#### 7.1.1 Tactic Quality

We first evaluate the quality of the adjusted tactics. We invited four experts (E1-E4) to conduct this experiment, testing whether our system could discover the expected tactics pre-set by experts.

**Setup.** We first required the experts to provide 100 tactics as benchmarks. Each expert was required to select five matches they were familiar with and manually list five tactics for each match based on their knowledge about the specific players. After experts repeatedly watched the game video and ensured the correctness of the listed tactics, we regarded these tactics as the benchmark tactics.

**Process.** We asked the experts to use our system to mine tactics, from the matches selected by other experts studying the same sport (e.g., both E1 and E2 studies in tennis, therefore E1 needed to mine tactics from the matches selected by E2), ensuring that the expert did not know the benchmark tactics in advance. After experts finished mining, we recorded the number of benchmark tactics discovered by our system and the number of constraints needed to adjust certain initially mined tactics to each benchmark tactic (*note: if there existed a mined tactic that had at most one value different from a benchmark tactic, we regarded this benchmark tactic to be discovered by our system*).

**Results.** The results were analyzed from three angles as follows.

- **Effectiveness:** Following experts’ suggestions, our system discovered 94 tactics from a total of 100 benchmark tactics, which proved the effectiveness of our algorithm. The other 6 tactics were all used in the tiebreaker (i.e., the last game to decide who was the winner of the match), crucial (hence found by experts as the benchmark tactics) but not frequent (hence not captured by our system).
- **Efficiency:** For each captured benchmark tactic, experts applied 0.38 global constraints and 3.11 local ones on average, which proved that our system can quickly meet experts’ expectations.
- **Necessity:** Only 3 out of benchmark tactics were directly mined by the initial generation process without any constraints, which proved the necessity of interactive pattern mining.

#### 7.1.2 Runtime

Given that experts need to continually adjust tactics, we further evaluated our algorithm’s runtime, to ensure that experts can interact with the system smoothly. We ran our algorithm on multi-scaled synthetic datasets and recorded the runtimes required to mine the initial tactic set, process global constraints, and process local ones (Table 1).

**Setup.** We generated six synthetic datasets varying over five parameters, and generated random sequences and random tactics (each with three hits and seven non-null values) for each dataset. Each tactic was embedded into 10% of the sequences, therefore we can regard the generated tactic set as an appropriate best model for the generated dataset. For each tactic set, we generated 4 global constraints (one for each type, preventing conflicts like expecting long tactics and expecting short ones) and 30 local ones (five for each type).

**Process.** We applied the constraints for each dataset in random order, as experts would do in an actual analysis scenario. The time necessary to process each constraint is recorded for analysis.

**Results.** The runtime shows that our algorithm can support smooth interactions. For each dataset, the runtime required to process global

