

Getting NBA Shots in Context: Analysing Basketball Shots with Graph Embeddings

Marc Schmid, Moritz Schöpf and Otto Kolbinger

Technical University Munich

Abstract

Evaluating the quality of shots in basketball is crucial and requires considering the context in which they are taken. We introduce a graph neural network to process a graph based on player and ball tracking data to compute expected shot quality. We evaluate this model against other models focusing on calibration. The messages between spatial and temporal features are separated, and an attention mechanism is implemented, making the graph neural network interpretable. We use the GNNExplainer to further show the importance of node features. To demonstrate possible practical applications, we analyse the embeddings of the graph neural network concerning different situations like the mean of all player predictions or similarity between created shots and compare this to existing methods.

KEYWORDS: BASKETBALL ANALYTICS, GRAPH NEURAL NETWORKS, EMBEDDINGS

.

Introduction

Over the last decade, there has been an increasing amount of spatio-temporal tracking data in sports provided by different providers like StatsPerform or Kinexon. Scientists and sports analysts started to use this data to create insights for coaches and fans by creating metrics like expected goals (xG) in soccer or expected shot quality (ESQ) in basketball. Many of those metrics and systems are valuable in the workflow of game analysts by accelerating their search process through a dataset of plays with trajectory encodings (Miller & Bornn, 2017), giving more granular insights into the game (Schmid et al., 2021) or helping athletic coaches to monitor player load (Ravé et al., 2020).

Recently, there has been increased coverage of the xG metric in soccer TV broadcasts and the scientific community (Anzer & Bauer, 2021). xG describes how many goals can be expected regarding the input parameters of the model, which are the position of the shooter, the closest defenders, the angle to the goal, and many more. Furthermore, the players can be ranked for their effectiveness regarding the metric. Field goal percentage (FG%) is a key statistic in basketball, that reports the number of made field goals divided by attempted field goals. Over time, analysis of shooting has evolved to incorporate spatial and temporal dependencies, which provide deeper insights into shooting performance. Recent studies have leveraged FG% to analyze spatial shooting patterns, such as the visually intuitive shot charts (Goldsberry, 2012). These insights has proven to be useable for player recruitment and game analytics. Miller et al. (2014), introduced a double stochastic model to construct shooting intensity surfaces, condensing these into a set of spatial basis vectors with different factorization methods. The factorized shooting maps allow computing the similarity among players on the basis of the shooting behavior in a low dimensional space.

Subsequently, expected possession value (EPV) has been introduced that are rooted in Markov chains and decision-making frameworks. Cervone et al. (2014) use tracking data and event data to construct an estimate of a value for every action involving a ball in a basketball game. The model then enables observers to quantify how valuable specific actions are, by investigating the change in EPV for a specific action.



Figure 1 Left: Tracking data provided by SportsVU, Right: Video footage of Stephen Curry shooting a 3pt shot over Anthony Davis in a game of the Golden State Warriors against the New Orleans Pelicans

This study focuses on the expected shot quality (ESQ), a key component of a EPV. ESQ represents the probability of a shot being successful and has been previously studied through shot chart analyses (Reich et al., 2006) and machine learning models utilizing event data (Chang et al., 2014; Oughali et al., 2019). Traditional ESQ measures, such as dividing field goal makes (FGM) by field goal attempts (FGA) on a spatial distribution of the court, provide a static view of shooting ability. In contrast, models from commercial providers like the Spax (Cheema, 2019) introduced an expected efficient field goal percentage, which models the ESQ by including time

left on the shot clock and distance to the closest defender. Despite the proliferation of ESQ models by companies like Second Spectrum and ESPN, their proprietary methods remain unpublished. Notably, existing models share a common limitation: they focus on static time frames and fail to capture the underlying process by which shots are generated, aside from potentially including shot type as a feature.

Famous players like Russell Westbrook, were criticized during a bad series of the LA Lakers in the 21/22 NBA season, but hypothetically hit more shots than expected from those highly contested shots. Another player whose shot effectiveness is hard to capture is, for example James Harden, as he needs to create many of his shots for himself and did not shoot many catch and shoots in the previous seasons. The temporal data evolution must be added to the model to capture shot creation and execution for such players to account for the special circumstances they are in.

Figure 1 shows the players' positional data and the respective play from video footage. We can recover an expected shot-made value according to a binary classification problem from the positional tracking data at a specific time frame. This can be extended by taking into account the spatial and temporal history of the play to provide additional context.

Recently, graph neural networks (GNNs) have arisen as a way to study and predict graph-structured data like proteins (Battaglia et al., 2018), molecules (Fang et al., 2021), or soccer passes (Stöckl et al., 2021) effectively. Instead of connecting all the input data, like a fully connected neural network, GNNs can capture unstructured locality by aggregating information via a neighbourhood function. For example images, represent a ordered graph with uniform distance between the nodes, which make convolutional neural network an instantiation of graph neural networks due to the structural design of an image. In contrast, for unstructured data, the single nodes (representing images' pixels) are connected via edges or a neighbourhood function. In the case of tracking data, a graph could be represented by the position of the players, and the neighbourhood function could be the Euclidean distance.

While prior research used boosting, trees, logistic regression or artificial neural networks and major feature engineering to build a probability surface for successful shots based on the time frame of the shot (Oughali et al., 2019), our approach is to use the graph-like structure of the tracking data to efficiently compute the probability surface of making a shot. Using GNNs enables us to investigate graph embeddings and compare different shots, players and positions in a transformed space, which is impossible with tree-based models or logistic regression.

Therefore, our contributions are:

- We propose an end-to-end trainable Graph neural network architecture that captures space and time in two different dimensions by separation of information aggregation via space and time edges.
- We, provide a solution that does not require investing in heavy feature engineering to achieve competitive results.
- We analyze and visualize embeddings of players, positions and teams with t-SNE (Maaten & Hinton, 2008) and allow coaches and scouts to compare players with a nearest neighbor search based on shot selection/success and creation.

Next, we will introduce related work in graph neural networks and field goal prediction. Then, we will specify this approach's methodology and present the results of the training procedures. Finally, we conduct experiments with the graph embeddings and showcase use cases of the proposed ESQ model.

Related Work

Graph Neural Networks

Graph neural networks are consolidating tools to perform tasks, including link prediction, node classification or graph classification. Bronstein et al. (2017) surveys the problems in applying deep learning to graph-structured data. Under Gilmer et al. (2017), different GNN structures are homogenized under the term message passing neural networks (MPNN) and the connections between different aggregation functions, nodes and edges. In contrast to previous work (Kipf & Welling, 2017; Schütt et al., 2017) that builds on concepts and ideas around spectral filtering and graph Fourier transformation, Gilmer et al. (2017) worked with a more geometrical, graphical understanding of the underlying algorithms. Recent work shows extensive progress in the performance of MPNN in quantum chemistry (Unke et al., 2021), social networks (Tan et al., 2019), traffic forecasting (Li et al., 2018) or football pass completion prediction (Stöckl et al., 2021). All those implementations used the naturally graphical representation of the data in spatial or temporal dimensions. In traffic forecasting, spatio-temporal graph convolutional networks (STGCN) (Yu et al., 2018) and diffusion convolutional recurrent neural networks (DCRNN) (Li et al., 2018) are dominant approaches. STGCN layers are "sandwiched" layers consisting of spatial and temporal convolutions that can be optimized in parallel and more effectively than recurrent units. In contrast, DCRNNs spatial structure is based on bidirectional graph random walks and decouples the temporal dependencies with encoder-decoder structures. Velićkovic et al. (2018) introduced a graph attention mechanism that allows generalizing well on different data sets and across multiple special networks with implemented attention mechanisms (Duan et al., 2017). Dick and Brefeld (Dick & Brefeld, 2023) use recurrent graph neural networks to model inter-dependencies between players for football. Rahimian et al. (Rahimian et al., 2024) use GNNs to predict pass success probabilities in football. Eventually, graph networks are well suited for modeling interactions between unordered sets of datapoints that preserve a structural integrity like player trajectories, the interaction between the players or the geometrical dependencies between them.

