

# OpenSTARLab: Open Approach for Spatio-Temporal Agent Data Analysis in Soccer

Calvin Yeung<sup>1</sup>, Kenjiro Ide<sup>1</sup>, Taiga Someya<sup>2</sup>, Keisuke Fujii<sup>1,3\*</sup>

<sup>1</sup>Graduate School of Informatics, Nagoya University, Nagoya, Japan.

<sup>2</sup>Graduate School of Arts and Sciences, The University of Tokyo, Tokyo, Japan.

<sup>3</sup>Center for Advanced Intelligence Project, RIKEN, Osaka, Japan.

\*Corresponding author(s). E-mail(s): [fujii@i.nagoya-u.ac.jp](mailto:fujii@i.nagoya-u.ac.jp);

Contributing authors: [yeung.chikwong@g.sp.m.is.nagoya-u.ac.jp](mailto:yeung.chikwong@g.sp.m.is.nagoya-u.ac.jp);

[ide.kenjiro@g.sp.m.is.nagoya-u.ac.jp](mailto:ide.kenjiro@g.sp.m.is.nagoya-u.ac.jp); [taiga98-0809@g.ecc.u-tokyo.ac.jp](mailto:taiga98-0809@g.ecc.u-tokyo.ac.jp);

## Abstract

Sports analytics has become both more professional and sophisticated, driven by the growing availability of detailed performance data. This progress enables applications such as match outcome prediction, player scouting, and tactical analysis. In soccer, the effective utilization of event and tracking data is fundamental for capturing and analyzing the dynamics of the game. However, there are two primary challenges: the limited availability of event data primarily restricted to top-tier teams and leagues, and the scarcity and high cost of tracking data, which complicates its integration with event data for comprehensive analysis. Here we propose OpenSTARLab, an open-source framework designed to democratize spatio-temporal agent data analysis in sports by addressing these key challenges. OpenSTARLab includes the Pre-processing Package that standardizes event and tracking data through Unified and Integrated Event Data and State-Action-Reward formats, the Event Modeling Package that implements deep learning-based event prediction, alongside the RLearn Package for reinforcement learning tasks. These technical components facilitate the handling of diverse data sources and support advanced analytical tasks, thereby enhancing the overall functionality and usability of the framework. To assess OpenSTARLab's effectiveness, we conducted several experimental evaluations. These demonstrate the superior performance of the specific event prediction model in terms of action and time prediction accuracies and maintained its robust event simulation performance. Furthermore, reinforcement learning experiments reveal a trade-off between action accuracy and temporal difference loss and show comprehensive visualization. Overall, OpenSTARLab serves as a robust platform for researchers

and practitioners, enhancing innovation and collaboration in the field of soccer data analytics.

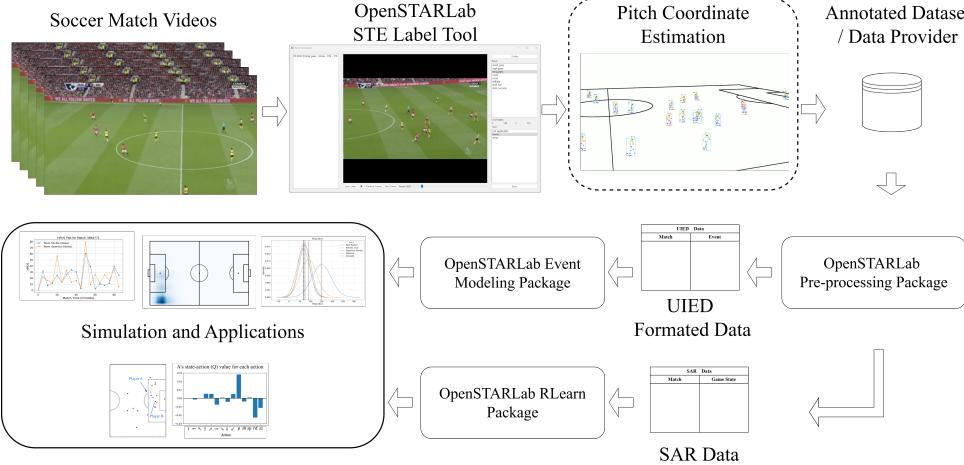
**Keywords:** Machine learning, Reinforcement learning, Sports, Football

## 1 Introduction

The increasing availability of sports data has developed analytics, driving applications such as match result prediction [1–3], player scouting [4–6], and tactical analysis [7–9] through data-driven approaches. A cornerstone of these advancements in soccer analytics is the utilization of match data, particularly the event data and tracking data, which are essential for capturing and analyzing the game’s dynamics. Event data records key occurrences such as passes, shots, and fouls and tracking data captures player and ball positions over time. The growing availability of such data, combined with advancements in machine learning techniques, has led to the development of numerous methods for play evaluation using predictive modeling (e.g., [4, 7]) and reinforcement learning (RL; e.g., [10, 11]). Here we call the data for such analytical tools *spatio-temporal agent data*, and we focus on soccer, a sport characterized by significant investment in data collection and the presence of several prominent data providers.

Currently, two primary challenges hinder the universal accessibility of spatio-temporal agent data analysis. First, event data, which forms the foundation of many soccer analytics tasks, is sometimes publicly available from data provider companies, but predominantly available only for top-tier teams and leagues. This limitation creates a significant barrier for researchers and analysts working on amateur competitions, where such data is scarce. It should be noted that video footage is primarily effective for understanding the game. However, its accessibility is often limited by copyright and privacy concerns, which restrict its widespread use in research. While open-source datasets like SoccerNet [12–14] provide a substantial amount of publicly available footage, though sufficient for computer vision tasks such as tracking [15–17] and event detection [12–14, 18, 19], they often lack the extensive annotations required for predictive modeling and RL applications. Regardless of whether the data is richly annotated or limited in quantity, there remains a lack of accessible infrastructure for obtaining data on amateur teams, often the focus of interest for many researchers and practitioners. To democratize sports analytics, it is essential to establish an open-source platform from annotating data of video footage to training machine learning models.

Second, tracking data is even more limited than event data, with publicly available datasets being exceptionally scarce, which remains primarily accessible only to professional teams through data providers due to higher measurement costs. Due to the limitation of access, integrating tracking data with event data presents significant challenges. Although temporal synchronization methods have been discussed in some approaches (e.g., [20]), there is a notable lack of accessible pipelines for preprocessing tracking and event data, especially for tasks that require feeding data at each time frame into deep learning models. This limitation is particularly pronounced in tasks



**Fig. 1** Overview of the OpenSTARLab application on soccer. The dashed rectangle indicates steps that may be required depending on the specific soccer match video. Soccer match image from SoccerNet [13].

involving deep RL (DRL), where data from every time frame is crucial for modeling [11] (for details, see Section 2).

To address these issues, we propose OpenSTARLab (i.e., *Open approach Spatio-Temporal Agent data Research platform*), a unified framework designed to streamline event annotation, data standardization, and various deep learning modeling for soccer analytics as illustrated in Figure 1. For the first challenge, OpenSTARLab provides an *Event Modeling Package*<sup>1</sup>, which offers a holistic workflow for deep learning-based event prediction models. This package provides tools for model training, applications, and visualization, enabling effective interpretation and analysis of results. To accommodate the diverse data formats from different providers, the platform introduces a *Unified and Integrated Event Data* (UIED) format in *Preprocessing Package*<sup>2</sup> to standardize and prepare the data. These tools ensure that data from both public and private sources can be seamlessly integrated. For cases where annotated data is unavailable, OpenSTARLab also offers an *Spatial-Temporal Event (STE) label Tool*<sup>3</sup>, enabling users to label video data with event details efficiently, thereby democratizing the ability to prepare data for machine learning tasks.

For the second challenge, OpenSTARLab extends its Preprocessing Package to handle tracking data and introduces the SAR (*State-Action-Reward*) format specifically tailored for deep learning tasks, including DRL. The SAR format enables the sequential processing of data at each time frame, which is critical for modeling tasks that involve temporal dynamics. To further support DRL, the platform includes the *RLearn Package*<sup>4</sup>, which provides an integrated pipeline for training DRL models,

<sup>1</sup>OpenSTARLab Event Package: <https://github.com/open-starlab/Event>

<sup>2</sup>OpenSTARLab Pre-processing Package: <https://github.com/open-starlab/PreProcessing>

<sup>3</sup>OpenSTARLab STE label Tool: [https://github.com/open-starlab/Event\\_Data\\_LabelTool](https://github.com/open-starlab/Event_Data_LabelTool)

<sup>4</sup>OpenSTARLab RLearn Package: <https://github.com/open-starlab/RLearn>

from data preparation to visualization of results. By aligning event and tracking data within a unified framework, OpenSTARLab enables seamless integration and supports advanced DRL applications that leverage spatiotemporal information.

The contributions of this study are summarized as follows. OpenSTARLab offers a comprehensive and unified framework that democratizes spatio-temporal agent data analysis by effectively addressing key challenges in data accessibility, preprocessing, and modeling. This enables researchers and practitioners to conduct advanced analytics and provides greater innovation and collaboration within the field of soccer data science. The framework includes specialized packages such as the Preprocessing Package, which standardizes event and tracking data through the UIED and SAR formats, and the STE Label Tool, which facilitates efficient event annotation from video data. These technical components provide solutions for handling diverse data sources and support sophisticated deep learning and reinforcement learning tasks. Experimental evaluations demonstrate the superior performance of the specific event prediction model in terms of action and time prediction accuracies and maintained its robust event simulation performance. Reinforcement learning experiments reveal a trade-off between action accuracy and temporal difference loss and show comprehensive visualization.

The rest of this paper is organized as follows: Section 2 reviews related work; Section 3 introduces the OpenSTARLab framework; Section 4 presents experimental results; Section 5 concludes this paper with future directions.

## 2 Related work

In this section, the literature on event annotation, data formatting, and predictive modeling for soccer is reviewed, with attention to recent advancements and existing challenges. Section 2.1 examines methods for soccer event annotation, including semi-automatic, automatic, and deep-learning approaches. Section 2.2 addresses the diversity of soccer event data formats across providers. Finally, Sections 2.3 and 2.4 review predictive modeling and RL approaches in soccer.

### 2.1 Event annotation in soccer

In soccer, analyzing player interactions and game events is essential for strategic insights. However, the sport’s complexity and fast pace make accurate event annotation challenging. Reliable and detailed event data, including event types and spatiotemporal information, is crucial for coaches, analysts, and players. Over the years, various methods have been developed to automate the detection and annotation of soccer events from broadcast videos.

Initial approaches relied on semi-automatic methods, which include clustering techniques [21] and rule-based algorithms [22], often enhanced with external text sources [23]. These approaches required extensive domain knowledge, as rule accuracy and cluster reliability are sensitive to factors like camera angles and filming conditions. Subsequently, automatic methods were introduced to improve efficiency. These leveraged audiovisual features and fuzzy rule-based reasoning systems [24], visual cues paired with finite state machines [25], and synchronized match reports and video analysis [26].

The rapid advancement of deep learning has further expanded the capabilities of automatic event detection in soccer. Recent approaches include CNN-based methods [27, 28], combinations of inertial sensor data with Random Forest classifiers [29], self-attention mechanisms [30], and hybrid CNN-RNN architectures [31]. Since 2021, the SoccerNet Challenge [32] has provided a competitive platform for testing and improving these techniques. Additionally, simulated data has been explored to enhance event detection [33].

