Deep soccer analytics: learning an action-value function for evaluating soccer players

Article *in* Data Mining and Knowledge Discovery · September 2020

DOI: 10.1007/s10618-020-00705-9

Guiliang Liu
Simon Fraser University

Oliver Schulte
Simon Fraser University

Noname manuscript No.
(will be inserted by the editor)

Deep soccer analytics: Learning an action-value function for evaluating soccer players

Guiliang Liu · Yudong Luo ✉ · Oliver Schulte · Tarak Kharrat

the date of receipt and acceptance should be inserted later

Abstract Given the large pitch, numerous players, limited player turnovers, and sparse scoring, soccer is arguably the most challenging to analyze of all the major team sports. In this work, we develop a new approach to evaluating all types of soccer actions from play-by-play event data. Our approach utilizes a Deep Reinforcement Learning (DRL) model to learn an action-value Q-function. To our knowledge, this is the first action-value function based on DRL methods for a comprehensive set of soccer actions. Our neural architecture fits continuous game context signals and sequential features within a play with two stacked LSTM towers, one for the home team and one for the away team separately. To validate the model performance, we illustrate both temporal and spatial projections of the learned Q-function, and conduct a calibration experiment to study the data fit under different game contexts. Our novel soccer Goal Impact Metric (GIM) applies values from the learned Q-function, to measure a player's overall performance by the aggregate impact values of his actions over all the games in a season. To interpret the impact values, a mimic regression tree is built to find the game features that influence the values most. As an application of our GIM metric, we conduct a case study to rank players in the English Football League (EFL) Championship. Empirical evaluation indicates GIM is a temporally stable metric, and its correlations with standard measures of soccer success are higher than that computed with other state-of-the-art soccer metrics.

Keywords Deep Reinforcement Learning · Action-Value Q-function · Goal Impact Metric · Fine-Tuning · Player Ranking

1 Introduction: Valuing Actions and Players

A major task of sports statistics is *player evaluation*, which provides insight into the performance of a player [Schumaker et al., 2010]. Performance evaluation is important for team

The final publication is available at link.springer.com `http://link.springer.com/article/10.1007/s10618-020-00705-9`
Corresponding Author: Yudong Luo
School of Computing Science, Simon Fraser University, and Sportlogiq Predictive Analytics
Burnaby, British Columbia, Canada
E-mail: yudong_luo@sfu.ca

management and fan engagement. For instance, fantasy leagues allow fans to draft or build their favourite team, based on the skills and the performance of players.

With the arrival of high-frequency tracking systems and object detection algorithms, ever more data on the movement of players in professional sports have become available. There is an increasing opportunity for large-scale machine learning to model complex sports dynamics and evaluate players' performances. Many evaluation metrics have been proposed in recent years. The most common approach has been to evaluate players via quantifying the values of the actions they took [McHale et al., 2012; Decroos et al., 2019].

Traditional sports evaluation metrics face two major problems: 1) Many player evaluation metrics (e.g., expected goals) focus only on the actions with *immediate* impact on goals, such as shots, but omit other actions that have significant long-term effects. This limitation is more severe when scoring is sparser; for example, soccer games are very likely to end with zero or one goal. 2) Traditional methods tend to assign fixed values to actions, regardless of the playing circumstances. To tackle these issues, Routley and Schulte [2015] built a Markov model to capture the game context for ice hockey and calculated a Q-value for each action. The Q-values estimate, for each action, the probability that a team scores the next goal after the action, given the current game context.

Soccer is arguably the most challenging to analyze of all the major team sports [Bornn et al., 2018]. The game context of soccer is even more complicated than that of ice hockey, given that soccer has more players (22 players), larger pitch (350 feet long and 150 feet wide) and longer playing time (90 minutes), all which lead to complex spatio-temporal distribution patterns for each team. In this paper, we apply *Deep Reinforcement Learning* (DRL) to learn an action-value Q-function from events in a soccer game. We introduce a stacked two-tower LSTM to capture the playing dynamics for home and away teams separately. Unlike the traditional *control* problem in reinforcement learning aiming to learn the optimal policy, we solve the *prediction* problem in the passive learning (on policy) setting.

Based on the learned Q-function, we introduce two metrics to measure the performance of players and theoretically justify their consistency. First, the Goal Impact Metric (GIM) ranks a player by aggregating the impacts of all his actions, where the *impact* of an action is the change of consecutive Q values due to this action. In empirical comparison with four comparison metrics, GIM shows the highest correlation with most standard success measurements. Generalizing from an initial sample of season matches, GIM is the best predictor of season total goals and assists. Second, an alternative to the action value approach is to compare a player to a random or league-average player (e.g., Cervone et al. [2014]). This compares the expected success (e.g. the number of team wins) between the situations where the player is fielded and the situation if the player is replaced by a random or average player. We adopt this idea to introduce a new approach for play-by-play data that defines a natural Q-value-above-average-replacement metric for player performance measurement. Our main theorem states that a player's Q-value-above-average-replacement gives the same score as their total action impact value. This means that the DRL framework unifies the two fundamental approaches to player evaluation; the plausibility of the average replacement approach supports our total action value metric (GIM).

To compute the action values for all players, we build a large dataset consisting of over 4.5M action events by pooling data from several soccer leagues. This dataset allows the model to learn general estimates for actions values. However, as the game context within a specific league may differ from that of the general soccer game, player assessment should be adjusted for different leagues. To address the trade-off between generalizing across leagues and specializing to a specific one, we propose a *fine-tuning* approach: beginning with the general model as an initialization, then train the model on the specific data from a certain

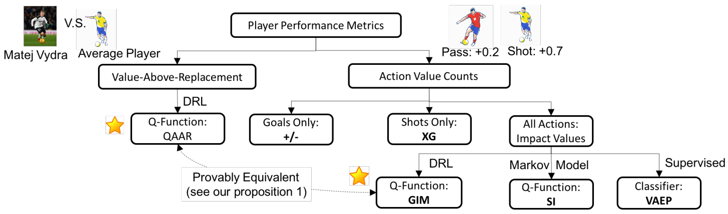

Fig. 1: A tree diagram to position our work in the research landscape. An important factor is whether a metric considers all actions or only a subset of them. Our approaches assign a value to *all* on-the-ball actions. Methods in bold are evaluated in our experiments and the star marks the proposed metrics.

league. Given the English Football League (EFL) Championship data, we use fine-tuning to improve the model's fitting performance as well as the evaluation results for players in this league.

Contribution. The main contributions of this paper can be summarized as follows.

1. The first neural Markov game model for soccer play-by-play event data. We utilize deep reinforcement learning to estimate a context-aware Q-function.
2. A novel two-tower neural network architecture to capture the spatio-temporal complexity of the home and away teams separately in a soccer game.
3. A fine-tuning approach that learns a general action value model from a very large dataset that combines different leagues, while capturing statistical patterns for specific leagues. While versions of fine-tuning have been applied in computer vision image datasets, to our knowledge, fine-tuning is new in deep sports analytics.
4. Two new soccer performance metrics based on the Q-function: Goal Impact Metric and Q-value-above-average-replacement (QAAR). To the best of our knowledge, QAAR is the first replacement-based metric for soccer play-by-play data. We prove that they are numerically identical, unifying the two fundamental approaches to player evaluation in an RL framework.

2 Related Work

2.1 Evaluating Soccer Players

The handbook by Albert et al. [2017] provides several up-to-date survey articles on player evaluation.

+/- (Plus-Minus) is a commonly applied player evaluation metric using *goals only*. It qualifies the influence of a player's presence on the goal scoring opportunity for his team. The basic version awards a player +1 if a goal is scored by the player's own team when the player is on the pitch, and -1 if the other team scores. Some recent works modify the basic plus-minus metric, by weighting the goals according to their importance, based on expected win probability, game time and game frequency [Schultze and Wellbrock, 2018], or with machine learning and survival models to estimate both expected goals and expected points to assess a player's overall defensive and offensive influence [Kharrat et al., 2019].

Expected Goals (XG) uses *shot information* to quantify the value of a shot by the probability of a goal given shot features (e.g. angle to goal). Players are ranked by their total expected goals [Ali, 2011]. Many recent works have applied a similar method to study passes rather than shots, where the quality of a player's passes is quantified by their influence on expected scoring opportunities. Passing is one of the most frequent actions in soccer. For each pass, Brooks et al. [2016] measured its value as the estimated probability of resulting in a successful shot. Bransen and Van Haaren [2018] measured its value as the difference between the goal-scoring probability before and after the pass. A drawback of these ratings is that they evaluate only one type of action without modeling a player's overall performance.

Several recent works rate players by evaluating *all* their actions. The *Expected Possession Value (EPV)* [Cervone et al., 2016] evaluated all the actions in basketball within a possession by estimating the expected number of points from the possession. Following this framework, Fernández et al. [2019] built a deep model from the full resolution spatiotemporal data to compute the EPVs for all actions during a game. They study the action impacts of individual soccer players under different game situations. Their approach requires tracking data, which assume the complete observability of all players. Many other play-by-play datasets, including ours, provide only partial observability of game context: they record only actions of the players who possess the ball at a given time. For on-ball action data, Decroos et al. [2019] introduced the VAEP (Valuing Actions by Estimating Probabilities) framework that evaluates all on-ball actions of soccer players based on their influence on the game outcome. However, instead of explicitly representing the game environment, their model considers a set of hand-crafted action features from the recent game history, and whether an action will lead to a goal within a constant number of future steps.

