PIVOT: A Parsimonious End-to-End Learning Framework for Valuing Player Actions in Handball using Tracking Data

Oliver Müller[1], Matthew Caron[1], Michael Döring[1,2], Tim Heuwinkel[1], and Jochen Baumeister[1]

[1] Paderborn University, Paderborn, Germany
{oliver.mueller, matthew.caron, michael.doering, tim.heuwinkel, jochen.baumeister}@uni-paderborn.de
[2] SG Flensburg-Handewitt, Flensburg, Germany

Abstract. Over the last years, several approaches for the data-driven estimation of expected possession value (EPV) in basketball and association football (soccer) have been proposed. In this paper, we develop and evaluate PIVOT: the first such framework for team handball. Accounting for the fast-paced, dynamic nature and relative data scarcity of handball, we propose a parsimonious end-to-end deep learning architecture that relies solely on tracking data. This efficient approach is capable of predicting the probability that a team will score within the near future given the fine-grained spatio-temporal distribution of all players and the ball over the last seconds of the game. Our experiments indicate that PIVOT is able to produce accurate and calibrated probability estimates, even when trained on a relatively small dataset. We also showcase two interactive applications of PIVOT for valuing actual and counterfactual player decisions and actions in real-time.

Keywords: expected possession value · handball · tracking data · time series classification · deep learning

1 Introduction

Team handball, hereafter simply referred to as handball, is an action-packed invasion sport that is considered to be one of the fastest in the world. Although it is a high-scoring sport, the task of objectively assessing the impact that individual decisions and actions have on the overall game remains, nonetheless, challenging.

Despite the fact that several data-driven approaches to value player contributions in various team sports, such as basketball and association football (soccer), have been proposed over the last years [2, 3, 10, 13, 6, 5, 12], none of these approaches can, due to the unique characteristics defining this highly dynamic game [18], easily be applied to handball. For instance, compared to football, handball is played on a smaller field, generates way more goals (between 20 and 35 goals), and most of the action takes place around the goal area (6-meter

line). Rule-wise, handball shares many common characteristics with basketball while being, however, much more physical. Lastly, and most importantly, the data revolution has yet to happen in handball. In truth, there are currently no official providers that collect event data for handball, and only recently, some European handball leagues have started implementing tracking systems.

Against this background, we propose PIVOT – i.e., a framework for valuing player actions in handball purely based on tracking data. To the best of our knowledge, PIVOT is the first expected possession value (EPV) approach for handball. By leveraging deep neural networks and spatio-temporal features, our framework can predict the probability of a team scoring in the immediate future, given all players' and the ball's actual and past positions on the court. Since our framework does not require additional event data (annotations) and learns in an end-to-end fashion – i.e., without complex ensembles of specialized sub-models for evaluating different facets of the game – we argue that this approach is more parsimonious than existing approaches. Furthermore, our empirical experiments reveal that PIVOT produces accurate and calibrated probability estimates, even when trained with only a limited amount of data. In close collaboration with SG Flensburg-Handewitt[3], a top-tier German first division team and former EHF Champions League winners, we also showcase two interactive applications of PIVOT for valuing actual and counterfactual player decisions and actions in real-time.

2 Related Work

Over the last years, the evaluation of players' individual decisions and actions has gained increased attention in invasion sports, especially basketball and football. Driven by the increased availability and quality of in-game data, several statistical and machine learning approaches have been proposed to estimate the value of various offensive and defensive actions – i.e., with and without the ball. In the following, we concentrate on approaches that are primarily based on tracking data (as opposed to event data, like in [1, 4]).

Concentrating on basketball, Cervone et al. introduced a concept known as expected possession value (EPV) which refers to the number of points the attacking team is expected to score at the end of a given possession [2, 3]. The EPV of a given possession is computed by a multi-resolution process as the weighted average of the ball carrier's predicted probability of making a specific action and the estimated value of each of these potential actions. The model distinguishes between discrete actions like passing or shooting (referred to as macro-transitions) and continuous actions like moving with the ball (referred to as micro-transitions). The latter type of actions is derived from optical tracking data and the former from annotated events based on the tracking data.

