One xG model to rule them all?

Challenging the most accurate xG model

Robert Bajons & Tobias Harringer

1. Motivation

Expected Goals (xG) models are ubiquitous in football. They are used across a wide range of applications, including match and team performance analysis, player evaluation, as well as for predicting match results and simulating league outcomes. In contrast to many other advanced data-based models, xG has gained broad acceptance in the football community. Not only are xG a standard part of match facts provided for broadcast (see e.g. the DFL[1]), but more and more managers nowadays rely on xG values to analyze their team (e.g., Mikel Arteta[2], Arne Slot[3], and many more). Hudl-Statsbomb is one of the most prominent providers of xG values, and their xG model is labeled the "most accurate xG model"[4]. In this project, we contribute to the highly debated topic of xG models from multiple perspectives. First, we thoroughly analyse Hudl-Statsbomb's claim of having the most accurate xG model. To do so, we establish how to evaluate xG models and analyze the performance of Hudl-Statsbomb's model. Next, using openly available data as well as data provided exclusively for this project, we derive our own model to show that we can improve upon Hudl-Statsbomb's model with respect to several evaluation criteria. Finally, we tackle the question of whether there is one xG model that should be used uniquely, or whether certain use-cases call for their own variant of an xG model. In particular, we argue that there exists no ultimate xG model to rule them all, but that each use case benefits from an xG variant tailored to it. We present various examples of such use cases, discussing team strengths adjusted xG models, xG models used for game prediction, xG models for player evaluation, and xG models when only a limited amount of data is available. We aim to guide analysts on how to choose a reasonable model and strategies to improve upon existing xG models.

2. Data and Preprocessing

This section describes the dataset used in this project and the preprocessing steps applied before analysis.

2.1 Data Description

Our dataset consists of two main components. First, we use data from Hudl-Statsbomb provided exclusively for this project, which includes both event data and aggregated physical data. The data come from the 2024/25 season of five professional leagues: Allsvenskan, Championship, Eredivisie, Jupiler Pro League, and Major League Soccer. In addition, we complement this dataset

[1] https://www.dfl.de/en/topics/match-data/bundesliga-match-facts/
[2] https://www.thesun.co.uk/sport/
[3] https://liverpooloffside.sbnation.com/english-premier-league/
[4] https://www.youtube.com/watch?v=P-0slh66ekE&t=50s

by using publicly available event data from the 2015/16 season of the so-called "top five" European leagues: Bundesliga, La Liga, Ligue 1, Premier League, and Serie A.

The event data provides detailed information about on-ball actions during matches, such as passes, shots, dribbles, and more. Each event is annotated with rich contextual information such as the possession sequence of the event, timestamp, player and team identifiers, (x, y)-coordinates indicating where the event occurred on the pitch, and many more. Furthermore, all shot events contain a "shot freeze frame", which captures the (x, y)-coordinates of all the visible players in the instant of a shot. For a detailed description of the event data, see the Hudl-Statsbomb data specification[5].

The physical data provides information across 14 different physical metrics, derived from player tracking data and aggregated in 15-minute intervals. These metrics include, among others, total running distance, sprint count and distance, and accelerations. Again, for a more detailed description, see the Wyscout physical metrics glossary[6].

2.2 Data Preprocessing

To prepare our dataset for analysis, we perform several preprocessing steps. As we are interested in analyzing and modelling expected goals (xG), we first filter the event data to include only shot events. Of those, we further exclude penalties, shots from corners, and kick-off situations, keeping only shots from open play and free kicks. Any shots with missing location information (e.g., goalkeeper position or freeze-frame data) are removed as well. In total, 1,025 shots are excluded in this process, leaving 88,737 shots for the subsequent analysis. Of these, 43,596 shots come from the Hudl-Statsbomb data provided exclusively for this project, while the remaining 45,141 originate from Hudl-Statsbomb's publicly available data for the "top five" European leagues.

Next, we use the StatsBombR package (Yam, 2025) to derive additional contextual features, such as distance and angle from the shot location to the goal and goalkeeper, how many attackers and defenders are behind the ball, and further spatial features. Furthermore, we determine each player's strong foot using all pass events and identify the foot used more frequently through a function provided in the StatsBombR package. Based on this, we add to each shot a label indicating whether it is taken with the player's strong or weak foot.

We then add the scoreline from the shooting team's perspective to each shot. In order to derive this, we first use goal and own-goal events to compute the scoreline for the home team at each moment in the match. We then map these values to the timestamp of each shot and adjust for shots taken by the away team by inverting the scoreline.

Finally, we add an indicator if a shot occurs in a rebound situation. We follow the definition by Litwitz, Memmert and Wunderlich (2024), who describe a rebound as a shot attempt where the ball bounces back from another player, the goalkeeper, or the goal (i.e., post or bar). As such situations

[5] https://support.hudl.com/s/article/data-specifications?language=en_US
[6] https://dataglossary.wyscout.com/physical_metrics/

cannot be identified with certainty from the available data, we adopt a simple yet intuitive approach: we label each shot as occurring in a rebound situation if it takes place within the same possession sequence and within 3 seconds of the prior shot. This threshold captures immediate rebound situations while excluding shots that result from new build-up play within the same possession sequence.

3. Analyzing Hudl-Statsbomb's xG model

In this section, we analyze Hudl-Statsbomb's xG model by comparing the model's predicted xG values for all the individual shots with the observed binary shot outcomes (goal or no goal).

There are different ways to evaluate the quality of an xG model, with no single best approach. Hence, in this analysis, we consider several aspects of model performance. We start by evaluating the model's predictive accuracy using conventional loss metrics. Then, we inspect its calibration graphically and the goal distribution from a theoretical standpoint. Finally, we compare predicted versus actual outcomes in specific situations to identify potential systematic biases.

3.1 Predictive accuracy

Assessing predictive accuracy provides directly comparable benchmark measures of how well the model distinguishes between goals and non-goals. As metrics, we use Accuracy, Area Under the Curve (AUC), and logloss. While accuracy is simple and highly intuitive, it depends on the chosen cutoff point and is therefore not the most informative measure for probabilistic models. In contrast, AUC quantifies the model's discrimination ability, i.e., its ability to assign higher predictions to goals than non-goals, independent of any specific threshold. Further, logloss evaluates the quality of predicted probabilities by penalizing inaccurate and overconfident predictions while rewarding well-calibrated estimates. As a proper scoring rule, it is particularly well-suited for probabilistic models such as xG.

These evaluation metrics serve as benchmarks for comparing different models, which we will use later to evaluate our own xG model. As an initial baseline, we compare against a naïve predictor that always assigns the average empirical goal probability of 9.6%.

Metric	Naïve baseline	H-SB xG
Accuracy	0.9040	0.9108
AUC	-	0.8053
logloss	0.3162	0.2535

Table 1: Comparison of loss metrics naïve baseline vs. H-SB's xG model

We observe in Table 1 that Hudl-Statsbomb's (H-SB) xG model clearly outperforms the naïve baseline as expected, improving accuracy slightly and logloss substantially. However, without further comparison to other models on the same dataset, these absolute numbers provide limited

insight into the model's relative performance. Although many papers publish their xG evaluation results, such comparisons are not meaningful as they are evaluated on different shot datasets.

3.2 Model calibration

Calibration assesses whether the model's predicted probabilities align with observed outcomes, i.e., if shots assigned a x% goal probability result in goals roughly x% of the time. This is particularly of interest in an xG model, as we want to use these predictions to obtain estimates of how many goals a player or team should have scored on average, given their shots.

We evaluate calibration graphically by comparing predicted xG values against observed goal rates across different probability bins. To do so, we sort all individual xG predictions from lowest to highest and bin them into twenty equally sized groups. Then, we compute in each group the average xG value and the average observed goal rate.

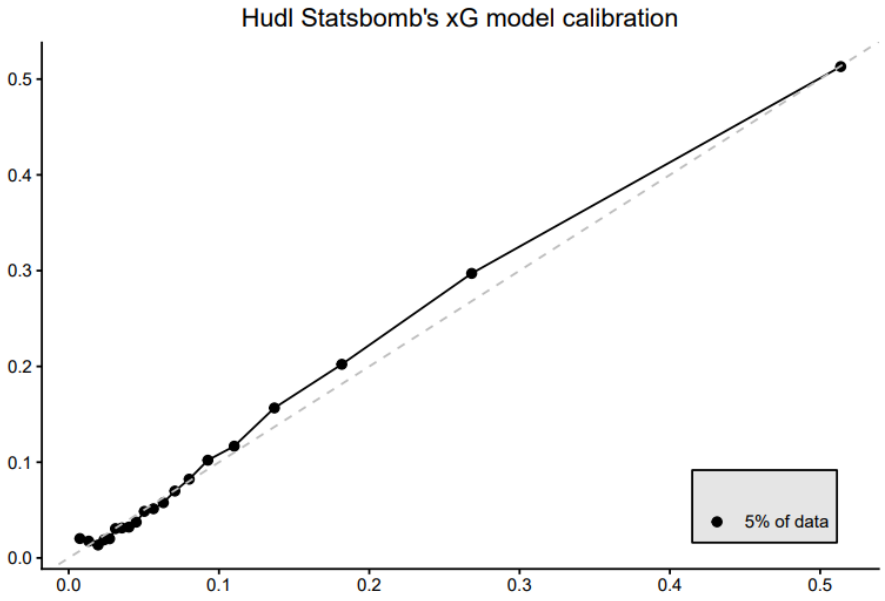

Figure 1: Calibration plot of H-SB's xG model

Figure 1 displays the calibration plot for Hudl-Statsbomb's xG model. If it were perfectly calibrated, all dots would lie exactly on the grey dashed line, indicating observed and predicted goal rates are equal within the groups. Most dots are clustered in the low-probability region due to class imbalance. In this region, the dots are either on the line or very close to it, suggesting good calibration. However, in the 0.1 to 0.3 range, the dots consistently lie above the dashed line, meaning that empirical goal rates are higher than predictions. This indicates systematic underprediction in this region. It is worth noting that calibration biases have been observed in other xG models as well. For instance, an analysis of Opta's model found the exact opposite pattern, namely systematic overprediction in the higher probability area (ElHabr, 2023b).