Another approach to evaluating players is quantifying their value-above-replacement (VAR). The most common VARs include *Goals/Wins Above Replacement (GAR/WAR)* which measure the player's contribution to his or her team by estimating the difference of team's scoring/winning chances when the target player is on the field, vs. compared to a replacement-level player. In this paper we take the replacement-level player to be a statistical league-average or random player. In other works, replacement-level represents a player of common skills available for minimum cost to a team.

2.2 Reinforcement Learning in Sport Analytics.

Reinforcement Learning (RL) models event data of the form $s_0, a_0, r_1, s_1, a_1, \dots, s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1}$: environment state s_t occurs, an action a_t is chosen, resulting in a reward r_{t+1} and state s_{t+1}. At the next time step, another action a_{t+1} is chosen. The data are often separated into local *transitions* of the form $T\{s, a, r', s', a'\}$. Reinforcement Learning has been applied to evaluating the actions of players. Schulte et al. [2017a] applied an ice hockey play-by-play dataset to build a Markov model, where actions record the player movements and states capture the game context. They measured players performance by their expected Scoring Impact (SI). The expected scoring probabilities of player actions under different game context are modeled by a Q-function using dynamic programming [Puterman and Patrick, 2017] based on the Bellman equation:

$$\begin{aligned} Q(s,a) &= \mathbb{E}_{s',a'}[r' + Q(s',a')|s,a] && (1) \\ &= \sum_{r'} \Pr(r'|s,a)r' + \sum_{s',a'} \Pr(s',a'|s,a)Q(s',a') && (2) \end{aligned}$$

This recurrence allows us to estimate the Q value at a current context s, a given an estimate for the next Q values and transition probabilities $\Pr$. Schulte et al. [2017a] discretized location and time coordinates, and used maximum likelihood estimates for the resulting discrete transition probabilities. The XThreat model is a discrete Markov model for soccer that divides the pitch into 192 zones and uses the Bellman equation to assess the expected scoring changes and resulting impact values [Van Roy et al., 2017]. The XThreat model considers only two action types, passes and dribbles. Discretization leads to loss of information and undesirable spatial-temporal discontinuities in the Q-function. The discontinuities prohibit the model from generalizing to the unobserved part of the state space.

Instead of explicitly modeling transitions in a discrete MDP, our work employs a *model-free approach* which learns Q values without explicitly estimating transition and reward probabilities [Sutton and Barto, 2018]. Many previous model-free RL works [Mnih et al., 2015] applied model-free learning with deep neural networks to capture continuous action and state features. These works mainly focused on *controlling* in continuous-flow games (e.g., Atari games). However, the real agents—players—in professional sports games are subject to evaluation, but not subject to control by an RL method.

Dick and Brefeld [2019] applied model-free RL to value match states in soccer according to the chance that the team currently in position will bring the ball close to the other team's goal. They assume tracking data (specifying the location and ball at each time step), rather than event data as our model does. Also, they did not apply the learned value function to assess player performance. To evaluate players performance, Liu and Schulte [2018] applied a deep recurrent model to capture the features of game history in ice hockey. Their model computes Q values to measure a player's expected probability of scoring the next goal with the Sarsa temporal difference learning method. Our work extends the approach of Liu and Schulte [2018] from ice hockey to a more complex model designed for the more complex sport of European soccer. We show can the resulting impact values can be interpreted through mimic learning and provide a theoretical justification for the learned impact values.

3 Dataset

Sports analytics uses several different formats of data: *box score* data, which provide total action counts per player and match (e.g., number of goals scored), *play-by-play* data, which are logs of discrete action events specifying various properties of the action (e.g. action type, acting player, time and location), and *tracking* data, which record the location of each player at dense time intervals (e.g. for every broadcast video frame, or more frequently with stadium cameras). In this paper, we utilize the F24 play-by-play soccer game dataset provided by Opta[1]. The dataset records the play-by-play information of game events and player actions for the entire 2017-2018 game season from multiple soccer leagues, including English Premier League, Dutch Eredivisie, EFL Championship, Italian Serie A, German Bundesliga, Spanish La Liga, French Ligue 1 and German Bundesliga Zwei. Table 3 shows dataset statistics. The dataset records the actions of on-the-ball players and the spatial and the temporal context features. The complete feature set is listed in Table 2. Table 1 lists a series of events describing a goal sequence for the home and away teams. The dataset utilizes adjusted spatial coordinates. Both the X-coordinates and Y-coordinates are adjusted to [0, +100]. The adjusted soccer pitch is shown in Figure 2, where play flows from left to right for either team. To adjust coordinates, we reverse them when the team in possession attacks towards the left, so in this case $X_{Adjusted} = -rescale(X)$ and $Y_{Adjusted} = -rescale(Y)$.

[1] https://www.optasports.com/

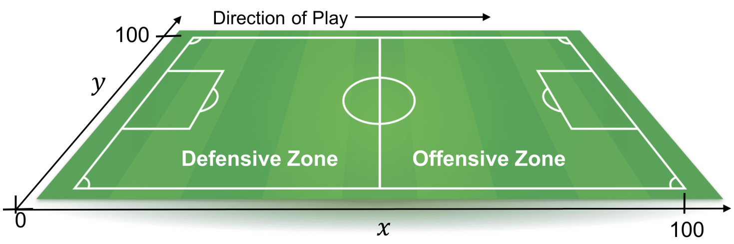

Fig. 2: Soccer pitch layout with adjusted coordinates. Coordinates are adjusted so that for the home/away team performing an action, its offensive zone is on the right

The adjusted coordinates accelerate model convergence during training and improve the model fit for spatial features (Section 6.1).

MP=Manpower, GD=Goal Difference, OC = Outcome, S=Succeed,
F=Fail, H=Home, A=Away, T=Team who performs action, GTR = Game Time Remain, ED = Event Duration

GTR	X	Y	MP	GD	Action	OC	Velocity	ED	Angle	T	Reward
35m44s	87	26	Even	1	simple pass	S	(2.2, 1.7)	11.0	0.19	H	[0,0,0]
35m42s	90	17	Even	1	standard shot	F	(1.5, -4.5)	2.0	0.11	H	[0,0,0]
35m42s	99	44	Even	1	save	S	(0, 0)	0.0	0.06	A	[0,0,0]
35m9s	100	1	Even	1	cross	S	(0.0, -1.3)	33.0	0.0	H	[0,0,0]
35m7s	85	56	Even	1	simple pass	S	(-7.3, 27.6)	2.0	0.39	H	[0,0,0]
35m5s	92	67	Even	1	simple pass	S	(3.6, 5.4)	2.0	0.28	H	[0,0,0]
35m4s	97	50	Even	1	corner shot	S	(5.1, -16.2)	1.0	1.74	H	[0,0,0]
35m4s	100	50	Even	1	goal	S	(0, 0)	0.0	0.0	H	[1,0,0]
....	...	...	....	...	....	...	....	...	.....	.	......
3m41s	62	96	Even	2	long ball	F	(4.5, 9.3)	9.0	0.08	A	[0,0,0]
3m39s	19	89	Even	2	clearance	S	(-21.5, -3.2)	2.0	0.07	H	[0,0,0]
3m35s	24	100	Even	2	throw in	S	(1.3, 2.7)	4.0	0.09	A	[0,0,0]
3m33s	27	96	Even	2	simple pass	S	(1.1, -2.2)	2.0	0.1	A	[0,0,0]
3m31s	12	95	Even	2	cross	S	(-7.5, -0.5)	2.0	0.07	A	[0,0,0]
3m28s	6	46	Even	2	simple pass	S	(-1.7, -16.3)	3.0	0.79	A	[0,0,0]
3m26s	14	48	Even	2	standard shot	S	(3.8, 1.3)	2.0	0.44	A	[0,0,0]
3m26s	0	50	Even	2	goal	S	(0, 0)	0.0	0.0	A	[0,1,0]

Table 1: A data sample featuring team scoring: a sequence of events where home team scores and then away team scores. The rewards [1,0,0] and [0,1,0] indicate the scoring event of home team and away team respectively (see Section 4.1). We skip some events in the middle due to space issues.

4 Modeling Play Dynamics

This section introduces our approach to defining a Markov model for soccer games and a Q-function to evaluate actions of players under different game context.

4.1 Markov Game Model for Sports Game

Similar to [Liu and Schulte, 2018], we apply the Markov Game Framework to model the play dynamics for sports games. The basic building blocks of the model are:

Name	Type	Range
Game Time Remaining	Continuous	[0, 100]
X Coordinate of ball	Continuous	[0, 100]
Y Coordinate of ball	Continuous	[0, 100]
Manpower Situation	Discrete	[-5, 5]
Goal Differential	Discrete	$(-\infty, +\infty)$
Action	Discrete	one-hot representation
Action Outcome	Discrete	{success, failure}
Velocity of ball	Continuous	$(-\infty, +\infty)$
Event Duration	Continuous	$[0, +\infty)$
Angle between ball and goal	Continuous	$[-\pi, +\pi]$
Home or Away Team	Discrete	{Home, Away}

Table 2: Complete feature list. For the feature manpower situation, negative values indicate short-handed, positive values indicate power play.