Focusing merely on passes in football, Power et al. developed an approach for evaluating the risk – i.e., the likelihood of making a successful pass – based on the

[3] https://www.sg-flensburg-handewitt.de

potential reward – i.e., the likelihood that a pass made will result in a shot within the next 10 seconds [10]. To do so, they used a logistic regression classifier to estimate both likelihoods given micro features – e.g., speed of players, distance to the nearest opponent, pressure applied to passer and receiver – tactical features – e.g., build-up, counter-attack, unstructured play – and formation features (high, medium or low block). These features were derived from both tracking and event data. Based on the risk and reward models, they propose several new metrics such as passing plus/minus and dangerous passes.

Spearman proposed an off-ball scoring opportunity model that, given data about the current position and velocity of all players and the ball, can predict the probability that a football player not currently in possession of the ball will score with the next on-ball event [13]. To estimate this probability, the model has to estimate three distinct probabilities: (1) the probability that the attacking team passes to each possible point on the field, (2) the probability that the team can control the ball at each of those possible locations, and (3) the probability that the team scores from these locations. The parameters of these separate models are estimated using tracking and event data, and, at prediction time, their outputs are combined to produce a single off-ball scoring opportunity metric.

Recently, Fernandez et al. developed what is, arguably, the most holistic approach to EPV in football [6, 5]. This framework, which is based on tracking and event data, comprises a series of sub-components, each targeted at evaluating a specific type of action – e.g., passes, ball drives, shots. These sub-components are implemented either through specialized machine learning models – e.g., deep neural networks for extracting spatio-temporal features from tracking data – or through handcrafted algorithms developed by domain experts. According to the authors, the main advantage of this decomposition approach is the increased interpretability of the individual components of the framework.

Sicilia et al. proposed an end-to-end learning approach for estimating the probability and value of individual actions in basketball [12]. Their deep learning architecture learns a joint function incorporating all available spatio-temporal information at once. More specifically, they combine (1) a recurrent neural network trained on a multivariate time series of the positions of all players and the ball and (2) an embedding of player identities to predict the terminal action – e.g., field goal, turnover, foul – of a given possession. Based on these terminal action probabilities and the long-term averages of points per action, the expected value of individual actions can be determined. Like all other approaches, the framework requires the availability of linked tracking and event data.

3 The PIVOT Framework

Our proposed framework, named PIVOT, builds on the end-to-end learning idea presented by Sicilia et al. [12] while however, only requiring tracking data and minimal data preparation. Therefore, we argue that our framework is more parsi-

monious than existing EPV estimation approaches and, as a result, more suitable for practical applications in handball.

3.1 Features

We represent a game G as a sequence of frames $[f_1, f_2, ..., f_n]$, where n is the total number of frames in a game. Each frame f is a tuple comprised of a timestamp (only used for indexing) and the (x, y) coordinates of the ball, the seven home team players, and the seven away team players. When a player is substituted, G is updated so that the incoming player's data is stored in the columns formerly occupied by the outgoing player.

Instead of using the entire sequence of frames from the beginning of the game up to a given point in time t, we only focus on a sliding window composed of p frames. Hence, our features are represented by the fixed-length two-dimensional array $X_t = [f_{t-p}, ..., f_{t-2}, f_{t-1}, f_t]$, where X_t contains the spatio-temporal distribution of the ball and all players present on the field during a given window (Figure 1). Thereby, this array accounts for the obvious temporal dependency of positions between neighboring frames. Moreover, it also has the advantage of having a fixed length – i.e., a requirement for many machine learning algorithms.

