PIXLBALL: USING CONVOLUTIONAL NEURAL NETWORK TO PREDICT FOOTBALL OUTCOMES*

JONAS BRUNO†

Abstract. This capstone project develops a multi-task deep learning framework to quantify possession utility in professional football by predicting outcomes within a six-step forward window. Utilizing StatsBomb 360 data discretized into 12×8 spatial grids, I evaluate a hierarchy of architectures—ranging from a baseline 2D CNN to a motion-enriched Kinetic Context model and a spatio-temporal 3D CNN. My results demonstrate that static spatial configurations provide a robust signal for threat assessment. Performance is significantly enhanced through the integration of kinetic features (past ball positions), with the Kinetic Context model achieving a peak Balanced Accuracy of 62.3%. Furthermore, the dual-head architecture effectively estimates goal probability (xG), yielding a stable AUC-ROC score across 2D variants. Conversely, the 3D CNN implementation suffered from significant classification instability, failing to achieve meaningful predictive utility. This failure is attributed to the inherent problems of spatial sparsity within the discretized voxel volume and the high computational overhead required to optimize 3D kernels on sparse event-level data, suggesting that explicit kinetic vectors provide a more efficient proxy for temporal flow in low-resolution tactical environments.

Key words. Sports Analytics, Convolutional Neural Networks, Multi-Task Learning, Possession Utility, Spatiotemporal Modeling, Football Data Science.

AMS subject classifications. 68T05, 62M10, 91F99

1. Introduction.

Sports represent one of the most significant social, cultural and economic pillars of modern society, with the global sports industry being valued at approximately $2.65 trillion according to the Global Institute of Sports (Jess, 2024). A substantial portion of this valuation is driven by fan engagement and sports products. Under these conditions, teams that have global brand equity profit the most, as demonstrated by the fact that Lionel Messi kits are worn all across the world or by New York Yankees caps being worn by people wo have never even seen a single second of baseball. The reach of sports iconography is universal.

Estimating the intrinsic value of a sports franchise is a complex endeavor. According to Forbes, the Dallas Cowboys NFL Franchise currently holds the title of the world's most valuable sports team. While European football clubs are global titans in terms of fandom, the American sports market remains more heavily commercialized, benefiting from deeply entrenched domestic revenue streams and sophisticated media rights structures.

A primary driver for increasing the value of a franchise is athletic success. In the United States, winning enhances a team's profile and local engagement; however, the impact is somewhat moderated by structural mechanisms such as salary caps and league wide revenue sharing, which aim to maintain parity. In contrast, European football revenues are inextricably tied to on-pitch performance. The financial windfall from elite competitions, such as the UEFA Champions League, can dictate a club's trajectory—providing the capital to secure world-class talent rather than being forced to liquidate assets to maintain solvency.

In this high-stakes environment, analytics, statistics, and data science have transitioned from peripheral tools to integral components of the decision-making process. While baseball "led the charge", a paradigm shift famously chronicled in *Moneyball*,

*Thanks to Hudl Statsbomb for their Open Data access that made this project possible

†University of Lausanne, Lausanne, Switzerland (jonas.bruno@unil.ch, https://github.com/theshufflebee).

other sports were initially slower to adapt. Soccer (from here on out football), in particular, has only recently seen significant analytical progress. This delay is largely due to the fluid and continuous nature of the game; unlike the discrete, "stop-start" actions of baseball, soccer is characterized by low scoring outcomes and a high degree of stochasticity, making it notoriously difficult to disentangle individual contributions from collective dynamics, coaches decisions and pure luck.

Today the Sports Analytics Market is valued at almost 6$ billion according to Fortune Business Insights and is projected to more than double in the near future. Key drivers are technology advancement and an icreased adaption of analytical tools. At the same time, with the overturning of PASPA (Professional and Amateur Sports Protection Act) by the US Supreme court, sports-gambling has been legalized by many US states, thus creating a large market for Data Analysts working in sports.

1.1. Research Question and Objectives. The primary objective of this project is to bridge the gap between static spatial configurations, namely player positions, and dynamic possession outcomes in professional football. To this end, the project is structured around the goal of "Predictive Utility Modeling": I develop a deep learning framework capable of accurately predicting the 6-step forward outcome of a possession sequence. By utilizing a multi-task learning objective, the model seeks to classify whether a possession will be maintained, lost, or culminate in a shot, while simultaneously estimating the underlying goal probability (xG). This gives us the utility of any given snapshot of a game, as it allows to both quantify the chance to get a shot, a positive utility event, but also how dangerous the current field position is to a negative utility event (turnover).