Expected Field Goal/Shot Quality Models

The earliest models of field goal attempts in basketball relied on a straightforward division of the court into broad areas, such as the two-point and three-point zones, and aggregated basic statistics of made shots within these regions (Jaime Sampaio & Alberto, 2006). Although these early models provided some insights that were expected to aid in player recruitment, the granularity of data was quite limited, restricting their practical application.

As the field progressed, the level of detail in the data increased with more precise annotations of shooter positions, enabling richer analyses of shooting performance (Goldsberry, 2012; Reich et al., 2006). While these advancements resulted in visually engaging graphics for fans, sports analysts and researchers continued to push beyond visualization by developing more sophisticated models that incorporated additional variables, such as the distance to defenders, remaining time on the shot clock, and various other contextual features.

In the publication by Chang et al. (2014), extensive heatmaps are generated for count statistics, multiple statistical models are fitted with the generated features and the evaluation of expected shot quality models (ESQ) is discussed. Here, a specific valuation of the decrease in Brier-loss is discussed with respect to an improvement in points per game. Several models estimate the shot quality or the FG% (Franks et al., 2015) (Skinner, 2012). In Daly-Grafstein and Bornn (2018), it is investigated how additional information about the shot (such as shooting angle, entry angle and shot depth) helps to infer FG% and stabilize this information earlier in the season,

corresponding to a smaller sample size.

In Rolland et al. (2020), the origin of 3-point shots is characterized by a physical space occupation map and time occupation model based on point-mass moving equations. They show catch-and-shoot events need more time than pull-ups for the same ESQ. Pelechrinis and Goldsberry (2021) built on this work by analyzing FG% in conjunction with player tracking data, demonstrating that the efficiency of corner three-pointers is not solely due to their proximity to the basket but also to the high frequency of assisted attempts—marking an important step in disentangling ESQ from underlying factors contributing to shot success. Sandholtz and Bornn (2020) developed a more detailed time-dependent ESQ model that incorporates dynamic transition models based on Markov Decision Processes (MDP) to investigate shot selection and simulate single episodes of the game.

Cervone et al. (2014) uses the ESQ to build the game's expected point value map. This was recently extended by (Sa-Freire et al., 2024) using a GNN to predict the immediate expected points a team can score from a specific action combined with a more granular action subdivision, by building different models for different actions e.g. passing, turnover or ESQ.

Methods

Problem Identification

In basketball, ESQ can be described in a probabilistic fashion by predicting the probability that a shot is successful with $p(y = t|x)$ where x describes the game's current state and y denotes if the shot was a success (t) or not ($\neg t$). The feature vector x can contain multiple physical quantities since the emergence of tracking data. We use a tracking data set provided by SportsVU, which includes player and ball coordinates at 25 Hz for event sequences of all 2015/16 NBA season games and annotations like shot type, shot outcome, player names, and player positions over time. As the shot time stamps of annotations are inaccurate in the original data, they are recomputed based on the acceleration of the ball and the shooter coordinates. The shot time is chosen when the ball is closest to the shooter and accelerates fast towards the hoop. Tracking data is converted from full-court coordinate space into half-court coordinate space by mirroring along the half-court line.

For each time frame, players are represented as nodes with respective features. Each node is connected with spatial edges to their k closest neighbours. Spatial edge features indicate the distance between the nodes. Per-frame spatial graphs are connected over time into single graphs by adding temporal edges between a node and its corresponding neighbours in the time axis (Figure 2 left). We use more frames closer to the time of the shot than further back in time to have both short-term and long-term information while keeping the size of the graph small. Temporal edge features indicate the time difference in the number of frames between nodes. Figure 2 displays the tracking data in its graphical representation.

To avoid the possibility that the model can calculate the shot's outcome based on the ball's motion, no frame after the shot time stamp is added. The model should learn to focus on shot selection, defensive pressure and other aspects of the game state; otherwise, we could model the shot by methods that model the physics of the basketball (Daly-Grafstein & Bornn, 2018). Predicting if a shot goes in (or not) boils down to a binary classification problem, resulting in heavy feature engineering in previous works. The GNN approach allows us to exploit the graph structure of tracking data, enabling us to extract new information from the data with similarity-search in embedding space and interpret the data with visualization of node and feature characteristics in new ways.

Graphs and the Computational Model

A graph $G = (V, E)$ is structured in nodes $V = \{v_1, \dots, v_n\}$ and edges $E = \{(x, y) | x, y \in V \wedge x \neq y\}$. Graphs can either be classified as homogeneous or heterogeneous graphs. Heterogeneous means that the edges in the graph are different, while all nodes and edges have the same type in a homogeneous graph. Our graph is heterogeneous; we denote spatial edges as E^S that model geometric dependencies between players and temporal edges as E^T , modeling the temporal evolution of the players' positions. The task is to classify the given computational graph, made shot. This results in optimizing either the Brier loss or the log loss.

We process the graph through a graph neural network with multiple spatio-temporal graph layers. These layers, help to process information and counter the effect of oversmoothing, as using a homogenous graph, results in a large neighborhood (Rusch et al., 2023). For a spatio-temporal graph layer L , each node v_i has node features $F_{i,L}^V$, each spatial edge $e_{x,y}^S$ from a node x to a node y has features $F_{x,y,L}^S$ and each temporal edge $e_{x,y}^T$ has features $F_{x,y,L}^T$. We use shared node and edge functions to update the features from one layer to the next. In our case, these are represented by neural networks N_L^V , N_L^S and N_L^T , corresponding to the node and edge features. Additionally, a per-edge attention value $a_{x,y}^{s|t}$ for spatial and temporal edges are evaluated with the help of a neural network $A^{s|t}$. This is highly related to GAT and adapted to our specific problem.