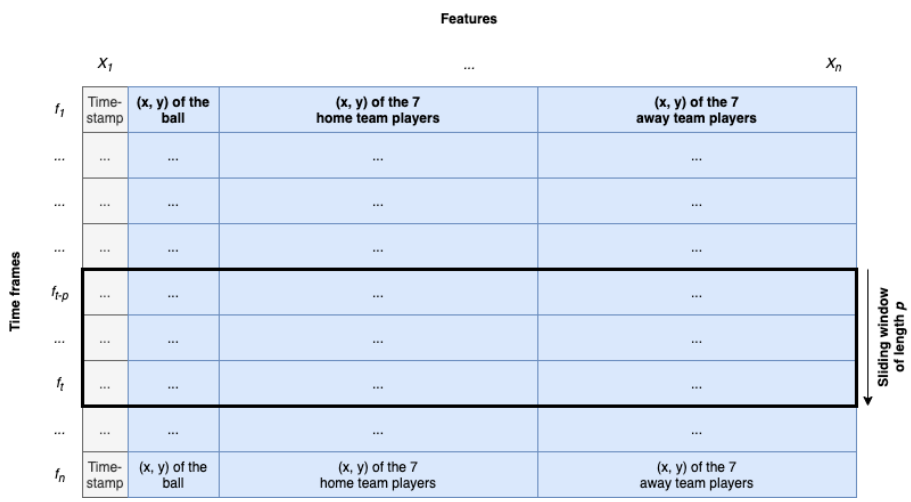

Fig. 1. Representation of (parts of) a game by a fixed-length two-dimensional array capturing the spatio-temporal distribution of the ball and all players present on the field during a given time window.

3.2 Response Variable

Our response variable, or target, is the binary variable Y_t which indicates whether the home team will score a goal in the immediate future. It takes the value 1, if

the home team scores a goal within the next k frames $[f_{t+1}, ..., f_{t+k}]$; otherwise it takes the value 0.

Recall that our tracking data does not contain any event annotations – e.g., goals, shots, passes. Hence, in order to construct the target variable, we implemented a simple rule-based approach capable of determining the exact frame of a goal from raw tracking data. The algorithm is illustrated in Figure 2 and executed for each frame of the game during the data preparation phase.

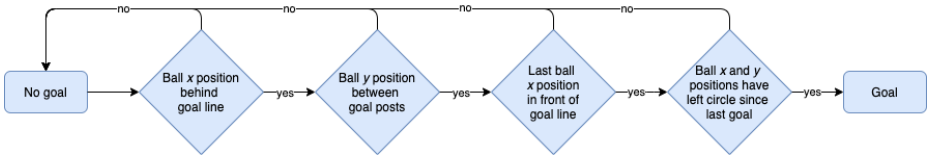

Fig. 2. Rule-based approach for determining the exact frame of a goal from tracking data.

3.3 Learning Task

Given the above-defined features and labels, our learning task is to predict, at frame t, the probability that the home team will score a goal within the next k frames, given the previously-defined sequence X_t: $P(Y_t = 1|X_t)$. We define this conditional probability as the current EPV of the home team. Theoretically, it can be estimated using any probabilistic classification algorithm. However, with X_t being a two-dimensional array, multivariate time series or sequence classification methods seem especially suited for the problem.

Considering that a team's probability of scoring a goal within the next k frames is, arguably, equal to the other team's probability of conceding a goal within the same time window, the proposed framework allows for a holistic evaluation of offensive as well as defensive plays.

3.4 Undersampling, Smoothing and Calibration

Class imbalance represents a serious challenge when it comes to estimating EPV. Hence, various techniques, such as over or undersampling [14], have been proposed to overcome this problem. However, applying such sampling techniques to a training set tends to cause bias in the estimated class probabilities for unseen observations – i.e., an issue that is especially severe when using neural networks for probability estimation [11]. With only 2.5% of the frames in our dataset being positive, we address this problem by implementing the following three steps.

First, we perform undersampling by randomly deleting a share of the training observations that did not result in an immediate goal – i.e., the majority class. The amount of undersampling is a hyperparameter that should be tuned

empirically. In our experiments, decreasing the negative class by a factor of 0.9 showed to be effective.