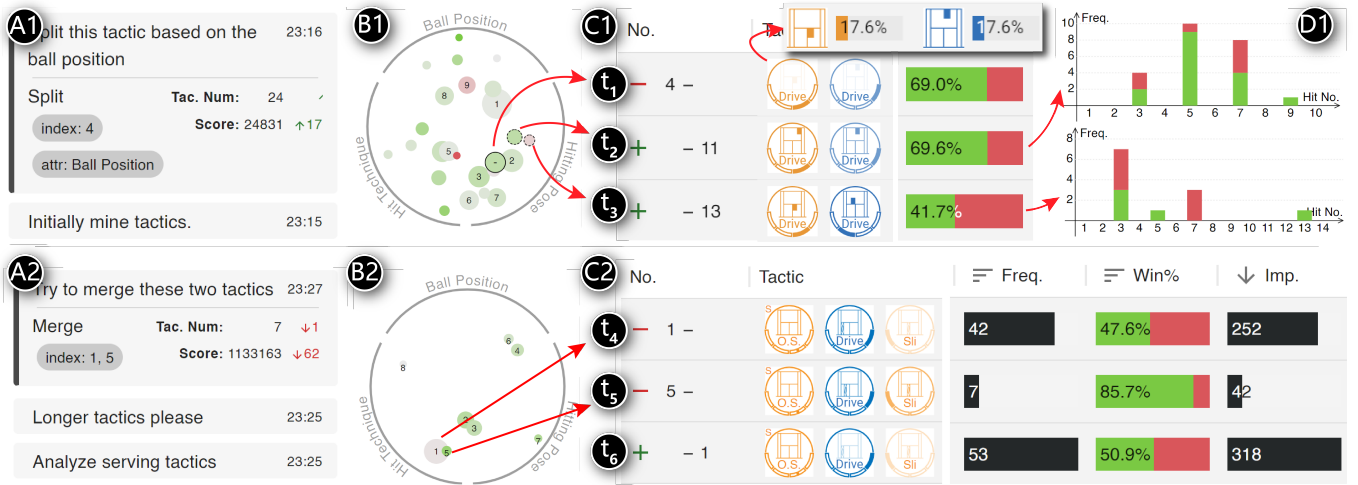


Fig. 9. Screenshots of the tennis case, where experts split a seesaw-battle tactic (A1-D1) and merged two serving tactics (A2-C2). We present the screenshots of Suggestion Panel (A1, A2), Projection View (B1, B2), Tactic View (C1, C2), and the stacked bar chart in Rally View (D1).

constraints was similar to the time needed for initial mining (about one minute), which is acceptable for experts. The runtime required to process local constraints was almost negligible because our generator can accurately generate only a few candidates expected by experts.

## 7.2 Case Studies

We invited four out of the five experts to conduct two case studies each in tennis (E1, E2) and badminton (E3, E4).

### 7.2.1 Tennis

The initial tactic set contained 23 tactics from 604 rallies played by Djokovic against Thiem (334), Nadal (143), and Pouille (127) since 2019. The tactics involved three hit features (encoded by the glyph on the right of Fig. 8) — the ball position (i.e., the position where the ball bounced on the court, encoded by the court at center), the hitting technique (which is displayed by the abbreviation text), and the hitting pose (i.e., four types of hitting pose encoded by the donut below). The analysis process occurred as follows.

**Splitting a seesaw-battle tactic based on the ball position.** Experts were first attracted by the text “Drive” in the top tactic in Tactic View ( $t_1$  in Fig. 9C1), which represents the hit technique *drive* — “Texts are easier to understand than visualizations.” Tactic  $t_1$  involved both players consecutively driving the ball, indicating a seesaw-battle tactic. Based on their knowledge, experts thought that the ball position decided the outcome of a seesaw battle. However, they found a low opacity of the courts in the glyphs, indicating high uncertainty around ball positions. Experts further expanded the tactic and found that the ball position with the highest frequency was used in only 17.6% of rallies. E1 said that, “the tactic is so abstract that I could not obtain insights from it.” Thus, they selected the tactic and suggested “splitting the tactic based on the ball positions” in Suggestion Panel (Fig. 9A1).

**Analyzing the split tactics.** Then, experts previewed the results and compared the two new tactics ( $t_2$  and  $t_3$ ) with the original ones. They noticed that, in Projection View (Fig. 9B1), the new points were closer to the sector of ball position, indicating that new tactics contained more non-null values of ball position. E2 praised the design, saying “the projection helps me track the changes.” Experts found that, although  $t_2$  and  $t_3$  were similar, they had different win rates — hitting the ball to the baseline ( $t_2$ ) tended to bring more wins for Djokovic than hitting it to the service line ( $t_3$ ). To uncover the reason, experts further explored  $t_2$  and  $t_3$  in Rally View (Fig. 9D1). The bar chart showed that Djokovic preferred to use  $t_2$  after the fifth hit, but to use  $t_3$  at the third hit, which is the first attack after service. Experts thought that the opponents might return the ball weakly at the second hit, making it difficult for Djokovic to return powerfully at the third hit, “like a ball falling on the ground, the faster it hits the ground, the faster it bounces.” However, his opponents can return powerfully at the fourth hit in an attempt to

gain the advantage. By viewing the rallies and watching the videos, experts confirmed their thoughts and then applied the adjustment.

**Analyzing serving tactics.** After splitting the seesaw-battle tactic, a serving tactic became the top tactic in Tactic View. Because Djokovic excels at serving tactics, experts suggested “analyzing serving tactics” in Suggestion Panel (Fig. 9A2). However, experts were not satisfied with the new tactic set because almost all serving tactics contained only two hits. Experts further suggested “analyzing longer tactics” and obtained some three-hit serving tactics. E2 re-ranked the tactics in Tactic View (Fig. 9C2) based on the tactical importance and commented that, “the tactical importance is more valuable than the frequency as frequent tactics are usually short and lack analytical value.”