3.3 Goal distribution

Next, we investigate a key theoretical assumption underlying xG models. In general, xG models assume goals are independent conditional of the features used in the model. Under feature sufficiency, i.e., when all relevant features are included in the model, and when the model is calibrated reasonably well, i.e. $p_i \approx \mathrm{xG}_i$, where p_i are the true probabilities and xG_i are the model predictions, the total number of goals G from all shots follow a Poisson-Binomial distribution

$$G \sim Poisson\text{-}Binomial(xG_1, \cdots, xG_n)$$

where n is the number of shots and xG_i is the xG value of shot i.

Hence, treating the xG values as the true underlying pre-shot goal probabilities, we can derive a high-probability range (prediction interval) for the total number of goals. We can then compare this to the observed goals and if the total goals lie outside the interval, it suggests a violation of underlying assumptions.

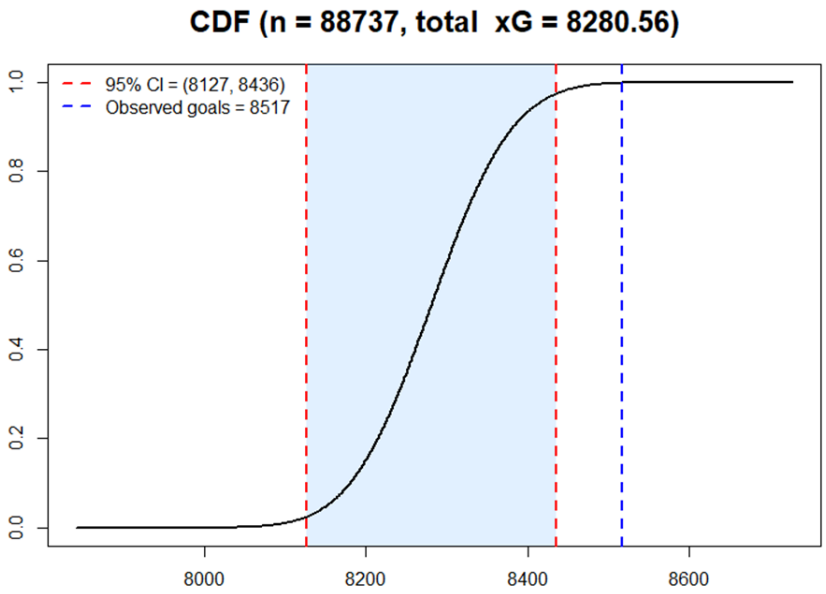

Figure 2: CDF of Poisson-Binomial distribution for all shots using xG values of H-SB's xG model

Figure 2 shows the cumulative distribution function (CDF) of the resulting Poisson-Binomial distribution. The red dashed lines mark the lower and upper bounds of the 95% prediction interval, while the blue dashed line represents the observed number of total goals. The observed total lies right of the upper bound, indicating that significantly more goals were scored than predicted by the model. This systematic underprediction suggests a violation of the conditional independence assumption, indicating that the features used to construct the xG model are insufficient to capture all relevant information for predicting goals.

3.4 Potential biases

In this section, we investigate whether the model shows systematic bias in specific situations. We focus on factors that, to the best of our knowledge, are not included in the features of Hudl-

Statsbomb's xG model. To do so, we group the shots into different categories and compute the average Goals Above Expectation (GAX) for each category as:

$$\frac{1}{n_c}\sum_{i \in c}(G_i - xG_i)$$

where n_c is the number of shots in the given category c, G_i the observed binary outcome (1 = goal, 0 = no goal) and xG_i the xG value of shot i.

If different categories show substantially different average GAX values, this suggests the model is situationally biased due to omitting the variable. Similar versions of some of the following factors have shown predictive value in prior work (see e.g., Hewitt and Karakuş, 2023; Mead et al., 2023).

Game state.
The game state is defined as the scoreline from the shooting team's perspective (e.g., +1 means leading by one goal). Values are capped at +/- 3 to avoid categories with few shots. The underlying hypothesis is that favorable game states reduce pressure on the shooter and thereby potentially increase the average GAX, while unfavorable game states may have the opposite effect.

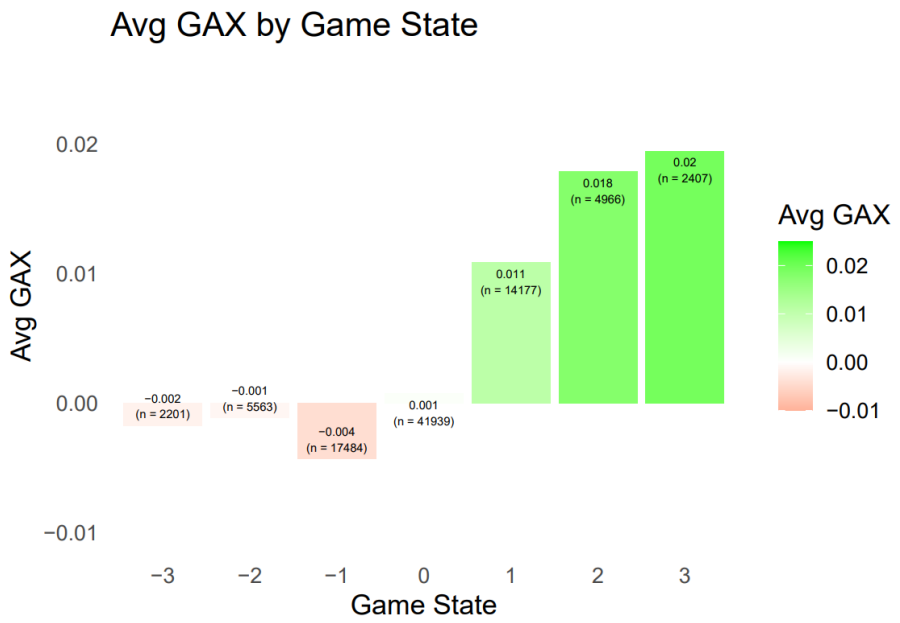

Figure 3: Avg. GAX by Game State

We observe in Figure 3 that the average GAX indeed increases with the game state. When teams are trailing, shooters tend to underperform slightly relative to xG, while there is clear overperformance when leading. Game state thus appears to be an important factor influencing shooting performance.

We further investigate whether this effect interacts with the game clock. To do so, we create three time intervals: early in the game (0-45 minutes), mid-game (45-75 minutes), and late game

(75+ minutes). The idea is that the influence of the game state may increase towards the end of a match. For instance, when a team is trailing late in the game, pressure might increase and further reduce shooting performance.

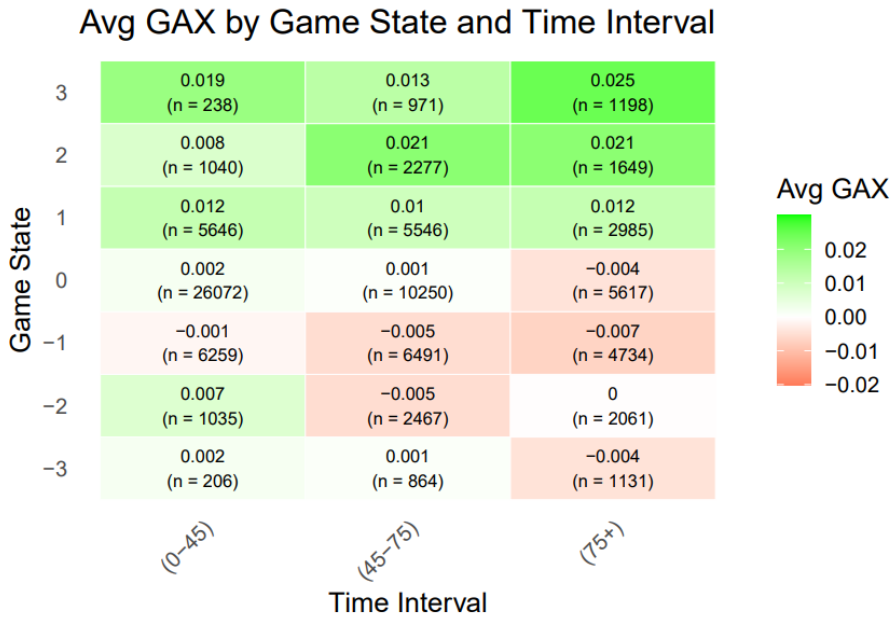

Figure 4: Avg. GAX by Game State and Time Interval

Figure 4 shows that this effect tends to become stronger later in the game, for both favorable and unfavorable game states. Further, there is also a slight underperformance in neutral game states toward the end of matches. This suggests that both scoreline and time influence shooting performance.

Dominant foot.

The dominant foot is defined for each player as the foot with which they attempt the majority of their passes. The hypothesis is that shooting performance is better when using the "strong" foot. Each shot is categorized as "Strong", "Weak", or "Other", where "Other" refers to shots not taken with either foot, i.e., headers or any other body part.

category	avg. GAX	n shots
Strong	0.0091	57,485
Weak	-0.0106	15,746
Other	-0.0078	15,506

Table 2: Avg. GAX by dominant foot categories

Table 2 reveals clear over- and underperformance for shots with the player's strong and weak foot, respectively. Moreover, we also find that shots in the category "Other" show a substantial

underperformance. Hence, the dominant foot variable seems to carry important information not captured by the xG model.