Second, we use a loss function with label smoothing instead of a conventional loss function to train our networks. More specifically, we smooth the standard one-hot encoded vector with ones and zeros using the Label Smoothing Cross-Entropy loss function [9]. In short, this decreases the values of the ones by a small amount and, respectively, increases the zeros by a small amount; therefore, preventing the network from becoming overconfident in its predictions. The parameter epsilon (ϵ) of the Label Smoothing Cross-Entropy function is another hyperparameter that should be tuned for a given dataset. In our experiments, we stuck to the default value – i.e., $\epsilon = 0.1$.

Finally, we calibrate the raw probability estimates generated by our classifiers using the method proposed by [11], which is based on a Bayesian framework and takes the class imbalance and undersampling ratios into account. A classifier is said to be well-calibrated when the predicted probability of a class matches the expected frequency of that class. Some statistical models, such as logistic regression, are naturally calibrated; others, especially neural networks, tend to be uncalibrated and overconfident – i.e., they typically output probabilities close to 0 or 1. While calibration is often unnecessary for discrete classification problems, it becomes critical when exact probability estimates are required. The estimation of EPV represents such a case. .

4 Experiments

In this section, we demonstrate that with the help of real-world data, PIVOT can generate accurate and calibrated probability estimates. We also show that using the latest deep learning architectures for time series classification, namely Transformers, can substantially improve predictive accuracy over standard Convolutional Neural Networks (CNN) or Long Short-Term Memory networks (LSTM).

4.1 Dataset

In the 2019/20 season, the German elite handball league (Liqui Moly HBL) started rolling out a sensor-based tracking system capable of collecting location data at a frequency of 20 Hz (or frames per second). Through a collaboration with the team SG Flensburg-Handewitt, we obtained ball and player tracking data for a total of 15 games (the season was ended prematurely due to the COVID-19 pandemic). We performed minimal data cleaning by removing all observations with coordinates outside the field – e.g., players sitting on the bench – or timestamps associated with events taking place before/after the game or during the halftime break. Furthermore, we rotated the data for the second halftime by 180 degrees so that the home team always plays from left to right and the away team from right to left. Finally, we augmented the dataset by mirroring each game; therefore, enabling us to train the classifier on both the home and away team data.

After the above transformations, our dataset contained more than 2 million observations (frames), each composed of 32 variables. To derive the features and labels described in Section 3, we applied a sliding window with window length p, horizon k, and stride one and split the data into training (approx. 70%), validation (approx. 15%), and test (approx. 15%) sets. For the following experiments, we set k to 60 frames (3 seconds) – i.e., a decision based on domain knowledge and discussion with the club – and experimented with various p ranging from 20 to 60 frames (1-3 seconds).

4.2 Network Architectures

Theoretically, the framework outlined in Section 3 could be instantiated with any machine learning classifier capable of producing probability estimates. However, because of the sequential nature of our features, one would have to flatten the two-dimensional arrays before feeding them into a standard classification model – e.g., logistic regression or random forest. Hence, to maintain the original structure of the data, we tackle this problem by using time series classification models that can handle two-dimensional feature arrays. All of the three following models were implemented using the TSAI[4] library and trained for 10 epochs with early stopping regularization with a patience of 3 and a minimum delta of 0.005 (AUC) on the validation set.

First, we implemented the Fully Convolutional Network (FCN) architecture proposed by Wang et al. [16]. We chose this relatively simple model as a baseline for gauging the performance of the more complex architectures, as it proved to be a strong benchmark for end-to-end time series classification in prior experiments [16]. The FCN architecture comprises three convolution blocks, each followed by a batch normalization layer and a ReLU activation layer. Following these convolution blocks, the features are fed into a global average pooling layer, and a linear classification layer produces the final label. The default TSAI values were used for all hyperparameters – i.e., filter size, kernel size, stride, and padding.