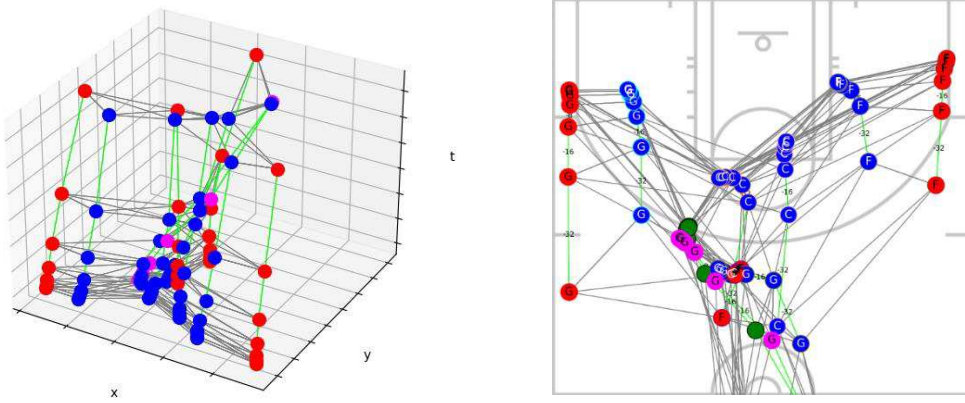


Figure 2: Left: Spatio-temporal subgraph of tracking data from multiple time frames. Spatial edges are in grey (respective on the x and y plane), and tempo-ral edges are in lime (in the time dimension). Right: Spatio-temporal graph for a game situation with seven frames before the shot. The red and blue nodes are for attackers and defenders, respectively. Green nodes are for the ball. Player position is annotated with their position (G: Guard, F: Forward, C: Center). Similar colored rings around the players indicate the players' identity. Spatial edges are grey and indicate the distance between the nodes. The shooter node has an outline in magenta. Temporal edges are green and connect nodes from neighboring frames. The numbers indicate the temporal delay in time frames.

First, we update spatial and temporal edge features by applying the respective neural networks over the concatenation of the current edge features and the connected node features. Given an edge from node x to node y , the update is performed as follows

$$F_{(x,y),L+1}^{s|t} = N_L^{s|t}(F_{x,y,L+1}^{s|t}, F_x^V, F_y^V)$$

For a node V_x the per-edge attention value is calculated for all k edges $\{(y_1, x), (y_2, x), \dots, (y_k, x)\}$ that connect nodes v_{y_i} with $i = \{1, \dots, k\}$ to node v_x based on the already updated edge features.

$$\begin{aligned}
a_{(y_i,x),L}^{s|t} &= \sigma(A_L^{s|t}(F_{x,y_i,L+1}^{s|t})) \\
&= \frac{e^{a_{(y_i,x),L}^{s|t}}}{\sum_{j=1}^k e^{a_{(j,x),L}^{s|t}}}
\end{aligned}$$

The node v_x collects all connected spatial and temporal edge features from all nodes into a spatial or temporal reduced message $m_{x,L}^{s|t}$ by applying the attention weighted sum over the respective updated edge features

$$m_{x,L}^{s|t} = \sum_{j=1}^k a_{(y_j,x),L}^{s|t} \circ F_{(y_j,x),L+1}^{s|t}$$

Finally, we obtain the updated node features of a node v_x by applying the neural network over the concatenation of node features, its spatial reduced message and its temporal reduced message

$$F_{x,L+1}^V = N_L^V([F_{x,L}^V, m_{x,L}^s, m_{x,L}^t])$$

In order to get a scalar value in the range $[0,1]$ for the probabilistic prediction, we first apply attention-weighted reduction over all the nodes and edges in the graph. This operation also uses attention functions similar to the per-edge attention calculation in the spatio-temporal layer. After iteratively applying all the spatio-temporal layers to the input graph, we obtain the final node, spatial edge and temporal edge features. The result is a single node, a single spatial edge and a single temporal edge with respective aggregated information. We then evaluate a final linear layer with a sigmoid activation function over the concatenation of the aggregated features to obtain the classification prediction.

Training and Evaluation

The training of the graph models is a graph classification task. Robberechts et al. (2021) argue that training with a Brier loss is superior when optimizing for probabilities that are further used for models that are summing over probabilities. At the same time, cross-entropy is an advantage when further multiplying the probabilities. Both losses return different optimization surfaces, leading to a superior loss for a specific problem. To investigate this behaviour with respect to calibration, we trained the same model with Brier- and cross-entropy-loss.

In basketball, models must be well-calibrated. When a model outputs a probability, this corresponds to the actual number of outcomes this event might occur. If a well-calibrated model predicts 80% success probability, meaning 80 of 100 shots lead to a field goal in the reference data. This turned out to be especially challenging in basketball regarding the distribution's tails at zero and one. We inspect the calibration of our models with the expected calibration error (ECE) and reliability diagrams (Niculescu-Mizil & Caruana, 2005). The ECE takes the weighted average over the difference between n binned accuracy and binned confidence. The binned (B) confidence (conf) and accuracy (acc) correspond to

$$\begin{aligned}
conf(B_i) &= \frac{1}{|B_i|} \sum_{n \in B_i} p_n \\
acc(B_i) &= \frac{1}{|B_i|} \sum_{n \in B_i} \mathbb{I}(y_n = \hat{y}_n)
\end{aligned}$$

and the ECE is the sum over the bins.

$$ECE(B) = \sum_{n \in B_i} \frac{|B_i|}{n} |acc(B_i) - conf(B_i)|$$

In our results, we used 30 bins to calculate the ECE, as in basketball, the quality of a shot can vary by around three percent and makes the difference between a bad, average, good and excellent shooter.

Results

In this section, we describe the optimization results of the GNN and compare them to four other classifiers, namely Logistic Regression, naive Bayes, a feed-forward neural network, and a gradient-boosted decision tree, representing the main classes of classifiers for the given data.

The dataset consists of 32272 shots of the NBA season 2015-2016 with an average hit percentage of 44.7%. In general, all tracking data for the plays is available, and the timestamp is computed as described above. The data is temporally split between regular season as training data and the playoffs for testing the different classifiers. Within the training set, we optimized the classifiers with a repeated k-fold cross validation approach (k=5 folds and n=2 repeats) (Kim, 2009) that is stratified by shot outcome. The batch size of 1024 was chosen for efficient computation on our hardware.

For the interpretability and embedding experiments, we leveraged the total dataset.

Each classifier receives numerical and categorical data as input (except the GNN):

Numerical data consist of shooter position, velocity and acceleration on the playing field in Cartesian coordinates as well as polar coordinates, where the coordinate system's centre corresponds to the bucket's position. The same data is provided for the first and second closest defenders. We provide the height and span of the arms as numerical features.

As *categorical features*, we introduce the type of the shot (jump shot, layup) and the nominal position (G, F, C) of the shooter to the classifiers. All the data with its specific types and ranges is depicted in Table 3 in the Appendix.

The GNN receives a downsampled version of the spatio-temporal tracking data as input, consisting of all players' positions, velocity, and acceleration as node features. We incorporate time difference as temporal edge features and distance between each player as spatial edge features. The KNN-algorithm builds the graph with k neighbours, and self-loops are removed. The only categorical features we give to the GNN are the nominal position and the binary feature of the shooting player, so the network knows which player is shooting the ball. This is unnecessary in theory, yet experiments showed that the "shooter" feature helps the networks convergence.

The final GNN was trained with three hidden layers, with a learning rate of $5e-4$ and an exponential learning rate decay of 0.99. We used a dropout rate of 0.2 and for every node and edge MLP we used two layers with 24 hidden features. The number of neighbours in the graph results in k=2. The GNN was trained with both brier and negative log-likelihood loss.

In Table 1, we can see the results for the optimization of the classifiers. The classifiers with feature engineering are worse than the raw graph data results processed by our GNN. This might be due to missing features or oversimplified aggregation engineering, but it highlights the advantage of our approach: feature engineering might lead to wrong assumptions or need an advanced mapping of specific features, like using templates or combinations to improve the fitting of the classification surface.