Despite these advancements, both semi-automatic and automatic methods face limitations. Semi-automatic methods require deep domain expertise in both computer science and soccer to ensure the accuracy of rules and clusters, and their performance can be highly affected by variations in video capturing. On the other hand, automatic methods struggle with precision, with the current state-of-the-art method in SoccerNet Challenge 2024 [32] achieving only a mean average precision of 73. A significant gap in most approaches is the lack of attention to spatial elements in event data, even though modern datasets often include both event types and spatiotemporal information.

Currently, no publicly available annotation tool allows users to comprehensively label event types along with their spatial and temporal dimensions. To address this, we propose the STE Label Tool<sup>3</sup>, designed to support the detailed and accurate annotation of event types and integrate spatiotemporal data for more holistic analysis. This tool aims to bridge the current gap in soccer event data annotation, providing an essential resource for researchers and analysts seeking accurate and spatially aware event data for all types of soccer matches.

## 2.2 Event data format in soccer

In professional soccer, numerous sports data vendors provide human-annotated event data, which is indispensable for modern soccer analytics, spanning applications in performance analysis, fan engagement, and machine learning-based research. However, using event data from multiple providers presents several challenges [34]:

1. **Diverse Objectives:** Vendors structure data differently to meet diverse objectives, such as audience engagement, in-depth performance analytics, and research, leading to variability in data formats.
2. **Inconsistent Terminology:** Each vendor uses distinct terminologies and definitions to describe in-game events, making it challenging to standardize the data for comparative or integrative studies.
3. **Fragmented Event Information:** Many vendors offer additional, optional information for specific event types, creating variability in the data and complicating the application of automated analysis tools consistently across events.

Existing Python packages SoccerAction (SPDAL) [34] and Kloppy<sup>5</sup> have aimed to address these issues by grouping event data into broader categories, facilitating standardized event handling. However, these packages still face limitations: common actions, such as different pass types, are often aggregated under a single category, while rare actions (e.g., penalty shots) are classified separately, limiting flexibility in data

---

<sup>5</sup>Kloppy: <https://github.com/PySport/kloppy/tree/master>

analysis [35]. Moreover, deep learning and statistical analyses benefit from features that capture transitions between events, which are underdeveloped in these tools.

To address these limitations, we introduce the Unified and Integrated Event Data (UIED) format, implemented in the OpenSTARLab Pre-processing package<sup>2</sup>. UIED unifies data from various vendors—such as StatsBomb, Wyscout, and DataStadium—into a consistent format that also aligns with the Google Research Football (GRF) RL environment [36]. The UIED format integrates essential spatial and temporal features, such as changes in distance, time, and the distance and angle to goal, which have proven valuable in previous studies [7, 9, 35, 37]. These features support advanced analyses without dependence on optional, vendor-specific fields, making UIED a robust tool for consistent event data analysis. Further details on the UIED format are provided in Section 3.2.

### 2.3 Predictive modeling for soccer event

Model (Year)	Data	Input Variables	Model Architecture (Cost Function)	Target Variables (Event at $t + 1$ )
Seq2Event [7] (2022)	Wyscout [38]	- xy coordinates - action - time - score advantage - derived features <sup>2</sup>	Transformer encoder, MLP decoder (CE + RMSE)	- xy coordinates - action
NMSTPP [37] (2023)	Wyscout [38]	- zone - action - inter-event time - derived features <sup>2</sup>	Transformer encoder, NPP decoder (CE + RMSE)	- zone - action - inter-event time
NMSTPP+360 [9, 35] (2023, 2024)	StatsBomb 360 Data	- zone - action - inter-event time - player coordinates - derived features <sup>2</sup>	Transformer encoder, NPP decoder (CE + RMSE)	- zone - action - inter-event time
LEM [39, 40] (2024, 2024)	Wyscout [38]	- xy coordinates - action - time - home team indicator - accuracy indicator - goal indicator - match scores	3 MLPs (non end-to-end) (CE)	- xy coordinates - action - time - home team indicator - accuracy indicator - goal indicator
FMS [8] (2024)	StatsBomb Open Data <sup>1</sup>	- xy coordinates - action - team	Transformer decoder (CE)	- action

**Table 1** Comparison of soccer event prediction models in recent studies. The models are ranked by publication year, and studies that use similar models but differ only in factors other than architecture, inputs, target, or cost function are grouped into the same row. In the model architecture and cost function, MLP denotes multi-layer perceptron, NPP denotes neural point process, CE denotes cross-entropy, and RMSE denotes root mean square error.

<sup>1</sup>StatsBomb open data available at <https://github.com/statsbomb/open-data>

<sup>2</sup>Derived features include the change in time and distance, angle to goal, and distance to goal.

Predictive modeling in soccer provides quantitative insights into tactical dynamics, team performance, and potential outcomes by forecasting on-field events. A prominent example is the expected goals (xG) model [41], which predicts the probability of a goal. This concept has been extended to models such as expected assists (xA)<sup>6</sup> and expected threat (xT)<sup>7</sup>, which evaluate the likelihood of creating scoring opportunities or increasing the potential threat of a play.

The Valuing Actions by Estimating Probabilities (VAEP) framework [4] further enhances predictive capabilities by modeling the likelihood of scoring or conceding in future time frames. Variants such as VDEP [42] and GVDEP [43] extend these principles to focus specifically on defensive evaluations. However, these traditional approaches often lack comprehensive representations of future events, limiting their predictive scope. Recent advances in deep learning, particularly with transformer models [44], have revolutionized soccer analytics by enabling the capture of complex sequences in match data. These methods provide more robust and comprehensive predictions of future events, offering new possibilities for analyzing and optimizing team performance.

The Seq2Event model [7] marked a foundational step in this field. Utilizing a transformer encoder for feature embedding and a multi-layer perceptron (MLP) for decoding, Seq2Event predicts the xy coordinates and action probabilities at the next time step t+1. This model introduced a novel metric, the possession utility (poss-util) score, which assesses team possession.

Building on Seq2Event, the Neural Marked Spatio-Temporal Point Process (NMSTPP) model [37] integrates a neural marked spatio-temporal point process [45] framework. Replacing the MLP decoder, NMSTPP predicts event timing and spatial attributes sequentially, extending the possession utility score into a Holistic Possession Utilization Score (HPUS) by adding temporal considerations. The NMSTPP+360 [9] further enhances predictive accuracy by incorporating player coordinate data, and its potential in RL applications is explored in [35].

Further advancements include the Large Events Model (LEM) model [39], which forecasts soccer events by independently training three MLPs to predict event attributes sequentially. LEM also introduces the Expected Points Added (xP+) metric to quantify a player's contribution toward match outcomes. The model has demonstrated potential for both short- and long-term match simulations [40]. Lastly, the Foundation Model for Soccer (FMS) [8], a transformer decoder-based model, predicts upcoming actions with potential applications in action simulations. Table 1 summarizes these soccer event prediction models.

Despite these advancements, existing studies use varying frameworks and evaluation metrics, complicating direct comparisons. Additionally, most models lack comprehensive simulation performance evaluations. To address these gaps, we introduce the UIED format alongside the OpenStarLab Event package<sup>1</sup>. Together, they establish a standardized platform for event data, facilitating consistent training, evaluation, and optimization methods, thereby enabling reliable model comparison and performance benchmarking, as detailed in Section 4.

---

<sup>6</sup>Expected Assist Model <https://www.statsperform.com/opta-analytics/>

<sup>7</sup>Expected Threat Model <https://karun.in/blog/expected-threat.html>

## 2.4 Reinforcement learning in soccer

The application of agent-based modeling in sports analytics has grown significantly, particularly for analyzing player behaviors and interactions. Data-driven analysis involves estimating various variables and functions (such as state variables, reward and policy functions, and action evaluation) in Markov decision processes (MDPs) from observed data representing a sub-problem of RL known as the inverse approach. In contrast, constructing an RL model to generate data in a virtual environment (i.e., the forward approach such as [36, 46]) provides another way to understand the behavior of the modeled agents; however, the forward approach is beyond the scope of this study because it does not usually use the measured data. In this paper, we focus on introducing deep RL (DRL) based methods, these inverse approaches leverage large-scale data to evaluate actions based on their potential to maximize future rewards (for other approaches including the forward approach, see [47, 48]).

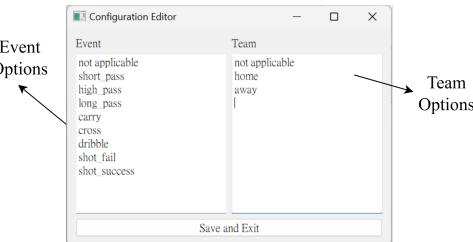
In the context of on-ball actions for team sports, numerous studies have focused on estimating state-action value functions (Q-function) or other policy-related functions using soccer event data [10, 49, 50] and tracking data [51–53], as well as in sports like ice hockey [54, 55], badminton [56], and basketball [57, 58]. Despite these advancements, most models primarily represent teams using on-ball events, often neglecting evaluations of off-ball player dynamics at all time steps. To address this limitation, Nakahara et al. [11] proposed a holistic RL framework capable of estimating Q-functions for multiple players, enabling simultaneous evaluation of both on- and off-ball players, even in the absence of explicit events. In this paper, we introduce the SAR format and RLearns package<sup>4</sup> for the first time, which includes preprocessing, modeling, and visualization tools to improve accessibility and facilitate easier integration and analysis. For other topics, research in inverse RL (IRL) has explored estimating reward functions directly from data [59, 60]. To model policies, trajectory predictions have been conducted using imitation learning [61–64] and behavioral modeling [65–68], which focus on mimicking observed behaviors through neural networks rather than optimizing them.

## 3 OpenSTARLab

In this section, we present OpenSTARLab, a comprehensive framework for soccer analytics and research that integrates event annotation, data standardization, and predictive modeling capabilities. Section 3.1 introduces the STE Label Tool, a specialized software application for efficient soccer event annotation from match videos. Section 3.2 describes the Unified and Integrated Event Data (UIED) format, which standardizes event data across various providers such as GRF, StatsBomb, Wyscout, and DataStadium. Section 3.4 details the Pre-processing package that facilitates the conversion of diverse data formats into UIED while maintaining data integrity and consistency. Section 3.5 presents the Event Modeling package, which enables event prediction and analysis through state-of-the-art models, along with their practical applications. Finally, Section presents the RLearns Modeling package, which enables multi-agent decision-making analysis. The framework’s limitations and future directions are discussed in Section 3.7.



**Fig. 2** Interface of STE label tool. Soccer match image from SoccerNet [13].



**Fig. 3** Configuration of STE label tool.

The framework's complete workflow is illustrated in Figure 1. The pipeline begins with raw soccer match videos, which are processed through the STE Label Tool for systematic event annotation. An intermediate step of pitch coordinate mapping may be required (see Section 3.7 for details). The annotated data is then standardized into the UIED format using the OpenSTARLab Pre-processing Package. Finally, the OpenSTARLab Event Modeling Package enables comprehensive analysis through model training, inference, simulation, and various analytical applications such as heat maps and possession metrics.

### 3.1 Spatial-Temporal Event (STE) label tool

The STE Label Tool<sup>3</sup> is a specialized software application inspired by the SoccerNet [13] action annotation tool, designed to facilitate efficient and accurate annotation of soccer event data. While the original tool focused primarily on annotating isolated player action types, the STE Label Tool extends these capabilities to include broader event details such as event type, position, and team information. To enhance user intuitiveness and accessibility, the tool provides precise frame-by-frame navigation,

displays the annotation workspace directly on the main interface, and allows users to edit annotation options directly on-screen. These enhancements make the tool particularly well-suited for labeling data critical to soccer analysis. It enables researchers and analysts to label key aspects of events in soccer match videos, which is essential for training machine learning models, especially in tasks related to predicting player actions, generating tactical insights, and evaluating player performance.