Second, we tested a recurrent neural network, more specifically, the LSTM architecture introduced by Hochreiter and Schmidhuber [8]. The main advantage of LSTMs is their capability to model long-term dependencies in sequences, which might enable the detection of patterns in build-up plays. We chose this architecture as it is close to the architecture used by Sicilia et al. [12], which inspired our work. Akin to Sicilia et al. [12], we stacked three LSTM layers with hidden states of size 32, followed by a dropout layer ($\delta = 0.2$) and a linear classification layer.

Third, we used a transformer-based architecture, namely the Time Series Transformer (TST) by Zerveas et al. [17]. The TST architecture uses only the encoder layers of the original Transformer architecture [15]. We selected this architecture as it has outperformed many other models in recent experiments by a significant margin while requiring only relatively few training samples [17]. Our TST network has three stacked encoder layers, each consisting of a multi-head

[4] https://github.com/timeseriesAI/tsai

self-attention layer (128), dropout ($\delta = 0.2$), batch normalization, a feed-forward layer, dropout ($\delta = 0.2$), and batch normalization. The outputs of the encoder layers are then flattened, another dropout ($\delta = 0.2$) is applied, and the results are fed into a linear classification layer.

4.3 Results

Table 1 summarizes the results of our experiments. Following [12, 4], we evaluate the predictive performance of our models using the probability scoring metrics AUC, Brier Score (BS), and Brier Skill Score (BSS). For the BSS, we use a naive model, which predicts the base rate of the positive class (in our case 2.5%) as the probability for every observation in the test set, as the reference model [7]. The results provide strong empirical evidence for the superiority of the TST model over the other models. For instance, for the window length 20, TST outperforms the FCN in terms of BSS by 30% and the LSTM by 40%. Regarding the window length p, our results suggest that windows of 20 or 40 frames – i.e., 1 or 2 seconds – work best. It seems that longer windows introduce noise into the learning process, which is probably due to the dynamic and fast-paced nature of handball. Conversely, longer windows may work best for sports like football.

Table 1. Predictive performance for different neural network architectures and sliding window lengths. For AUC and Brier Skill Score values closer to 1 are better; for the Brier Score values closer to 0 are better. Best results in bold.

Model	Window	AUC	Brier Score	Brier Skill Score
FCN	20	0.808	0.027	0.222
FCN	40	0.810	0.027	0.225
FCN	60	0.761	0.028	0.185
LSTM	20	0.845	0.028	0.190
LSTM	40	0.827	0.026	0.230
LSTM	60	0.836	0.027	0.214
TST	20	**0.909**	**0.023**	**0.318**
TST	40	0.882	0.024	0.284
TST	60	0.884	0.025	0.278

In addition to probability scoring metrics, we used calibration plots to assess the accuracy of our EPV estimates. For a perfectly calibrated model, when considering all frames with a predicted EPV of x%, one would expect that x% of these frames actually resulted in a goal. Figure 3 shows the calibration plot for PIVOT, suggesting that the model is well-calibrated for probabilities below 50%, but underestimates situations with true high goal probabilities, which might be a result of the label smoothing.