Dataset	F24
Events	4,679,354
Players	5,510
Games	2,976
Teams	164
Leagues	10
Season	2017-18
Place	Europe

Table 3: Dataset statistics. The basic unit of this dataset is *event*, which describes the game context and the on-the-ball action of a player at a time step.

- There are two agents, $Home$ and $Away$, representing their respective teams.
- The **action** a_t denotes the movements of players who control the ball. Our model applies a discrete action vector using one-hot representation.
- An **observation** is a feature vector x_t specifying a value of the features listed in Table 2 at a discrete time step t. We use the complete sequence $s_t \equiv (x_t, a_{t-1}, x_{t-1}, \ldots, x_0)$ to represent the **state** [Mnih et al., 2015].
- The **reward** r_t is a vector of goal values g_t that specifies which team ($Home, Away$) scores. We introduce an extra $Neither$ indicator for the eventuality that neither team scores until the end of a game. For readability, we use $Home, Away, Neither$ to denote the team in a 1-of-3 vector of goal values $r_t = [g_{t,Home}, g_{t,Away}, g_{t,Neither}]$ and $g_{t,Home} = 1$ indicates the home team scores at time t (see Table 1).

4.2 The Next-Goal Q-Function

Several value functions have been used to evaluate player actions. One option is to measure actions by whether they increase the winning chances [Routley, 2015]. More recent works focus on an action's more immediate impact regarding scoring points or goals [Cervone et al., 2016; Schulte et al., 2017b]. For soccer, we formalize this idea in terms of the **next-goal Q function**, which is defined as follows.

We divide a soccer game into **goal-scoring episodes**, so that each episode 1) starts at the beginning of the game, or immediately after a goal, and 2) terminates with a goal or at the end of the game. The next-goal Q-function represents the probability that the home resp. away team *scores the goal at the end of the current goal-scoring episode* ($goal_{Home} = 1$ resp. $goal_{Away} = 1$), or neither team scores ($goal_{Neither} = 1$):

$$Q_{team}(s, a) = P(goal_{team} = 1 | s_t = s, a_t = a) \tag{3}$$

where $team$ is a placeholder for one of $Home, Away, Neither$. This Q-function represents *the probability that a team scores the next goal*, given current play dynamics in a sports game [Schulte et al., 2017a; Routley and Schulte, 2015]. For player evaluation, the next-goal Q-function has several advantages over win probabilities.

- Compared to final match outcome, the Q values model the probability of scoring the next goal that is a relatively short time away and thus easier to explain and understand.

- Increasing the probability that a player's team scores the next goal *captures both offensive and defensive value*. For example, a defensive action like tackling decreases the probability that the other team will score the next goal, thereby increasing the probability that the player's own team will score the next goal.
- The next-goal reward captures what a coach expects from a player. For example, instead of thinking about how the game will end, a coach prefers his players to focus on defending against their opponent's strike and creating the next scoring opportunities at the moment.

5 Learning Q Values: Model Architecture and Training

This section introduces a neural network architecture and the weight training methods to learn a Q-function $(Q_{team}(s, a))$.

5.1 Model Architecture: Function Approximation with Neural Network

We discuss the model architecture for learning the Q values. Given a *discrete* state space, it is possible to use dynamic programming for computing Q-values [Schulte et al., 2017b; Van Roy et al., 2017]. But our soccer model contains continuous observation features derived from continuous time stamps and spatial locations. A common solution is to discretize spatio-temporal indices [Gudmundsson and Horton, 2017]. However, the resulting discontinuities undermine the precision of state values and impugn predictive accuracy. In this paper, we develop a neural network approach that can directly incorporate continuous observation features.

To generate Q-values, our model applies the two-tower design [Song et al., 2017] to fit the data of home/away teams separately and a recurrent neural network to capture the sequential features in play history. Figure 3 shows our model structure. The model fits home and away data separately, because from domain knowledge we expect the Q values to be different depending on whether a team plays at home or away (for a discussion of the home team advantage see [Swartz and Arce, 2014]). Each tower captures the play history with a stacked LSTM, which is a multi-layer LSTM, where outputs of LSTM cells in lower layers are used as the input for higher layers. Compared to the single layer LSTM, stacking adds levels of abstraction for the input features of sequences. This increases the model's ability to generalize across complex game contexts. The complete play history of game contexts and actions (s_t, a_t) is summarized in the last hidden state of the top LSTM layer. Our model uses a team identifier unit to select the hidden state from the home or the away tower according to who controls the ball in the current play. The selected hidden state values are sent to hidden layers whose outputs are normalized by a softmax function and considered as our estimates of $\hat{Q}_{Home}(s, a)$, $\hat{Q}_{Away}(s, a)$, and $\hat{Q}_{Neither}(s, a)$.

5.2 Weight Training

We train the two-tower neural network with an Temporal Difference (TD) prediction method **Sarsa** [Sutton and Barto, 2018, Ch.6.4] and apply a dynamic-possession LSTM to control the trace length during training. Our goal is to learn a function that estimates $Q_{team}(s, a)$ for the play dynamics observed in our dataset, with which we evaluate the performance of players. The training details are as follows.

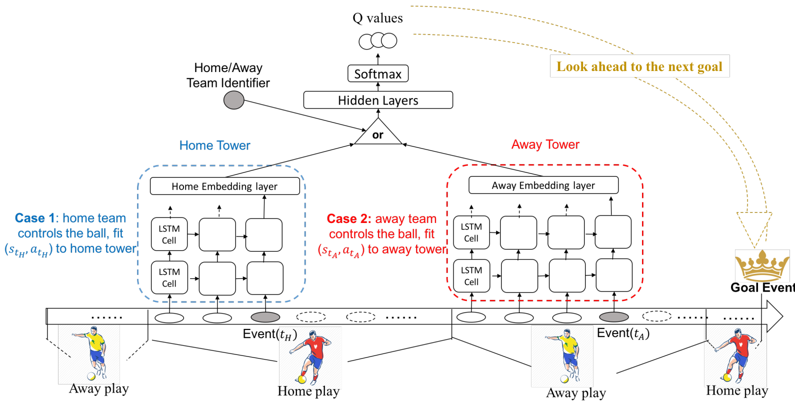

Fig. 3: The architecture of our Two-Tower Dynamic Play LSTM (TTDP-LSTM). The figure shows how the model processes two generic time instances, one associated with home team, is analyzed by the home tower, and the other from away team, is analyzed by the away tower.

Home/Away Tower Weight Training. At training time step t, our model feeds the output from the home/away tower to the hidden layers if the home/away team controls the ball at time t. During one training step, the hidden layers estimate the Q values for two continuous actions and states within one transition $T\{s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1}\}$. The estimated Q values are applied to compute the TD loss:

$$\mathcal{L}(\theta) = \sum_{team \in T} \mathbb{E}\left[\left(r_{team,t+1} + \hat{Q}_{team}(s_{t+1}, a_{t+1}) - \hat{Q}_{team}(s_t, a_t)\right)^2\right] \tag{4}$$

We use mini-batch gradient descent with backpropagation to find weights of our neural model that minimize this loss function (Figure 3). As for each transition, an error signal is sent only to either the home or the away tower, the flow of gradients will only influence one of the two towers and thus their weights are updated independently. This independence separates home and away signals and helps the network to learn their impact.

Dynamic Possession-LSTM. Team sports like soccer have a turn-taking aspect where one team is on the offensive and the other defends; one such turn is called a *play*. A play ends when possession passes from the team at time t to the opposing team at time $t+1$ [Liu and Schulte, 2018]. In a sports game, events within a play are highly correlated, but when a team loses control of the ball (meaning the play ends), the attacking team switches to defense. The dependence between actions from successive plays is therefore much weaker. The turn-taking aspect inspires a natural way of determining the trace length tl_t, which controls how far back in time the LSTM propagates the error signal from the current time at the input history. Instead of fixing the trace length, our model dynamically computes it and sets tl_t to the number of time steps from current time t to the beginning of the current *play* (with a maximum of 10 steps), so that the LSTM can restrict the history traces to the continuous possession of one team. Using possession changes to define episodes for temporal models has been proven to be successful in many continuous-flow sports, especially basketball [Cervone et al., 2016; Gudmundsson and Horton, 2017].

Training Settings. For our TTDP-LSTM model in Figure 3, both home and away towers apply a two-layer LSTM, whose outputs are sent to two hidden layers with three output nodes. The number of nodes in LSTM hidden states and hidden layers are both 256. The max trace length of LSTM is 10 [Hausknecht and Stone, 2015b]. During training, we minimize the loss function $\mathcal{L}(\theta)$ with Adam optimizer with an initial general learning rate of 10E-04 on the entire dataset (containing over 4.5M event data) and a fine-tuning initial learning rate of 10E-05 on the league-specific datasets.