This project specifically investigates whether the integration of contextual and kinetic features—such as ball velocity and situational metadata—significantly improves the predictions compared to baseline spatial models. The ultimate research question asks:

To what extent can convolutional neural networks, leveraging spatial grids of player position, capture the stochastic nature of football to provide a stable metric for possession utility?

1.2. Literature Review. The evolution of football analytics has transitioned from simple counting statistics to complex probabilistic frameworks that account for the fluid nature of the game. Early research primarily focused on discrete, high-impact events. An example of very early football analytics is the work of Ensum and Taylor (2004), who utilized logistic regression to estimate the probability of a shot resulting in a goal based on spatial characteristics such as distance and angle. However, as shots represent only a small fraction of total match events, subsequent research sought to value the sequences leading to these opportunities. One of the most influential frameworks in this regard is Expected Threat (xT), introduced by Singh (2018). The xT model discretizes the pitch into a grid and computes a probability matrix for ball transitions—either via passes or carries—between squares, while accounting for the probability of turnovers. By utilizing Markov chains, the model iterates backwards N times from scoring events to determine the probability of a goal occurring within the next N actions. This allows for the evaluation of non-shooting actions by measuring the "threat gained" between the start and end points of a ball movement.

Further progress was achieved by Decroos et al. (2019), who introduced the "Actions Speak Louder Than Goals" framework and the VAEP (Valuing Actions by Estimating Probabilities) metric. Their approach estimates the probability of both scoring and conceding a goal within a user-defined window of future events. By calculating the change in these probabilities after every action, any event on the pitch can be assigned a specific value. This framework was designed to be modular, allowing it to be integrated with various machine learning architectures. In contrast to event-based models, Fernandez et al. (2021) developed a framework for Expected Possession Value (EPV) that estimates the likelihood of the next goal being scored at any given moment. Utilizing high-frequency tracking data (10Hz), their approach integrates low-level spatio-temporal details with contextual features.

Overmeer et al. (2025) expanded upon this by introducing a U-Net-type convolutional neural network, which allows for the calculation of optimal pass locations to maximize possession impact. Defensive evaluation has also seen significant advancement through the work of Merhej et al. (2021), who studied the threat of passages of play preceding defensive actions to value what those actions prevented. Additionally, Stöckl et al. utilized Graph Convolutional Networks (GCNs) to represent expected receivers and pass receptions, thereby measuring defensive performance through the lens of prohibited offensive value. In summation, contemporary research increasingly relies on Machine Learning and Deep Learning to derive interpretable probabilities from high-dimensional data. While approaches vary between event-data modeling and tracking-data analysis, the common objective of mapping the stochasticity of football into actionable metrics that influence game outcomes, remains.

2. Data and Methodology. This study utilizes two primary datasets provided by StatsBomb through the `statsbombpy` Python package to model the spatial and temporal dynamics of football events.

2.1. StatsBomb Event Data. The core dataset consists of high-frequency event data, which records both administrative events and every on-ball action occurring on the pitch. We remove all administrative events and focus only on on-ball action. These events are organized into a sequential chain of actions. A typical sequence may be represented as a player receiving a ball at coordinates A, carrying the ball forward to B ($A \to B$), followed by a pass from B to a teammate at coordinates C. This chain continues through duels, dribbles, and final actions such as shots or goalkeeper saves. For each of these events, the dataset also has an indication of what team is currently in possession, which I use for the assignment of the possession outcomes.

The event-data provides granular attributes for each action, many of which aren't used in this project:

- **Event Type:** Precise identification of the action (e.g., Pass, Carry, Shot) including the actor.
- **Contextual Metadata:** Indicators of whether a player is for example under pressure.
- **Technical Details:** The specific body part used to play the ball (e.g., foot, head), coordinates, time,...