Table 1. AUC, accuracy, F1-score, log-loss, Brier Loss and ECE for logistic regression, naive Bayes, gradient boosted classifier, multi-layer perceptron, GNN and GNN trained with Brier-loss. Arrows point upwards (downwards) if a higher (lower) value represents better performance.

Model	AUC $\uparrow$	Accuracy $\uparrow$	F1 $\uparrow$	Log loss $\downarrow$	Brier loss $\downarrow$	ECE $\downarrow$
Logistic Regression	0.5914	0.5861	0.6188	0.6727	0.2399	0.0294
Naïve Bayes	0.5777	0.4699	0.6187	2.0678	0.4238	0.4887
Gradient-Boosted Classifier	0.5989	0.5895	0.6205	0.6696	0.2384	0.0278
Multi-Layer Perceptron	0.5690	0.57822	0.6187	0.7915	0.2712	0.1573
GNN - NII	0.6102	0.6069	0.6245	0.6693	0.2379	0.073
GNN - Brier	0.6174	0.6093	0.6259	0.6522	0.2287	0.0263

For the ESQ model, models must be well calibrated so that the follow-up metrics like eFG%, EPV or an opportunity value (Amour et al., 2015) can be computed with a specific expressiveness. Graph neural networks have the reputation of not being calibrated well (Minderer et al., 2021). Table 1 shows that this is not the case for our data. This might be due to the massive number of training examples and the statistical distribution of shots made and missed on the total playing field.

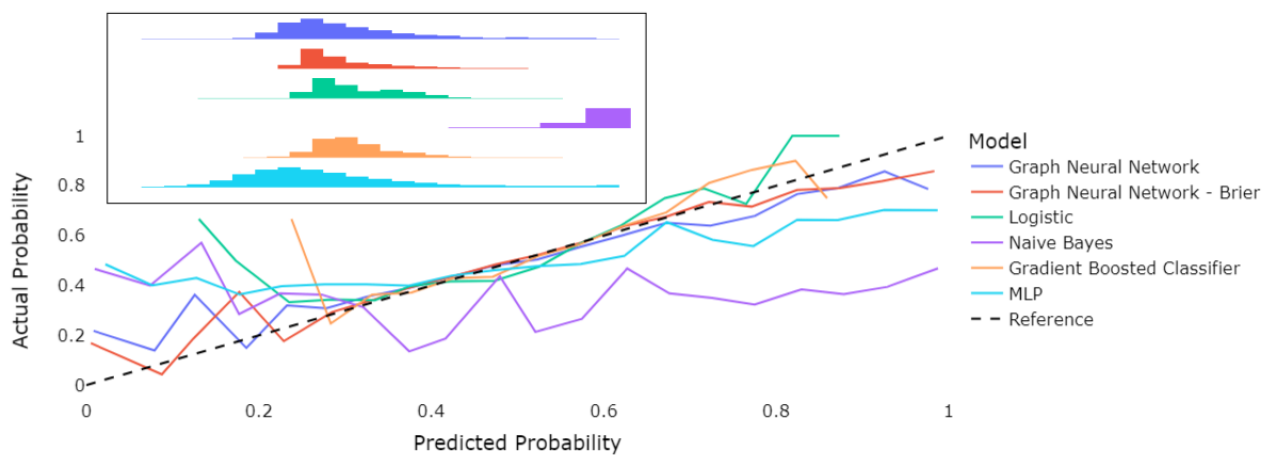


Figure 3. We can see the calibration probabilities in the main plot and the corresponding population of predicted probabilities in the top left detail box of the figure for the six differently trained models to predict the expected shot quality.

In Figure 3, we can see the reliability diagram of the respective classifiers and the probability density function of the respective mean predicted probabilities. For the reliability diagram, well-calibrated classifiers are close to the diagonal between zero and one, which means that the fractions of occurrences of a specific event are close to the event's predicted probabilities. For the gradient-boosted classifier, we can see a relatively well-calibrated graph. Nevertheless, there are no values for prediction probabilities below 0.2, and its probabilities are generally too small for shots with low probability. The same is true for shots with a high probability. For the logistic regression, almost the same applies, but roughly a little worse. The naive Bayes classifier and MLP are not well calibrated. The GNN that is trained with log loss is quite well-calibrated. However, it underestimates shots made with low probability and overestimates shots made with high probability. Finally, according to ECE and the calibration diagrams, the GNN trained with

Brier loss is the model with the best calibration. Besides that, we must be careful with extreme predictions as they tend to be worse calibrated. For further interpretation, we need to consider that the ESQ of players with high values is generally a little underestimated, and ESQ of players that have lower values are overestimated. Hence, if players take shots that result in probabilities $p < 20\%$ or $p > 80\%$, the models no longer represent the correct estimation. In general, if models are not well calibrated, methods exist for performing post-hoc calibration (Silva Filho et al., 2023).

Discussion

In this part, we will argue about the applications of graph neural networks in basketball analytics and show how they are related to other statistical measures if applicable. First, we will show how we can interpret predictions made by graph neural networks. Then we aim to identify the application of searching by example and compare it to existing or engineered approaches.

Interpretability

A common problem with applying neural networks is the explainability or interpretability of the data. In logistic regression, tree models or gradient boosting models, this is inherent in the model due to feature weights, splitting or feature importance, respectively. In neural networks it started to get investigated with LIME (Ribeiro et al., 2016), SHAP framework (Lundberg et al., 2017) or newly for graphs there are multiple models proposed (Duval & Malliaros, 2021; Huang et al., 2020; Ying et al., 2019) to explain single predictions. All those models focus on the interpretability of features and structures in the graph. The attention mechanism is a proxy for the interpretability of the edge importance (Ying et al., 2019). Using attention in our model as a proxy for the interpretability corresponds to the temporal and spatial evolution of the nodes. The proximity of the nodes is hence captured by the attention weight and can be visualized as described in Figure 6. We can see that the temporal edge attention is mainly on the previous positions of the shooters and the closest defenders' previous positions. The spatial edge attention contributes with large weights to the shooters from the closest players around him, including defence and offence. The same behaviour is expected from other classifiers (and common sense) that the closest defenders contribute to the shooter's performance.

To analyse the impact of the respective features, we used the GNNExplainer to give some interpretability to the features. We iterate over the test set of shot predictions and take the mean of the respective node feature mask with respect to the shooter node. When using all nodes for computation of the node feature mask, the values of the mask collapse at approximately the same values for different observations. As our prediction is yet made on the shooter node, we take the respective features of the shooter node to evaluate the associate graph features. This increases diversity in node feature mask values, yet the obvious happens: Node class features are less critical than positional features. Respectively, we receive much more interpretable results from the attention mechanism to give us additional information about the connectivity of the respective nodes.