Developed using PyQt5, the STE Label Tool provides a flexible and user-friendly graphical interface, with an application layout focused on ease of use and efficient labeling workflows. The interface comprises three main columns, as shown in Figure 2:

- **Left Column:** This section is dedicated to data input and displays previously annotated data (if any). Users can easily navigate through annotated events and corresponding frames, or remove annotations with minimal effort.
- **Middle Column:** This section enables frame-by-frame navigation, allowing users to precisely control time-based events. This feature is particularly useful for aligning annotations with exact moments in the video.
- **Right Column:** This serves as the primary workspace for annotating events. It displays options for configuration, event type selection, position selection (by simply clicking on the frame), and team selection. Additionally, the configuration section includes a pop-up window (Figure 3) that allows users to edit options in a text editor-like format, where each row represents one customizable option.

This modular layout allows users to manage multiple functionalities without leaving the main screen, enhancing efficiency and minimizing task-switching. Event data collected using the tool are stored in formats compatible with the OpenStarLab packages, facilitating seamless integration with event prediction models.

The STE Label Tool is a critical component in the OpenStarLab pipeline for soccer analytics, enabling efficient data preprocessing and standardization for downstream model training and analysis. Its design and compatibility with OpenStarLab packages make it a valuable asset in advancing research in soccer event prediction and performance analysis.

### 3.2 UIED format

The Unified and Integrated Event Data (UIED) format is designed to standardize event data across various soccer datasets, enabling seamless integration and comparative analysis. It provides a unified schema for encoding game events, allowing analysts and machine learning models to work with a consistent data format regardless of the source. (The distinction between UIED and existing formats is discussed in Section 2.2.) This standardization is crucial for accurate event prediction and performance evaluation across datasets that may have distinct naming conventions and structures.

To address differences across popular soccer datasets, the UIED format includes a comprehensive mapping of event types from datasets such as GRF, StatsBomb, Wyscout, and DataStadium. Table 2 illustrates these mappings for common event types, including passes, shots, and dribbles. For instance, a “Short Pass” in UIED corresponds to various terms across datasets, such as “Ground Pass” in StatsBomb and “Simple Pass” in Wyscout. By unifying these event types under a single nomenclature,

UIED	GRF [36]	Statsbomb	Wyscout	DataStadium
Short Pass	Short Pass	Ground Pass Low Pass Half Start	Goal Kick Free Kick Simple Pass Hand Pass Head Pass Smart Pass Throw In	Direct FK Pass Indirect FK KickOff HomePass AwayPass PKPass Through Pass ThrowIn Feed
High Pass	High Pass	High Pass	High Pass	-
Long Pass	Long Pass	Ground Pass Low Pass	Goal Kick Free Kick Simple Pass	Direct FK Pass Indirect FK KickOff HomePass AwayPass PKPass Through Pass ThrowIn Feed
Shot	Shot	Shot	Shot Free Kick Shot	Shoot Direct FK Shot
Carry	Sprint	Carry	Acceleration	-
Dribble	Dribble	Dribble	Touch	Dribble Touch
Cross	-	Cross Corner	Cross Free Kick Cross	CK Cross

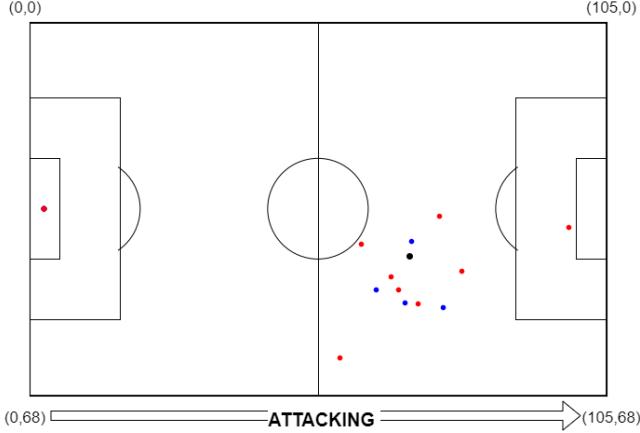
**Table 2** Event type mappings across different soccer event data format. Short Pass and Long Pass are classified based on a pass length threshold of 45 meters. Additionally, special event markers—such as `end_of_possession` (denoted as “-”), `period_over`, and `game_over`—are appended at the end of each possession, period, and game to signal these transitions.

UIED allows researchers and analysts to compare and aggregate events across sources without the need for manual re-labeling.

Additionally, UIED uses specific thresholds to differentiate event types. For example, passes are classified as either “Short Pass” or “Long Pass” based on a pass length threshold of 45 meters. Special event markers, such as `end_of_possession`, `period_over`, and `game_over`, are also included to signal key transitions in the match, facilitating timeline-based analysis.

Different soccer datasets tend to use different coordinate systems for event location. In the UIED format, the `start_x` and `start_y` coordinates represent the starting location of each event on the soccer pitch, standardized to a specific coordinate system for consistency. This coordinate system assumes a pitch width of 105 meters and a height of 68 meters, following FIFA’s (Fédération Internationale de Football Association) recommended field dimensions. The coordinates are scaled accordingly to fit within this layout, as illustrated in Figure 4.

The origin point, (0, 0), is defined at the top-left corner of the pitch, with:



**Fig. 4** UIED standardized pitch coordinates format. The red, blue, and black points represent the positions of the attacking team, defending team, and event location, respectively. The pitch is oriented such that the attacking direction is always from the left ( $x = 0$ ) to the right ( $x = 105$ ).

Variables	Type	Description
match_id	int	Unique identifier for each match.
poss_id	int	Unique identifier for each possession within a match.
team	str	The team associated with the event.
home_team	int	Indicator of whether the team is the home team (1: home, 0: away).
<b>action</b>	str	Simplified and standardized description of the event action.
success	int	Indicator of whether the event action was successful (1: success, 0: failure).
goal	int	Indicator of whether the event resulted in a goal (1: goal, 0: no goal).
home_score	int	The current score of the home team.
away_score	int	The current score of the away team.
goal_diff	int	The goal difference ( $\text{home\_score} - \text{away\_score}$ ).
Period	int	The period of the match (1: 1st half, 2: 2nd half, etc.).
Minute	int	The minute within the current period.
Second	float	The second within the current minute.
<b>seconds</b>	float	The total seconds elapsed since the start of the match, adjusted for different periods. (+15 minutes time for each period passed)
<b>delta_T</b>	float	The time difference between the current and previous events in seconds.
<b>start_x</b>	float	The x-coordinate of the event's starting location (scaled).
<b>start_y</b>	float	The y-coordinate of the event's starting location (scaled).
deltaX	float	The change in the x-coordinate from the previous event.
deltaY	float	The change in the y-coordinate from the previous event.
distance	float	The distance covered by the event.
dist2goal	float	The distance from the event's starting location to the center of the goal.
angle2goal	float	The angle between the event's starting location and the goal, in radians.

**Table 3** Available variables for UIED format data. The bolded variables are the most basic and common variable in the event prediction model and performance analysis, as shown in Table 1.

- The **x-axis** increasing horizontally from left to right, spanning from  $x = 0$  (left end) to  $x = 105$  (right end).
- The **y-axis** increasing vertically from top to bottom, spanning from  $y = 0$  (top end) to  $y = 68$  (bottom end).

This setup facilitates consistent analysis of event locations, player positioning, and ball movement across datasets. For example:

- The center of the pitch is located at approximately (52.5, 34).
- The **goal areas** are positioned near  $x = 0$  and  $x = 105$ , with events closer to these ends indicating attacking or defensive plays, depending on the team's perspective.
- Events in the **attacking direction** flow from the left to the right end of the pitch, as indicated by the "ATTACKING" arrow at the bottom of Figure 4.

This standardized system simplifies the evaluation of each event's spatial context, such as distance to the goal (calculated from `start_x` and `start_y` to the goal's coordinates) or the angle relative to the goal, which are essential metrics in shot analysis and predictive modeling.

Finally, the UIED format defines a comprehensive set of variables that capture essential aspects of each event. Table 3 provides an overview of these variables. Key variables in the UIED format, such as `action`, `seconds`, `delta_T`, `start_x`, and `start_y`, and derived variables are designed to support in-depth analysis of gameplay and facilitate robust model training. This standardized structure allows for precise evaluations of gameplay dynamics and player performance across datasets, forming a reliable foundation for advanced soccer analytics.

### 3.3 SAR Format

The State-Action-Reward (SAR) format standardizes event and tracking data across diverse soccer datasets, facilitating seamless integration and comparative analysis. Similar to the UIED format, this standardization is essential for accurate reward prediction and performance evaluation, especially when datasets use different naming conventions and structures.

The SAR format is available in datasets such as StatsBomb, SkillCorner, and DataStadium. While it maintains the same pitch width and height as the UIED format (Figure 4), it also aligns with the UIED format in both structure and direction of attack, which progresses from left to right on the pitch (Figure 4). However, its coordinate system differs, with the origin point positioned at the center of the pitch. The coordinate system in the SAR format is defined as follows:

- The **x-axis** increases horizontally from left to right, ranging from  $x = -52.5$  (left end) to  $x = 52.5$  (right end).
- The **y-axis** increases vertically from top to bottom, ranging from  $y = -34$  (top end) to  $y = 34$  (bottom end).

For tracking data, the velocity and acceleration of all players and the ball were calculated, with missing data supplemented where necessary. Event and tracking data were synchronized based on `frame_id`. Additionally, attack sequences were segmented using "attack\_history\_num", following the approach of a previous study [11]. By structuring data into distinct attack sequences, rather than treating an entire match as a single sequence, this method enables more granular learning and evaluation of offensive plays. To ensure consistency, only sequences with a minimum of 50 frames and a maximum of 300 frames were included in the dataset.

Variables	Type	Description
match_id	int	Unique identifier for each match.
frame_id	int	Unique identifier for each frame within a match.
team	str	The team associated with the event.
team_id	int	Unique identifier for each team.
home_team	int	Indicator of whether the team is the home team (1: home, 0: away).
player_name	str	The player associated with the event.
<b>player_id</b>	int	Unique identifier for each player.
jersey_number	int	The player jersey_number associated with the event.
player_role_id	int	The Identifier for each position of each player (1: GK, 2: DF, 3: MF, 4: FW).
action	str	Simplified and standardized description of the event action.
<b>action_id</b>	int	Unique identifier for each action.
success	int	Indicator of whether the event action was successful (1 for success, 0 for failure).
is_goal	int	Indicator of whether the event resulted in a goal (1: goal, 0: no goal).
is_shot	int	Indicator of whether the event is a shot (1: shot, 0: no shot).
is_pass	int	Indicator of whether the event is a pass (1: pass, 0: no pass).
is_dribble	int	Indicator of whether the event is a dribble (1:dribble, 0: no dribble).
is_cross	int	Indicator of whether the event is a cross (1: cross, 0: no cross).
is_through_pass	int	Indicator of whether the event is a through_pass (1: through_pass, 0: no through_pass).
is_ball_recovery	int	Indicator of whether the event is a recovery (1: recovery, 0: no recovery).
is_block	int	Indicator of whether the event is a block (1: block, 0: no block).
is_clearance	int	Indicator of whether the event is a clearance (1: clearance, 0: no clearance).
is_interception	int	Indicator of whether the event is a interception (1: interception, 0: no interception).
Period	int	The period of the match (1: 1st half, 2: 2nd half, etc.).
seconds	float	The total seconds elapsed since the start of the match, adjusted for different periods.
start_x	float	The x-coordinate of the player location when event's starting (scaled).
start_y	float	The y-coordinate of the player location when event's starting (scaled).
ball_x	float	The x-coordinate of the ball location when event's starting (scaled).
ball_y	float	The y-coordinate of the ball location when event's starting (scaled).
ball_touch	int	Only event valid as play 1 Ball out, fouls, etc. 0.
series_num	int	Sequential number of the sequence of in-play of the match.
history_num	int	No. of the history in chronological order of the play of the match (the start of the match is 1, and after that, the number is +1 or higher than the previous history No.).
<b>attack_history_num</b>	int	Number for a single series of attacks.
attack_start_num	int	First history No. within the same attack_history_num.
attack_end_history_num	int	Last history No. within the same attack_history_num.