2.2. StatsBomb 360 Data. To provide spatial context, I incorporate StatsBomb 360 data, which captures the coordinates of all visible players at the timestamp of a specific event from the event data. Unlike event data, this provides a comprehensive threat map of the pitch for (almost) every on-ball action. For the project, I turn this data into a 12×8 grid, as a football pitch is usually $120m \times 80m$ and Statsbomb normalizes its coordinates to these measures. Then player counts are binned by these $10m \times 10m$ cells. Figure 1 serves as an example.

2.3. Data Integration and Enrichment. Both datasets are synchronized via a unique Event ID, allowing for direct mapping between on-ball actions and the spatial distribution of players. Information from the event dataset is used to enrich the 360-degree spatial grids. By mapping player attributes—such as pressure—onto the 360-grid, I create a multi-layered input tensor that captures both the physical positioning of the players and the specific context of the ball-carrier's action.

2.4. Evaluation Framework and Replicability. To assure replicability and temporal leakage, I use a train / test split. However this is done at the match level (i.e 80% of matches are training data, 20% of matches are test data). This helps to prevent temporal data leakage, as traditional random splits often lead to inflated performance metrics because tactical signatures or specific looks from a single match might appear in both the training and test sets. By splitting at the match level, I ensure the model is evaluated on entirely fresh tactical environments. This methodology ensures that the reported results represent true generalization rather than memorization of specific match contexts. In total we have 231 train matches and 58 test matches totaling 919'077 events, of which *Keep Possession* accounts for 635'414 observations, *Lose Possession* for 245'066 and *Shot* for 38'597. These matches are from the Women's World Cup (2023), the Women's Euros (2022 & 2025), the Men's World Cup (2022) and the Men's Euros (2020 & 2024)[1].

2.4.1. Data Limitations. Despite its granular spatial insights, the StatsBomb 360 dataset is subject to several constraints inherent to its collection methodology.

Broadcasting Context and Occlusion. Because the 360-degree coordinates are derived from broadcast video, data availability is contingent on the camera's zoom level. During close-up shots or replays, spatial positions are not recorded, resulting in temporal gaps within the dataset.

Field of View Constraints. The recorded frame is a direct reflection of the broadcasting angle. Consequently, players positioned at significant distances from the ball, most notably Center Backs and Goalkeepers, are frequently excluded from the frame even during wide-angle shots. This leads to a systematic under-representation of defensive positioning in certain phases of play.

Computational Constraints. While the 360 metadata includes information regarding the specific visible area of the pitch (the visible polygon), this feature was excluded from the current analysis. Integrating these polygonal coordinates would significantly increase the dimensionality of the input tensors, exceeding the memory and computational capacity of the Nuvolos environment utilized for this study. Especially the Voxels (3D Pixels) create significant computational issues, which led me to keeping the resolution of all inputs to 12×8 and additionally using uint8 encoding to reduce memory usage for this model.

These systemic data constraints are clearly reflected in the spatial decomposition shown in Figure 1. While the first grid precisely isolates the ball position, the second and third grids illustrate the broadcasting "blind spots," containing only five teammates and six opponents, respectively. This confirms that a significant portion of the 22 players are absent from the data due to camera zoom and occlusion. However broadcast angles are generally similar between games and therefore the model should be able to pick up on this constant absence reducing it's detrimental impact on this endeavor.

[1] The results I showed during my presentation only contained a subset of these matches, accounting for the varying results

3. Research Strategy. The primary empirical objective is to emulate and potentially improve upon the Expected Threat (xT) metric using deep neural networks (NNs). While my methodology is not directly comparable to xT, it does expand upon the idea of Threat by including both the Threat of a Shot and the Threat of Losing the ball. Further the capabilities of Convolutional Neural Networks allow me to incorporate rich spatial, situational, and temporal features that are absent from xT.

3.1. Empirical Methodology. The Expected Threat (xT) framework traditionally uses a Markov Chain approach to value actions based on how much they increase a team's probability of scoring. It identifies that a player's decision at any moment is binary: shooting or moving the ball to a better position to continue the possession. I adopt this logic as the primary motivation for my neural network architectures, but with a crucial shift in perspective. While the classic xT model is purely "location-based", valuing only the x, y coordinates of the event and therefore the grid cell these coordinates belong to, ignoring the "context", i.e. the specific positioning of all 22 players. I therefore utilize the dual-nature of xT to define the Multi-Task learning objective:

1. **Emulating Transition Probabilities:** The NN's **Event Head** predicts the probability of keeping possession, losing it, or taking a shot in the next 6 events, based on the full spatial distribution of players. It represents the player choice of continuing the possesion.
2. **Quantifying Immediate Reward:** The NN's **Goal Head** acts as an integrated Expected Goals (xG) model, emulating the shot decision of a player.