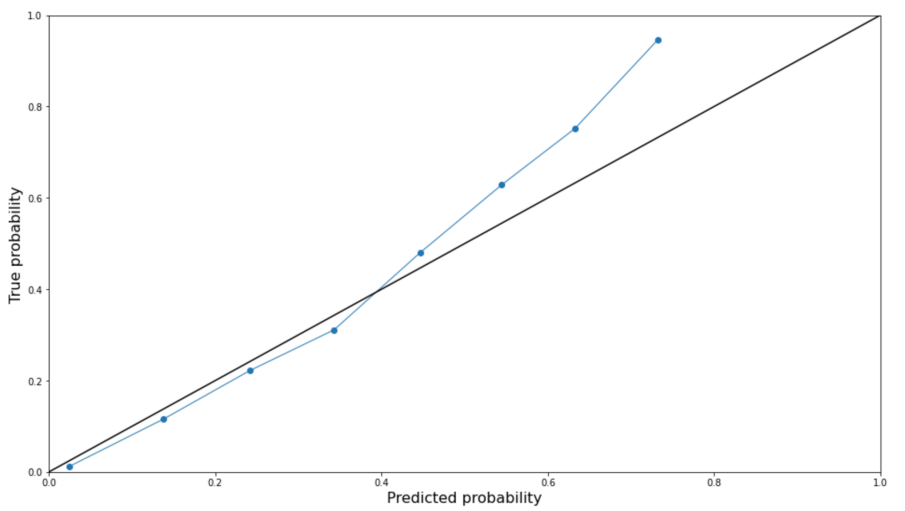

Fig. 3. Calibration plot for PIVOT. The model is well-calibrated for EPV values below 0.5, but underestimates high goal probabilities.

5 Applications

In this section, we showcase two exemplary applications built on top of PIVOT, which were co-created with members of the coaching staff of SG Flensburg-Handewitt.

5.1 Application 1: Augmented Instant Replay

Following the stock ticker idea of Cervone et al. [2], we can use PIVOT to calculate and monitor EPV in near real-time continuously. The resulting timeline, which we call *Augmented Instant Replay*, uncovers what in-game decisions have the most significant impact on EPV. Figure 4 shows the evolution of EPV during a possession of SG Flensburg-Handewitt in a 2019/20 league game against TSV Hannover-Burgdorf. After an unsuccessful attack from Hannover, Flensburg's goalkeeper Bergerud initiates a fast break, which results in a steep rise in EPV. Gottfridsson then quickly drives the ball towards Hannover's goal area. When he comes under pressure by Olson, the EPV drops momentarily but promptly recovers when Gottfridsson makes a successful pass to the right-winger Svan, who then finishes the possession with a successful falling jump shot.

5.2 Application 2: What-If

Another application that builds on top of PIVOT is a sensitivity analysis called *What-If*. Instead of only valuing plays that actually happened, it can be used to perform a counterfactual analysis of plays that could have happened. Using

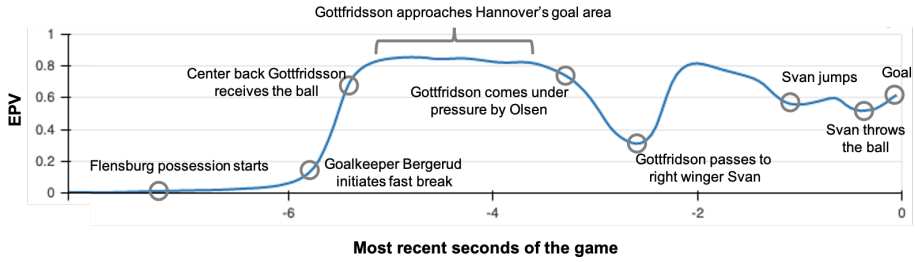

Fig. 4. Development of EPV for an attack of SG Flensburg-Handewitt during a league match against TSV Hannover-Burgdorf.

an interactive dashboard similar to a tactics board, an analyst can simulate and assess the expected value of any play or combination of moves by dragging the players and/or the ball around the court. The upper half of Figure 5 depicts a moment from the match between Flensburg and Hannover when the right-back Rød, currently in possession of the ball, comes under pressure by Hannover's Garcia. The four arrows represent four different passing options along with their expected change in EPV. Passing to Gottfridsson or Johannessen would result in a loss of EPV, while passes to Svan or Golla would dramatically improve the chances of scoring in the near future. Of course, the latter two passing options are much riskier, which is not accounted for in the current version of the analysis. The bottom half of Figure 5 shows another counterfactual analysis, this time focusing on off-ball moves. The heatmap around Rød visualizes how the EPV would change if the player moved to the respective locations (the size of the heatmap roughly spans the distance a player can reach within one second). The model suggests that Rød should run towards the right wing to increase the likelihood of Flensburg scoring a goal within the next three seconds.

6 Conclusion and Outlook

This paper introduced PIVOT – i.e., a deep learning framework for estimating the expected value of possession in handball purely based on tracking data. Compared to existing approaches, PIVOT is less resource-intensive, as it does not require the availability of linked tracking and event data and learns in an end-to-end fashion, even with a limited amount of data.