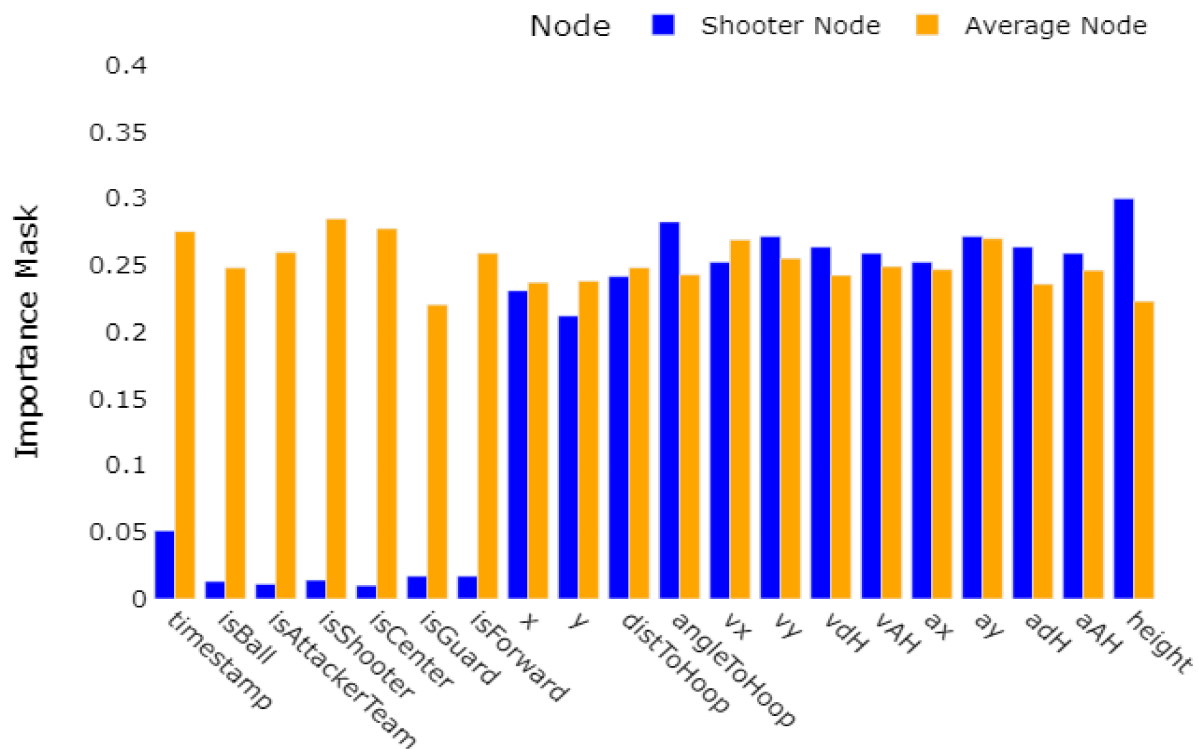


Figure 4. Normalized values of the learned feature mask by the GNNExplainer model for the shot classification task with output node shooter and mean of all embeddings (orange). The graph shows the difference in importance for an average aggregate over all nodes, and the average shooter node.

In Figure 4 we can see the node feature importance for the shot taken in Figure 6. This helps coaches to interpret and communicate specific shot outcomes first through the attention mechanism, which gives insights into the strength of the signal from different nodes and secondly through the weighting of the respective node features. Hence, storytelling is made easy with the interpretability of the model.

Embeddings

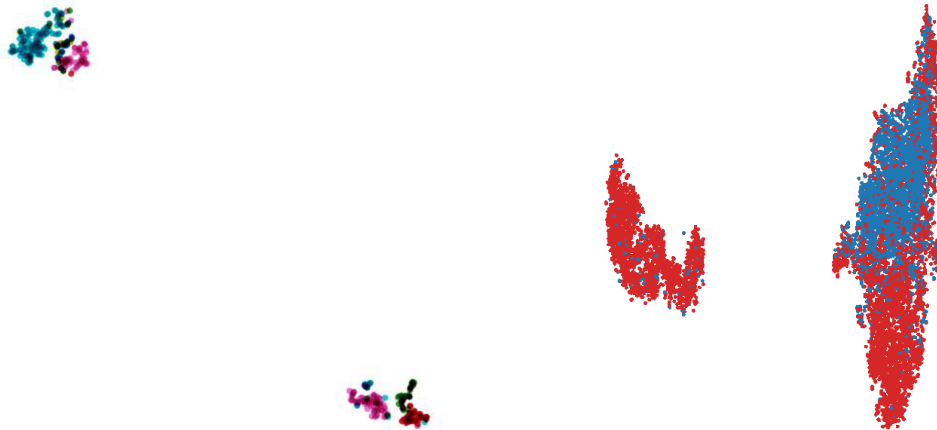


Figure 5. In the left frame, we can see the embeddings for the neural network predictions with a mean over different shooters with coloring in positions. The pure guards are colored in turquoise, the pure forwards are colored in pink, and the pure centers are colored in red. Guard-Forwards are colored in blue and the Forward-Centers are colored in green. On the right, the embeddings of every single situation are taken with coloring in 2-point (red) and 3-point shots (blue), both displayed with the TSNE-Embedding visualization.

Since the GNN only updates the features of the input graph and keeps the nodes and edges, we can investigate the features after the last layer (embeddings). Using the knowledge about the structure of the graph before passing it through the network, we can derive various kinds of embeddings by focusing on certain nodes/ edges. We compute and compare three kinds of embeddings:

- Shooter embeddings: a vector containing the average of all temporal shooter node embeddings, averaged over all shots in the dataset, which results in one embedding per player. In the following, for clarity, player embedding corresponds in the wider sense to shooter embedding.
- Attacker or defender embeddings: a vector containing the average of all temporal attacker/defender node embeddings, averaged over all shots in the dataset (attacker/defender team), which results in one embedding per team
- Situation embeddings: a vector containing the attention weighted node, spatial edge and temporal edge embedding, resulting in one embedding per shot

The respective embeddings are averaged by the equation:

$$h_{p|t} = \frac{1}{|H_{p|t}|} \sum_{i \in H_{p|t}} h_i$$

where $|H_p|$ and $|H_t|$ are the number of plays and H_p and H_t the number of sets corresponding to the respective player or team. Especially the situation and shooter embeddings are interesting. For situation embeddings, we can use nearest neighbors search to find similar plays to compare to and visualize the corresponding graph with attention and the respective feature importance (see Figure 6).

Similarity in Shooters

Player similarity analysis holds significant value in the NBA, especially for team managers when

building rosters, replacing departing players, or devising defensive strategies. Popular similarity measures, such as FiveThirtyEight’s RAPTOR score (Silver, 2023) and Regularized Adjusted Plus-Minus (RAPM), are often used to assess player contributions (Gong & Chen, 2024). These measures evaluate a player’s impact on their team’s success, either by looking at their performance over 100 possessions or by comparing their plus-minus score using three seasons of NBA data. However, these traditional metrics focus primarily on team success rather than individual shot-making patterns. Miller et al. (2014) uses non-negative matrix factorization (NMF) to compute base vectors for spatial shot selection. Our approach introduces an alternative method by comparing players based on shot-making similarities, which can offer a more nuanced understanding of their offensive play style. We validate our similarity analysis using the well-known “Messi” case study or also known as the “face validity test” (Davis et al., 2024), which illustrates the effectiveness of evaluating individual contributions beyond standard metrics, especially for offensive contributions. In Figure 5, we visualize player embeddings using the t-SNE model. The color coding in the visualization highlights player positions, revealing that guards, forwards, and centers tend to cluster together, indicating that these positions often exhibit similar shooting patterns. Nonetheless, there are instances of players classified as guards who perform more like forwards, and vice versa, as well as overlapping characteristics between forwards and centers. Table 2 provides a comparison of Stephen Curry’s nearest neighbors according to different similarity metrics. Our embeddings emphasize shot-making similarity, which tends to highlight players with a similar offensive playing style. For instance, basketball experts would generally agree that Kyle Lowry and Damian Lillard are more similar to Stephen Curry in terms of offensive skill set compared to players like LeBron James, Dwyane Wade, or Rudy Gobert. This observation suggests that utilizing seasonal or advanced statistics, combined with our embeddings, could provide a more accurate measure of player similarity. Table 2 also includes the L2-distance between players’ offensive statistics based on data from Basketball-Reference. This distance measure averages season statistics without necessarily factoring in variables like playing time or total points scored. Additionally, NMF-weight-correlation is the correlation coefficient between the decomposed player weight matrix from Miller et al. (2014). This approach focuses on comparing shot location from specific players and less on the volume. Our approach, therefore, extends beyond existing statistical methods by focusing on specific skills, and pre-shot behavior, making it more useful for identifying players with similar capabilities. A detailed calculation of “Normalized Stats” can be found in the appendix.