**Table 4** Available variables for SAR format event data. The bolded variables are the most essential variable in the reward prediction model and performance analysis.

Finally, the SAR format defines a comprehensive set of variables capturing key aspects of each event. Table 4 provides an overview of the event data variables, while Table 5 details the tracking data variables. Key variables such as **player\_id**, **action\_id**, **attack\_series\_num**, **x**, and **y**, along with derived variables, are designed to support in-depth gameplay analysis and enhance model training.

Variables	Type	Description
match_id	int	Unique identifier for each match.
frame_id	int	Unique identifier for each frame within a match.
home_team	int	Indicator of whether the team is the home team (1: home, 0: away).
jersey_number_id	int	The player jersey_number associated with the event.
x	float	The x-coordinate of the player location scale by the field size.
y	float	The y-coordinate of the player location scale by the field size.

**Table 5** Available variables for SAR format tracking data. The bolded variables are the most essential variable in the reward prediction model and performance analysis.

### 3.4 Pre-processing package

The OpenStarLab Pre-processing package<sup>2</sup> is a comprehensive tool designed to standardize and streamline event data and tracking data in soccer across various providers. It allows users to easily convert raw event data from different formats into the Unified and Integrated Event Data (UIED) format or other formats used in event predictive studies [7, 9, 35, 37, 39, 40], and State-Action-Reward (SAR) format in reinforcement learning, providing a cohesive and standardized dataset for analysis.

#### 3.4.1 Key features

The key features of OpenSTARLab Pre-processing package are as follows:

- **Multi-Provider Compatibility:** The package supports multiple data providers, including DataFactory, DataStadium, Metrica, Opta, Robocup 2D Simulation, SoccerTrackv2 (BePro), Sportec, Statsbomb, Statsbomb with Skillcorner Tracking Data, and Wyscout, allowing seamless integration regardless of the original data format. For consistency, most providers' (see Section 3.7) data structures could be mapped to the UIED standard or the formats used in previous studies<sup>8</sup>.
- **Automated Preprocessing Pipeline:** With functions tailored to each provider, the package can handle large volumes of data by loading, parsing, and transforming it into the UIED format. Users specify parameters such as file paths and IDs, and the package handles the necessary transformations, including coordinate standardization, event categorization, and temporal adjustments.
- **Scalable Processing:** The package is built with scalability in mind, supporting parallel processing through the `max_workers` parameter. This allows users to customize the level of parallelization based on available computing resources, optimizing performance when dealing with large datasets.
- **UIED-Standardized Outputs:** The resulting data is output in the UIED format (detailed in Section 3.2), with standardized columns for match and possession IDs, event actions, spatial information, and game statistics. This uniform structure simplifies subsequent analysis and model training tasks by providing consistent data regardless of the source.
- **SAR Format Option:** The package also supports the SAR format 3.3 specifically tailored for deep learning tasks (in particular DRL by computing state, action, and

---

<sup>8</sup>Details of other formats: [https://openstarlab.readthedocs.io/en/latest/Pre\\_Processing/Sports/Event-data/Data\\_Format/Football/other.html](https://openstarlab.readthedocs.io/en/latest/Pre_Processing/Sports/Event-data/Data_Format/Football/other.html)

reward). By aligning event and tracking data at each time frame within a unified framework, it enables seamless integration and supports advanced deep learning applications.

### 3.4.2 Customization and flexibility

The OpenStarLab Pre-processing package allows users to specify provider-specific parameters and preprocessing options, offering flexibility for different project requirements. Its modular structure enables easy integration into larger workflows, while broad compatibility with various file formats ensures minimal setup effort.

Each provider's function parameters can be customized to match specific file paths or other requirements, producing standardized output ready for UIED-based analysis. For further details, refer to the documentation of OpenStarLab<sup>9</sup>. Below is an example demonstrating how to use the OpenStarLab Pre-processing package to convert Wyscout data [38] into the UIED format.

Python code for Example usage

```
from preprocessing import Event_data

wyscout_df = Event_data(data_provider="wyscout",
                        event_path="path/to/event/folder",
                        match_folder="path/to/match/folder",
                        preprocess_method="UIED",
                        max_workers=1).preprocessing()
```

Similar to the UIED-based analysis, the SAR-based analysis enables the customization of function parameters for each provider to accommodate specific file paths and other requirements, ultimately generating standardized output suitable for SAR-based analysis. For more details, please refer to the OpenStarLab documentation<sup>10</sup>. Below is an example demonstrating how to use the OpenStarLab Pre-processing package to convert DataStadium data [38] into the SAR format.

Python code for SAR Example usage

```
from preprocessing import SAR_data

datastadium_df = SAR_data(data_provider="datastadium",
                           event_path="path/to/event/folder",
                           tracking_h_path="path/to/tracking_h/folder",
                           tracking_a_path="path/to/tracking_a/folder",
                           max_workers=1).preprocessing()
```

<sup>9</sup>Comprehensive documentation on UIED data providers: [https://openstarlab.readthedocs.io/en/latest/Pre\\_Processing/Sports/Event\\_data/Data\\_Provider/Football/index.html](https://openstarlab.readthedocs.io/en/latest/Pre_Processing/Sports/Event_data/Data_Provider/Football/index.html)

<sup>10</sup>Comprehensive documentation on SAR data providers [https://openstarlab.readthedocs.io/en/latest/Pre\\_Processing/Sports/SAR\\_data/Data\\_Provider/index.html](https://openstarlab.readthedocs.io/en/latest/Pre_Processing/Sports/SAR_data/Data_Provider/index.html)

In summary, this package provides a powerful and efficient tool for soccer event and tracking data preprocessing, transforming disparate data sources into a standardized UIED format and SAR format for streamlined analysis and application in machine learning models.

### 3.5 Event modeling package

The OpenSTARLab Event package<sup>1</sup> is an essential toolkit for event prediction-based modeling, offering users streamlined functionalities for training, inference, simulation, and applications in the sports domain. This package is optimized to work with data formatted by the OpenSTARLab PreProcessing package, facilitating smooth integration and ensuring a consistent workflow across OpenSTARLab repositories.

#### 3.5.1 Key functionalities

The Event package provides comprehensive support for several event prediction models tailored for soccer analytics, including Seq2Event [7], NMSTPP [37], LEM [39], FMS [8], and the Majority Class (MAJ) model, a simple baseline that predicts the most frequent event class from the training data. In addition, the package offers a variety of functions that optimize the event modeling workflow. Key features include:

1. **Data Loading and Preprocessing Functions for UIED Format:** These functions facilitate the loading, cleaning, and organization of UIED data, ensuring seamless integration with new model architectures. They provide researchers and analysts with an efficient foundation for developing and testing event prediction models.
2. **Training and Hyperparameter Tuning Functions:** The package includes built-in training functions that support hyperparameter optimization using Optuna [69], one of the most widely used frameworks for hyperparameter tuning. For convenience, hyperparameter configurations can be easily specified in a simple YAML file, minimizing user effort. More details are provided in Appendix A.
3. **Inference and Simulation Functions:** Apart from traditional inference functions, the package provides simulation functions to simulate events with greedy selection (select the one with the highest probability) or probabilistic selection, supporting the application of counterfactual analysis. For example, in FMS [8], researchers can simulate how different actions or player movements might impact the game's outcome.
4. **Evaluation Metrics:** The package includes built-in functions to track and display key metrics such as loss function value, accuracy, F1-score, mean absolute error (MAE), and cross-entropy loss during training, inference, and simulation. These metrics are crucial for assessing model reliability and precision, enabling cross-model comparisons and a detailed evaluation of simulation performance, which was previously unavailable. More details are provided in Section 4.

#### 3.5.2 Application of the event model in Soccer

Event prediction models have a wide application in soccer analytics, particularly for generating visualizations and metrics that capture spatial patterns, possession

efficiency, and overall team performance. These Event Model applications enable a deep understanding of soccer gameplay dynamics, supporting data-driven decisions and performance evaluations tailored to the nuances of soccer. As a result, the package encompasses the most important applications, with available functionalities including<sup>11</sup>:

- **Heat Map of the Predicted Next Event Location [40]:** The event model allows analysts to generate heat maps predicting the location of subsequent events. By visualizing the probability distribution of the next event, this tool provides valuable insights into potential hotspots for player actions, helping teams to analyze spatial trends and refine strategies. For example, in defensive planning, understanding likely zones for opponent actions can guide positioning and movement planning. Similarly, offensive analyses can reveal opportunities to exploit open spaces based on opponent tendencies.
- **Holistic Possession Utilization Score (HPUS) and HPUS+ [37]:** The HPUS and HPUS+ metrics quantify possession effectiveness over time. HPUS captures the overall value of each possession within a match, while HPUS+ refines this assessment to reflect the realization of that possession value. These metrics are particularly useful for tracking team strategy across intervals, as they highlight periods of high or low possession productivity. For instance, a steady HPUS score may indicate a team's control over the game, while a fluctuating HPUS+ score could suggest varied effectiveness in creating goal-scoring opportunities.

Through segment-based visualization, HPUS and HPUS+ illustrate how teams capitalize on their possessions across the match duration, offering insights into momentum shifts, defensive or offensive efficiency, and situational adjustments. These metrics are effective tools for both in-match tactical evaluations and post-match performance summaries.

- **Possession Utilization Score (Poss-Util) and Poss-Util+ [7]:** The Poss-Util and Poss-Util+, enhance performance insights by measuring a team's success in converting possession into scoring chances. Poss-Util highlights a team's efficiency by comparing their possession time to meaningful actions, helping to gauge each team's productivity and strategic approach. An elevated Poss-Util score typically indicates efficient use of ball control to generate offensive opportunities, whereas a low score might suggest prolonged but unproductive possession phases.

This metric is also applicable to individual and comparative analyses, allowing analysts to track performance across multiple teams. By observing Poss-Util scores over a season, coaches can evaluate tactical changes, assess opponent weaknesses, and adjust training focus to optimize possession strategies.

In conclusion, the OpenSTARLab Event package is a comprehensive tool for event modeling in soccer, designed to facilitate the analysis and prediction of key in-game events. It supports various models, enabling users to generate visualizations such as heat maps and calculate advanced metrics. These metrics provide valuable insights into team performance, possession efficiency, and tactical strategies throughout a match.