**Merging two similar serving tactics.** When analyzing the most important tactics recommended by our system ( $t_4$ ), experts found that the point representing this tactic in the Projection View was overlapped with a point representing another tactic ( $t_5$ ), indicating the two tactics were similar (Fig. 9B2). Experts expected to merge them to form one tactic. However, the point representing  $t_4$  was nearly gray, while the point for  $t_5$  was green, indicating different outcomes and leaving experts hesitant. Given that they would be able to undo the adjustment, experts decided to try. E1 commented that, “trial-and-error tactical analysis is essential.” Experts found that our system outputted a new tactic set with a score similar to the original one, which supported the adjustment and made experts confident (Fig. 9A2). In Tactic View, experts noticed that the new tactic ( $t_6$ ) had a similar win rate to  $t_4$ , with little impact from  $t_5$ . Experts guessed that  $t_5$  had a low frequency, which may lead to a misleading win rate — “Djokovic might win these rallies by chance”. Experts applied this merging adjustment and obtained a more accurate estimate of the win rate for this serving tactic.

The two experts mainly gave positive comments on the interactive analysis workflow in the following interview. But E2 mentioned a limitation in fine-tuning tactics — “The system accurately mined the tactics I wanted, but a significant adjustment might require numerous commands. It can be helpful to control the extent of an adjustment.”

### 7.2.2 Badminton

The dataset contained 377 rallies played by Momota Kento against Viktor Axelsen (147), Anders Antonson (104), Srikanth Kidambi (63), and Yuqi Shi (63) since 2019. Experts mainly considered three hit features (encoded by the glyph on the left of Fig. 8) — the 2-D ball position (i.e., the position where the player hit the shuttle, encoded by the top-left court), the ball height (i.e., a four-level quantization of the maximum ball height encoded by the right bars), and the hitting technique (which is displayed by the bottom-left abbreviation text). The algorithm mined 13 initial tactics from these rallies for experts to analyze. The exploration process was as follows.

**Finding a tactic inconsistent with experts’ knowledge.** Tactical analysis in badminton usually focuses on both the 2-D ball position



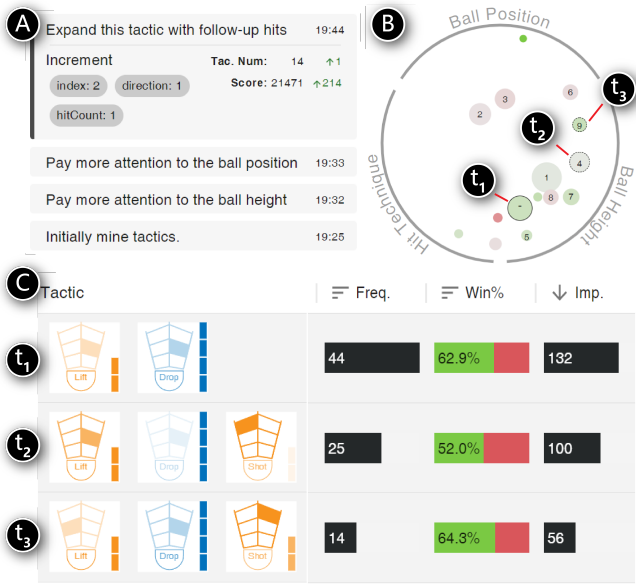


Fig. 10. Case in badminton. Experts gave suggestions for adjusting tactics (A). When experts suggested expanding the tactic, the system enabled the preview mode of Projection View (B) and Tactic View (C).

and the ball height. Hitting techniques are also named based on the ball position (e.g., *lift* is a technique hitting the ball from a low position at the player’s backcourt to a high position at the opponents’ backcourt). Thus, experts first suggested paying more attention to the ball position and the ball height, leading to two constraints on the importance of these two attributes (Fig. 10A). Then, experts started from Projection View to explore the tactics in the sector of the ball height, i.e., the tactics most relevant to the ball height (Fig. 10B). E4 praised the semantic-based projection, “different directions corresponding to different tactics is intuitive to me, helping me start analyzing.” Experts were quickly attracted by a point (with a black border in Fig. 10), which was large (high importance) and green (high win rate). Turning to Tactic View, experts further explored the tactic with two hits (t<sub>1</sub> in Fig. 10C) — Kento first lifted the shuttlecock, and his opponents then dropped the shuttlecock (i.e., an offensive technique that involves hitting the ball from a high position to a low position in the opponents’ court). Experts were confused by the tactic because “Kento gave his opponents a chance to attack but finally won many points.”