As part of our future work, we aim to extend the framework with additional player features. Currently, our approach neither takes the identity nor skills of individual players into account. The reason for this is that our models were trained using only 15 games and, therefore, would easily overfit with such features (for SG Flensburg-Handewitt players). The situation would be even worse with players of opposing teams, as our models have seen them for a maximum of one game. Likewise, after collecting data for more games, we plan to integrate advanced spatial features. Examples include players' speed and direction

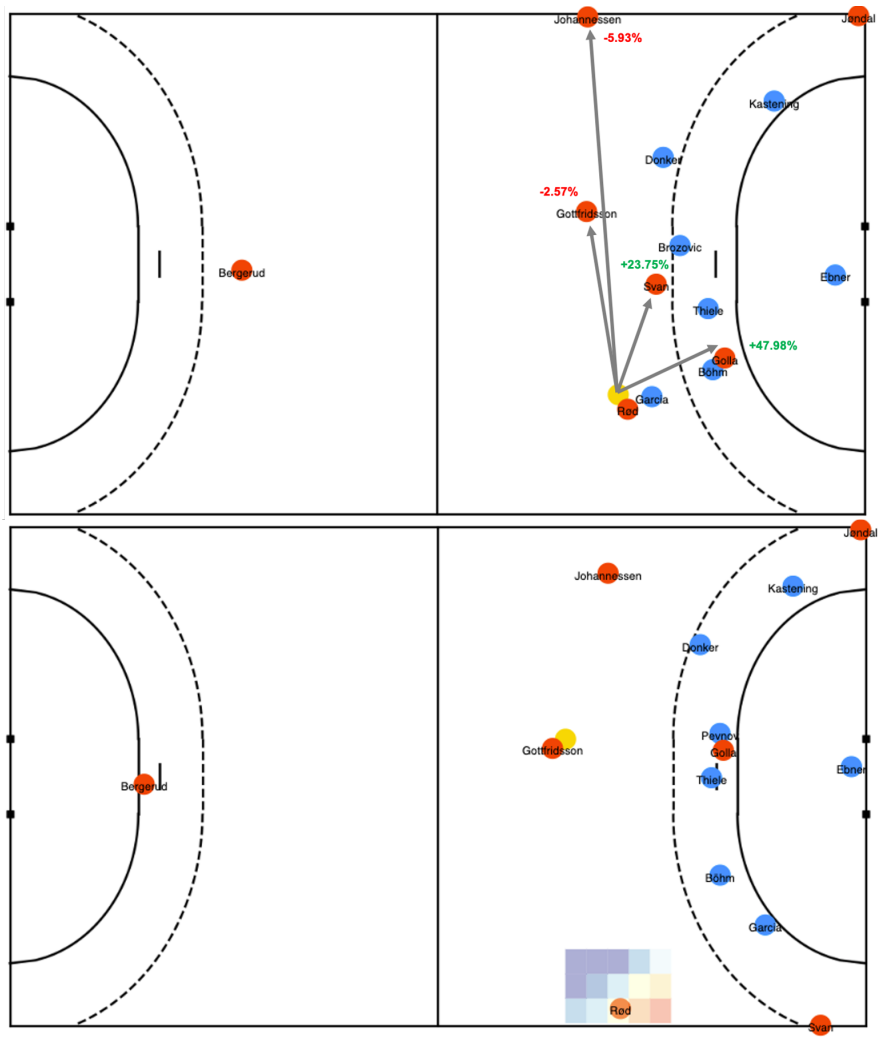

Fig. 5. Upper half: Change in EPV for four different passing options. According to the model, a pass from Rød to Svan or Golla would increase EPV, while a pass to Gottfridsson or Johannessen would decrease EPV. Bottom half: EPV surface for right back Rød. Red color indicates increase in EPV, blue color indicates decrease in EPV. According to the model, Rød should run towards the right wing.

of movement, their orientation on the field, the pressure put on a player, and their available passing options. Finally, we currently explore different ways for distributing overall EPV to the individual players involved in a play. A straightforward approach, which produced promising results in the first tests, would be to calculate the difference in EPV between the start and end of a player's ball possession. The obvious drawback of this approach is that it only values the actions of the ball-carrying player. An extension to off-ball players could be based on our What-If application. For each frame, one could calculate the difference in EPV between the estimated best possible action and the action taken.