---

<sup>11</sup>For more details on Event Model applications in soccer analytics: [https://openstarlab.readthedocs.io/en/latest/Event\\_Modeling/Sports/Soccer/application.html](https://openstarlab.readthedocs.io/en/latest/Event_Modeling/Sports/Soccer/application.html)

The package is a powerful resource for both analysts and researchers seeking to enhance their understanding of soccer gameplay through data-driven methodologies.

### 3.6 Reinforcement learning modeling (RLearn) package

The OpenSTARLab RLearns package<sup>4</sup> is a toolkit designed for reinforcement learning (RL) in multi-agent decision-making models, operating at each timestep. Similar to the Event Modeling package, RLearns is compatible with data formatted by the OpenSTARLab PreProcessing package, ensuring a streamlined and consistent workflow across OpenSTARLab repositories.

While RL models in soccer are diverse, the RLearns package focuses on multi-player RL at each timestep [11]. In contrast, many other RL models—such as event-driven single-agent models [10, 49–53]—are very similar to those in Event Modeling package, provided the data and output is formatted appropriately. In the context of imitation learning [61, 64] and behavioral modeling [65–68], these models solely utilize tracking data, and disregard the event data.

For these reasons, the RLearns package is particularly valuable in its ability to handle both event and tracking data of multi-player at each timestep, offering a comprehensive solution for RL in soccer. Key features of the RLearns package include:

1. **Data Loading and Preprocessing Functions for SAR Format:** These functions, similar to those for the UIED format in event modeling, facilitate the loading, cleaning, and organization of SAR data (see Section 3.3). This ensures seamless integration with new model architectures and facilitates data preparation for RL tasks.
2. **Modeling:** The provided baseline models include Multi-Layer Perceptron (MLP), Gated Recurrent Unit (GRU), and Long Short Term Memory (LSTM). While previous study [11] has focused solely on GRU, this study further examines the effectiveness of MLP or LSTM to enable a comparative analysis of whether the use of time-series data improves performance.
3. **Evaluation Metrics:** The package includes a built-in functionality to monitor key performance indicators, such as loss function values and action prediction accuracy, both during training and inference. The action prediction accuracy was evaluated using TorchMetrics, a library provided by PyTorch, to assess the model’s classification performance across 16 action types. Following [11], the Temporal Difference (TD) error was adopted as a loss function to evaluate the validity of the Q-values learned by the model.
4. **Visualization Function:** The package offers a visualization tool that enables users to observe evaluation metrics for continuous actions of multiple agents. This allows for quantitative evaluation alongside expert qualitative assessments, by visualizing the effectiveness of players’ actions in real time.

### 3.7 Limitations

For event labeling tools, when using broadcast video, the annotated coordinates may not be directly convertible to pitch coordinates with the OpenStarLab packages. However, many existing studies [70–74] and repositories provide methods for such

conversions, offering alternative solutions. For a fixed camera angle, pitch coordinate mapping can be achieved using simple homography estimation, as demonstrated in [16].

Regarding the UIED format, while the preprocessing package supports loading data from most major soccer data providers to generate a standardized event data version, full mapping to the UIED format is currently only supported for GRF [36], StatsBomb, Wyscout, and DataStadium. For instance, the RoboCup 2D data includes only pass events, making it impossible to categorize other types of events, such as shots and carry.

As for the event package, although significant effort has been made to optimize the code and models, performance may still be influenced by the specific version of PyTorch being used. Additionally, while the codebase is standardized, there is still room for further optimization. Despite these factors, the results should remain consistent and provide a fair basis for comparison across different setups.

## 4 Experiments and Results

In this section, the experimental setup and results for evaluating the proposed framework for soccer match data were presented. Section 4.1 introduces the datasets used in the event modeling experiments, including details on the Wyscout and StatsBomb datasets, which serve as the foundation for model training and evaluation. Section 4.2 outlines the preprocessing and configuration steps, including data preparation, model setup, and hyperparameter tuning. Section 4.3 provides a detailed comparison of the performance of different models, assessing error metrics for event predictions. Section 4.4 discusses the simulation performance of the best-performing model, focusing on event count, action classification, and predictive accuracy for spatiotemporal coordinates at each timestep. Sections 4.5, 4.6, and 4.7 described the RL experiments using RLearn package. Finally, Section 4.8 demonstrates the application of the event model with real-world soccer analytics.

### 4.1 Dataset in Event Modeling

In the event modeling experiments, two event datasets were used, each capturing actions performed by players in control of the soccer ball. These datasets document a variety of action types, such as passes, shots, fouls, and more. For each recorded action, additional details are provided, including the position on the soccer pitch (in (x, y) coordinates), the time of the event, and the outcome of the action. More detailed information about the datasets is provided below:

**Wyscout dataset:** It includes soccer match event data from the 2017/2018 season across the top five leagues: the Premier League, La Liga, Ligue 1, Serie A, and Bundesliga. This dataset was obtained from the WyScout Open Access Dataset [38] and is currently recognized as the largest publicly available collection of soccer match event data, aimed at advancing research in soccer data analytics. For computational efficiency, the Premier League was specifically chosen due to its high-intensity gameplay, which leads to greater variation between events.

**StatsBomb Dataset:** To test the models on the latest data, the 2023/2024 La Liga season dataset was purchased from StatsBomb, as La Liga teams frequently compete at the highest level, including winning the Champions League against teams from other major soccer leagues in the 2023/2024 season. However, due to terms and conditions, this dataset could not be published as part of this study. It includes all 380 matches played by the 20 La Liga teams.

## 4.2 Preprocessing and Configuration in Event Modeling

The two datasets in Section 4.1 were split into training, validation, and test sets with a 60/20/20 ratio. The data were then converted to the UIED format (see Section 3.2) using the OpenSTARLab Pre-processing package (see Section 3.4). Afterward, all models were trained with the OpenSTARLab Event package (see Section 3.5).

For each model, hyperparameter tuning was conducted with a maximum of 100 trials, and training was performed on an NVIDIA GeForce RTX 3090 GPU, with training for a single model taking approximately three to four days. Detailed configuration settings, training code, and pre-trained model parameters are available in the OpenSTARLab documentation<sup>12</sup>.

The prediction task for each event includes determining the action type, time, and (x, y) coordinates. To evaluate performance, the following metrics were utilized: accuracy and F1-score for action type prediction, when accounting for the class imbalance in action types, F1-score provides a more accurate reflection of performance. Mean Absolute Error (MAE) was used to assess the predictions of time and spatial coordinates, which offer an intuitive understanding of model accuracy within the context of a football match. In addition, the Floating-point operations per second (FLOPs) and number of parameters (Num Prams) were included to show the computational efficiency and model complexity of each approach, providing insights into their scalability and resource requirements when applied to large-scale soccer datasets.

The models evaluated in this study include the baseline majority class model (MAJ), Seq2Event [7], NMSTPP [37], LEM\_1 [39], LEM\_3 [40], and FMS [8]. The key distinction between LEM\_1 and LEM\_3 lies in their model input: LEM\_1 uses only the most recent event, whereas LEM\_3 leverages events from the three most recent timesteps. Finally, NMSTPP+360 [9, 35] was excluded from this evaluation due to its dependency on player coordinate data, which is not available in any current large public soccer datasets.

## 4.3 Comparison of event model performance

This section compared the performance of the existing event model with the Wyscout and StatsBomb datasets, as detailed in Section 4.1, using the preprocessing and configuration outlined in Section 4.2. The comparison provided a benchmark for future studies. A summary of the performance of various models on soccer event prediction was presented in Table 6.

---

<sup>12</sup>OpenSTARLab document on model training: <https://openstarlab.readthedocs.io/en/latest/Event-Modeling/Sports/Soccer/model.html>

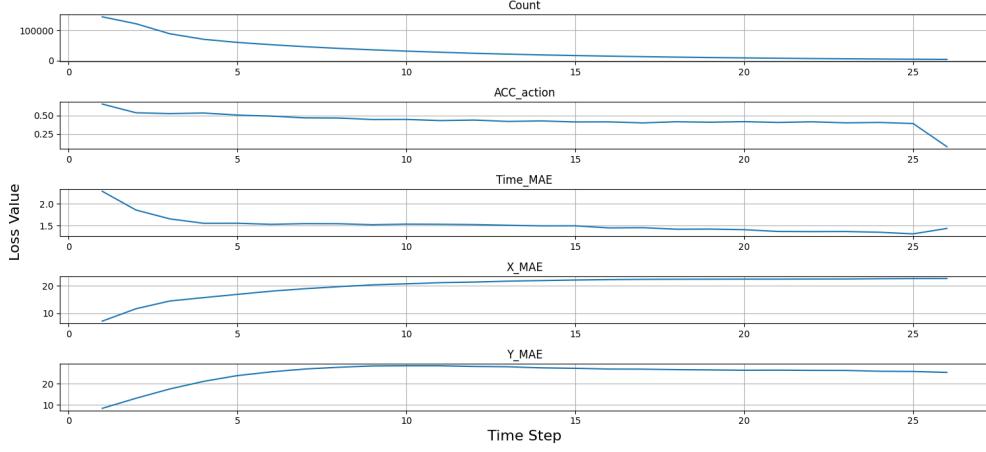
Model (Publication Year)	Action <sub>↑</sub> Acc.	Action <sub>↑</sub> F1	Time -MAE ↓	X-MAE ↓	Y-MAE ↓	FLOPs	Num Prams
<b>Wyscout Dataset</b>							
MAJ	0.57	0.08	3.60	18.97	52.55	-	-
Seq2Event (2022) [7]	0.67	0.16	3.41	7.11	15.72	112M	135K
NMSTPP (2023) [37]	0.67	0.17	3.34	<b>6.94</b>	<b>15.08</b>	296M	121K
LEM_1 (2024) [39]	<b>0.67</b>	0.17	3.07	8.34	21.44	50M	98K
LEM_3 (2024) [40]	0.67	<b>0.20</b>	<b>2.69</b>	7.62	21.83	20M	39K
FMS (2024) [8]	0.67	0.16	3.27	11.27	24.19	930M	782K
<b>StatsBomb Dataset</b>							
MAJ	0.40	0.06	2.76	20.72	33.32	-	-
Seq2Event (2022) [7]	0.65	0.23	2.43	7.22	6.86	4.03B	413K
NMSTPP (2023) [37]	0.65	0.23	2.53	7.38	<b>6.86</b>	2.02B	217K
LEM_1 (2024) [39]	0.65	0.24	2.23	7.36	8.21	66M	128K
LEM_3 (2024) [40]	<b>0.66</b>	<b>0.25</b>	<b>2.07</b>	<b>7.07</b>	8.32	19M	38K
FMS (2024) [8]	0.65	0.24	2.35	7.77	8.82	3.66B	1.29M

**Table 6** Comparison of model performance on soccer event prediction. Arrows indicate whether a higher ( $\uparrow$ ) or lower ( $\downarrow$ ) value is better for each metric. Notation: B = billion, M = million, K = thousand. The models are ranked by publication year. The bold value indicates the performance, determined by the unrounded value.

First, considering **Action Accuracy and F1-Score**, the LEM\_3 model achieved the highest performance for both metrics on both datasets. This trend indicated the effectiveness of LEM\_3 in handling soccer event prediction tasks, likely due to its use of three previous timestep events, which provided more contextual information compared to the LEM\_1 model, which only utilized one timestep. Additionally, the LEM\_3 model’s architecture proved more effective, as models like FMS, Seq2event, and NMSTPP, which used 9, 40, and 40 previous events respectively, also contributed to its superior performance. Overall, the accuracy of all models was similar, but the LEM models outperformed the others in terms of the F1-score, demonstrating better handling of imbalanced action classes.