**Exploring the tactic to find the reason.** Because Kento did not mainly rely on lift technique to score points, experts thought that the tactic must have further follow-up development and suggested “expanding the tactic with follow-up hits” (Fig. 10A). Our system presented two new tactics with three hits (t<sub>2</sub> and t<sub>3</sub> in Fig. 10C). In the third hit of two of the tactics, Kento hit the shuttlecock at a low height and to the opponents’ backcourt (to the left in t<sub>2</sub> and the right in t<sub>3</sub>). However, the text “Shot” in the two tactics (i.e., a technique that involves hitting the ball a little higher than the net) attracted experts, as it indicated Kento’s counterattack. Experts continued to watch the video and found that Kento was familiar with the tactic, indicating that he might train for it. He used a lift for the first hit to lure his opponents to drop the shuttlecock. Then, relying on the quick reaction developed through training, he can counterattack with the shot technique, often throwing off his opponents and earning him points. Experts said that “the system helped us find Kento’s secrets behind the tactic.”

The two experts gave positive comments on our system in the following interview. E3 said that, “it is like I am communicating with the computer and teaching it to find tactics.” E4 suggested supporting tactic searching — “We need to search for some crucial tactics that are not mined by the system due to the low frequency.”

## 8 DISCUSSION

**Scalability.** We discuss the scalability of the system’s two main views — Projection View and Tactic View. First, the projection view cannot

precisely encode multiple related hit features of a hit. Although we can scale the projection view to more hit features by simply splitting the polar coordinates into more sections, users can hardly judge which hit feature is the second most important. But the projection view can accurately encode the most important parameters, which can help experts explore a certain type of tactics, crucial and sufficient for experts.

Second, when many hit features exist in one hit, we cannot always draw all of the components in a single glyph in the Tactic View. In the future, experts may expect to analyze other hit features that are not presented in the current dataset, such as the speed of the ball. Drawing many hit features in one glyph can lead to high visual clutter. A possible solution is to encode the hit features that experts most care about in the glyph and display others in tooltips.

**Generalizability.** Recently, we have witnessed that many domains like sports and EHR recorded and analyzed multivariate event sequences rather than only analyzing the event type and the timestamps. Our work can inspire researchers in other domains to study interactive pattern mining and obtain better mining results. Meanwhile, our workflow with close collaboration with domain experts can be instructive.

**Design Implications.** We summarize several findings made during our study, which may be relevant to interactive visual analytics in other domains. 1) Comparative analysis is crucial for interactive tactical analysis, during the process of which data-driven metrics play an essential role to provide hard evidence about new results, helping users to better trust the system. 2) Semantics could help domain experts with little visualization experience explore a projection view. If the position of the points presents semantic information, this could help experts recognize and identify each point from their domain perspective. 3) The uncertainty arising from the interactive analysis process is an issue, especially when users interact with the system through natural language. When users do not understand how the system works, they may make intractable suggestions, preventing the system from producing results that meet users’ expectations. We tried to address this issue by exposing how user suggestions are translated to constraints step by step. Further research is required to better address the problem.

**Limitations and future work.** Although we summarized nine constraints and hundreds of natural language templates through the pre-study to support interactive tactic mining, we still found them limited the diversity of tactics in practice. For example, players prefer to hide some tactics as weapons used in tiebreakers, which are crucial (hence requiring analysis) but infrequent (hence tricky for algorithms to discover). Without similar tactics to modify, experts can hardly obtain these tactics in the tactic set for further analysis. Natural language can be hardly used to precisely define a complex tactic with multiple detailed hit features due to the limited natural language templates and the ambiguity of natural language. Compared with NL-based interactions, Eventpad [6] proposed efficient and intuitive interaction widgets to help users construct multivariate patterns for further queries based on regular expressions. Our future work includes mining these crucial but infrequent tactics and integrating a pattern searching function.