Rebound.
We define a shot to occur in a rebound situation if it follows within three seconds of another shot in the same possession sequence. Rebound situations are often chaotic situations with the goalkeeper and defender not being in their optimal position, as they reacted to the first shot. Although this is partially reflected by the positional features of the goalkeeper and defenders, we believe there is extra information in knowing if a shot happens in a rebound situation.

category	avg. GAX	n shots
Rebound	0.0104	4,293
No rebound	0.0023	84,444

Table 3: Avg. GAX by rebound categories

In Table 3, we observe strong overperformance for shots in rebound situations. Hence, the rebound variable seems to capture important information that is not fully represented in the current xG model.

Further variables.
We also investigated additional factors, such as whether the shooting team was playing home or away, and several aggregated physical performance metrics of the shooter over the 15-75 minutes before the shot (e.g., sprint distance, total running distance). For the continuous physical metrics, we analyzed correlations with GAX rather than using categorical analysis. None of these factors has a meaningful effect on GAX, suggesting they do not add additional value to the existing model features.

3.5 Summary of findings
Overall, Hudl-Statsbomb's xG model performs well, showing good predictive accuracy and generally reasonable calibration. However, there are also some potential issues, such as slight miscalibration in higher probability ranges and total goals exceeding model expectations, suggesting a violation of the feature sufficiency assumption. Further, the predictions show some situational biases related to game state, dominant foot, and rebounds. Together, these findings indicate strong overall performance, but also potential for further improvement.

4. Developing own xG model
Motivated by the findings in Section 3, we develop our own xG model addressing the identified biases. We start by describing the feature set used, then outline the model fitting and tuning procedure, and finally evaluate our model's performance against the benchmark of Hudl-Statsbomb's xG model.

4.1 Model features

Based on the preprocessing steps described in Section 2.2, we build our xG model using a feature set that combines common xG predictors with additional variables to address the biases identified in Section 3. Standard features include distance and angle to both the goal and goalkeeper, as well as basic spatial information about defenders and attackers. We further incorporate features directly from the event data annotations, such as shot type, technique, and body part used. In addition, we include all variables for which a bias was identified in the Hudl-Statsbomb model in Section 3: Game state (scoreline difference), time interval (three distinct time periods), a rebound indicator, and a dominant foot feature. Finally, we extend an idea from Hudl-Statsbomb of representing defenders as bivariate Gaussian distributions (Vatvani, 2022), which we describe in more detail in Section 4.1.1. A complete list of all features used in the model is provided in Table 7 in the appendix.

4.1.1 Advanced player position features

Hudl-Statsbomb identified that using exact defender positions leads to potentially unrealistic jumps in xG predictions for marginal positional changes. This issue arises because features that count the number of defenders in some area increase in integer values, as a defender is either in an area or not. To address this, they represent defenders as bivariate Gaussian distributions and then sum up the Gaussian mass within an area (Vatvani, 2022). We believe this is a sensible approach, as a player's defensive influence area is not limited to their exact position. However, we extend this idea by incorporating additional domain knowledge. We argue that defenders further from the shooter should have a wider influence area due to more possible movement until the ball passes them. Simultaneously, defenders closer to the shooter exert more pressure and have a better chance to block a shot. Hence, we introduce two scaling factors: The first increases the standard deviation of each defender's distribution with their distance from the shooter, while the second downscales the weight of defenders further away when summing up the Gaussian mass within an area.

Mathematically, we represent defender i with position $p_i = (x_i, y_i)$ as

$$D_i \sim N\left(p_i, \sigma_i^2 I\right)$$

where $\sigma_i = \sigma_0(1 + \alpha d_i)$ and $d_i = ||p_i - p_{shot}||$.

In essence, α is the factor scaling the baseline standard deviation σ_0 with distance. Based on this representation, we can compute the defensive mass within any desired area A. To do so, we numerically integrate by discretizing the field into subfields. We first compute the Gaussian mass in each subfield $s \in A$ and then sum up over all subfields. In the summation step, we incorporate the second scaling factor β by downweighing each defender's influence based on their distance d_i to the shooter.

We calculate the Gaussian mass within area A as

$$\sum_i w_i \sum_{s \in A} g_i(s)$$

where $w_i = (1 - \beta d_i, 0)$ is the distance scaled influence weight of defender i and $g_i(s)$ represents the Gaussian mass of defender i in subfield s. To determine the parameters σ_0, α, and β, we use freely available event data from Hudl-Statsbomb not included in this project and tune the parameters by doing a grid search to maximize the correlation between defensive mass and blocked shots, as these features serve as proxies for blocking probability. After this grid search, our final chosen values $\sigma_0 = 0.2$, $\alpha = 0.03$ and $\beta = 0.05$.

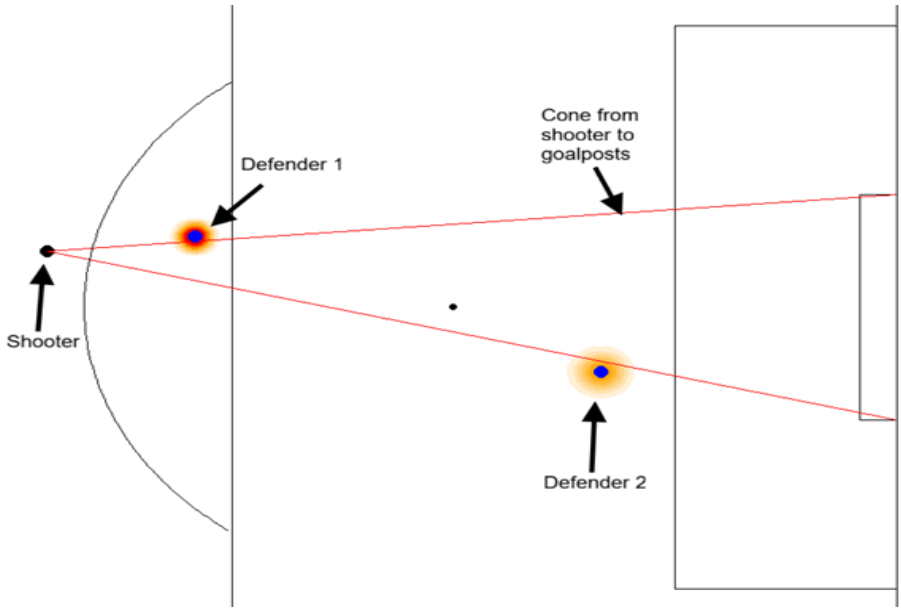

Figure 5: Example of defender representation as a bivariate Gaussian distribution

Figure 5 shows an example of the defender representation approach. Defender 1, positioned closer to the shooter, has a smaller influence area than Defender 2, positioned further away, reflecting their different movement possibilities before the shot passes them. The figure also demonstrates a direct application: calculating defensive mass within the area in the cone from the shooter to the goalposts. Using the standard representation with exact (x, y) coordinates would lead to 0 defenders in this scenario, as both are positioned just outside the cone. Small positional changes could then cause jumps to 1 or 2 defenders, leading to potentially substantially different xG predictions. However, with our Gaussian representation the mass within the cone area is already non-zero and small positional changes would only lead to small changes in the total mass.

We apply the same approach to compute defensive mass in a cone from the shooter to the goalkeeper's y-coordinate ± 1 meter. Furthermore, we compute the same features for attacking players. Although the intent of attackers is not to block shots, they still occupy space and block certain shooting lanes and thereby influence chance quality.

4.2 Model fitting

We considered multiple modelling approaches, including logistic regression, random forests, neural networks, and gradient boosting methods. The XGBoost algorithm (Chen and Guestrin, 2016) achieved the best out-of-sample performance and is therefore selected for our xG model.

We employ a 10-fold cross-fitting strategy to obtain out-of-sample predictions for the entire dataset. The data is partitioned into 10 approximately equal-sized folds. For each fold, we tune the model on the remaining 9 folds and generate predictions for the left-out fold, ensuring all predictions are out-of-sample.

We tune hyperparameters via grid search within each cross-fitting iteration. To reduce computational cost, we first perform a preliminary search to identify promising regions of the hyperparameter space. We then conduct a refined grid search over the following ranges: learning rate $\eta \in \{0.01, 0.05, 0.1, 0.2, 0.5\}$, maximum tree depth $\in \{4, 5, 6\}$ minimum child weight $\in \{10, 20\}$, subsample ratio $\in \{0.8, 1\}$ and column sampling ratio $\in \{0.6, 0.8\}$. The objective is to minimize logloss (binary cross-entropy). The number of boosting rounds is not explicitly tuned but determined via early stopping with 5-fold cross-validation, stopping when no improvement in logloss is observed for 20 consecutive rounds.

4.3 Model evaluation

We evaluate our xG model using the same evaluation criteria as established in Section 3 and compare its performance against the Hudl-Statsbomb xG model as a benchmark.

We start by analyzing the predictive accuracy. Table 4 presents the metrics for both models.

Metric	H-SB xG	Own xG model
Accuracy	0.9108	0.9105
AUC	0.8053	0.8098
logloss	0.2535	0.2523

Table 4: Model evaluation H-SB vs. own xG model

While overall accuracy is slightly higher for the Hudl-Statsbomb model, our model improves in both AUC and logloss. The increase in AUC indicates that our model better distinguishes between goals and non-goals, independent of any specific threshold. The lower logloss further demonstrates better probability estimation.

Next, we look at the model calibration. Figure 6 compares predicted probabilities against observed goal rates using the same binning approach as in Section 3.