Computational Complexity : Applying the neural network approximation function, the Sarsa prediction algorithm learns the Q function by updating the weights of a neural network through backpropagation. Our model applies a two-layer stacked LSTM with trace length 10 plus an embedding layer for each team and two hidden layers to generate the Q values. The sizes of hidden layers (or state) for both dense layers and LSTM cells are set to 256. Assuming we have m training examples in a batch and the dimension of input space is n, the time complexity of finishing training a neural network for one batch is therefore $\mathcal{O}(mn)$. While the cost of each training step is linear in the batch size, the number of gradient steps required until convergence depends on the dataset and the hyperparameter settings and cannot be bounded a prior.

6 Model Validation: Q Values

Our case studies illustrate the learned Q-function with temporal and spatial projections. To validate the model performance, we show that the learned Q values are well-calibrated, meaning that they offer a satisfactory fit to empirical scoring frequencies observed under different game contexts.

6.1 Illustration of Temporal and Spatial Projection.

Temporal Projection. We illustrate the estimated Q values for actions and states across game times. Figure 4 shows a *value ticker* [Cervone et al., 2016] that represents the evolution of the Q values during a randomly sampled game from our dataset. The figure plots values of the three output nodes representing $\hat{Q}_{Home}(s,a)$, $\hat{Q}_{Away}(s,a)$, and $\hat{Q}_{Neither}(s,a)$, according to which we highlight critical events to show the context-sensitivity of the Q-function. We observe that: 1) High scoring probabilities for one team decrease those of its opponent. 2) The probability that neither team scores rises significantly at the end of the match.

Spatial Projection. To study the influence of players' positions on scoring probability, we generate Q values for the entire soccer pitch. Our neural model can generalize from observed states and actions to those that have not occurred in the observed game season. Our model's generalization ability allows us to estimate a Q value for any action performed at any position. Figure 5 shows the learned smooth Q-function surface $\hat{Q}_{Home}(s,a)$ over possible game trajectories for several actions of the home team including shot, pass, cross, and tackle. We select these actions because they occur frequently and have been studied in previous work [Brooks et al., 2016; Van Haaren et al., 2016]. For the selected actions, we observe that the Q value of offensive actions like shots, passes, and crosses *increases with proximity* to the opponent's goal. The value of defensive tackling increases with proximity to the team's own goal. *Angles from the left side of the goal* appear slightly more promising than from the right. The plots for $\hat{Q}_{home}(s, pass)$ and $\hat{Q}_{home}(s, cross)$ show the same phenomena. An explanation for the first observation is that players have more chance to score when they approach their opponent's goal. For the second observation related to shot angle, inspection of our dataset reveals several goals scored on the upper corner (e.g. successful banana kick) but none on the lower corner. The left/right asymmetry also explains why the defensive action tackle made near the bottom left corner is more valuable (the last plot): tackles disturb opponents' actions that might lead to successful shots on their upper corner.

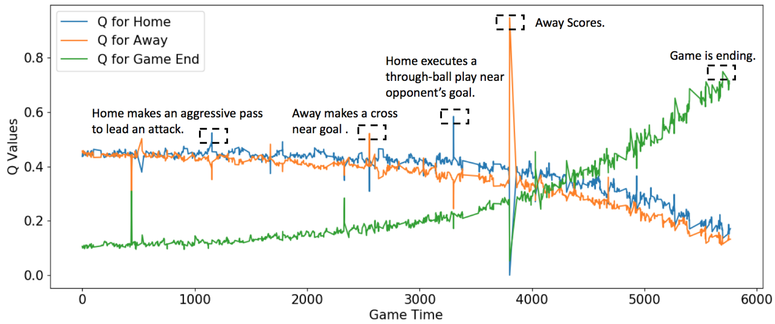

Fig. 4: Temporal Projection of the learned Q-function. The game is between Fulham (Home) and Sheffield Wednesday (Away), which has happened on Aug.19th, 2017.

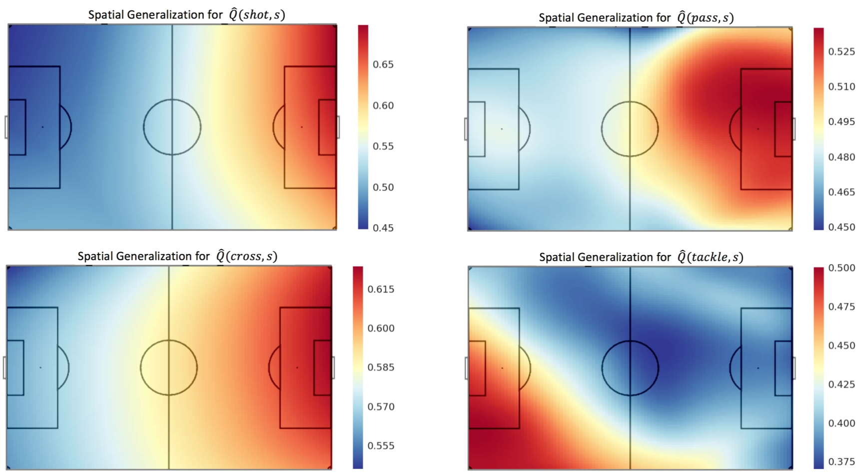

Fig. 5: Spatial Projections for estimated Q values: $\hat{Q}_{Home}(s, shot)$, $\hat{Q}_{Home}(s, pass)$, $\hat{Q}_{Home}(s, cross)$ and $\hat{Q}_{Home}(s, tackle)$ over the entire soccer pitch. We use the adjusted coordinate described in Section 3.

6.2 Calibration Quality for the learned Q-function

The calibration studies evaluate how well our learned Q-function fits the observed next-goal scoring frequencies under different game discrete contexts. Our approach to defining discrete game contexts is to divide the continuous state space into discrete bins. To calculate the empirical scoring frequency associated with each bin, we assign an observed state to a bin according to the values of three discrete *context features* in the last observation: Manpower (Short Handed (SH), Even Strength (ES), Power Play (PP)), Goal Differential (≤ -3, -2, -1, 0, 1, 2, $\geq$ 3) and Period (1 (first half), 2 (second half)). The total number of bins is $3 \times 7 \times 2 = 42$. This partition has two advantages. 1) The context features are well-studied and important for soccer experts [Decroos et al., 2019], so the model predictions can be checked against domain knowledge. 2) The partition covers a wide range of match contexts, and each bin aggregates a large set of play histories. If our model exhibits a systematic bias, the aggregation should amplify it and the bias should become detectable.

Given the set of bins where each bin A contains a total of $|A|$ states, the empirical and estimated scoring probabilities for each bin are defined as follows:

- *Empirical Scoring Probabilities* : for each observed state s, we set $goal^{obs}_{team}(s) = 1$ if the observed episode containing state s ends with a goal by team $team = Home, Away$ or neither ($team = Neither$). Then $Q^{obs}_{team}(A) = \frac{1}{|A|}\sum_{s \in A} goal^{obs}_{team}(s)$
- *Estimated Scoring Probabilities*: we apply our TTDP-LSTM model to estimate a Q value for each observed sequence and average the resulting estimates to compute the estimated scoring probabilities : $\hat{Q}_{team}(A) = \frac{1}{|A|}\sum_{s \in A} \hat{Q}_{team}(s, a)$

We evaluate the fit as the difference between the average empirical scoring probability $Q^{obs}_{team}(A)$ and the average estimated scoring probability $\hat{Q}_{team}(A)$. We show the results in Table 4 where the context features Manpower (Man.), Goal Differential (Goal.) and Period (P.) define a bin, and $|A|$ records the number of actions in each bin A in our dataset. The estimated Q-function matches several well-known phenomena: 1) The chance of either team scoring another goal decreases in the second period. 2) A clear *home team advantage* [Swartz and Arce, 2014]: Comparing two match contexts with the home and away team roles exchanged, the relative advantage of the home team is greater than that of the away team. 3) Manpower advantage by the home team means a lower scoring chance for the away team.

Our conclusions are as follows. 1) The model fit is satisfactory (i.e., the average MAE for all bins is below 0.1), except for some relatively rare game contexts. (For instance, the context where the home team is trailing with a manpower advantage in the first period, whose corresponding bin count is only 876 out of 3M match states). 2) Our model significantly outperforms the Markov Model with a discrete state space. This shows the advantage of a function approximation model that can utilize continuous space-time information without losing information due to discretization.

7 Player Evaluation Metric Based on Q values

In this section, we show how a player evaluation metric can be derived from the Q-function. Our paper's main approach to measuring player performance is assigning impact values (the difference between two consecutive Q values) to a player's action. To understand when the neural network will assign a high value to a player action, we fit a regression tree with the

Man.	Goal.	P.	$\|A\|$	TT_Home	TT_Away	TT_MAE	Markov_MAE
ES	-1	1	73176	0.4374	0.4159	0.0052	0.1879
ES	-1	2	96408	0.3496	0.3025	0.0782	0.1783
ES	0	1	356597	0.4437	0.4272	0.026	0.1908
ES	0	2	160080	0.356	0.3077	0.0814	0.1792
ES	1	1	88726	0.4402	0.4128	0.0335	0.1899
ES	1	2	119901	0.3459	0.295	0.077	0.1787
PP	-1	1	876	0.4366	0.4045	0.1752	0.1937
PP	-1	2	3319	0.352	0.2911	0.0668	0.1685
PP	0	1	3183	0.4414	0.403	0.1308	0.187
PP	0	2	7183	0.3579	0.2855	0.0841	0.1804
PP	1	1	1316	0.4391	0.3949	0.115	0.1825
PP	1	2	7676	0.356	0.2862	0.1121	0.1792

Table 4: Calibration Results. TT_Home and TT_Away report the average scoring probability $\hat{Q}_{team}(A)$ estimated by our TTDP-LSTM model. Here we compare only Q values for pass and shot as they are frequent and well-studied actions. TT_MAE is the Mean Absolute Error (MAE) between estimated scoring probabilities from our model and empirical scoring probabilities. For comparison, we also report a Markov_MAE which applies the estimates from a discrete-state Markov model [Schulte et al., 2017b].

state-action features and the corresponding impact values. To provide a theoretical foundation for our impact metric, this section introduces another Q-value-Above-Replacement metric to evaluate a player's action. By proving both metrics are equivalent, we show that Q-values unify the two main approaches to player evaluation.