References

[1] Lotte Bransen and Jan Van Haaren. "Measuring Football Players' On-the-Ball Contributions from Passes During Games". In: *International Workshop on Machine Learning and Data Mining for Sports Analytics*. 2019, pp. 3–15.

[2] Dan Cervone et al. "POINTWISE: Predicting Points and Valuing Decisions in Real Time with NBA Optical Tracking Data: A New Microeconomics for the NBA". In: *Proceedings of the 8th MIT Sloan Sports Analytics Conference*. 2014.

[3] Daniel Cervone et al. "A Multiresolution Stochastic Process Model for Predicting Basketball Possession Outcomes". In: *Journal of the American Statistical Association* 111 (514 2016), pp. 585–599.

[4] Tom Decroos et al. "Actions Speak Louder than Goals". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019, pp. 1851–1861.

[5] Javier Fernández, Luke Bornn, and Daniel Cervone. "A framework for the fine-grained evaluation of the instantaneous expected value of soccer possessions". In: *Machine Learning* 110 (6 2021), pp. 1389–1427.

[6] Javier Fernández et al. "Decomposing the Immeasurable Sport: A Deep Learning Expected Possession Value Framework for Soccer". In: *Proceedings of the 13th MIT Sloan Sports Analytics Conference*. 2019.

[7] Thomas M. Hamill and Josip Juras. "Measuring forecast skill: is it real skill or is it the varying climatology?" In: *Quarterly Journal of the Royal Meteorological Society* 132 (621C 2006), pp. 2905–2923.

[8] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9 (8 1997), pp. 1735–1780.

[9] Rafael Müller, Simon Kornblith, and Geoffrey Hinton. "When Does Label Smoothing Help?" In: *Advances in Neural Information Processing Systems*. Vol. 32. 2019.

[10] Paul Power et al. "Not All Passes Are Created Equal". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2017, pp. 1605–1613.

[11] Andrea Dal Pozzolo et al. "Calibrating Probability with Undersampling for Unbalanced Classification". In: *Proceedings of the 2015 IEEE Symposium Series on Computational Intelligence*. 2015, pp. 159–166.

[12] Anthony Sicilia, Konstantinos Pelechrinis, and Kirk Goldsberry. "DeepHoops: Evaluating Micro-Actions in Basketball Using Deep Feature Representations of Spatio-Temporal Data". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019, pp. 2096–2104.

[13] William Spearman. "Beyond Expected Goals". In: *Proceedings of the 12th MIT Sloan Sports Analytics Conference*. 2018.

[14] Y. Sun, A.K.C. Wong, and M.S. Kamel. "Classification of Imbalanced Data: A Review". In: *International Journal of Pattern Recognition and Artificial Intelligence* 23 (04 2009), pp. 687–719.

[15] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Vol. 30. 2017.

[16] Zhiguang Wang, Weizhong Yan, and Tim Oates. "Time series classification from scratch with deep neural networks: A strong baseline". In: *2017 International Joint Conference on Neural Networks*. 2017, pp. 1578–1585.

[17] George Zerveas et al. "A Transformer-based Framework for Multivariate Time Series Representation Learning". In: *arXiv preprint* (2020). URL: `https://arxiv.org/abs/2010.02803`.

[18] Kasper Zeuthen. *Team Handball: It's Not What You Think It Is*. Feb. 2002. URL: `https://www.washingtonpost.com/archive/lifestyle/2002/02/15/team-handball-its-not-what-you-think-it-is/09cee01c-9daa-4fdb-a17b-bb23fc60fd11/`.