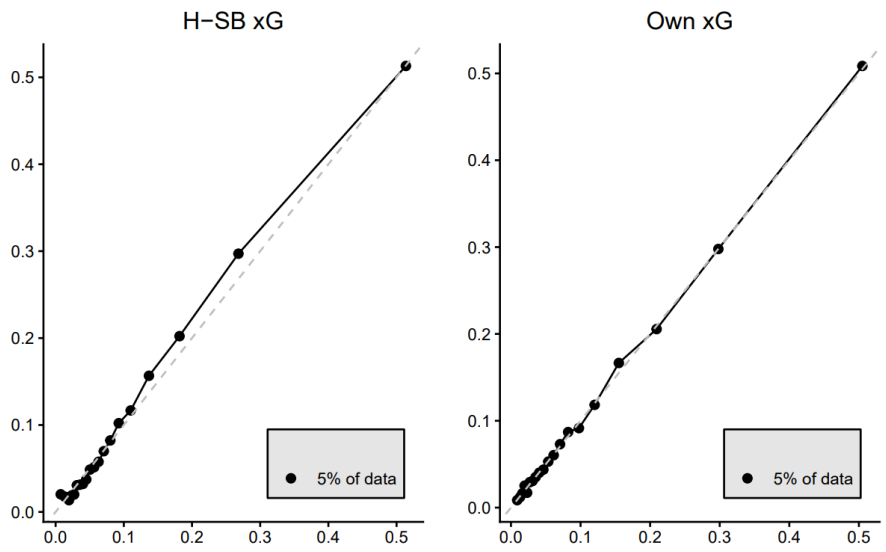

Figure 6: Model calibration H-SB (left) vs. own xG (right) model

Our model shows substantial improvement in calibration compared to the Hudl-Statsbomb model. While not all points lie exactly on the dashed line, they are very close across all probability ranges. Most notably, the issue of systematic underprediction observed in the 0.1 to 0.3 range for the Hudl-Statsbomb model is no longer present, suggesting that our model achieves better calibrated probabilities throughout.

We then examine the goal distribution. Figure 7 shows the CDF of the Poisson-Binomial distribution for both models derived from the xG predictions.

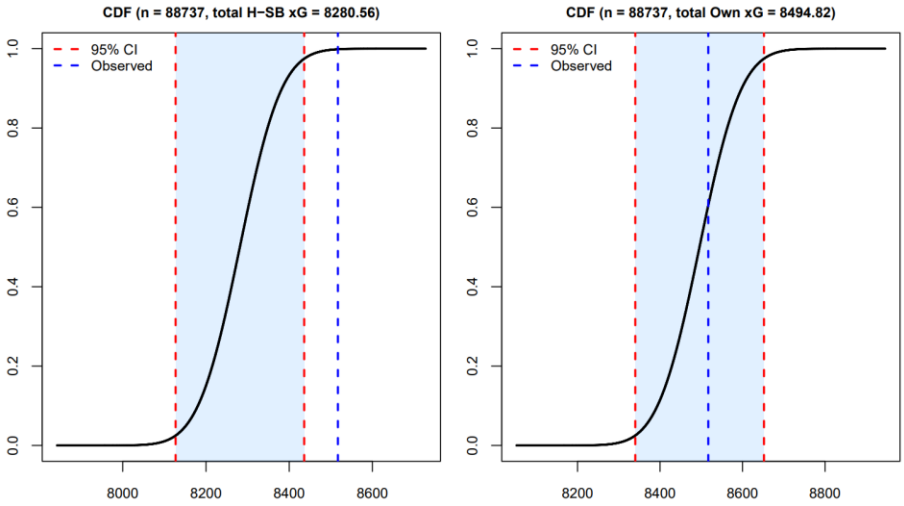

Figure 7: Goal distribution H-SB (left) vs. own xG (right) model

We observe that for our model, the total goals lie well within the prediction interval. This suggests that the feature sufficiency assumption is more reasonable for our model after accounting for the variables where we previously identified biases in the Hudl-Statsbomb model.

Finally, we assess whether the inclusion of the additional features also helps to remove the situational biases observed in the Hudl-Statsbomb model.

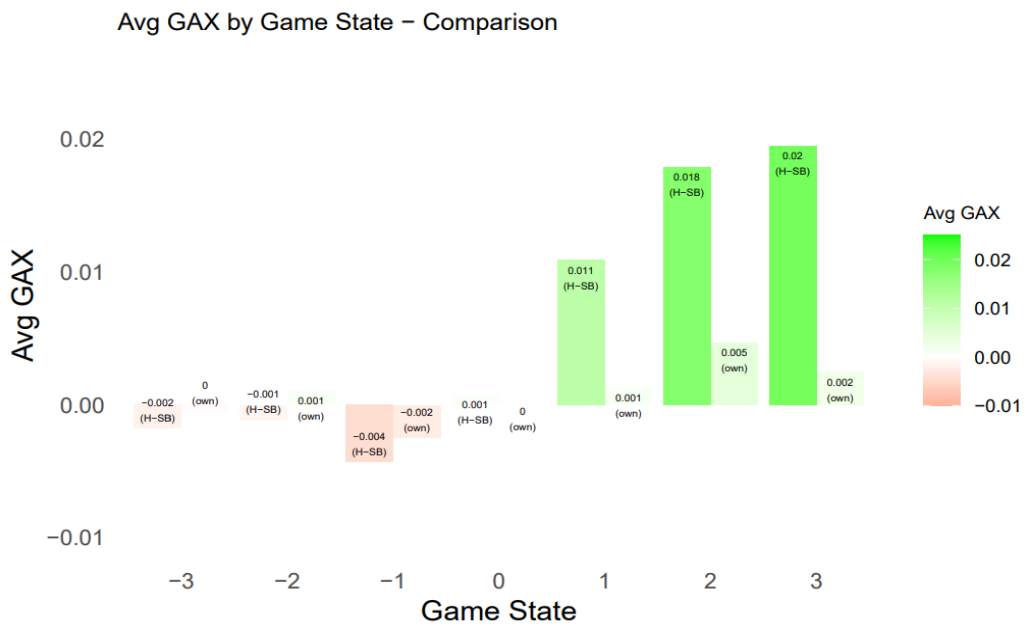

Figure 8: Avg. GAX by game state: H-SB (left) vs. own xG (right) model

Figure 8 shows the average GAX by game state. For each game state, the left bar represents the Hudl-Statsbomb model and the right bar our own model. We observe that the bars of our model are consistently close to zero across all game states, demonstrating that including the scoreline drastically reduces this bias.

category	avg. GAX (H-SB)	avg. GAX (own)	n shots
Strong	0.0091	0.0014	57,485
Weak	-0.0106	0.0024	15,746
Other	-0.0078	-0.0054	15,506

Table 5: Avg. GAX by dominant foot categories: H-SB vs. own xG model

Table 5 presents the results for the dominant foot categories. Similar to the findings for game state, the average GAX values in our model are close to zero across all three categories, indicating that this bias has also been effectively removed.

category	avg. GAX (H-SB)	Avg. GAX (own)	n shots
Rebound	0.0104	0.0081	4,293
No rebound	0.0023	0.0000	84,444

Table 6: Avg. GAX by rebound categories: H-SB vs. own xG model

Finally, Table 6 provides results for rebound situations. We observe an average GAX of exactly zero for non-rebound shots, but a relatively high value of 0.0081 for shots in rebound situations. While it represents a slight improvement compared to the Hudl-Statsbomb model's value of 0.0104, it suggests that a small bias remains in our model even after explicitly accounting for rebound situations in the feature set.

Altogether, we find that our model achieves improved predictive performance and calibration compared to the Hudl-Statsbomb benchmark xG model. Moreover, by explicitly incorporating further contextual information, we substantially reduce situational biases identified earlier, and the feature sufficiency assumption is more reasonable, as demonstrated by the theoretical goal distribution.

5. Use case specific xG models

While our own model developed in Section 4 shows promising results in terms of various evaluation criteria, we discuss the choice of xG models for specific use cases in this section. In particular, we argue that, depending on the interest of the analyst, it is not always optimal to stick to one xG model, but to adjust the xG model to the particular use case.

5.1 Team strengths

A commonly discussed topic is whether one should add team strength variables to xG models or not (ElHabr 2023a, and others). There are various ways to incorporate team strengths in an xG model. One popular approach (ElHabr 2023a) uses ELO ratings (see e.g. www.clubelo.com). In this paper, we take a different approach and use a bivariate Poisson model (Karlis and Ntzoufras, 2003) to derive team strengths. That is, using historic match data obtained from https://www.football-data.co.uk/, we estimate team strength parameters via maximum likelihood estimation (for more details, see e.g. Bajons and Kook, 2025). A particular advantage of this approach is that we obtain attacking and defending strengths for each team.