By structuring the models this way, I move from a static grid-value where the probabilities are exclusively determined by the start position (and in the case of continuing, the end position of the event) to a dynamic value that changes based on whether a defender is blocking the passing lane or a teammate is making a run. The goal of the following architectures is therefore to move away from just calculating a single fixed value for a cell but to learn the spatial "patterns" that represent threat in modern football.

The selection of a six-step lookahead window is primarily driven by the inherent temporal variance of event-level data, where the duration between recorded actions can fluctuate significantly. This choice finds its theoretical roots in the original Expected Threat (xT) framework, where Markov Chains were observed to converge effectively at the $N = 6$ threshold. A window that is too expansive risks "tactical dilution," where a team may navigate into a high-utility zone only to recycle possession into a non-threatening state before the target is reached, thereby introducing noise that explains why models often struggle with predicting final possession outcomes. Conversely, a lookahead that is too brief may suffer from "threat-blindness"; for example, a rapid counter-attack sequence might require several intermediate passes to move the ball into a shooting lane, yet a short window would erroneously label these high-threat setup actions merely as "Keep Possession". Consequently, the six-step target serves as a middle ground, forcing the convolutional neural network to move beyond immediate ball location and instead learn the underlying spatial patterns—such as defensive gaps or teammate runs—that represent latent threats likely to be exploited in the immediate future.

The six-step temporal look-ahead was implemented as follows: All events were assigned a default label of *Keep Possession*, which was subsequently overwritten as *Lose Possession* if they occurred within the terminal 6 event window of a possession sequence. Finally, any event occurring within the 6 steps window of a *Shot* was assigned the *Shot* utility label, overriding previous designations.

3.2. Neural Network Architecture. All models are designed as Multi-Task Networks, sharing a common feature backbone to predict two distinct outcomes simultaneously as defined in the previous section:

1. **Event Classification ($\mathbf{P_{outcome}}$):** The probability of the action resulting in one of three classes as defined by the 6 steps ahead outcome (*Keep Possession*, *Lose Possession*, or *Shot*).
2. **Goal Probability (xG):** The probability of the action resulting in a goal (conditional on the action being a *Shot*).

3.2.1. Input Feature Layers. All models receive spatial input data, discretized into bins (squares or pixels) of a 12×8 pitch grid. In total one input constitutes 3 layers as shown in Figure 1:

- **Layer 1: Ball Position ($\mathbf{C_B}$):** A binary layer where the cell containing the ball is set to 1, and all others are 0.
- **Layer 2: Teammate Positions (of the Team in Possesion) ($\mathbf{C_T}$):** A layer where each cell contains an integer count (0 to 11) corresponding to the number of teammates in that cell.
- **Layer 3: Opponent Positions ($\mathbf{C_O}$):** A layer where each cell contains an integer count (0 to 11) corresponding to the number of opposing players in that cell.

3.3. Model Architecture. I evaluate four distinct architectures of increasing complexity, incrementally integrating spatial, situational, and temporal features to identify the optimal configuration for utility prediction. Further structural details and specific hyperparameter configurations are provided in Table 2 and Table 3, respectively.

To address the high sparsity inherent in grid-based positional data, the models utilize a specific suite of stabilization layers designed to maintain gradient flow and prevent overfitting. The architecture employs LeakyReLU activation functions to mitigate the "dying neuron" problem, ensuring that the network continues to learn from subtle tactical signals despite the high frequency of zero-values in the input grids. Spatial dimensionality reduction is achieved through a hierarchical downsampling strategy using successive Max Pooling layers. This process summarizes player density within local regions of the pitch, effectively reducing the grid to a compact 3×2 spatial representation while preserving the most salient tactical features. To further stabilize the learning process, Batch Normalization is applied after each convolutional layer to maintain consistent activation scales across training batches, while Dropout layers provide necessary regularization to prevent the model from overfitting on routine, high-frequency possession sequences.