7.1 Goal Impact: Deriving Action Values from Q-values.

Our Q-function concept provides a novel AI-based definition for assigning a value to an action. Similar to Schulte et al. [2017b]; Routley and Schulte [2015], we measure the quality of an action by how much it changes the expected total reward of a player's team: *the difference in expected total reward before and after the player acts*. The scoring chance at a time measures the value of a state, and therefore depends on the previous efforts of the entire team, whereas the change in value directly measures *the impact of an action by a specific player.* For our specific choice of Next Goal as the reward function, we refer to *goal impact*. The total impact of a player's actions is his **Goal Impact Metric** (GIM) value.

The following equations show how the action impact can be computed for a transition $T\{s, a, r', s', a'\}$ given Q value estimates from our TTDP-LSTM model. The expected future total reward before s', a' is given by $r' + \mathbb{E}_{s',a'}[Q_{team}(s', a')|s, a]$ (here the expectation is taken over all possible successor states and actions). The expected future total reward after s', a' is given by $r' + Q_{team}(s', a')$. Therefore:

$$\begin{aligned} impact^{team}(s, a, s', a') &\equiv Q_{team}(s', a') - \mathbb{E}_{s',a'}[Q_{team}(s', a')|s, a] \\ GIM^{i}(D) &\equiv \sum_{s,a,s',a'} n[s, a, s', a', pl' = i; D] \cdot impact^{team}(s, a, s', a') \end{aligned} \tag{5}$$

where D indicates our dataset, $team_i$ denotes the team of player i, and

$$n[s, a, s', a', pl' = i; D]$$

is the number of occurrences that player i performs action a' at s' after s, a. The Bellman equation (1) implies that $\mathbb{E}_{s',a'}[Q_{team}(s',a')|s,a] = Q_{team}(s,a) - \mathbb{E}[r'|s,a]$. The expectation can therefore be computed from estimated Q values given an expected rewards model. In our data, scoring a goal is represented as a separate action $goal$, after which no transition occurs. This means that for every transition $T\{s,a,r',s',a'\}$, we have $a \neq goal$, $r' = 0$ and thus $\mathbb{E}[r'|s,a] = 0$. So in this representation, *the impact equation* (5) *reduces to the difference in Q values before and after the player acts.*

7.2 Understanding Impact Values with Mimic Decision Tree

The impact values are computed with the Q-function, which applies a black-box neural network to fit the state-action features. To understand why some actions have large impacts under certain game contexts, we apply Mimic Learning [Ba and Caruana, 2014] and train a transparent regression tree (CART) to mimic the behavior of the deep model.

This interpretability study consists of two main steps. 1) We feed states and actions of the players as input into a CART to fit the resulting impact values via supervised learning. At each splitting node, CART automatically selects the feature that contributes the largest variance reduction to impact values on the child nodes. We split until one of the child nodes contains fewer than 80/90 samples for shot/pass respectively. 2) After tree learning, we compute the importance of a feature by summing the variance reductions at the splits applying this feature [Liu et al., 2018].

We rank the state and action features by their importance values. Tables 5 and 6 show the top 10 important features for shot and pass. Figure 6 and Figure 7 illustrate the structure of the CART trees by plotting its top three layers. The trees for both shot and pass impacts place at the root action outcome (a binary feature marking success or failure of an action), which intuitively is one of the most important action features. We also find that the shot impact significantly increases as a player approaches the goal, which is consistent with our finding in the spatial projection for Q values. For passing, its impact increases with game velocity. An explanation is that a quick pass prevents potential interruptions from opponents. When the game is close to the end, we observe that although the average passing impact decreases, the variance of impact among different passes significantly increases. Our CART in Figure 7 accurately locates the time when this phenomenon starts to occur (Time Remain (t-1)<39.45). Another important observation is that in addition to features from current time t, the historical features (e.g. X Coordinate (t-1)) are also considered as important for predicting the impact of the current action.

Feature	Influence
X distance (t)	0.6632
outcome (t)	0.2275
Y distance (t)	0.0469
Game Time Remain (t)	0.0242
duration (t)	0.0062
X Coordinate (t-1)	0.0059
Game Time Remain (t-1)	0.0035
interrupted (t)	0.0035
X velocity (t)	0.0030
outcome (t-1)	0.0019

Table 5: Feature influence for the impact of shot.

Feature	Influence
X Velocity (t)	0.1355
Distance to Goal(t)	0.1264
Game Time Remain (t-1)	0.1082
Game Time Remain (t)	0.0816
Outcome (t)	0.0773
Outcome (t-1)	0.0760
Distance to Goal (t-1)	0.0411
Angle (t)	0.0373
Angle (t-1)	0.0298
X Velocity (t-1)	0.0174

Table 6: Feature influence for the impact of pass.

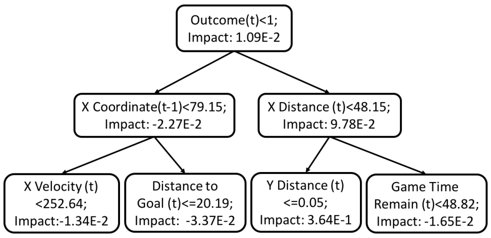

Fig. 6: Regression tree for the impact of shot.

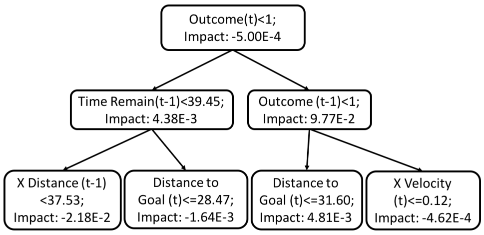

Fig. 7: Regression tree for the impact of pass.

7.3 Q Value Above Average Replacement

We compare the goal impact metric to deriving a player metric from a Q-function using an above-average-replacement framework. The fact that the same player performance ranking can be derived using two fundamentally different approaches supports the conceptual foundations of our metric.

The QAAR metric, compares the expected total future reward given that player i acts next, to the expected total future reward given that a random replacement player acts next:

$$QAAR^i(D) \equiv \sum_{s,a} n[s, a, pl' = i; D]\Big(\mathbb{E}_{s',a'}[Q_{team}(s', a'|s, a, pl' = i)] - \mathbb{E}_{s',a'}[Q_{team}(s', a')|s, a]\Big) \quad (6)$$

where $n[s, a, pl' = i; D]$ is the occurrence number that player i performs an action after s, a. The QAAR metric can be computed for a dataset by using the maximum likelihood estimates of transition probabilities. QAAR and GIM are natural definitions for the value-above-replacement and action-value approaches, respectively. Our main result is that they are equivalent:

Proposition 1 *For each player i recorded in our play-by-play dataset D, his Q-value-above-replacement is equal to his goal impact metric: $QAAR^i(D) = GIM^i(D)$.*

The complete proof is in our Appendix. This equation indicates that by summing a player's impact over an entire game season (GIM), we measure how much his general playing skill exceeds that of an average player (a replacement player with average Q-value) in the same league. Thus the same method for ranking players can be derived from a Q-function using two fundamentally different approaches. In the next section, we show some ranking examples by applying GIM to rate players.

8 Player Ranking: Case Study

To illustrate GIM, we discuss the ranking results for several players. We rank the EFL Championship players by their GIMs over the entire 2017-2018 game season. Our case study only ranks players in one league because they face the same level of competition and therefore their contributions are comparable. We chose the EFL Championship, which is just below the Premier League in the league hierarchy, because it has a large number of players in our data set and it has been much less studied than the Premier League.

Fine-Tuning. Different leagues have their own characteristics including competition level, season length, and playoff agenda. Therefore we apply a fine-tuning technique in order to achieve a better adaptation to the EFL Championship games.

1. Train a general model to evaluate actions in European soccer using games from multiple European Soccer leagues.
2. Fine-tune the initial weight values from the general model, with a smaller learning rate and using only EFL Championship game data.

Fine-tuning refines the general model and improves its ability to capture the behaviour of players. Compared to training the model from scratch, fine-tuning significantly reduces training time and prevents over-fitting. In the following assessment, we describe GIM values computed with the fine-tuned model and present both a general ranking for all actions and action-specific rankings.

8.1 All-Actions Assessment

Table 7 lists the 10 players with highest GIM for all actions. Our ranking includes the players with the most goals and assists. We investigate the positive correlation between our metric and standard success measures further in the next section.