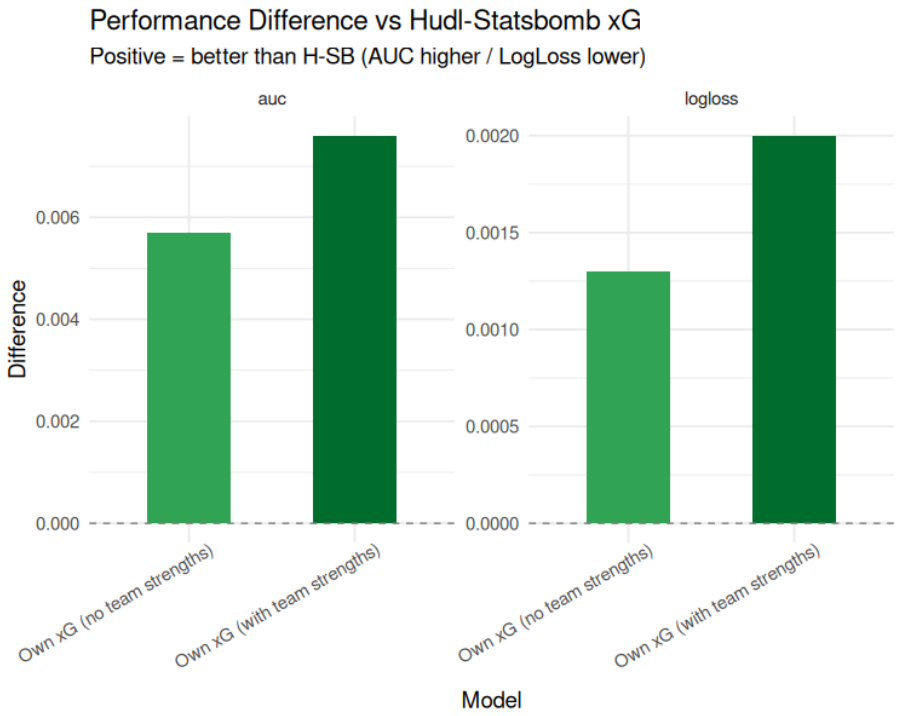

Figure 9: Performance of xG models with and without team strengths as compared to Hudl-Statsbomb xG as baseline. On the left we compare differences in AUC value, on the right, we compare differences in logloss. In both cases, differences are scaled such that positive bars represent better performance than the baseline.

In the following, we aim to shed light on the controversial question of adding team strengths to xG models. In particular, we argue that the inclusion of team strength variables depends heavily on the particular use case. When the goal is to predict match outcomes or to simulate future games based on xG estimates, then adding team strength variables is sensible and improves performance, as they provide more informed estimates for xG values. On the contrary, evaluating players using xG models with team strength variables can obscure individual contribution, as the team strength largely depends on player strength.

To demonstrate our point, we first analyze the performance of models using team strength variables against models without team strengths. More specifically, Figure 9 shows the performance difference for two loss metrics (AUC and logloss) of our own models fitted as discussed in Section 4 against Hudl-Statsbomb's xG model as a baseline. We observe that adding team strengths to our model derived in Section 4, improves upon the model without team strengths, while both perform better than the Hudl-Statsbomb model. This illustrates that from a purely predictive standpoint, team strength variables enhance the model. Additionally, we can perform a statistical test to detect whether the attacking and defensive strength parameters significantly impact the outcome of a shot. We use the Generalised Covariance Measure (GCM) test developed by Shah and Peters (2020), a non-parametric conditional independence test. This test allows us to identify whether a variable X adds power to predicting the outcome variable Y given other potentially relevant factors Z, i.e., it tests whether Y and X are conditionally independent given Z. In our case, X is the team strength parameter of interest (either attacking or defensive

strength), Y is the outcome of an xG model (i.e., whether a goal ended in a shot or not), and Z contains all other potentially relevant covariates for the xG model (all features derived in Section 4.1). Figure 10 shows the resulting p-values of the GCM test for the attacking and defensive team strength variable on a -log_10 scale. All tests are highly significant, indicating that team strength variables are strongly adding predictive power to the xG model even when conditioning on all other relevant features.

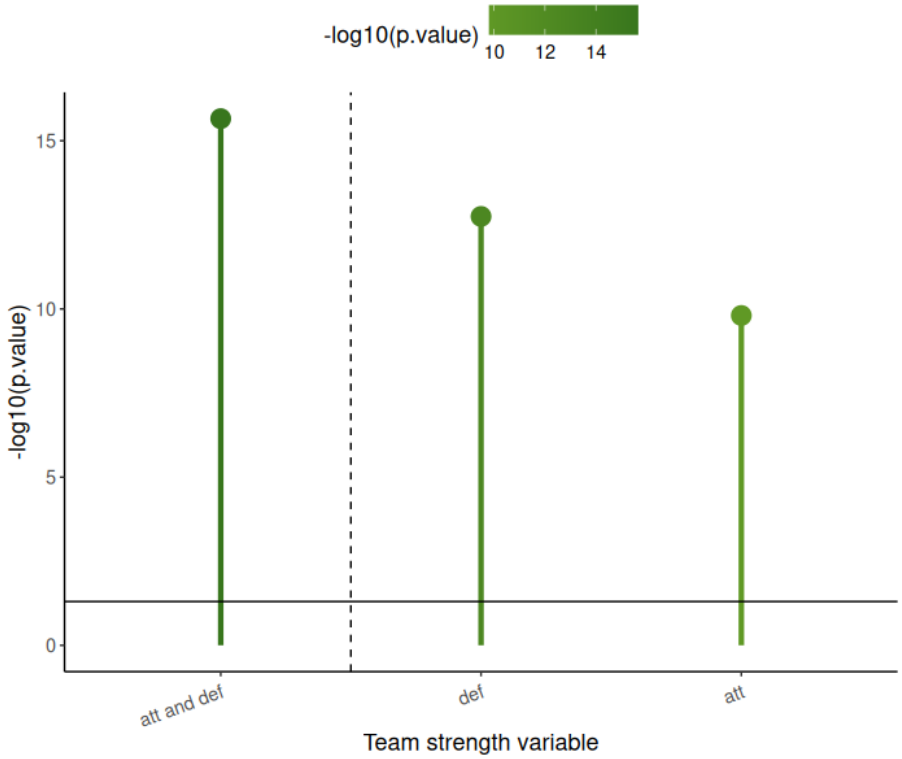

Figure 10: p-values of attacking and defensive parameters using the GCM test to test for conditional independence. p-values are displayed on a -log10 scale, i.e. higher means lower p-value. The horizontal solid black line represents a p-value of 5%, i.e. all dots above this line represent significant values at the 5% level. On the left of the vertical dashed line, we test on attacking and defensive strengths simultaneously, on the right, we test the two strength parameters individually.

Finally, we consider a specific use case, where an xG model with team strengths is superior to models not using team strengths. For this toy use case, we are interested in predicting future match performance based on xG values. The aim is to predict the outcome of a match (home win/draw/away win) using four features: the average xG for and against of the home team (averaged over the last five matches), and the same for the away team. Applying a multinomial regression model, we can then use these four features to predict probabilities for each of the three possible outcomes. We train five different models using the four xG-based predictors derived from:

- Hudl-Statsbomb's xG model
- Our own model without team strengths
- Our own model with attacking strength only
- Our own model with defensive strength only

- Our own model with attacking and defensive strengths

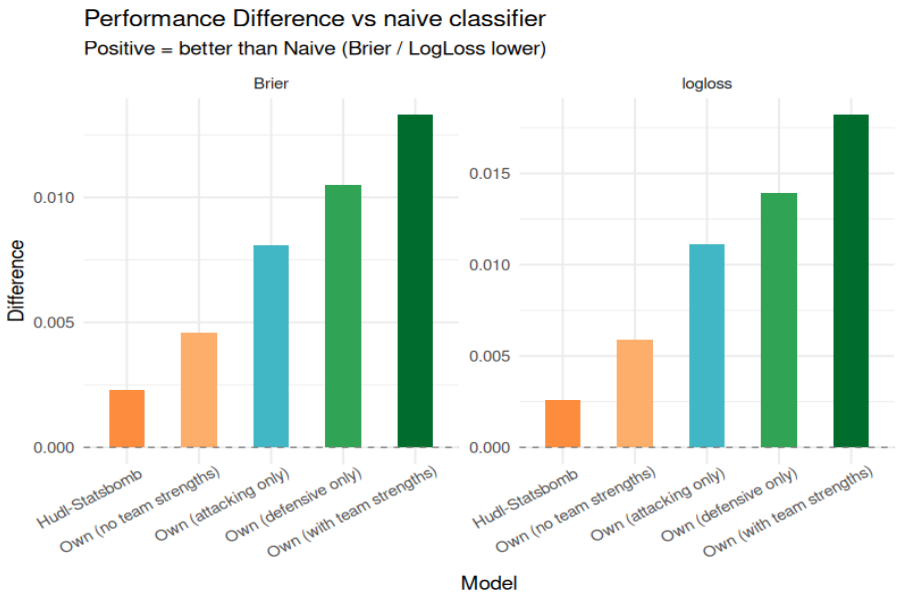

Figure 11: Performance of match prediction models using xG values from different models as predictors as compared to a Naive model as baseline. On the left, we compare differences in Brier score, on the right, we compare differences in logloss. In both cases, differences are scaled such that positive bars represent better performance than the baseline.

Figure 11 shows the result of our game prediction exercise. We compare model performance against a naive baseline (using the empirical goals-to-shots ratio of all shots, i.e. 0.096, as naive xG proxy for each shot) based on two commonly used loss metrics for multi-class problems, the Brier score and the logloss. The figure shows that adding team strength variables allows for better prediction of the future match outcome, where the best-performing model is the one that uses both attacking and defending team strength variables for the computation of the xG values. This clearly demonstrates the superiority of team strength-enhanced xG models for future match outcome prediction.

5.2 Player evaluation

Next, we turn to the problem of evaluating players by using xG values. A particularly common approach is to consider goals above expectation (GAX), i.e., for all of the shots of a player in a season, we compare the actual number of goals to the expected number of goals. While GAX are a very intuitive way to identify over- or underperformance in shooting skills, they have recently been criticized and labeled a bad metric for various reasons in the sports analytics community (Baron et al., 2024; Davis and Robberechts, 2024). To address these issues, Bajons and Kook (2025) derived an extension of GAX called residualized GAX (rGAX), which we will use in this paper to analyze

shooting skills. The main focus is to show that, when using a reasonable metric to analyze shooting skills, it is of utmost importance to choose the correct xG model.