Table 2. Stephen Curry’s similarity measures compared. Embedding Similarity (ours), RAPTOR, RAPM, averaged season statistics and the correlation coefficient of the weight matrix of the NMF .

Rank	Embedding Similarity	RAPTOR	RAPM	Normalized Stats	NMF-weight-correlation
1	K. Lowry (0.008)	C. Paul (26)	C.Paul (4.01)	B. Beal (0.074)	D. Augustin (0.85)
2	Se. Curry (0.009)	D. Wade (22)	L. James (3.9)	K. Irving (0.074)	E. Gordon (0.80)
3	P. Mills (0.011)	L. James (20)	D. Green (3.56)	I. Thomas(0.077)	Ryan Anderson (0.80)
4	M. Williams (0.012)	K. Leonard (18)	R.Gobert (3.39)	K. Walker (0.081)	Jodie Meeks (0.79)
5	D. Lillard (0.013)	J. Harden (17)	P. Patterson (3.33)	D. Lillard (0.088)	Ersan Ilyasova (0.78)

Similarity in Plays

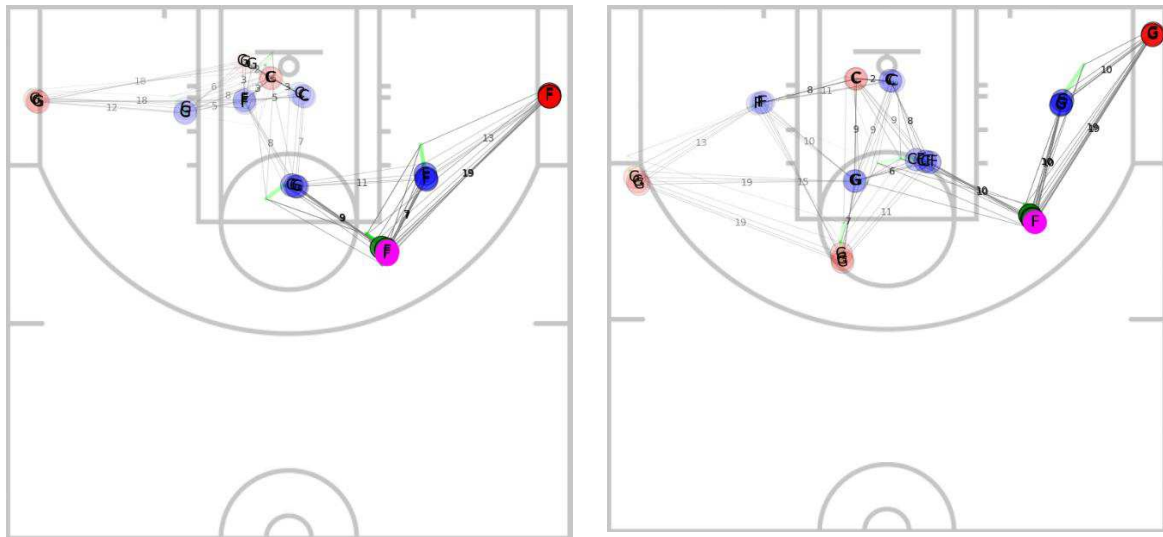


Figure 6 . Left: Shot of Serge Ibaka. Right: Shot from Blake Griffin. The attacking team is displayed in red with the shooter in magenta. The defending team is displayed in blue and the ball in dark green. Spatial edges are drawn in black/grey and temporal edges in lime. The thickness of the lines and nodes indicates the attention values, while the darker colour indicates the stronger attention of the network.

While the similarity in player shot type chart is quite important for managers to support decision making, either for role players or superstars, similarity in plays with the help of attention mechanism, might help coaches to understand specific shot types in more detail. Assuming we have two shots that have similar last seconds, this should result in similar embeddings. Hence, comparing these shots can lead to insights into how to defend specific situations or create space for a good bucket. These shots are already labelled with a lot of semantics (Action Type, Shot Zone etc.), yet the positional data gives another dimension of semantics to the shot, as labels might be slightly different but the shots have been created similarly.

In Figure 6 we compare shots of Serge Ibaka (left) and Blake Griffin (right). Those shots are closest to each other in the play embeddings with a Euclidean distance of 0.197. Serge Ibakas shot was classified as a jump shot from the left side center, while Blake Griffins attempt was classified as a center Jump shot. From tracking data we can see that the interesting part is not the shot classification of center or left side center. In both possessions, the network focuses on the closest defender as well as on the open shooter on the right, while the position of the other players is less important to the network in this situation, based on the edge and node attention of the graph.

A practical application, which comes with that, is the possibility to use the embeddings to automatically select videos for match analysis. E.g., coaches who want to create certain (promising) shot opportunities for their players, can look at how similar plays have been created against the upcoming opponent. Such data-driven detection and labelling of game situations is a valuable tool for coaches and performance analysis, as it could already be shown in other sports. Vice versa, coaches can be equipped with reports that show the most common (successful) similar plays, which can be used to adapt the defensive game plan. This could even be combined with previously suggested tools that allow coaches to simulate the behaviour of the opposing team against specific plays (Bauer & Anzer, 2021; Seidl et al., 2018).

Conclusion

In this work, we implemented a graph neural network (GNN) to predict the field goal probability of basketball shots. The GNN achieves similar values as state-of-the-art methods in this field like gradient-boosted models and even outperformed them regarding calibration. This draws a meaningful distinction, as ESQ models have to be well-calibrated to ensure proper validity for follow-up metrics like eFG% or EPV. With our approach, we differentiate between spatial and temporal connections and are able to give insights into the inner workings of the neural network via the attention mechanism and GNNExplainer. Similarity search in the embeddings of the graph can be used to identify similar players and present information to coaches and analysts in a different and interpretable way.