This hierarchical strategy forces the network to move beyond discrete player counts and instead internalize abstract tactical activations. The spatial data is processed through 32 learned filters that quantify patterns such as defensive pressure or passing lane gravity, which are then flattened into a 192-unit feature vector. This process effectively distills sparse inputs into a dense tactical score that serves as the foundation for the multi-task prediction heads, allowing for the simultaneous optimization of possession utility and scoring probability. All models have a batch size of 64 and are trained for 5 epochs. Other configurations were tested, but these yield the best results.

3.3.1. Static Spatial Models.

Model 1: Baseline CNN (`TinyCNN_MultiTask_Threat`). This model serves as the foundational spatial baseline. It processes three input channels representing the discretized pitch state (C_B, C_T, C_O) through a series of 2D convolutional layers. The primary objective of this model is to identify purely spatial relationships, such as player density and local numerical superiority.

Model 2: Context CNN (`TinyCNN_MultiTask_Context_Threat`). Building on the baseline, this architecture introduces a feature vector to process static situational features, such as event-specific flags ("under pressure", "counterpress", "dribble nutmeg"[2]). These high-level contextual features are concatenated with the flattened spatial features from the CNN block. This allows the model to weight spatial patterns dynamically; for example, a high-density area may be interpreted differently during a counter-press than during a settled defensive block.

3.3.2. Integrating the Temporal Dimension. While static models capture the state of the pitch at a specific moment, football is inherently fluid. Introducing time into a CNN framework is challenging, as CNNs usually struggle with time; however, this project explores two distinct methods to bridge the gap between spatial snapshots and temporal flow.

Model 3: Kinetic CNN (`TinyCNN_MultiTask_Context_Ball_Vector`). This model addresses the temporal dimension by explicitly providing the network with the ball's current and previous coordinates (x, y) as an auxiliary context vector, the model gains kinetic awareness. The vectors coordinates are normalized by dividing the fields length (120) and width (80) respectively. This allows the network to derive the speed and direction of play, enabling it to distinguish between a retreating defense, a stationary build-up, and a high-velocity counter-attack without the computational overhead of a full temporal sequence.

Model 4: 3D-Voxel CNN (`Tiny3DCNN_MultiTask`). The final architecture treats time as a native dimension by utilizing 3D convolutions. Instead of a single grid, the input is a "voxel" (a 3D Pixel) consisting of $T = 4$ consecutive 2D inputs, allowing kernels to extract features across both space and time simultaneously. This approach can identify complex dynamic patterns, such as a defensive line's "step" or a player's diagonal run, that are invisible to static models. The primary trade-off is the significant increase in parameter count and memory requirements.

3.4. Loss Functions and Optimization Strategy. To effectively train the multi-task architecture, I define a joint objective function that balances event classification with goal probability estimation.

3.4.1. Multi-Task Learning Objective. All four models utilize a shared-trunk architecture that branches into two distinct output heads. The total loss is calculated as the sum of the Event Classification loss and the Goal Probability loss. This joint optimization allows the model to learn shared spatial representations that are useful for both immediate event prediction and long-term threat assessment.

3.4.2. Weighted Focal Loss for Event Classification. The event classification head employs a Focal Loss to address the dominance of the Keep Possession class. By utilizing a focusing parameter of $\gamma = 2.0$, the loss function down-weights easy, routine examples and forces the model to focus on harder, misclassified instances like Shots.

To further stabilize training, we apply a square-root inverse frequency weighting

[2]A Dribble Nutmeg is when a player plays the ball through the legs of an opponent.

scheme to the class weights to address class imbalances. Importantly however the 3D Voxel Model uses a *WeightedRandomSampler*, and weights consequently are 1 for each class.

3.4.3. Binary Cross-Entropy for Goal Prediction. For the secondary task of predicting goal probability (Expected Goals), I utilize Binary Cross-Entropy with Logits. This approach is chosen for its numerical stability when generating probabilities for rare outcomes.

3.4.4. Code Implementation. All of the code is in the repo PIXLBALL. I use a structure (in detail described in the README.md file) with an src folder that contains all the relevant classes and function to run the code. The whole code is laid out in the 01_MASTER.ipynb Notebook and the 00_setup notebook for the data download. To run the full code, the main.py file is preferable as it runs the same code and generates the same results while doing a better job at clearing memory, reducing chances of a crash. I used pip freeze to create the requirements.txt so another person who runs the project has an easy time getting all packages. If all packages are loaded, it should be a one click run. The main.py also includes an option to re download the data. However by default in config.py, the FORCE_REDOWNLOAD is set to false, meaning there is only a re-download if the data doesn't already exist. As Statsbomb data is sometimes updated, to replicate these results exactly a re-download is not suggested. Additional information is available in the notebooks and the .py files as well.