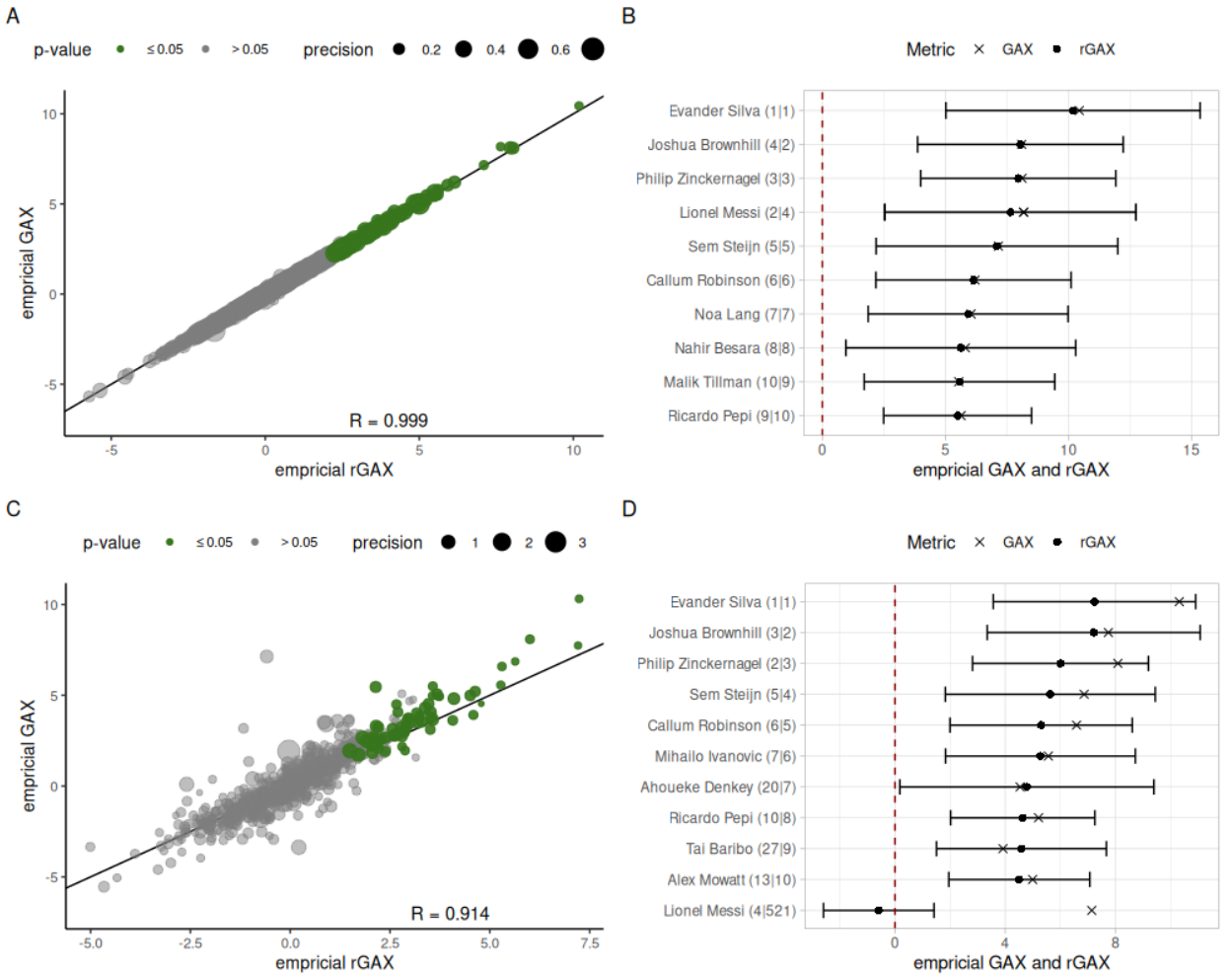

Figure 12: Comparison of GAX against rGAX values. A: Scatterplot of GAX and rGAX computed from an xG model using defensive team strengths only (additional to all other relevant features). The solid line indicates the identity. The correlation coefficient R is added to the plot. B: Player-wise GAX and rGAX for the defensive team strength only xG model with one-sided 95% confidence intervals for rGAX for the top 10 players. C: Scatterplot of GAX and rGAX computed from an xG model using both attacking and defensive team strengths (additional to all other relevant features). D: Player-wise GAX and rGAX for the xG model using attacking and defensive team strengths with one-sided 95% confidence intervals for rGAX for the top 10 players and Lionel Messi.

To illustrate this point, we consider two different ways to compute GAX and rGAX. First, we create an xG model using only defensive team strengths in addition to all relevant features as described in Section 4.1. Second, we use an xG model with both attacking and defensive team strengths. Figure 12 shows the results of our analysis. In particular, Figures 12A and 12B compare GAX and rGAX values for the xG model using only defensive team strengths, whereas Figures 12C and 12D compare GAX and rGAX for the model using both attacking and defensive team strengths. For the defensive team strength only xG model, we observe a desirable behaviour: The GAX and rGAX values are highly correlated. This is a particularly nice result, because rGAX in essence measure shooting skills similar to GAX, i.e., an analyst can interpret GAX and rGAX in the same way. However, rGAX allow for valid uncertainty quantification, i.e., they allow for the computation of

confidence intervals (as seen in Figure 12B) and allow for testing whether a player significantly over- or underperformed in a particular season (see green dots in Figure 12A). Additionally, rGAX is much more robust to selection effects in the data, a key issue of GAX (see Bajons and Kook, 2025, for more details). Figure 12C and Figure 12D display similar results for the xG model using both attacking and defensive team strengths parameters. We observe interesting patterns: First, although still being high, the correlation between GAX and rGAX is lower than for the model before, indicating that for some players rGAX and GAX are still very comparable, while for other players, the values differ. Figure 12D shows how the values of rGAX and GAX differ for selected players. A particularly interesting case is Lionel Messi, whose rGAX is suddenly negative as opposed to his GAX. This is also in stark contrast to his rGAX using the defending team strength only xG model (Figure 12B), where Messi obtains a significantly positive rGAX value. The reason for this drastic change is precisely the inclusion of attacking strengths in the computation of rGAX. The attacking strength of a team is largely influenced by the strength of the offensive players on that team. Hence, when trying to identify player strength, it is circular to use attacking strength. Or put differently, the attacking strength parameter is a "bad" control variable, i.e., a variable that when added to the model produces a discrepancy between the estimated coefficient (in our case rGAX) and the effect that the coefficient is intended to represent (Cinelli et al., 2024). The effect of the "bad" control variable is particularly pronounced for Lionel Messi, who undoubtedly has the strongest impact on the attacking strength parameter of his team, Inter Miami. These results indicate the importance of choosing an appropriate xG model for shooting skill evaluation. A similar reasoning should be applied when using xG (or post-shot xG, psxG) models to evaluate the goal-stopping ability of goalkeepers. In this case, the xG (or psxG) model should not include the defensive strength parameter, because the goalkeeper potentially influences this parameter strongly.

5.3 League-specific models

In this section, we discuss the development of xG models for a team or analyst that is specifically interested in one particular league. We act on the premise that data is expensive and the team/analysts only have access to data from their own league (and potentially freely available data). That is, in order to train a good xG model, only limited data is available. In Section 4.1, we derived a large number of 27 potential features for xG models. Together with the two team strength variables, this leaves 29 variables on which we train the model. While this number is not necessarily large, especially in the full data set of shots ($N \approx 89000$), for models trained on only data from one league and seasons, where typically $N < 10000$, the feature set may pose an issue. Especially in the case of xG models, where the effective dimension can be a lot bigger due to dummy encoded categorical variables, multicollinearity, and complex interactions between features. Additionally, including noisy features when fitting complex machine learning algorithms (such as XGBoost models) on small data increases the risk of overfitting. While such variables should be included when more data is available, because collectively noisy or weakly important

features provide richer context and may help in generalization, the opposite holds for small datasets (see also bias-variance trade-off, Hastie, Tibshirani and Friedman, 2013).