References

- Amour, A. D., Cervone, D., Bornn, L., & Goldsberry, K. (2015). Move or Die: How Ball Movement Creates Open Shots in the NBA. *MIT Sloan Sport Analytics Conference*. https://www.lukebornn.com/papers/damour_ssac_2015.pdf
- Anzer, G., & Bauer, P. (2021). A Goal Scoring Probability Model for Shots Based on Synchronized Positional and Event Data in Football (Soccer). *Frontiers in Sports and Active Living*, 3. <https://doi.org/10.3389/fspor.2021.624475>
- Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gulcehre, C., Song, F., Ballard, A., Gilmer, J., Dahl, G., Vaswani, A., Allen, K., Nash, C., Langston, V., & Pascanu, R. (2018). *Relational inductive biases*. <https://arxiv.org/abs/1806.01261>
- Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gulcehre, C., Song, F., Ballard, A., Gilmer, J., Dahl, G., Vaswani, A., Allen, K., Nash, C., Langston, V., ... Pascanu, R. (2018). *Relational inductive biases, deep learning, and graph networks*. <https://arxiv.org/abs/1806.01261>
- Bauer, P., & Anzer, G. (2021). Data-driven detection of counterpressing in professional football: A supervised machine learning task based on synchronized positional and event data with expert-based feature extraction. In *Data Mining and Knowledge Discovery*, 35, , publisher = (pp. 2009–2049). Springer. <https://doi.org/10.1007/s10618-021-00763-7>
- Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., & Vandergheynst, P. (2017). Geometric Deep Learning: Going beyond Euclidean data. *IEEE Signal Processing Magazine*, 34(4), 18–42. <https://doi.org/10.1109/MSP.2017.2693418>
- Cervone, D., D'Amour, A., Bornn, L., & Goldsberry, K. (2014). POINTWISE: Predicting Points and Valuing Decisions in Real Time with NBA Optical Tracking Data A new microeconomics for the NBA. *MIT Sloan Sport Analytics Conference*. https://www.lukebornn.com/papers/cervone_ssac_2014.pdf
- Chang, Y.-H., Maheswaran, R., Su, J., Kwok, S., Levy, T., Wexler, A., & Squire, K. (2014). Quantifying Shot Quality in the NBA. *MIT Sloan Sports Analytics Conference*.
- Cheema, A. (2019). *Shooting Efficiency Reimagined: Expected Effective Field Goal Percentage*. The Spax. <https://www.thespax.com/nba/shooting-efficiency-reimagined-expected-effective-field-goal-percentage/>
- Daly-Grafstein, D., & Bornn, L. (2019). Rao-Blackwellizing field goal percentage. *Journal of Quantitative Analysis in Sports*, 15(2), 85–95. <https://doi.org/10.1515/jqas-2018-0064>
- Davis, J., Bransen, L., Devos, L., Jaspers, A., Meert, W., Robberechts, P., Van Haaren, J., & Van Roy, M. (2024). Methodology and evaluation in sports analytics: challenges, approaches, and lessons learned. *Machine Learning*, 113(9), 6977–7010. <https://doi.org/10.1007/s10994-024-06585-0>
- Dick, U., & Brefeld, U. (2023). Action rate models for predicting actions in soccer. *AStA Advances in Statistical Analysis*, 107(1), 29–49. <https://doi.org/10.1007/s10182-022-00435-x>
- Duan, Y., Andrychowicz, M., Stadie, B. C., Ho, J., Schneider, J., Sutskever, I., Abbeel, P., & Zaremba, W. (2017). One-Shot Imitation Learning. *Conference on Neural Information Processing Systems*. <https://arxiv.org/pdf/1703.07326>
- Duval, A., & Malliaros, F. D. (2021). GraphSVX: Shapley Value Explanations for Graph Neural

Networks. *The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. <http://arxiv.org/abs/2104.10482>

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. (2017). Neural Message Passing for Quantum Chemistry. *International Conference on Machine Learning*. <http://arxiv.org/abs/1704.01212>

Goldsberry, K. (2012). *CourtVision: New Visual and Spatial Analytics for the NBA*. <https://www.sloansportsconference.com/research-papers/courtvision-new-visual-and-spatial-analytics-for-the-nba>

Gong, H., & Chen, S. (2024). Estimating positional plus-minus in the NBA. *Journal of Quantitative Analysis in Sports*, 20. <https://doi.org/doi:10.1515/jqas-2022-0120>

Huang, Q., Yamada, M., Tian, Y., Singh, D., Yin, D., & Chang, Y. (2020). GraphLIME: Local Interpretable Model Explanations for Graph Neural Networks. *IEEE Transactions on Knowledge and Data Engineering*, 35(7). <http://arxiv.org/abs/2001.06216>

Jaime Sampaio, M. J. S. I., & Alberto, L. (2006). Discriminant analysis of game-related statistics between basketball guards, forwards and centres in three professional leagues. *European Journal of Sport Science*, 6(3), 173–178. <https://doi.org/10.1080/17461390600676200>

Kim, J.-H. (2009). Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Comput. Stat. Data Anal.*, 53(11), 3735. <https://doi.org/10.1016/j.csda.2009.04.009>

Kipf, T. N., & Welling, M. (2016). Semi-Supervised Classification with Graph Convolutional Networks. *International Conference on Learning Representations*. <http://arxiv.org/abs/1609.02907>

Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2017). Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. *International Conference on Learning Representations*. <http://arxiv.org/abs/1707.01926>

Lundberg, S., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. *Conference on Neural Information Processing Systems*. <http://arxiv.org/abs/1705.07874>

Maaten, L. V. D., & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9, 2579–2605. <http://jmlr.org/papers/v9/vandermaaten08a.html>

Miller, A. C., & Bornn, L. (2017). Possession Sketches: Mapping NBA Strategies. *MIT Sloan Sport Analytics Conference*. <https://www.sloansportsconference.com/research-papers/possession-sketches-mapping-nba-strategies>

Miller, A., Bornn, L., Adams, R., & Goldsberry, K. (2014). Factorized Point Process Intensities: A Spatial Analysis of Professional Basketball. *International Conference on Machine Learning*. <http://arxiv.org/abs/1401.0942>

Minderer, M., Djolonga, J., Romijnders, R., Hubis, F., Zhai, X., Houlsby, N., Tran, D., & Lucic, M. (2021). Revisiting the Calibration of Modern Neural Networks. *M*. <http://arxiv.org/abs/2106.07998>

Niculescu-Mizil, A., & Caruana, R. (2005). Predicting good probabilities with supervised learning. *ICML 2005 - Proceedings of the 22nd International Conference on Machine Learning*, 625–632. <https://doi.org/10.1145/1102351.1102430>

Oughali, M. S., Bahloul, M., & Rahman, S. A. E. (2019). Analysis of NBA players and shot prediction using random forest and XGBoost Models. *2019 International Conference on Computer and Information Sciences, ICCIS, 2019*.

<https://doi.org/10.1109/ICCISCI.2019.8716412>

Pelechrinis, K., & Goldsberry, K. (2021). *The Anatomy of Corner 3s in the NBA: What makes them efficient, how are they generated and how can defenses respond?* <https://arxiv.org/pdf/2105.12785>

Rahimian, P., Kim, H., Schmid, M., & Toka, L. (2024). *Pass Receiver and Outcome Prediction in Soccer Using Temporal Graph Networks* (pp. 52–63). https://doi.org/10.1007/978-3-031-53833-9_5

Ravé, G., Granacher, U., Boullosa, D., Hackney, A. C., & Zouhal, H. (2020). How to Use Global Positioning Systems (GPS) Data to Monitor Training Load in the “Real World” of Elite Soccer. *Frontiers in Physiology*, 11. <https://doi.org/10.3389/fphys.2020.00944>

Reich, B., Hodges, J., Carlin, B., & Reich, A. (2006). A Spatial Analysis of Basketball Short Charts. *American Statistical Association*, 60(1). <https://doi.org/DOI:10.1198/000313006X90305>