## 9 CONCLUSION

This work introduces a visual analytics system for interactive tactical analysis in racket sports, which allows experts to incorporate their knowledge into data mining algorithms to discover meaningful tactics. We propose a constraint-based pattern mining algorithm that discovers an initial set of tactics and then supports adjustments to them by translating experts’ written suggestions into further constraints. We also propose a user interface through which experts can interact with the algorithm — exploring tactics through metaphoric glyphs, giving suggestions, and verifying results through comparative visualizations. A quantitative experiment on synthetic datasets shows that our system supports smooth interaction between experts and the algorithm. Two real-world case studies demonstrate that our system can help experts use their knowledge to find meaningful tactics.

## ACKNOWLEDGMENTS

The work was supported by NSFC (62072400) and the Collaborative Innovation Center of Artificial Intelligence by MOE and Zhejiang Provincial Government (ZJU).

## REFERENCES

- [1] C. C. Aggarwal and P. S. Yu. Online generation of association rules. In *Proceedings of the International Conference on Data Engineering*, pp. 402–411. IEEE, 1998. <https://doi.org/10.1109/ICDE.1998.655803>.
- [2] 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. <https://doi.org/10.1109/TVCG.2019.2952129>.
- [3] N. Andrienko, G. Andrienko, S. Miksch, H. Schumann, and S. Wrobel. A theoretical model for pattern discovery in visual analytics. *Visual Informatics*, 5(1):23–42, 2021. <https://doi.org/10.1016/j.visinf.2020.12.002>.
- [4] M. Boley, M. Mampaey, B. Kang, P. Tokmakov, and S. Wrobel. One click mining: Interactive local pattern discovery through implicit preference and performance learning. In *Proceedings of the ACM SIGKDD workshop on interactive data exploration and analytics*, pp. 27–35, 2013. <https://doi.org/10.1145/2501511.2501517>.
- [5] A. Cao, X. Xie, J. Lan, H. Lu, X. Hou, J. Wang, H. Zhang, D. Liu, and Y. Wu. MIG-Viewer: Visual analytics of soccer player migration. *Visual Informatics*, 5(3):102–113, 2021. <https://doi.org/10.1016/j.visinf.2021.09.002>.
- [6] B. C. Cappers, P. N. Meessen, S. Etalle, and J. J. Van Wijk. Eventpad: Rapid malware analysis and reverse engineering using visual analytics. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security*, pp. 1–8, 2018. <https://doi.org/10.1109/VIZSEC.2018.8709230>.
- [7] B. C. M. 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. <https://doi.org/10.1109/TVCG.2017.2745278>.
- [8] 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. <https://doi.org/10.1109/TVCG.2021.3114861>.
- [9] M. Du and X. Yuan. A survey of competitive sports data visualization and visual analysis. *Journal of Visualization*, 24(1):47–67, 2021. <https://doi.org/10.1007/s12650-020-00687-2>.
- [10] J. J. Dudley and P. O. Kristensson. A review of user interface design for interactive machine learning. *ACM Transactions on Interactive Intelligent Systems (TiIS)*, 8(2):1–37, 2018. <https://doi.org/10.1145/3185517>.
- [11] J. A. Fails, A. Karlson, L. Shahamat, and B. Shneiderman. A visual interface for multivariate temporal data: Finding patterns of events across multiple histories. In *Proceedings of the IEEE Symposium On Visual Analytics Science And Technology*, pp. 167–174, 2006. <https://doi.org/10.1109/VAST.2006.261421>.
- [12] 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. <http://www.philippe-fournier-viger.com/dspr-paper5.pdf>.
- [13] T. Gao, M. Dontcheva, E. Adar, Z. Liu, and K. G. Karahalios. Datatone: Managing ambiguity in natural language interfaces for data visualization. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pp. 489–500, 2015. <https://doi.org/10.1145/2807442.2807478>.
- [14] R. Gove, L. Cadalzo, N. Leiby, J. M. Singer, and A. Zaitzeff. New guidance for using t-SNE: Alternative defaults, hyperparameter selection automation, and comparative evaluation. *Visual Informatics*, 2022. <https://doi.org/10.1016/j.visinf.2022.04.003>.
- [15] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit. Lineup: Visual analysis of multi-attribute rankings. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2277–2286, 2013. <https://doi.org/10.1109/TVCG.2013.173>.
- [16] P. D. Grünwald and A. Grunwald. *The minimum description length principle*. MIT press, 2007. <https://mitpress.mit.edu/books/minimum-description-length-principle>.
- [17] Y. Guo, S. Guo, Z. Jin, S. Kaul, D. Gotz, and N. Cao. A survey on visual analysis of event sequence data. *IEEE Transactions on Visualization and Computer Graphics*, 2021. <https://doi.org/10.1109/TVCG.2021.3100413>.