Matej Vydra tops our 2017-2018 season ranking. He dominated the scoring board of the England Championship league and won the 2017-18 Golden Boot award[2]. In the next season (2018-2019), the Premier League team Burnley recognized the talent of Vydra and signed him on a three-year deal from team Derby.

Another example is Tom Cairney, who has only 5 goals and 5 assists over the entire season but ranks 6th in GIM assessment. Although he does not lead by any standard success statistics (Goals, Assists), his impact was an indispensable factor of his team's success in winning the 2017-18 EFL playoffs. For example, he scored the only goal of the final in which Fulham beat Aston Villa by 1-0 in the Wembley stadium and earned promotion to the Premier league. Tom Cairney was nominated as the EFL's Championship Player of the Season award[3].

name	team	GIM	Goals	Assists
Matej Vydra	Derby	18.017	21	4
Leon Clarke	Sheffield United	17.785	19	5
Lewis Grabban	Sunderland	16.045	12	0
Bobby De Cordova-Reid	Bristol	15.976	19	7
Diogo José Teixeira da Silva	Wolverhampton	15.707	17	5
Tom Cairney	Fulham	15.24	5	5
Ivan Cavaleiro	Wolverhampton	14.979	9	12
Stefan Johansen	Fulham	13.565	8	8
James Maddison	Norwich	13.23	14	8
Gary Hooper	Sheffield Wednesday	11.953	10	3

Table 7: 2017-2018 season top-10 Player Impact Scores for players in EFL Championship game season.

[2] https://www.skysports.com/football/news/11688/11361634/
[3] https://www.bbc.com/sport/football/43641225

name	GIM	Goal
Matej Vydra	4.747	21
Leon Clarke	4.024	19
Lewis Grabban	3.775	12
Kouassi Ryan Sessegnon	3.657	15
Harry Wilson	3.135	7
Famara Diedhiou	3.015	13
Sean Maguire	2.5	10
Joe Garner	2.44	10
Jarrod Bowen	2.408	14
Callum Paterson	2.29	10

Table 8: Top-10 players with largest shot impact in 2017-2018 EFL Championship game season.

name	GIM	Assist
Leon Clarke	8.05	5
Matej Vydra	5.957	4
Bobby De Cordova-Reid	5.134	7
Chris Wood	4.732	1
Gary Hooper	4.694	3
Ivan Cavaleiro	4.533	12
Diogo José Teixeira da Silva	4.283	5
Gary Madine	4.202	2
Tom Cairney	4.123	5
Conor Hourihane	4.042	2

Table 9: Top-10 players with largest pass impact in 2017-2018 EFL Championship game season.

8.2 Action-Specific Assessment

An action-specific ranking evaluates only the impacts of action of interest. We compute two GIM rankings of EFL Championship players by shots and passes respectively. These are frequent actions in soccer with high impact. Table 8 and Table 9 list the top 10 players. GIM computed from shots only can be seen as an alternative to the popular expected goals (XG) metric. A shot with high impact will significantly increase the probability of scoring and thus top players in Table 8 also lead the goal scoring. For instance, Matej Vydra is the player with the highest scoring impact and he also dominated goal scoring during the 2017-18 game season. However, the relation between pass impact and the number of assists is more complex. There is some association, because assists are often high-valued passes. On the other hand, the number of assists is an incomplete measure of passing ability because it neglects midfield and defensive zone passes. Our ranking, in contrast, provides a comprehensive evaluation to *all* the passes of a player. For example, Conor Hourihane plays as Midfielder and managed only 2 assists over the entire season. But he makes many influential passes and is ranked as a top-10 passer by our metric.

9 Player Ranking: Empirical Evaluation

We describe our comparison methods and evaluation methodology. Similar to clustering and recommendation problems, there is *no ground truth* for player ranking. To assess a player evaluation metric, we follow previous work [Routley and Schulte, 2015; Liu and Schulte, 2018] and compute its correlation with statistics that directly measure success.

9.1 Comparison Player Evaluation Metrics

We compare GIM with baseline player evaluation metrics to show the advantage of 1) modeling game context 2) incorporating continuous context signal and history 3) separately handling home and away state action signals.

Our baseline player evaluation metrics are as follows. *Goal-based Metrics.* i) Plus-Minus **(PM)** is a commonly studied metric that measures how much the presence of a player influences the goals of his team [Macdonald, 2011]. ii) Expected Goal **(XG)** weights each shot by its chance of leading to a goal. Players are ranked by their total expected goal shots. Both PM and XG consider only very limited game context and action types. The next three baselines assign an impact value to all actions and evaluate players according to their total action impact.

All-Action Metrics. iii) Valuing Actions by Estimating Probabilities **(VAEP)** [Decroos et al., 2019] applies the difference of action values to compute the impact of on-the-ball actions. Instead of applying Temporal Difference learning to estimate Q values, VAEP uses a classifier[4] to estimate the probability that an action leads to a goal within the next k (window size) steps. iv) Scoring Impact **(SI)** is based on a Markov model with *pre-discretized spatial and temporal features* (e.g. x,y coordinate and game time) [Schulte et al., 2017a]. Dynamic programming is applied to estimate a Q-function and impact values for the discrete state-action space. v) DP-LSTM is a neural network architecture that was previously applied to estimate action values for ice hockey. It applies a recurrent model to capture game context and TD learning to train the model [Liu and Schulte, 2018]. The difference with our TTDP-LSTM is that it *merges* the home/away towers and fits all the states and actions with a single-layer network. We refer to the resulting impact score as (**M-GIM**) for "merge".

A league-specific study evaluates our Fine-Tuning GIM (FT-GIM) for players in the EFL Championship. Training a separate model with only EFL Championship data from scratch consumes more computational resources than fine-tuning the general model. Our experiment records 4,386,894 gradient steps to learn a reliable model from initial weights while fine-tuning requires only 818,120 gradient steps.

Significance Test. To assess whether GIM is significantly different from the other player evaluation metrics, we perform paired t-tests over all players. The null hypothesis is rejected with respective p-values: 9.33E-2, 5.27E-281, 8.03*E-218, 4.82E-14 and 1.02E-118 for PlusMinus, XG, SI, VAEP and M-GIM. This shows that GIM values are different from the values of other metrics.

9.2 Season Totals: Correlations with Standard Success Measures

We report the correlations between player ranking metrics and commonly used success measures over the entire 2017-18 game season and highlight the comprehensiveness of our GIM metric. The examined success measures include Goals, Assists, Shots per Game (SpG), Pass Success percentage (PS%) and Key Passes per game (KeyP). We also study two penalty measures: Yellow card received (Yel) and Red card received (Red). Table 10 shows the correlations between the comparison methods and the success/penalty measures, for the players in all 10 leagues. In addition to the general study, Table 11 shows the result of a *league-specific evaluation* where we compare only the correlations for players in the EFL Championship.

Our GIM achieves very good correlations compared to the other methods. Among the positive success measures, GIM has the highest correlation with 4 out of 5 success measures (Goals, Assists, SPG, and KeyP) and a competitive result for the other (PS%). Together, the Q-function based metrics GIM, M-GIM, and SI show the highest correlations with success measures. XG is only the fourth best metric, because it considers only the expected value of shots and does not correct for the team effort leading up to the shot. VAEP achieves only limited correlation with the success measures. This is because their model assigns similar expected values to all actions, which translates into all action impact values being close to 0. The traditional Plus-Minus metric correlates poorly with almost all success measures. We conclude that RL techniques that provide fine-grained expected action value estimates lead to performance metrics that better match traditional success statistics.

[4] The classifier is implemented with a neural network rather than CatBoost in [Decroos et al., 2019] due to the size of dataset. We discuss our VAEP implementation further in the limitations (section 10.2).

Comparing the different RL approaches, the neural network model allows GIM to handle continuous inputs without pre-discretization. This prevents the loss of game context information and explains why both GIM and M-GIM perform better than SI in most success measures. The higher correlation of GIM compared to M-GIM also demonstrates the value of separately modeling home/away data. For Yel and Red which reflect the number of received penalties—negative contributions by a player—only our GIM-based metrics (GIM, M-GIM) show a negative correlation with both of them. The model correctly recognizes that a penalty will significantly reduce the scoring probability, influencing the overall player GIM. In contrast, other metrics focus on the actions that are likely to lead to goals, which tends to reward aggressive players who incur more penalties.

Methods	Goals	Assists	SpG	PS%	KeyP	Yel	Red
PM	0.284	0.318	0.199	**0.288**	0.218	0.001	-0.069
VAEP	0.093	0.290	0.121	-0.111	0.116	0.024	0.133
XG	0.422	0.173	0.328	0.164	0.278	0.534	0.034
SI	0.585	0.153	0.438	-0.140	0.052	0.114	-0.089
M-GIM	0.648	0.367	0.573	0.153	0.417	-0.110	<u>-0.145</u>
GIM	**0.844**	**0.498**	**0.596**	0.16	**0.562**	<u>-0.181</u>	-0.137

Table 10: Correlation with standard success measures for all the players. We bold the highest correlations and underline the lowest ones for penalties.