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). “Why Should I Trust You?”: Explaining the Predictions of Any Classifier. *Explaining the Predictions of Any Classifier*. <http://arxiv.org/abs/1602.04938>

Robberechts, P., Van Haaren, J., & Davis, J. (2021). A Bayesian Approach to In-Game Win Probability in Soccer. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 3512–3521. <https://doi.org/10.1145/3447548.3467194>

Rolland, G., Vuillemot, R., Bos, W., & Rivière, N. (2020). Characterization of Space and Time-Dependence of 3-Point Shots in Basketball. *MIT Sloan Sport Analytics Conference*. <https://www.sloansportsconference.com/research-papers/characterization-of-space-and-time-dependence-of-3-point-shots-in-basketball>

Rusch, T. K., Chamberlain, B. P., Mahoney, M. W., Bronstein, M. M., & Mishra, S. (2022). Gradient Gating for Deep Multi-Rate Learning on Graphs. *International Conference on Learning Representations*. <https://doi.org/arXiv:2210.00513>

Sá-Freire, B. M., Barbosa, G. R. G., Gonçalves, J. L. L., Schuster, J., & Rios-Neto, H. (2025). GraphEIV: A Framework for Estimating the Expected Immediate Value in Basketball Using Graph Neural Networks. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases* (pp. 29–40). https://doi.org/10.1007/978-3-031-86692-0_3

Sandholtz, N., & Bornn, L. (2020). Markov decision processes with dynamic transition probabilities: An analysis of shooting strategies in basketball. *The Annals of Applied Statistics*, 14(3). <https://doi.org/10.1214/20-AOAS1348>

Schmid, M., Blauburger, P., & Lames, M. (2021). Simulating Defensive Trajectories in American Football for Predicting League Average Defensive Movements. *Frontiers in Sports and Active Living*, 3. <https://doi.org/10.3389/fspor.2021.669845>

Schütt, K. T., Arbabzadah, F., Chmiela, S., Müller, K. R., & Tkatchenko, A. (2017). Quantum-chemical insights from deep tensor neural networks. *Nature Communications*, 8(1), 1–8. <https://doi.org/10.1038/ncomms13890>

Seidl, T. (2018). Bhostgusters: Realtime Interactive Play Sketching with Synthesized NBA Defenses. *19th MIT Sloan Sports Analytics Conference*. <https://www.sloansportsconference.com/research-papers/bhostgusters-realtime-interactive-play-sketching-with-synthesized-nba-defenses>

- Silva Filho, T., Song, H., Perello-Nieto, M., Santos-Rodriguez, R., Kull, M., & Flach, P. (2023). Classifier calibration: a survey on how to assess and improve predicted class probabilities. *Machine Learning*, 112(9), 3211–3260. <https://doi.org/10.1007/s10994-023-06336-7>
- Silver, N. (2023). <https://fivethirtyeight.com/features/introducing-raptor-our-new-metric-for-the-modern-nba/>
- Skinner, B. (2012). The Problem of Shot Selection in Basketball. *PLoS ONE*, 7(1), e30776. <https://doi.org/10.1371/journal.pone.0030776>
- Stöckl, M., Seidl, T., Marley, D., & Power, P. (n.d.). Making Offensive Play Predictable -Using a Graph Convolutional Network to Understand Defensive Performance in Soccer. *MIT Sloan Sport Analytics Conference*. <https://www.sloansportsconference.com/research-papers/making-offensive-play-predictable-using-a-graph-convolutional-network-to-understand-defensive-performance-in-soccer>
- Tan, Q., Liu, N., & Hu, X. (2019). Deep Representation Learning for Social Network Analysis. *Frontiers in Big Data*, 2, 2. <https://doi.org/10.3389/FDATA.2019.00002/BIBTEX>
- Unke, O. T., Chmiela, S., Gastegger, M., Schütt, K. T., Sauceda, H. E., & Müller, K.-R. (2021). SpookyNet: Learning force fields with electronic degrees of freedom and nonlocal effects. *Nature Communications*, 12(1), 7273. <https://doi.org/10.1038/s41467-021-27504-0>
- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2017). Graph Attention Networks. *International Conference on Learning Representations*. <http://arxiv.org/abs/1710.10903>
- Ying, R., Bourgeois, D., You, J., Zitnik, M., & Leskovec, J. (2019). GNNExplainer: Generating Explanations for Graph Neural Networks. *Conference on Neural Information Processing Systems*. <http://arxiv.org/abs/1903.03894>
- Yu, B., Yin, H., & Zhu, Z. (2017). Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. *International Joint Conference on Artificial Intelligence*. <https://doi.org/10.24963/ijcai.2018/505>

Appendix

What is "Normalized Stats"?

For the similarity measure, we scale the following offensive statistics between zero and one and calculate the Euclidean distance between players.

- Field goal percentage: the number of made field goals, divided by all field goal attempts
- Three-point percentage: the number of three-point field goals made, divided by all three-point field goal attempts
- Two-point percentage: the number of two-point field goals made, divided by all two-point field goal attempts
- Effective Field Goal percentage: accounts for the fact that a three-point field goal is worth more than a two-point field goal¹
- Free throw percentage: Free throws made, divided by free throw attempts
- Points per Game: average points per game
- Minutes Played Per Game: average minutes played per game
- Assists per Minutes Played: total assists divided by total minutes played
- Ratio between two-point and three-point field goal attempts
- Ratio between two-point field goal attempts and resulting free throws
- Ratio between three-point field goal attempts and resulting free throws
- Offensive Rebounds per Game: offensive rebounds divided by games played

While we do not have control over RAPTOR and BR, the metric allows us to think about the production of players in an offensive scheme. This allows us, to categorize and rank our embedding similarity in context of a convolution of counting statistics as we described in Table 2.

¹ <https://www.nba.com/bucks/features/boeder-120917>

Features for Reference Models and configurations

Player	Position (x and y)	Velocity (x and y)	Acceleration (x and y)	Basket (distance and angle)	Nominal Position	Height
Shooter	x	x	x	x	x	x
Closest Defender	x	x	x	x	x	x
2.Closest Defender	x	x	x	x	x	x

Table 3 Features and number of players for each reference model.

In Table 3 we visualized the features used for the reference models. We supplied the model with 3 sets of features for the shooter, closest defender and second closest defender. Additionally, we supplied the model with the time left on the shot clock and the classification of the shot (e.g. jump shot, lay-up, etc.). We used optuna to tune the models according to accuracy with respect to the hyperparameter range depicted in table 4.

Model	Framework	Hyperparameter
MLP	sklearn	Layers: $1 < x < 6$ Neurons: $x < 400$ lr: $0.01 > x > 0.0001$
GBC	sklearn	Max depth: $3 < x < 10$, n_estimators: $10 < x < 200$
NB	sklearn	-
Log. Reg.	sklearn	-

Table 4 Hyperparameter boundary settings for the optimization of the baseline models via optuna.

(Re)calculation of the timestamp for the shot

We used two conditions for the timestamp selection to align the event data. We differentiated between the shot is in field and not a layup or dunk.

When iterating over the reversed timestamps

1. The distance between the shooter and the ball needs to be smaller than 3ft.
2. The ball needs to start accelerating towards the basket on an angle of 50 deg. relative to the shooter acceleration above 1.

We discarded shots from the dataset, where the timestamp could not be identified by the rule based alignment of the event data.