- [18] A. Hibbs and P. O'Donoghue. Strategy and tactics in sports performance. In *Routledge handbook of sports performance analysis*, pp. 266–276. Routledge, 2013. <https://www.taylorfrancis.com/chapters/edit/10.4324/9780203806913-32/strategy-tactics-sports-performance-angela-hibbs-peter-donoghue>.
- [19] W. Kim, C. Shim, and Y. D. Chung. SkyFlow: A visual analysis of high-dimensional skylines in time-series. *Journal of Visualization*, 24(5):1033–1050, 2021. <https://doi.org/10.1007/s12650-021-00758-y>.
- [20] 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. *Journal of Visualization*, 25(1):143–158, 2022. <https://doi.org/10.1007/s12650-021-00772-0>.
- [21] M.-Y. Lin and S.-Y. Lee. Improving the efficiency of interactive sequential pattern mining by incremental pattern discovery. In *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, pp. 8–pp. IEEE, 2003. <https://doi.org/10.1109/HICSS.2003.1173921>.
- [22] D. Liu, S. Alnegheimish, A. Zyttek, and K. Veeramachaneni. MTV: Visual analytics for detecting, investigating, and annotating anomalies in multivariate time series. *arXiv preprint arXiv:2112.05734*, 2021. <https://doi.org/10.48550/arXiv.2112.05734>.
- [23] D. Liu, P. Xu, and L. Ren. TPFlow: Progressive partition and multidimensional pattern extraction for large-scale spatio-temporal data analysis. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):1–11, 2019. <https://doi.org/10.1109/TVCG.2018.2865018>.
- [24] M. H. Loorak, C. Perin, N. Kamal, M. Hill, and S. Carpendale. Timespan: Using visualization to explore temporal multi-dimensional data of stroke patients. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):409–418, 2015. <https://doi.org/10.1109/TVCG.2015.2467325>.
- [25] Y. Luo, N. Tang, G. Li, J. Tang, C. Chai, and X. Qin. Natural language to visualization by neural machine translation. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):217–226, 2021. <https://doi.org/10.1109/TVCG.2021.3114848>.
- [26] E. Maguire, P. Rocca-Serra, S.-A. Sansone, J. Davies, and M. Chen. Taxonomy-based glyph design—with a case study on visualizing workflows of biological experiments. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2603–2612, 2012. <https://doi.org/10.1109/TVCG.2012.271>.
- [27] V. Menger. An experimental analysis of the pattern explosion. Master's thesis, 2015. <https://studenttheses.uu.nl/bitstream/handle/20.500.12932/19396/Experimental%20Analysis%20of%20the%20Pattern%20Explosion.pdf?sequence=2>.
- [28] Y. Ming, P. Xu, F. Cheng, H. Qu, and L. Ren. ProtoSteer: Steering deep sequence model with prototypes. *IEEE transactions on visualization and computer graphics*, 26(1):238–248, 2019. <https://doi.org/10.1109/TVCG.2019.2934267>.
- [29] Y. Ming, P. Xu, H. Qu, and L. Ren. Interpretable and steerable sequence learning via prototypes. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 903–913, 2019. <https://doi.org/10.1145/3292500.3330908>.
- [30] S. Parthasarathy, M. J. Zaki, M. Ogihara, and S. Dwarkadas. Incremental and interactive sequence mining. In *Proceedings of the 8th International Conference on Information and Knowledge Management*, pp. 251–258, 1999. <https://doi.org/10.1145/319950.320010>.
- [31] J.-D. Ren and J.-S. Zong. A fast interactive sequential pattern mining algorithm based on memory indexing. In *Proceedings of the International Conference on Machine Learning and Cybernetics*, pp. 1082–1087. IEEE, 2006. <https://doi.org/10.1109/ICMLC.2006.258564>.
- [32] M. Sedlmair, M. Meyer, and T. Munzner. Design Study Methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431–2440, 2012. <https://doi.org/10.1109/TVCG.2012.213>.
- [33] L. Shen, E. Shen, Y. Luo, X. Yang, X. Hu, X. Zhang, Z. Tai, and J. Wang. Towards natural language interfaces for data visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2022. <https://doi.org/10.1109/TVCG.2022.3148007>.
- [34] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the IEEE symposium on visual languages*, pp. 336–343, 1996. <https://doi.org/10.1016/B978-155860915-0/50046-9>.
- [35] C. Stoiber, D. Ceneda, M. Wagner, V. Schetinger, T. Gschwandner, M. Streit, S. Miksch, and W. Aigner. Perspectives of visualization onboarding and guidance in va. *Visual Informatics*, 6(1):68–83, 2022. <https://doi.org/10.1016/j.visinf.2022.02.005>.
- [36] N. Tatti and M. Mampaey. Using background knowledge to rank itemsets. *Data Mining and Knowledge Discovery*, 21(2):293–309, 2010. <https://doi.org/10.1007/s10618-010-0188-4>.
- [37] L. Van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11), 2008.