Table 1: Model Metrics Comparison for a 5 Epoch run

Model	**Acc**	**Bal. Acc**	**Rec. Keep**	**Rec. Loss**	**Rec. Shot**	**Goal AUC**
Baseline	**0.557**	0.613	**0.544**	0.563	0.732	0.620
Context	0.499	0.610	0.423	**0.664**	0.744	0.626
Kinetic	0.541	**0.623**	0.515	0.573	**0.780**	**0.629**
3D-Voxel	0.267	0.338	0.201	0.428	0.384	0.457

4. Results. All models are run for 5 Epochs and results are reported in Table 1 while confusion matrices are in Appendix A. We find that there is a constant improvement of all metrics when additional context information is added, except for the 3D-Voxel CNN whose balanced accuracy is at 33% and isn't better than random prediction.

4.1. Baseline CNN. The baseline CNN model achieved an Overall Accuracy of **55.7%** and a Balanced Accuracy of **61.3%**. Given the inherent complexity and stochastic nature of football events, these results demonstrate that static spatial configurations—captured via the discretized grid-pitch layers—provide a robust signal for predicting outcomes within a six-step forward window.

The Event Confusion Matrix (Figure 2a) reveals that while the model effectively distinguishes *Keep Possession* from *Shot* events, it faces challenges in differentiating between *Keep* and *Lose Possession*. It has the highest Overall Accuracy as it tends to favor the majority class (*Keep*), as evidenced by its high *Keep Recall* of 54.4%. Conversely, while the model identifies *Lose Possession* with 56.3% recall, it frequently misclassified these as the other two categories. Most notably, the model maintains a high recall for the *Shot* class at **73.2%**.

These patterns suggest that the Baseline architecture has effectively mapped the pitch into "zones of risk." It recognizes that play in the defensive third constitutes "Secure Possession," whereas entry into the final third increases the statistical probability of both a turnover and a shot. The high recall for shots confirms that the model has successfully identified the "hot zone" surrounding the goal, often defaulting to a shot prediction whenever the ball enters high-utility central areas of the penalty box, independent of the specific tactical sequence.

Furthermore, the model's performance in predicting *Lose Possession* identifies what can be termed a "Goldilocks zone." This area is characterized by a high threat of turnover despite a lack of immediate shooting opportunities, representing a high-risk transition state. However, the frequent confusion with both other classes suggests the model overestimates the spatial boundaries of this zone, occasionally mislabeling labeling stable possession or shooting threats.

Regarding the secondary task, the baseline model achieved a Goal AUC-ROC of 0.620. This indicates that the spatial layers alone are effective at distinguishing high-threat from low-threat scenarios. In the context of football analytics, this is a critical distinction; many "dangerous" positions do not culminate in a goal due to stochastic external factors—such as technical execution errors or defensive interventions—which the model correctly identifies as high-probability goal opportunities regardless of the final outcome.

4.2. Contextual Models: Situational and Kinetic. Interestingly the contextual architectures hasn't did not improve much upon the spatial baseline by incorporating metadata. The Context CNN decreased the Balanced Accuracy slightly to 61.0% while *Shot* recall increased to 74.4%, compared to the baseline model. Notably, the *Lose Possession* recall increased to 66.4% while the *Keep Possession* recall decreased to 42.3%. This shift suggests that the situational metadata helped the model successfully reclassify "risky" plays that the baseline model had previously over-labeled as *Keep Possession* although, it leads to the model prefering to *Lose Possession* as shown by the confusion matrix Figure 3a reducing it's accuracy.

The Kinetic CNN emerged as the superior architecture, achieving the highest Balanced Accuracy of all tested models with (62.3%) and an Overall Accuracy of (54.1%). It also has the highest *Shot* recall of 78.0%. By integrating ball velocity vectors, the model gains a fundamental understanding of trajectory and momentum. This physical context allows it to better differentiate between a controlled carry and erratic ball movement, facilitating the identification of secure possessions as well as high-velocity transitions that are likely to end in a turnover, leading to a more balanced recall of both *Keep Possession* and *Lose Possession* .