In terms of **Time and Location MAE**, LEM\_3 also demonstrated superior performance in predicting the timing across both datasets and the spatial x-coordinate of events in the StatsBomb dataset. The NMSTPP model performed the best in predicting the xy coordinates but lagged behind LEM\_3 in the x-coordinate prediction on the StatsBomb dataset. When considering **Computational Efficiency** in terms of FLOPs and Parameter Count, the LEM models, particularly LEM\_3, were notable for their efficiency, with lower FLOPs and parameter counts compared to other models, which could require billions of FLOPs. LEM\_3’s efficiency highlighted its strong performance without excessive computational overhead, making it a practical choice for large-scale implementations and simulations.

Finally, a **Dataset-Specific Performance** trend emerged, where models generally achieved better action accuracy and lower MAE values on the StatsBomb dataset compared to Wyscout. This difference suggested that the seasons and the league of choice could influence the performance of the event prediction task. Overall, the LEM\_3 model demonstrated the best balance between performance and computational efficiency across both datasets.



**Fig. 5** Simulation performance of the LEM\_3 model. The exact values for the simulation are available at [https://github.com/open-starlab/Event/blob/main/event/sports/soccer/examples/simulation\\_loss.csv](https://github.com/open-starlab/Event/blob/main/event/sports/soccer/examples/simulation_loss.csv)

#### 4.4 Event Model simulation performance

In this section, the simulation performance of the best-performing model LEM\_3 [40] on the Statsbomb dataset (see Section 4.1) was evaluated across various metrics at each time step. The simulation utilized the greedy selection method (see Section 3.4), and a maximum time step of 26 was chosen, as approximately 98 percent of the possessions' number of events fall below this threshold. For each possession, if the model simulated more events than the actual number, the excess events were discarded. The evaluated metrics included event count, action classification accuracy (ACC\_action), and MAE for time and XY-coordinate predictions. The results are shown in Figure 5.

The **Event Count** decreased steadily over the time steps, reflecting a natural decay in the number of events during a soccer possession. In contrast, the **ACC\_action** dropped from 0.66 to 0.5 and remained relatively stable across most time steps, indicating that the model could consistently maintain its ability to classify 9 types of actions reasonably over time.

For the **Time\_MAE**, a noticeable decrease occurred in the early time steps, with the error stabilizing as the simulation progressed. However, this did not imply that the model improved its temporal predictions as the sequence moved forward, as the number of events decayed over time. In contrast, both the **X and Y MAEs** showed an increase over time, signaling a gradual decline in spatial prediction accuracy. The increase was more pronounced in the X MAE, suggesting that the model faced greater challenges in accurately predicting the lateral positions of events.

In summary, the simulation results demonstrated that the event model effectively maintained stable performance up until timestep 10, after which its performance worsened compared to the baseline MAJ model, as shown in Table 6. This highlighted that

while the model could predict events with reasonable accuracy in the short term, long-term predictions still had room for improvement, particularly in terms of enhancing spatial predictions.

#### 4.5 Dataset in RLearns Modeling

In the RL modeling experiments, J-League 2019-2020 tracking and event data were used. The event data consists of a dataset that captures actions performed by players controlling the soccer ball, similar to the dataset used in Event Modeling. It records various types of actions, such as passes, shots, and fouls. For each recorded action, details are provided, including the position on the soccer pitch (given as  $(x, y)$  coordinates), event time, and action outcomes. The tracking data captures the position (represented by  $(x, y)$  coordinates) and velocity (denoted by  $(v_x, v_y)$  coordinates) of all players at all times on the soccer pitch, along with their jersey numbers and home-/away status. This dataset contains both event and tracking data from 55 matches of the 2019 season Meiji J1 League soccer matches, purchased from Data Stadium Inc (see also [11]).

#### 4.6 Overview of the RLearns Modeling

This section explained about the overview of Rlearn modeling. The dataset was split into training, validation, and test sets with a 50/5/45 ratio. The data were then converted to the SAR format using the OpenSTARLab Pre-processing package (see Section 3.4). Afterward, all models were trained with the OpenSTARLab SAR package (see Section 3.6). The GRU, implemented using the torch library’s GRU module, was employed in the reinforcement learning model following the framework of a previous study [11], while the MLP consists of two fully connected layers and the LSTM was implemented using the torch library.

The RL models consists of state  $s$ , action  $a$ , and reward  $r$ . The state  $s$  consists of the position and velocity of all players and the ball, following a previous study [11]. The action  $a$  consists of a total of 16 types: pass, through pass, shot, cross, dribble, defensive action, movement in 8 directions (in 45-degree increments), and idle. Since this study focuses on the evaluation of offensive players, defensive actions such as interception, tackle, clearance, and block were grouped together as a single “defensive action” category. Regarding the reward  $r$ , we followed the approach of the previous study [11]. The reward was set to 0 at all time steps except for the final moment of the attacking sequence. At the final time step, the reward was set to 1 if the attacking sequence ended with a goal,  $-1$  if the opponent’s attacking sequence immediately afterward resulted in a conceded goal, and the expected possession value (EPV)<sup>13</sup>

In the RL models, the value function  $Q(s, a)$  is updated using five elements: the current state  $s$ , the action  $a$  selected in that state, the reward  $r$  obtained for that action, the next state  $s'$ , and the next action  $a'$  selected in that state. The discount factor  $\gamma$  was set to 1.0 in this study. This update equation is given by Equation 1.

---

<sup>13</sup>Calculation of EPV: <https://github.com/Friends-of-Tracking-Data-FoTD/LaurieOnTracking>

$$\begin{aligned} Q^*(s_t, a_t) &= E_{s_{t+1}, r_{t+1}}[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1})|s_t, a_t] \\ &= \sum_{r_{t+1}} P(r_{t+1}|s_t, a_t)r_{t+1} + \gamma \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t)Q(s_{t+1}, a_{t+1}). \end{aligned} \quad (1)$$

The loss function based on this update equation is the TD loss, expressed as follows:

$$L_{td} = \sum_{t \in T} (r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))^2. \quad (2)$$

Furthermore, following [11], we trained the model by minimizing three loss functions: the TD loss ( $L_{td}$ ), the action-supervised loss ( $L_{as}$ ), and the L1 regularization loss ( $L_{L1}$ ), as defined in Equation 3. The purpose of  $L_{td}$  is to minimize the difference between the Q-value obtained from the current state-action pair and the actual reward obtained from the next state.  $L_{as}$  aims to maximize the Q-value for the action observed in the data.  $L_{L1}$  is used to prevent overfitting due to the relatively small real-world dataset. The overall loss function is defined as follows:

$$L_{total} = L_{td} + \lambda_1 L_{L1} + \lambda_2 L_{as}. \quad (3)$$

Since  $L_{td}$  aims to learn the “optimal actions” that maximize rewards, its optimization direction may not always align with the actual actions taken by players. On the other hand,  $L_{as}$  encourages the agent to maintain specific actions. This difference creates a trade-off, making it crucial to balance the priority of these losses during the model’s training process. Therefore, in our experiments, we tested different hyperparameters,  $\lambda_1$  at 0.01, 0.005, and 0.001, while  $\lambda_2$  was set to 0.0001, following [11]. Lastly, we used the Adam optimizer with a learning rate of 0.001.

#### 4.7 Reinforcement learning modeling performance

This section presents a comparison between the performance of the MLP, GRU, and LSTM models. Table 7 summarized their action accuracy and TD loss results with different hyperparameter values. The experimental results indicate that, in a comparison of MLP and GRU, the MLP model outperformed in both action accuracy and TD loss under  $\lambda_1 = 0.01$  and  $0.005$ . This suggests that while GRU has the ability to capture long-term dependencies in time-series data, simpler models like MLP might have been more suitable for tasks such as soccer.

At  $\lambda_1 = 0.001$ , all models showed similar performance, with accuracy close to the expected accuracy of 0.0625 (Random selection of 16 actions). This suggests that when  $\lambda_1$  is small, action loss is under-optimized, causing the model to prioritize reward stability over action accuracy. This highlights the need to balance TD loss and action accuracy. At  $\lambda_1 = 0.005$ , LSTM performed similarly to MLP and GRU at  $\lambda_1 = 0.001$ . However, LSTM’s higher complexity may lead to vanishing gradients

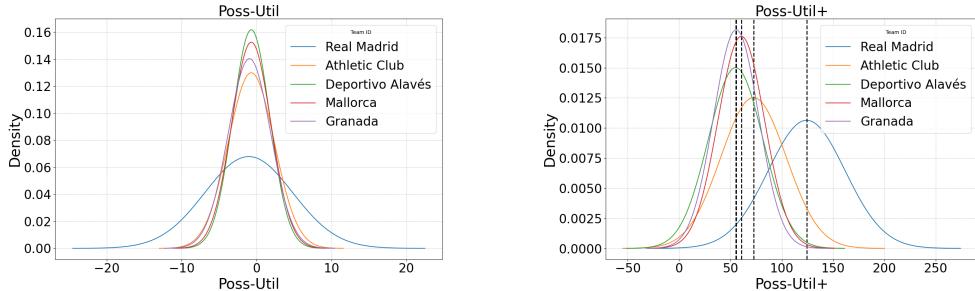
and irrelevant information accumulation, while MLP and GRU showed better action accuracy improvements with suitable  $\lambda_1$  adjustments.

The analysis of the impact of the  $\lambda_1$  revealed a trade-off relationship: increasing  $\lambda_1$  improved action accuracy but raised TD loss and destabilized reward prediction, while decreasing  $\lambda_1$  reduced action accuracy but lowered TD loss and enhanced reward stability. These results indicate that higher action accuracy does not necessarily improve reward prediction stability. This trade-off may stem from the reward function’s incomplete representation of player behavior.

Model	$\lambda_1$	Action Acc. $\uparrow$	TD Loss $\downarrow$
MLP	0.01	0.4939	1.8979
	0.005	0.1674	0.2406
	0.001	0.0714	0.0001
GRU	0.01	0.2607	2.1896
	0.005	0.1311	0.2552
	0.001	0.0714	0.0001
LSTM	0.01	0.1982	0.8730
	0.005	0.0714	0.0001
	0.001	0.0714	0.0001

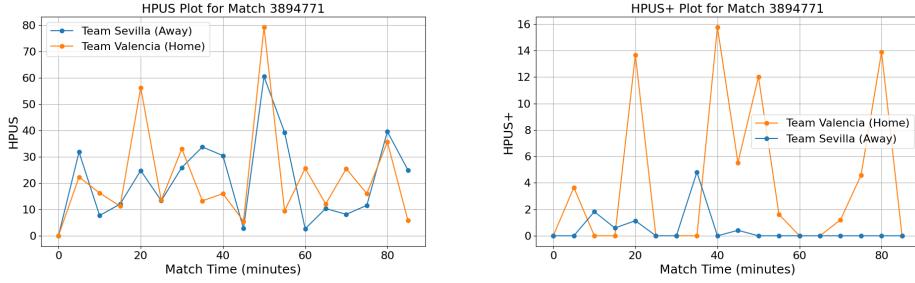
**Table 7** Comparison of Metrics for DataStadium. The value of  $\lambda_1$  represents the hyperparameter for each model. Arrows indicate whether a higher ( $\uparrow$ ) or lower ( $\downarrow$ ) value is preferred for each metric.