<https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf?fbclid=IwA>.

- [38] J. Wang and K. Mueller. The visual causality analyst: An interactive interface for causal reasoning. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):230–239, 2015. <https://doi.org/10.1109/TVCG.2015.2467931>.
- [39] J. Wang and K. Mueller. Visual causality analysis made practical. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 151–161. IEEE, 2017. <https://doi.org/10.1109/VAST.2017.8585647>.
- [40] 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. <https://doi.org/10.1109/TVCG.2021.3074576>.
- [41] 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. <https://doi.org/10.1109/TVCG.2019.2934630>.
- [42] M. Wang, J. Wenskovich, L. House, N. Polys, and C. North. Bridging cognitive gaps between user and model in interactive dimension reduction. *Visual Informatics*, 5(2):13–25, 2021. <https://doi.org/10.1016/j.visinf.2021.03.002>.
- [43] M. Wojciechowski. Interactive constraint-based sequential pattern mining. In *Proceedings of the East European Conference on Advances in Databases and Information Systems*, pp. 169–181. Springer, 2001. [https://doi.org/10.1007/3-540-44803-9\\_14](https://doi.org/10.1007/3-540-44803-9_14).
- [44] A. Wu, D. Deng, F. Cheng, Y. Wu, S. Liu, and H. Qu. In defence of visual analytics systems: Replies to critics. *IEEE Transactions on Visualization and Computer Graphics*, To Appear.
- [45] J. Wu, Z. Guo, Z. Wang, Q. Xu, and Y. Wu. Visual analytics of multivariate event sequence data in racquet sports. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 36–47. IEEE, 2020. <https://doi.org/10.1109/VAST50239.2020.00009>.
- [46] J. Wu, D. Liu, Z. Guo, Q. Xu, and Y. Wu. TacticFlow: Visual analytics of ever-changing tactics in racket sports. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):835–845, 2021. <https://doi.org/10.1109/TVCG.2021.3114832>.
- [47] 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. <https://doi.org/10.1109/TVCG.2017.2744218>.
- [48] Y. Wu, X. Xie, J. Wang, D. Deng, H. Liang, H. Zhang, S. Cheng, and W. Chen. Forvizor: Visualizing spatio-temporal team formations in soccer. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):65–75, 2018. <https://doi.org/10.1109/TVCG.2018.2865041>.
- [49] 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. <https://doi.org/10.1109/TVCG.2020.3030359>.
- [50] L. Ying, T. Tangl, Y. Luo, L. Shen, X. Xie, L. Yu, and Y. Wu. GlyphCreator: Towards example-based automatic generation of circular glyphs. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):400–410, 2021. <https://doi.org/10.1109/TVCG.2021.3114877>.
- [51] L. Ying, Y. Yang, X. Shu, D. Deng, T. Tang, L. Yu, and Y. Wu. MetaGlyph: Automatic Generation of Metaphoric Glyph-based Visualization. *IEEE Transactions on Visualization and Computer Graphics*, To Appear.
- [52] B. Yu and C. T. Silva. FlowSense: A natural language interface for visual data exploration within a dataflow system. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1–11, 2019. <https://doi.org/10.1109/TVCG.2019.2934668>.