Methods	Goals	Assists	SpG	PS%	KeyP	Yel	Red
PM	0.262	0.223	0.122	0.155	0.112	0.033	-0.046
VAEP	0.08	0.26	0.116	-0.126	0.137	-0.015	0.215
XG	0.420	0.165	0.394	0.149	0.254	0.578	-0.021
SI	0.574	0.124	0.408	-0.144	0.054	0.084	-0.147
M-GIM	0.629	0.309	0.551	**0.171**	0.388	-0.039	-0.132
GIM	0.638	0.382	0.553	-0.053	0.468	-0.026	-0.105
FT-GIM	**0.736**	**0.585**	**0.569**	0.082	**0.592**	<u>-0.110</u>	<u>-0.171</u>

Table 11: Correlation with standard success measures for players in the EFL Championship. We bold the highest correlations and underline the lowest ones for penalties.

The league-specific study demonstrates the benefit of fine-tuning the deep reinforcement learning models. Compared to the correlations for players in all 10 leagues, Championship League players' correlations generally decrease. Both traditional action-count metrics (PM, XG) and impact-based metrics (VAEP, SI, GIM, M-GIM) show the decrease, but it is more severe for our GIM metric whose correlations nearly drop 20% when the players in the Championship League are evaluated by the general model. Fine-tuning addresses this issue: the FT-GIM metric achieves a larger negative correlation with penalty counts (Yel and Red).

9.3 Round-by-Round Correlations: Predicting Future Performance From Past Performance

These results assesses the player performance metrics through round-by-round correlations. A sports season can be divided into **rounds**. In round n, a team or player has finished n games in a season. For a given performance metric, we measure the correlation between (i) its value computed *over the first n rounds*, and (ii) the value of the two main success measures, assists, and goals, computed *over the entire season*. This allows us to assess how quickly different metrics acquire predictive power for the final season total, so that future performance can be predicted from past performance. A good performance metric should be consistent with a player's overall performance in the early season, which provides the player and his team with evidence for trading or training.

Figure 8 shows the round-by-round correlations for the players in all 10 leagues.[5] The predictive power of GIM grows more quickly than with any other baseline: its correlation with both assists (left) and goals (right) dominates others before the first half of the season. M-GIM achieves the second highest correlations, for assists even higher than GIM in the first 5 rounds. However, its predictive power substantially drops after the first 10 rounds. The remaining two metrics XG and SI show only weak correlations with assists and goals.

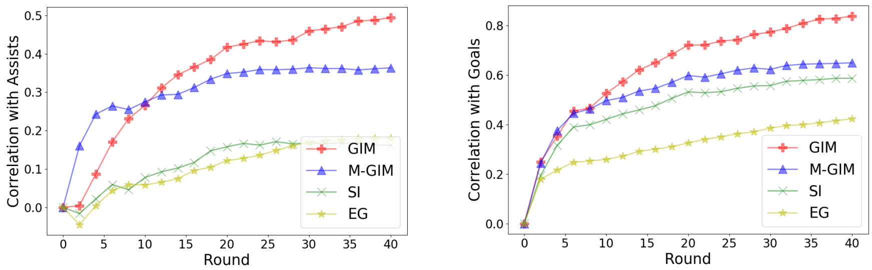

Fig. 8: Correlations between round-by-round metrics and season totals for all players.

The question for our next experiment is: *does fine-tuning help predict a player's final total performance from the past performance?* This experiment focuses on players in the EFL Championship. Figure 9 shows round-by-round correlations of the performance metrics with EFL Championship players' total assists and goals. We make the following observations. 1) Compared to the all-player setting of Figure 8, the metrics' correlations decline when restricted to EFL Championship players. This decline is more apparent for our GIM metric. The reason is that the neural network trained on the general player population does not fit the behaviour of players in the EFL Championship as well. 2) Fine-tuning significantly improves the correlations of GIM, especially for its correlation with assists, where the correlation of FT-GIM exceeds that of other metrics after the first 10 rounds.

10 Discussion

In this section, we discuss topics related to the sparsity of goals, model convergence and limitations of our method.

10.1 The Sparsity of Goals

A common method to evaluate soccer players' contribution is computing their influence on goal scoring. However, goals are rare in a soccer game. This issue is similar to the sparse

[5] In Figure 8 and 9, we omit players from teams that play less than 40 games in the 2017-18 season.

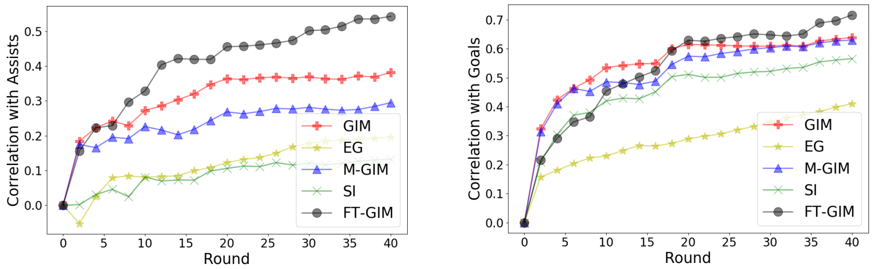

Fig. 9: Correlations between round-by-round metrics and season totals for the players in EFL Champion.

reward problem in Reinforcement Learning (RL). To address goal sparsity, many previous works on sport analytics suggested including other measures like assists, passes, and penalties in player evaluation. This is similar to *reward shaping* in RL, which adds some handcrafted indirect reward signals to accelerate training convergence Ng et al. [1999]. Reward shaping includes more information but raises the difficult issue of how to weight the relative importance of the indirect rewards (e.g., passes) of the real target reward (scoring). The Temporal Difference solution learns a Q-function that propagates the reward (scoring) signals to previous events, and assigns a value to all actions on the same expected rewards scale.

10.2 Model Convergence

We discuss the convergence of our TTDP-LSTM model. TTDP-LSTM is trained by the on-policy Temporal Difference (TD) method Sarsa. Previous work has guaranteed the convergence of on-policy TD with linear function approximators [Tsitsiklis and Van Roy, 1997]. However, in this paper, we apply a non-linear neural network function approximator. It is well-known that on-policy TD with a non-linear function approximator often exhibits unstable convergence in the traditional RL setting, when the action-value Q function is defined as the expected cumulative rewards with unlimited look ahead:

$$Q(s_t, a_t) = \mathbb{E}[\sum_{i=t}^{\infty} \gamma^{i-t} \cdot r(s_i, a_i)].$$

Here $\alpha \in (0, 1)$ is the discount factor and r is the reward function. To alleviate the instability of TD methods, in this work, we constrain the look-ahead to the next goal (rather than the end of game) and remove the discount factor, so $Q(s_t, a_t) = \mathbb{E}[r(s_T, t_T)]$ which is the expected scoring probability of the next goal. This is valid because, as discussed in Section 7.1, the reward $r(s_t, t_t) = 0$ except at goal occurrences T.

10.3 Limitations

We show some limitations in this work and discuss some potential solutions.

Partial observability for the players on pitch. At each time step, our dataset records only positions and actions of the player controlling the ball. The locations of the off-ball players are not known. The information of other players however, has influence on scoring probabilities, especially for a complex team sport like soccer. To alleviate this issue, our TTDP-LSTM model applies a recurrent model to fit the play history and includes the information of previous on-the-ball players. It has been previously observed in reinforcement learning that incorporating action history compensates for partial observability to some extent, because the model can infer missing current information from past information McCallum [1996]; Hausknecht and Stone [2015a]. For example, current player locations can be predicted to some extent from past player locations. Nonetheless, the model performance is limited due to partial observability. A direction for future work is to build a multi-agent reinforcement learning framework that combines fully observable tracking data with event categories. A possible approach is to combine the deep RL tracking model of Dick and Brefeld [2019] with our event-based deep RL model.

The problem of big input data. Our dataset has over 4M events including spatial and temporal features of players. Fitting the entire data requires substantial computational resources. The scalability challenges increase when we include the play history. Therefore it is difficult to utilize standard machine learning packages (such as decision tree, random forest or gradient boosting) that typically assume the entire data can be fit into a single working memory batch. In this work, we build a neural network with mini-batch gradients. In future work, we will explore *on-line* learning methods and evaluate their performance on big sports data. In addition to improving scalability, on-line methods are well-suited to sports data as teams want to update player assessments after every round.

11 Conclusion

This paper investigated Deep Reinforcement Learning (DRL) to learn complex spatio-temporal dynamics for professional soccer analytics. We designed a neural network architecture that, to our best knowledge, is the most complex deployed in sports analytics to date: A stacked two-tower LSTM architecture, with one tower each for home and away teams. The network was trained with on-ball action logs from several European leagues, comprising a total of over 4.5M action events. The trained neural network provides a rich source of knowledge about how a team's chance of scoring the next goal depends on the match context.

Based on the learned action values, we developed a new context-aware performance metric GIM for soccer players, taking *all* their actions into account. In our experiments, GIM computed over the entire season showed the highest correlation with most standard success measures. Generalizing from a sample of season matches, GIM was the best predictor of season total goals and assists. To improve the evaluation results for players in a specific league, we applied a fine-tuning approach to achieve an effective balance between generalizing across leagues and specializing to a specific league. Directions for future work include incorporating tracking data and developing on-line deep RL methods.