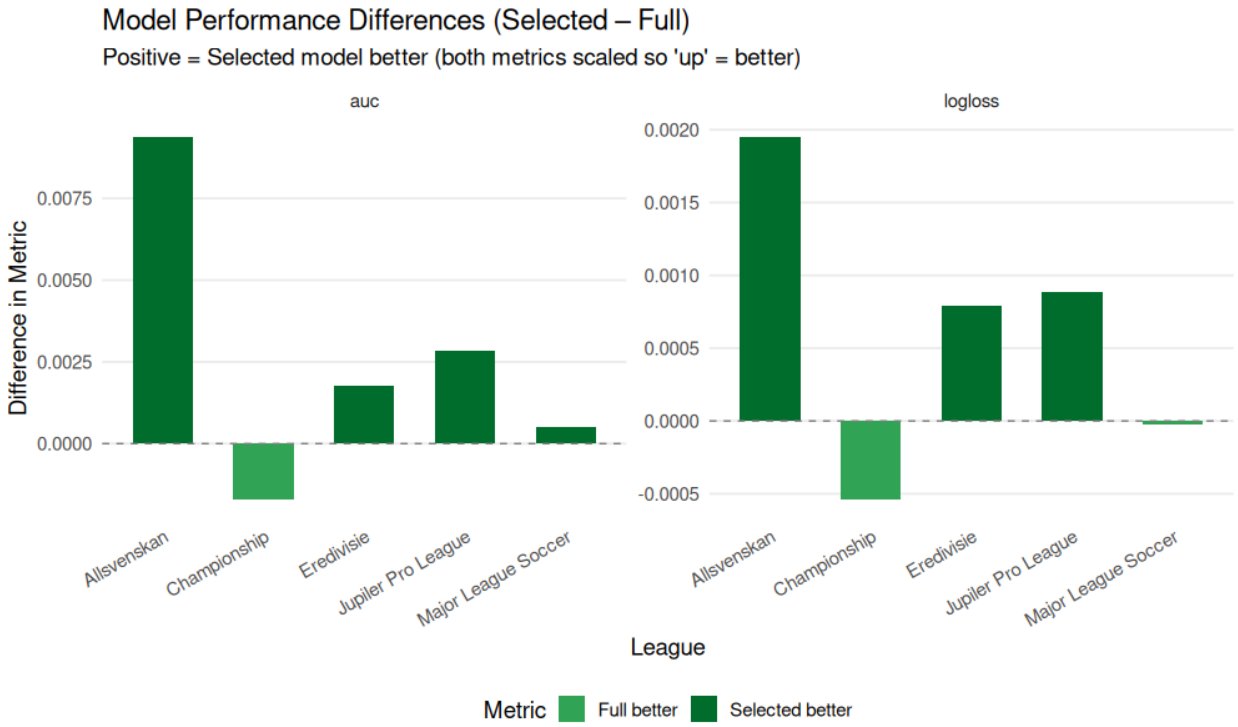

Figure 13: Performance of an xG model using only selected features against an xG model using all features for individual leagues. On the left, we compare differences in AUC. On the right, we compare differences in logloss. In both cases, differences are scaled such that positive bars represent better performance of the model using only selected features.

In a first step towards deriving league-specific models with limited data availability, we devise a strategy to identify important variables to retain for a small model. To do so, we develop a forward stagewise procedure, again relying on the previously described GCM test. While there are other strategies to identify a set of relevant variables, our approach using the GCM test has several advantages:

1. The GCM test measures the conditional contribution of a variable beyond those already selected, i.e., it checks at each step whether a variable significantly adds predictive information, conditional on the variables that are already in the model.
2. The GCM test is distribution-free, i.e., it allows for testing conditional independence without relying on strong modelling assumptions (such as in a linear regression model). Hence, our procedure allows for the choice of any flexible machine learning model capturing non-linear and complex relationships between variables. This is in stark contrast to classical stepwise regression for linear models.
3. Due to the selection of variables using valid statistical tests, we obtain a principled way of identifying relevant variables, without having to rely on subjective evaluation, as for example when keeping only variables that are important according to a variable importance measure.

More specifically, our proposed selection algorithm starts with a small model consisting only of well-known and highly important baseline features such as the distance to goal, the goal angle, and the body part (Pollard and Reep, 1997, Robberechts and Davis, 2020). Subsequently, at each step of the procedure, we add the variable that most significantly affects the outcome Y based on the GCM test. The procedure stops when all relevant (i.e., statistically significant) variables are added to the model. This procedure results in the selection of 19 relevant variables (see Table 7 in the Appendix for a full list of selected variables), which we use for the estimation of an xG model on league-specific data. As a validation, we compare the performance of the full model against the model using only the selected features in Figure 13. We observe that the smaller model, i.e., the model using only selected variables as opposed to the full set of variables, performs better for almost all leagues. Interestingly, only for the English Championship, the full model performs better than the model using selected variables only in both evaluation metrics. In our competition dataset, the Championship is also the league where we observe by far the most shots (N = 13112, as opposed to Allsvenskan, N = 4987, Eredivisie, N = 8036, Jupiler Pro League, N = 6362, and Major League Soccer, N = 11119). Additionally, the smaller model works best for the Swedish Allsvenskan, where we only observed 4987 shots, again reinforcing our findings that smaller models work better for smaller datasets.

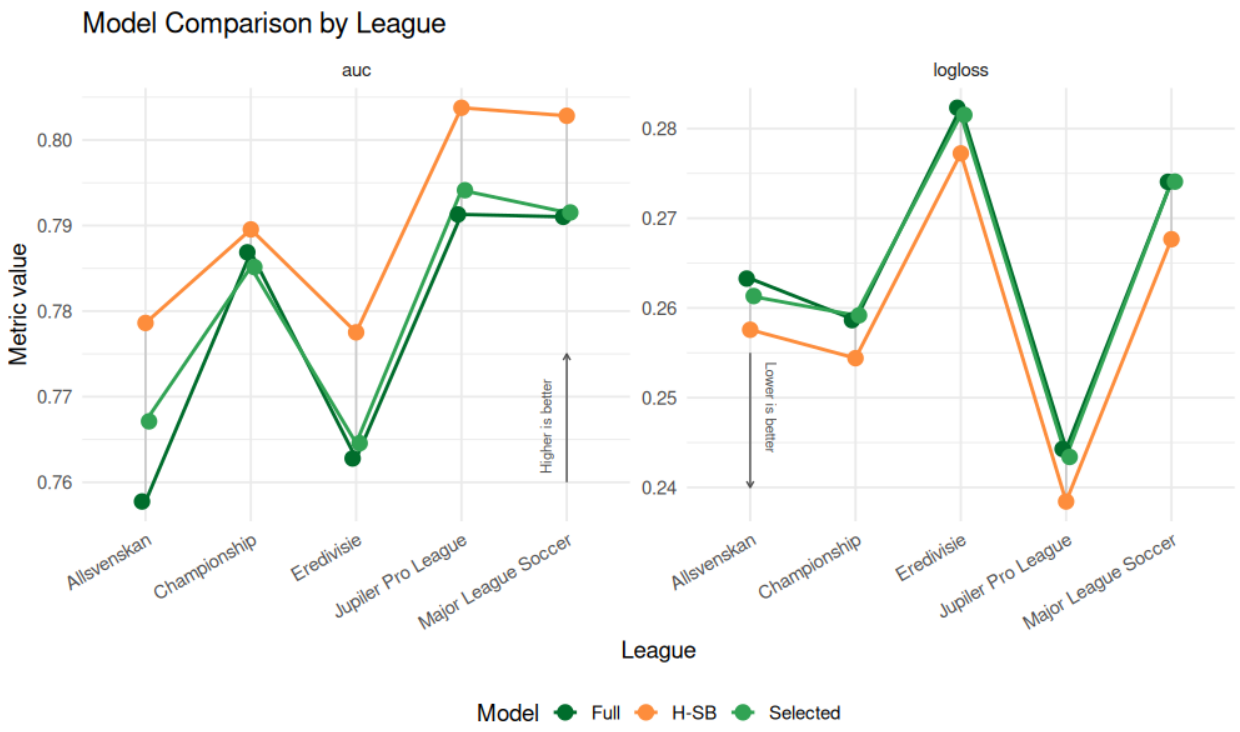

Figure 14: Performance of an xG model using only selected features, an xG model using all features, and Hudl-Statsbombs xG model for individual leagues. On the left, we display AUC values (a higher value is better). On the right, we display loglosses (a lower value is better).

In Figure 14, we see that, while the smaller model works better for most leagues, the best-performing model is the Hudl-Statsomb xG model. This comes to no surprise, as this model is trained on a much larger dataset. This raises the question whether we can improve on the global xG

model of Hudl-Statsbomb and find a model that fits the particular idiosyncrasies of the league of interest.

In order to improve the xG model, we make use of both the Hudl-Statsbomb model and the model trained on the league-specific data at hand. We use a stacking approach to combine each model's prediction to obtain a more refined prediction. There are various approaches to how one could stack models. The most straightforward idea is to use a convex combination of the predictions. A more refined approach is to use a meta-learner, i.e., a machine learning model to combine the predictions. In both cases, one needs to learn the optimal stacking parameters, i.e., either the weights for the convex combination or the parameters for the model. In this work, we use a logistic regression model on a logit transform of the probabilities from each model to obtain our stacked model. In order to learn the optimal parameters for this logistic regression model for each league, we trained a model on all other competition leagues. That is, for example, in order to derive the optimal parameter for the Swedish Allsvenskan, we used the predictions from the two xG models (the Hudl-Statsbomb and our xG model) for all four other leagues (Championship, Eredivisie, Jupiler Pro League, Major League Soccer) and fit a logistic regression model to all shots in these leagues. Then, we obtain the stacked probabilities for each shot of the Allsvenskan by using this model.