## 4.8 Application on soccer analytics

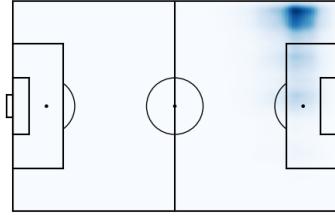


**Fig. 6** Poss-Util and Poss-Util+ distributions for LaLiga teams. Real Madrid, Athletic Club, Deportivo Alavés, Mallorca, and Granada, ranked 1st, 5th, 10th, 15th, and 20th, respectively.

This section demonstrated the application of the **LEM\_3 model** in soccer analytics, using the test set from the StatsBomb dataset. The analysis included **Poss-Util**, **HPUS**, a heatmap of predicted events (see Section 3.5 for more details on the metrics), and RLearn decision-making analysis.



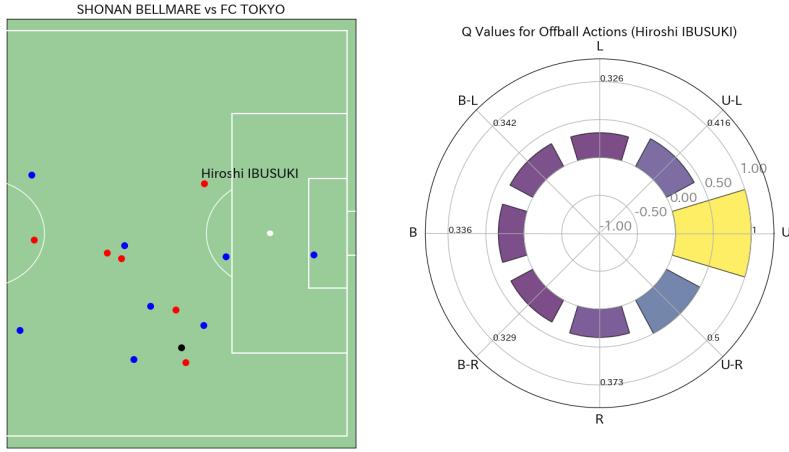
**Fig. 7** HPUS and HPUS+ for Valencia vs. Sevilla (0-0).



**Fig. 8** Heatmap of predicted cross event.

In the **Poss-Util analysis** (Figure 6), the Poss-Util and Poss-Util+ values were depicted for Real Madrid, Athletic Club, Deportivo Alavés, Mallorca, and Granada—teams ranked 1st, 5th, 10th, 15th, and 20th at the end of the season, respectively. In the left plot, Real Madrid demonstrated significantly higher variation in Poss-Util compared to other teams, reflecting their ability to consistently create possessions with high attacking potential. In the right plot, which focused on Poss-Util+ (positive Poss-Util values), Real Madrid displayed a higher mean and greater variation, emphasizing their superior attacking strength. In contrast, other teams showed relatively similar distributions for both Poss-Util and Poss-Util+, underscoring the gap between Real Madrid and even the 5th-ranked team.

In the **HPUS analysis** (Figure 7), the match between Valencia and Sevilla, which ended in a 0-0 draw, was examined. The left plot compared the HPUS values of the two teams, showing that both had similar capabilities in utilizing possession to generate potential attacks. However, the right plot revealed that Valencia was significantly better at converting potential attacks into actual attacks than Sevilla. This highlighted Valencia's superior attacking performance despite the match ending in a draw. Such insights provided analysts and fans with a deeper understanding of critical moments in the match and the overall performance of both teams.



**Fig. 9** A visualization of real-world soccer scenes and Q-values. The left figure plots the players and the ball. Attackers are plotted in red and defenders in blue. The right figure presents the Q-values (in 8 directions) of the off-ball actions of the player whose name is displayed in the left figure.

Finally, the **heatmap analysis** (Figure 8) presented a visualization of predicted cross events. The heatmap helped coaches and analysts identify the spatial distribution of likely events across the field. Unlike traditional methods that relied solely on historical event data, this approach offered a broader perspective on event probabilities, enabling more informed strategic planning. By understanding the patterns of potential events, coaches could prepare more comprehensive tactics beyond specific past instances.

Finally, the **RLearn Decision-Making Analysis** (Figure 9) provides a visualization of player positions and the Q-values for off-ball actions during the J1 League match between SHONAN BELLMARE and FC TOKYO. The Q-values for the players, whose names are displayed in the left diagram, are shown in the right figure. This figure is visualized in the model with  $\lambda_1 = 0.05$  for the RNN, which achieved a balanced trade-off between Action Accuracy and TD error based on the experimental results 7. In this scene, forward IBUSUKI has open space in the forward direction near the goal. Given that the Q-value in front of him is also high, it can be inferred that the model appropriately predicts rewards in this situation. This visualization aids coaches and analysts in evaluating the sequential behavior of all players throughout the match.

## 5 Conclusion

In this study, we introduced OpenSTARLab, an open-source framework designed to democratize soccer spatio-temporal agent data analysis by addressing key challenges in data accessibility, preprocessing, and modeling. This provides a Preprocessing Package

including UIED (event) format from different providers and SAR format specifically tailored for DRL tasks. For cases where annotated data is unavailable, OpenSTARLab also offers an STE label Tool for annotations. In the experiments, the comparative analysis revealed that the event modeling called LEM models [39, 40] consistently outperformed other event prediction models in terms of action and time prediction accuracies. Event simulation results demonstrated that the LEM\_3 model maintained robust event prediction performance up to the tenth timesteps. RL experiments highlighted a trade-off between action accuracy and temporal difference loss.

Overall, OpenSTARLab serves as a robust platform for researchers and practitioners, enhancing innovation and collaboration in the field of soccer data analytics. Future research can focus on enhancing libraries for player evaluation and spatial analysis, as well as extending the framework to encompass other sports. By improving usability, we aim to broaden the applicability of OpenSTARLab, making it a versatile tool for diverse sports domains. These advancements will facilitate more comprehensive performance assessments and strategic insights across various sporting contexts, providing greater innovation in sports analytics.

**Acknowledgments.** This work was financially supported by JST SPRING (Grant number JPMJSP2125), JST PRESTO (Grant number JPMJPR20CA), and NEDO Intensive Support Program for Young Promising Researchers (Grant number 24021654). The author Calvin Yeung would like to take this opportunity to thank the “Interdisciplinary Frontier Next-Generation Researcher Program of the Tokai Higher Education and Research System.”

## Declarations

- Funding: This work was financially supported by JST SPRING (Grant number JPMJSP2125), JST PRESTO (Grant number JPMJPR20CA), and NEDO Intensive Support Program for Young Promising Researchers (Grant number 24021654).
- Competing interests: The authors have no competing interests to declare that are relevant to the content of this article.
- Ethics approval: Not applicable
- Consent to participate: Not applicable
- Consent for publication: Not applicable
- Availability of data and materials: The data of the research is publicly available with details in [38]
- Code availability: The code for the study is available at <https://github.com/open-starlab>.
- Authors' contributions: All authors contributed to the study conception and design. Data preparation, modeling, and analysis were performed by all authors. The first draft of the manuscript was written by all authors and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

## Appendix A Example of training models with OpenSTARLab Event package

This section provides an example to train the NMSTPP model [37] and set up the YAML configuration for hyperparameter optimization using Optuna [69]. The following code snippets show how to initialize and train the NMSTPP model, followed by a YAML configuration template for optimizing hyperparameters with Optuna.

Python code for training the NMSTPP model

```
from event import Event_Model

% Initialize the NMSTPP model
model = Event_Model("NMSTPP", "path/to/NMSTPP_Configuration.yaml")

% Train the model
model.train()
```

### YAML configuration for Optuna hyperparameter optimization

```
# OpenStarLab Event Modeling, Apache-2.0 license
# UEID data and NMSTPP model

# Training parameters
train_path: /path/to/train.csv
valid_path: /path/to/valid.csv
save_path: /path/to/save
test: False

num_epoch: 50
print_freq: 10
early_stop_patience: 5
dataloader_num_worker: 4
device: None

# Input features
basic_features: ['action', 'delta_T', 'start_x', 'start_y']
other_features: ['team', 'home_team', 'success', 'seconds', 'deltaX',
                 'deltaY', 'distance', 'dist2goal', 'angle2goal']
use_other_features: True
num_actions: 9
seq_len: 40

# Model Hyperparameters
# (use lists to specify hyperparameters)
optuna: True
optuna_n_trials: 100

learning_rate: [0.01]
eps: [1e-16]
batch_size: [256]

action_embedding_out_len: [9]
scale_grad_by_freq: [True]
continuous_embedding_output_len: [12]

multihead_attention: [1]
hidden_dim: [16, 256, 512, 1024, 2048]
feature_embedding_output_len: [21]

NN_deltaT_num_layers: [1, 2, 3]
NN_location_num_layers: [1, 2, 3]
NN_action_num_layers: [1, 2, 3]
```

## References

- [1] Berrar, D., Lopes, P. & Dubitzky, W. A data-and knowledge-driven framework for developing machine learning models to predict soccer match outcomes. *Machine Learning* **113**, 8165–8204 (2024).
- [2] Yeung, C., Bunker, R., Umemoto, R. & Fujii, K. Evaluating soccer match prediction models: a deep learning approach and feature optimization for gradient-boosted trees. *Machine Learning* **113**, 7541–7564 (2024).
- [3] Yeung, C. C., Bunker, R. & Fujii, K. A framework of interpretable match results prediction in football with fifa ratings and team formation. *Plos one* **18**, e0284318 (2023).
- [4] Decroos, T., Bransen, L., Van Haaren, J. & Davis, J. Actions speak louder than goals: Valuing player actions in soccer. *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* 1851–1861 (2019).
- [5] Spearman, W. Beyond expected goals. *Proceedings of the 12th MIT Sloan sports analytics conference* 1–17 (2018).
- [6] Liu, G., Luo, Y., Schulte, O. & Kharrat, T. Deep soccer analytics: learning an action-value function for evaluating soccer players. *Data Mining and Knowledge Discovery* **34**, 1531–1559 (2020).
- [7] Simpson, I., Beal, R. J., Locke, D. & Norman, T. J. Seq2event: learning the language of soccer using transformer-based match event prediction. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* 3898–3908 (2022).
- [8] Baron, E., Hocevar, D. & Salehe, Z. A foundation model for soccer. *arXiv preprint arXiv:2407.14558* (2024).
- [9] Yeung, C. & Bunker, R. An events and 360 data-driven approach for extracting team tactics and evaluating performance in football. *Statsbomb Conference Proceedings* (2023).
- [10] Liu, G., Luo, Y., Schulte, O. & Kharrat, T. Deep soccer analytics: learning an action-value function for evaluating soccer players. *Data Mining and Knowledge Discovery* **34**, 1531–1559 (2020).
- [11] Nakahara, H., Tsutsui, K., Takeda, K. & Fujii, K. Action valuation of on-and off-ball soccer players based on multi-agent deep reinforcement learning. *IEEE Access* **11**, 131237–131244 (2023).
- [12] Giancola, S., Amine, M., Dghaily, T. & Ghanem, B. Soccernet: A scalable dataset for action spotting in soccer videos. *Proceedings of the IEEE Conference on*

*Computer Vision and Pattern Recognition Workshops* 1711–1721 (2018).