Deep RL methods have enjoyed spectacular success in board games. Our results show that the analysis of physical team sports is another highly promising application area.

Acknowledgements

This work was supported by Strategic Project Grant from the National Sciences and Engineering Council of Canada, and a GPU donation from NVIDIA Corporation. We are indebted for helpful discussion and comments to Norm Ferns and Bahar Pourbabee from Sportlogiq.

References

Albert J, Glickman ME, Swartz TB, Koning RH (2017) Handbook of Statistical Methods and Analyses in Sports. CRC Press

Ali A (2011) Measuring soccer skill performance: a review. Scandinavian journal of medicine & science in sports 21(2):170–183

Ba J, Caruana R (2014) Do deep nets really need to be deep? In: Advances in neural information processing systems, pp 2654–2662

Bornn L, Cervone D, Fernandez J (2018) Soccer analytics: Unravelling the complexity of “the beautiful game”. Significance 15(3):26–29

Bransen L, Van Haaren J (2018) Measuring football players’ on-the-ball contributions from passes during games. In: MLSA-ECML Workshop, Springer, pp 3–15

Brooks J, Kerr M, Guttag J (2016) Developing a data-driven player ranking in soccer using predictive model weights. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp 49–55

Cervone D, D’Amour A, Bornn L, Goldsberry K (2014) Pointwise: Predicting points and valuing decisions in real time with NBA optical tracking data. In: 8th Annual MIT Sloan Sports Analytics Conference, February, vol 28

Cervone D, D’Amour A, Bornn L, Goldsberry K (2016) A multiresolution stochastic process model for predicting basketball possession outcomes. Journal of the American Statistical Association 111(514):585–599

Decroos T, Bransen L, Haaren JV, Davis J (2019) Actions speak louder than goals: Valuing player actions in soccer. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019., pp 1851–1861

Dick U, Brefeld U (2019) Learning to rate player positioning in soccer. Big data 7(1):71–82

Fernández J, Barcelona F, Bornn L, Cervone D (2019) Decomposing the immeasurable sport: A deep learning expected possession value framework for soccer. In: MIT Sloan Sports Analytics Conference

Gudmundsson J, Horton M (2017) Spatio-Temporal Analysis of Team Sports. ACM Comput Surv 50(2):22:1–22:34, DOI 10.1145/3054132

Hausknecht M, Stone P (2015a) Deep recurrent Q-learning for partially observable MDPs. CoRR, abs/150706527

Hausknecht MJ, Stone P (2015b) Deep recurrent q-learning for partially observable mdps. In: 2015 AAAI Fall Symposia, Arlington, Virginia, USA, November 12-14, 2015, pp 29–37

Kharrat T, McHale IG, Peña JL (2019) Plus–minus player ratings for soccer. European Journal of Operational Research

Liu G, Schulte O (2018) Deep reinforcement learning in ice hockey for context-aware player evaluation. In: Proceedings of the Twenty-Seventh International Joint Conference on Ar-

tificial Intelligence, IJCAI-18, International Joint Conferences on Artificial Intelligence Organization, pp 3442–3448
Liu G, Zhu W, Schulte O (2018) Interpreting deep sports analytics: Valuing actions and players in the NHL. In: International Workshop on Machine Learning and Data Mining for Sports Analytics, Springer, pp 69–81
Macdonald B (2011) A regression-based adjusted plus-minus statistic for NHL players. Journal of Quantitative Analysis in Sports 7(3):29
McCallum A (1996) Learning to use selective attention and short-term memory in sequential tasks. In: From animals to animats 4: proceedings of the fourth international conference on simulation of adaptive behavior, MIT Press, vol 4, p 315
McHale IG, Scarf PA, Folker DE (2012) On the development of a soccer player performance rating system for the english premier league. Interfaces 42(4):339–351
Mnih V, Kavukcuoglu K, Silver D, et al. (2015) Human-level control through deep reinforcement learning. Nature 518(7540):529–533
Ng AY, Harada D, Russell S (1999) Policy invariance under reward transformations: Theory and application to reward shaping. In: ICML, vol 99, pp 278–287
Puterman ML, Patrick J (2017) Dynamic programming. In: Encyclopedia of Machine Learning and Data Mining, pp 377–388
Routley K (2015) A markov game model for valuing player actions in ice hockey. Master's thesis, Simon Fraser University
Routley K, Schulte O (2015) A markov game model for valuing player actions in ice hockey. In: Uncertainty in Artificial Intelligence (UAI), pp 782–791
Schulte O, Khademi M, Gholami S, Zhao Z, Javan M, Desaulniers P (2017a) A markov game model for valuing actions, locations, and team performance in ice hockey. Data Mining and Knowledge Discovery pp 1–23
Schulte O, Zhao Z, Javan M, Desaulniers P (2017b) Apples-to-apples: Clustering and ranking nhl players using location information and scoring impact. In: Proceedings MIT Sloan Sports Analytics Conference
Schultze SR, Wellbrock CM (2018) A weighted plus/minus metric for individual soccer player performance. Journal of Sports Analytics 4(2):121–131
Schumaker RP, Solieman OK, Chen H (2010) Research in sports statistics. In: Sports Data Mining, Integrated Series in Information Systems, vol 26, Springer US, pp 29–44
Song Y, Xu M, Zhang S, Huo L (2017) Generalization tower network: A novel deep neural network architecture for multi-task learning. arXiv preprint arXiv:171010036
Sutton RS, Barto AG (2018) Reinforcement learning: An introduction. MIT press
Swartz TB, Arce A (2014) New insights involving the home team advantage. International Journal of Sports Science & Coaching 9(4):681–692
Tsitsiklis JN, Van Roy B (1997) Analysis of temporal-diffference learning with function approximation. In: Advances in neural information processing systems, pp 1075–1081
Van Haaren J, Van den Broeck G, Meert W, Davis J (2016) Lifted generative learning of markov logic networks. Machine Learning 103(1):27–55
Van Roy M, Robberechts P, Decroos T, Davis J (2017) Valuing on-the-ball actions in soccer: A critical comparison of xt and vaep. In: Workshop on Team Sports AAAI 2020

A Proof of Proposition 1

The data record transitions from a state-action-player triple to another, possibly resulting in a non-zero reward (score or point in the context of sports). We denote the number of times such a transition occurs as

$$n_D[s, a, pl, s', a', pl']$$

where the $'$ indicates the successor triple. We freely use this notation for marginal counts as well, for instance

$$n_D[s', a', pl'] = \sum_{s,a,pl} n_D[s, a, pl, s', a', pl']$$

From the paper, we have the following equations for the Q-value-above-replacement and the GIM metrics:

$$QAAR^i(D) = \sum_{s,a} n_D[s, a, pl' = i]\big(\mathbb{E}_{s',a'}[Q_{team}(s', a'|s, a, pl' = i)] - \mathbb{E}_{s',a'}[Q_{team}(s', a')|s, a]\big) \tag{7}$$

$$GIM^i(D) = \sum_{s,a,s',a'} n[s, a, s', a', pl' = i; D] \cdot \Big[Q_{team}(s', a') - \mathbb{E}_{s'_E,a'_E}[Q_{team}(s'_E, a'_E)|s, a]\Big] \tag{8}$$

Now we have

$$\begin{aligned}
GIM^i(D) &\overset{Eq.2}{=} \sum_{s,a}\sum_{s',a'} n_D[s, a, s', a', pl' = i]\Big(Q_{team}(s', a') - \mathbb{E}_{s'_E,a'_E}[Q_{team}(s'_E, a'_E)|s, a]\Big) \\
&= \sum_{s,a} n_D[s, a, pl' = i] \sum_{s',a'} \frac{n_D[s, a, s', a', pl' = i]}{n_D[s, a, pl' = i]} Q_{team}(s', a') \\
&\quad - \sum_{s,a} n_D[s, a, pl' = i]\, \mathbb{E}_{s'_E,a'_E}[Q_{team}(s'_E, a'_E)|s, a] && (9) \\
&= \sum_{s,a} n_D[s, a, pl' = i] E[Q_{team}(s', a'|s, a, pl' = i)] && (10) \\
&\quad - \sum_{s,a} n_D[s, a, pl' = i]\, \mathbb{E}_{s'_E,a'_E}[Q_{team}(s'_E, a'_E)|s, a] \\
&= \sum_{s,a} n_D[s, a, pl' = i]\big(\mathbb{E}_{s'_E,a'_E}[Q_{team}(s'_E, a'_E|s, a, pl' = i)] - \mathbb{E}_{s'_E,a'_E}[Q_{team}(s'_E, a'_E)|s, a]\big) \\
&\overset{Eq.1}{=} QAAR^i(D) && (11)
\end{aligned}$$

Step (9) holds because the expectation $E[Q_{team}(s', a'|s, a)]$ depends only on s, a, not on s', a'. Line (10) uses the empirical estimate of the expected Q-value $Q_{team}(s', a')]$ given that player i acts next, computed from the maximum likelihood estimates of the transition probabilities:

$$\hat{\sigma}(s', a'|s, a, pl' = i) = \frac{n_D[s, a, s', a', pl' = i]}{n_D[s, a, pl' = i]}$$

The final conclusion (11) applies Equation (7).