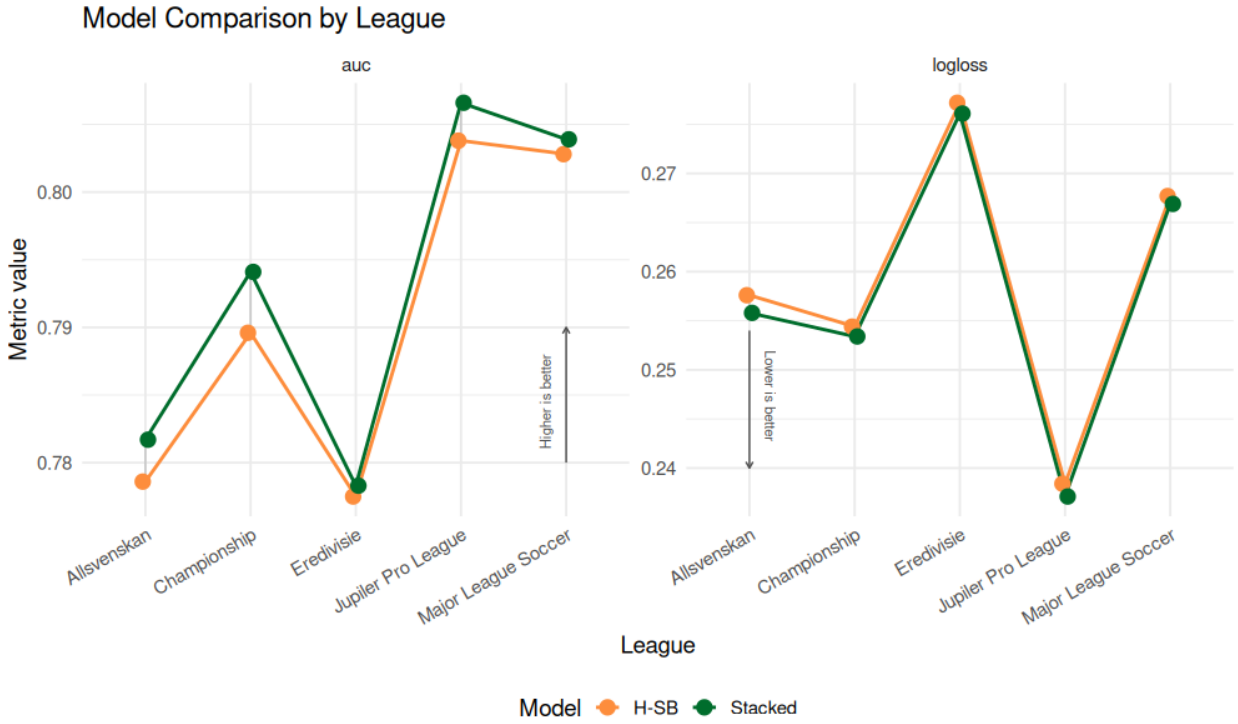

Figure 15: Performance of a stacked xG model and Hudl-Statsbombs xG model for individual leagues. On the left, we display AUC values (a higher value is better). On the right, we display loglosses (a lower value is better).

In Figure 15, we display the results from the stacked model and the Hudl-Statsbomb model for AUC and logloss. In all five leagues used for the competition, the stacked model is able to outperform Hudl-Statsbomb's xG model.

6. Summary and Conclusion

In this paper, we contribute to the hot topic of xG models in football. In particular, we analyze Hudl-Statsbomb's global xG model in detail. While the model performs reasonably well, our analyses show that using this globally trained model does not fully capture the idiosyncrasies of the specific leagues of interest. In particular, the model underpredicts total xG for the available data in certain scenarios and seems to react to biases from not accounting for variables such as the game state. Motivated by this finding, we derive our own xG model, which shows improvements over the Hudl-Statsbomb's xG model in almost all evaluation metrics. The features used for our xG model are motivated by standard features used in the literature and also adjusted for potentially biasing factors such as the game state or whether a shot resulted from a rebound. Having obtained a well-performing model, we tackle the question of whether there are specific use cases in which we can obtain an even more potent model or whether there are use cases where the model should be adjusted. More specifically, we find that, when interested in team analyses, match prediction, or game simulation, adding team strengths to the model improves performance. However, when interested in evaluating players, e.g., their shooting skill via GAX, adding (attacking) team strengths can drastically obscure finishing because especially good players influence their team's attacking strength strongly. Finally, if only a small amount of data is available, it can make sense to limit the number of features that are used for the xG model. In our results, we show that when only analysing one specific league, models using only selected relevant features perform in general better than models trained on all features. Furthermore, when there are more xG values available, such as e.g., the ones from Hudl-Statsbomb, one can drastically improve on model performance by using a stacking approach, which combines the predictions of two (or more) models. In total, our analyses show that there is not one xG model to rule them all, but that it makes sense to adjust the xG model to the specific needs of the analyst.

References

Bajons, R., & Kook, L. (2025). *Rethinking player evaluation in sports: Goals above expectation and beyond*. https://arxiv.org/abs/2509.20083

Baron, E., Sandholtz, N., Pleuler, D., & Chan, T. C. Y. (2024). Miss it like Messi: Extracting value from off-target shots in soccer. *Journal of Quantitative Analysis in Sports*, *20*(1), 37–50. https://doi.org/10.1515/jqas-2022-0107

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Cinelli, C., Forney, A., & Pearl, J. (2024). A Crash Course in Good and Bad Controls. *Sociological Methods & Research*, *53*(3), 1071–1104. https://doi.org/10.1177/00491241221099552

Davis, J., & Robberechts, P. (2024). *Biases in Expected Goals Models Confound Finishing Ability*. https://arxiv.org/abs/2401.09940

ElHabr, T. (2023a). *xG Model Calibration*. Tony's Blog. Retrieved from https://tonyelhabr.rbind.io/posts/opta-xg-model-calibration/

ElHabr, T. (2023b). *Should we account for team quality in an xG model?* Tony's Blog. Retrieved from https://tonyelhabr.rbind.io/posts/xg-team-quality/

Hastie, T., Tibshirani, R., & Friedman, J. (2013). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer New York. https://books.google.at/books?id=MuNmawEACAAJ

Hewitt, J. H., & Karakuş, O. (2023). A machine learning approach for player and position adjusted expected goals in football (soccer). *Franklin Open*, *4*, 100034. https://doi.org/10.1016/j.fraope.2023.100034

Karlis, D., & Ntzoufras, I. (2003). Analysis of sports data by using bivariate Poisson models. *Journal of the Royal Statistical Society: Series D (The Statistician)*, *52*(3), 381–393. https://doi.org/10.1111/1467-9884.00366

Litwitz, K., Memmert, D., & Wunderlich, F. (2024). Rebounds in football: A systematic investigation of characteristics of goals scored after rebounded balls in English Premier League seasons 2012/2013 to 2018/2019. *International Journal of Sports Science & Coaching*, *19*(6), 2476–2488. https://doi.org/10.1177/17479541241269007

Mead, J., O'Hare, A., & McMenemy, P. (2023). Expected goals in football: Improving model performance and demonstrating value. *PLOS ONE*, *18*(4), e0282295. https://doi.org/10.1371/journal.pone.0282295

Pollard, R., & Reep, C. (1997). Measuring the effectiveness of playing strategies at soccer. *Journal of the Royal Statistical Society: Series D (The Statistician)*, *46*(4), 541–550. https://doi.org/10.1111/1467-9884.00108

Robberechts, P., & Davis, J. (2020). How Data Availability Affects the Ability to Learn Good xG Models. In U. Brefeld, J. Davis, J. Van Haaren, & A. Zimmermann (Eds.), *Machine Learning and Data Mining for Sports Analytics* (pp. 17–27). Springer International Publishing. https://doi.org/10.1007/978-3-030-64912-8_2

Shah, R. D., & Peters, J. (2020). The hardness of conditional independence testing and the generalised covariance measure. *The Annals of Statistics*, *48*(3), 1514–1538. https://doi.org/10.1214/19-AOS1857

Vatvani, D. (2022). *Upgrading Expected Goals*. Hudl Blog. Retrieved from https://www.hudl.com/blog/upgrading-expected-goals

Yam, D. (2025). *StatsBombR: Cleans and pulls StatsBomb data from the API*. https://github.com/statsbomb/StatsBombR

Appendix

Feature List:

Table 7 shows a list containing all derived features for our own xG model. Variables "att" and "def" are not included in the model derived in Section 4.1, but in the team-strength specific models. All variables which are marked with * are selected for the league specific model using the GCM variable selection procedure.

variable	type	description
shot.type.name	categorical	Shot type (one of Open Play, Free Kick)
shot.technique.name*	categorical	Shot technique (one of Normal, (Half) Volley, Diving Header, Backheel, Lob, Overhead kick
shot.body_part.name*	categorical	Shot body part (one of Head, Foot, Other)
DistToGoal*	numeric	Distance of shooter to center of the goal
DistToKeeper*	numeric	Distance of goalkeeper to goal
DistSGK*	numeric	Distance of shooter to goalkeeper
distance.ToD1	numeric	Distance of shooter to closest defender (in front of shooter)
distance.ToD2	numeric	Distance of shooter to 2nd closest defender (in front of shooter)
distance.ToD1.360*	numeric	Distance of shooter to closest defender (general)
distance.ToD2.360*	numeric	Distance of shooter to 2nd closest defender (general)
AngleToGoal	numeric	Angle between shot and the center of the goal (in degree)
AngleToKeeper	numeric	Angle between goalkeeper and centre of goal (in degree)
AngleDeviation*	numeric	Absolute difference of the two angles
angle*	numeric	Angle between shot and goal posts (in radians)
AttackersBehindBall	integer	Attackers behind the ball (in x coordinate)
DefendersBehindBall*	integer	Defenders behind the ball (in x coordinate)
DefendersInCone*	integer	Defenders in cone drawn drawn from shot to goal posts
density*	numeric	Free space for shooter, sum over the inverse of distances from shooter to defenders
density.incone	numeric	Sum over the inverse of distances from shooter to defenders in cone drawn from shot to goal posts

rebound*	boolean	Indicator if a shot is in a rebound situation, 1 if shot happens within 3 seconds of prior shot in same possession sequence, 0 else
game_state*	integer	Score differential of shooting team's perspective, capped at +/- 3
time_interval	categorical	Time interval (one of (0-45), (45-75), (75+))
shot.strong_foot*	categorical	Strong foot (one of Strong, Weak, Other)
gauss_mass_conegoal_def	numeric	Sum of Gaussian mass from defenders within cone from shot to goal posts
gauss_mass_conegk_def*	numeric	Sum of Gaussian mass from defenders within cone from shot to goalkeeper y +/- 1
gauss_mass_conegoal_att	numeric	Sum of Gaussian mass from attackers within cone from shot to goal posts
gauss_mass_conegk_att*	numeric	Sum of Gaussian mass from attackers within cone from shot to goalkeeper y +/- 1
att*	numeric	attacking strength of team shooting
def*	numeric	defensive strength of team conceding shot

Table 7: Full feature list of own xG model