- [13] Delige, A. *et al.* Soccernet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* 4508–4519 (2021).
- [14] Cioppa, A. *et al.* Soccernet 2023 challenges results. *Sports Engineering* **27**, 24 (2024).
- [15] Cui, Y. *et al.* SportsMOT: A large multi-object tracking dataset in multiple sports scenes. *Proceedings of the IEEE/CVF International Conference on Computer Vision* 9921–9931 (2023).
- [16] Scott, A. *et al.* Soccertrack: A dataset and tracking algorithm for soccer with fish-eye and drone videos. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 3569–3579 (2022).
- [17] Scott, A. *et al.* Teamtrack: A dataset for multi-sport multi-object tracking in full-pitch videos. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 3357–3366 (2024).
- [18] Wu, T., He, R., Wu, G. & Wang, L. Sportshhi: A dataset for human-human interaction detection in sports videos. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 18537–18546 (2024).
- [19] Yeung, C., Ide, K. & Fujii, K. Autosoccerpose: Automated 3d posture analysis of soccer shot movements. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop* 3214–3224 (2024).
- [20] Van Roy, M., Cascioli, L. & Davis, J. Etsy: A rule-based approach to event and tracking data synchronization. *International Workshop on Machine Learning and Data Mining for Sports Analytics* 11–23 (2023).
- [21] Ballan, L., Bertini, M., Del Bimbo, A. & Serra, G. Semantic annotation of soccer videos by visual instance clustering and spatial/temporal reasoning in ontologies. *Multimedia Tools and Applications* **48**, 313–337 (2010).
- [22] Stein, M. *et al.* Director’s cut: Analysis and annotation of soccer matches. *IEEE computer graphics and applications* **36**, 50–60 (2016).
- [23] Alan, O., Akpinar, S., Sabuncu, O., Cicekli, N. & Alpaslan, F. Ontological video annotation and querying system for soccer games. *2008 23rd International Symposium on Computer and Information Sciences* 1–6 (2008).
- [24] Hosseini, M.-S. & Eftekhari-Moghadam, A.-M. Fuzzy rule-based reasoning approach for event detection and annotation of broadcast soccer video. *Applied Soft Computing* **13**, 846–866 (2013).

- [25] Assfalg, J., Bertini, M., Colombo, C., Del Bimbo, A. & Nunziati, W. Semantic annotation of soccer videos: automatic highlights identification. *Computer vision and image understanding* **92**, 285–305 (2003).
- [26] Wang, Z., Yu, J. & He, Y. Soccer video event annotation by synchronization of attack-defense clips and match reports with coarse-grained time information. *IEEE Transactions on Circuits and Systems for Video Technology* **27**, 1104–1117 (2016).
- [27] Theagarajan, R. & Bhanu, B. An automated system for generating tactical performance statistics for individual soccer players from videos. *IEEE Transactions on Circuits and Systems for Video Technology* **31**, 632–646 (2020).
- [28] Cioppa, A. *et al.* A context-aware loss function for action spotting in soccer videos. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 13126–13136 (2020).
- [29] Barra, S. *et al.* Footapp: An ai-powered system for football match annotation. *Multimedia Tools and Applications* **82**, 5547–5567 (2023).
- [30] Ma, S., Shao, E., Xie, X. & Liu, W. Event detection in soccer video based on self-attention. *2020 IEEE 6th International Conference on Computer and Communications (ICCC)* 1852–1856 (2020).
- [31] Sorano, D., Carrara, F., Cintia, P., Falchi, F. & Pappalardo, L. Automatic pass annotation from soccer video streams based on object detection and lstm. *Machine Learning and Knowledge Discovery in Databases. Applied Data Science and Demo Track: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part V* 475–490 (2021).
- [32] Cioppa, A. *et al.* Soccernet 2024 challenges results. *arXiv preprint arXiv:2409.10587* (2024).
- [33] Morra, L., Manigrasso, F. & Lamberti, F. Soccer: Computer graphics meets sports analytics for soccer event recognition. *SoftwareX* **12**, 100612 (2020).
- [34] Decroos, T., Bransen, L., Van Haaren, J. & Davis, J. Vaep: An objective approach to valuing on-the-ball actions in soccer. *IJCAI* 4696–4700 (2020).
- [35] Yeung, C., Bunker, R. & Fujii, K. Unveiling multi-agent strategies: A data-driven approach for extracting and evaluating team tactics from football event and freeze-frame data. *Journal of Robotics and Mechatronics* **36**, 603–617 (2024).
- [36] Kurach, K. *et al.* Google research football: A novel reinforcement learning environment. *Proceedings of the AAAI conference on artificial intelligence* **34**, 4501–4510 (2020).

- [37] Yeung, C., Sit, T. & Fujii, K. Transformer-based neural marked spatio temporal point process model for football match events analysis. *arXiv preprint arXiv:2302.09276* (2023).
- [38] Pappalardo, L. *et al.* A public data set of spatio-temporal match events in soccer competitions. *Scientific data* **6**, 1–15 (2019).
- [39] Mendes-Neves, T., Meireles, L. & Mendes-Moreira, J. Towards a foundation large events model for soccer. *Machine Learning* 1–23 (2024).
- [40] Mendes-Neves, T., Meireles, L. & Mendes-Moreira, J. Forecasting events in soccer matches through language. *arXiv preprint arXiv:2402.06820* (2024).
- [41] Eggels, H., van Elk, R. & Pechenizkiy, M. Expected goals in soccer: Explaining match results using predictive analytics. *The machine learning and data mining for sports analytics workshop* **16** (2016).
- [42] Toda, K., Teranishi, M., Kushiro, K. & Fujii, K. Evaluation of soccer team defense based on prediction models of ball recovery and being attacked: A pilot study. *Plos one* **17**, e0263051 (2022).
- [43] Umemoto, R., Tsutsui, K. & Fujii, K. Location analysis of players in uefa euro 2020 and 2022 using generalized valuation of defense by estimating probabilities. *arXiv preprint arXiv:2212.00021* (2022).
- [44] Vaswani, A. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).
- [45] Shchur, O., Turkmen, A. C., Januschowski, T. & Günnemann, S. Neural temporal point processes: A review. *IJCAI 2021* (2021).
- [46] Fujii, K. *et al.* Adaptive action supervision in reinforcement learning from real-world multi-agent demonstrations. *16th International Conference on Agents and Artificial Intelligence (ICAART' 24)* **2**, 27–39 (2024).
- [47] Fujii, K. Data-driven analysis for understanding team sports behaviors. *Journal of Robotics and Mechatronics* **33**, 505–514 (2021).
- [48] Fujii, K. *Machine Learning in Sports: Open Approach for Next Play Analytics* (Springer, 2025).
- [49] Van Roy, M., Robberechts, P., Yang, W.-C., De Raedt, L. & Davis, J. Leaving goals on the pitch: Evaluating decision making in soccer. *Proceedings of the MIT Sloan Sports Analytics Conference* (2021).
- [50] Van Roy, M., Robberechts, P., Yang, W.-C., De Raedt, L. & Davis, J. A markov framework for learning and reasoning about strategies in professional soccer. *Journal of Artificial Intelligence Research* **77**, 517–562 (2023).

- [51] Rahimian, P., Oroojlooy, A. & Toka, L. Towards optimized actions in critical situations of soccer games with deep reinforcement learning. *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)* 1–12 (2021).
- [52] Rahimian, P., Van Haaren, J. & Toka, T. A.-L. Beyond action valuation: A deep reinforcement learning framework for optimizing player decisions in soccer. *16th Annual MIT Sloan Sports Analytics Conference* (2022).
- [53] Rahimian, P., Van Haaren, J. & Toka, L. Towards maximizing expected possession outcome in soccer. *International Journal of Sports Science & Coaching* **19**, 230–244 (2024).
- [54] Liu, G. & Schulte, O. Deep reinforcement learning in ice hockey for context-aware player evaluation. *Proceedings of the 27th International Joint Conference on Artificial Intelligence* 3442–3448 (2018).
- [55] Schulte, O. *et al.* A markov game model for valuing actions, locations, and team performance in ice hockey. *Data Mining and Knowledge Discovery* **31**, 1735–1757 (2017).
- [56] Ding, N., Takeda, K. & Fujii, K. Deep reinforcement learning in a racket sport for player evaluation with technical and tactical contexts. *IEEE Access* **10**, 54764–54772 (2022).
- [57] Yanai, C., Solomon, A., Katz, G., Shapira, B. & Rokach, L. Q-ball: Modeling basketball games using deep reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence* **36**, 8806–8813 (2022).
- [58] Chen, X. *et al.* Reliable: Offline reinforcement learning for tactical strategies in professional basketball games. *Proceedings of the 31st ACM International Conference on Information & Knowledge Management* 3023–3032 (2022).
- [59] Luo, Y., Schulte, O. & Poupart, P. Inverse reinforcement learning for team sports: Valuing actions and players. *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20* 3356–3363 (2020).
- [60] Rahimian, P. & Toka, L. Inferring the strategy of offensive and defensive play in soccer with inverse reinforcement learning. *Machine Learning and Data Mining for Sports Analytics* 26–38 (2022).
- [61] Le, H. M., Yue, Y., Carr, P. & Lucey, P. Coordinated multi-agent imitation learning. *Proceedings of the 34th International Conference on Machine Learning-Volume 70* 1995–2003 (2017).
- [62] Le, H. M., Carr, P., Yue, Y. & Lucey, P. Data-driven ghosting using deep imitation learning. *Proceedings of MIT Sloan Sports Analytics Conference* (2017).

- [63] Teranishi, M., Fujii, K. & Takeda, K. Trajectory prediction with imitation learning reflecting defensive evaluation in team sports. *2020 IEEE 9th Global Conference on Consumer Electronics (GCCE)* 124–125 (2020).
- [64] Fujii, K., Takeishi, N., Kawahara, Y. & Takeda, K. Decentralized policy learning with partial observation and mechanical constraints for multiperson modeling. *Neural Networks* **171**, 40–52 (2024).
- [65] Zhan, E., Zheng, S., Yue, Y., Sha, L. & Lucey, P. Generating multi-agent trajectories using programmatic weak supervision. *International Conference on Learning Representations* (2019).
- [66] Yeh, R. A., Schwing, A. G., Huang, J. & Murphy, K. Diverse generation for multi-agent sports games. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 4610–4619 (2019).
- [67] Li, L. *et al.* GRIN: Generative relation and intention network for multi-agent trajectory prediction. *Advances in Neural Information Processing Systems* **34**, 27107–27118 (2021).
- [68] Fujii, K. *et al.* Estimating counterfactual treatment outcomes over time in complex multi-agent scenarios. *IEEE Transactions on Neural Networks and Learning Systems* 1–15 (2024).
- [69] Akiba, T., Sano, S., Yanase, T., Ohta, T. & Koyama, M. Optuna: A next-generation hyperparameter optimization framework. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2019).
- [70] Chen, J. & Little, J. J. Sports camera calibration via synthetic data. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops* (2019).
- [71] Sha, L. *et al.* End-to-end camera calibration for broadcast videos. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* 13627–13636 (2020).
- [72] Chu, Y.-J. *et al.* Sports field registration via keypoints-aware label condition. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 3523–3530 (2022).
- [73] Theiner, J. & Ewerth, R. Tvcilib: Camera calibration for sports field registration in soccer. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* 1166–1175 (2023).
- [74] Gutiérrez-Pérez, M. & Agudo, A. No bells just whistles: Sports field registration by leveraging geometric properties. *Proceedings of the IEEE/CVF Conference on*