The confusion matrices for these models (Figure 3a and Figure 4a) exhibit similar structural improvements over the baseline. This suggests that both situational metadata and kinetic vectors primarily enhance the model's ability to resolve the ambiguity between *Keep* and *Lose Possession*.

Finally, both contextual models achieved a higher goal prediction AUC than the baseline, though the margin of improvement was narrow (Kinetic AUC: 0.631). This suggests that while situational and kinetic context significantly sharpens the classification of possession *Outcomes*, the underlying probability of a *Goal* remains heavily dictated by the spatial geometry already captured in the baseline grid layers.

Furthermore, the performance of these models demonstrates that the systematic absence of players—a result of the "broadcast view" constraints in the StatsBomb 360 data—can be partially mitigated. The neural network learns to infer tactical value from the visible "tactical clusters," effectively treating the missing defensive periphery as a constant feature rather than a source of prohibitive noise.

4.3. The 3D-Voxel CNN. The results for the **3D-Voxel CNN** were the least effective across all tested architectures, yielding a Balanced Accuracy of 33.8% and a Shot recall of 38.4%. This performance level is effectively equivalent to a random classifier in a three-class problem ($1/k \approx 33.3\%$). While the implementation of a *WeightedRandomSampler* successfully prevented the architecture from collapsing into a single majority-class prediction, it could not overcome the model's inability to extract meaningful spatiotemporal features. This stagnation persisted despite extensive hyperparameter tuning, weight adjustments, and architectural modifications. Furthermore, the model struggled significantly with the goal prediction task, recording a Goal AUC-ROC of 45.7%, which is slightly below the threshold of a non-informative coin-flip.

The primary reason for this failure stems from the inherent challenges of 3D convolutional kernels when applied to sparse tactical data. 3D-CNNs generally require high-density data environments (such as high-resolution video) to learn motion gradients effectively. In the context of StatsBomb 360 grids, the data is exceptionally sparse: out of 288 total pixels per event-layer, typically fewer than 23 contain non-zero values. Stacking these sparse matrices into a temporal volume did not provide sufficient signal for the 3D kernels to identify "motion" or "intent" amidst the noise of the empty grid space. While a more comprehensive dataset—accounting for all 22 players at all times—might marginally improve feature density, the fundamental limitation remains the high ratio of zero-values to tactical signals.

The failure of this architecture highlights the efficiency of the **Kinetic CNN**; rather than forcing the model to infer motion from raw volumetric sequences, the Kinetic approach explicitly provides the temporal change as an engineered vector. This suggests that for sparse event-based data, encoding the "time aspect" through late-fusion kinetic features is a far more robust strategy than volumetric 3D modeling.

5. Conclusion. This capstone project demonstrates that spatial geometry, captured through 12×8 grid layers, provides a robust foundation for predicting multi-step tactical outcomes in football. By evaluating a hierarchy of models, I identified a clear performance plateau for raw volumetric architectures versus the efficiency of engineered kinetic features. The Kinetic Context model emerged as the best model with both the highest balanced accuracy and higher Goal AUC-ROC Score.

In contrast, the 3D-Voxel CNN performed no better than a random classifier, proving that for sparse event-based data, 3D kernels are highly inefficient. Explicitly providing the model with motion vectors (Kinetic features) is far more effective than forcing the network to infer temporal dynamics from sparse voxel volumes.

Furthermore, the dual-head architecture confirmed that possession utility and goal probability (xG) can be modeled simultaneously. The stable Goal AUC-ROC indicates that while situational context sharpens the prediction of possession outcomes, the probability of a goal is largely dictated by the spatial configurations already present in the baseline layers.

To improve upon these results, future efforts should move beyond grid-based representations. Graph Neural Networks (GNNs) offer a promising alternative by treating players as nodes, which could handle the sparsity issues that crippled the 3D CNN. Additionally, integrating player-specific metadata or high-resolution tracking data—where available—could refine the model's understanding of individual technical quality, potentially breaking the ceiling established in this project.

6. Bibliography.

References.

Tom Decroos, Lotte Bransen, Jan Van Haaren, and Jesse Davis. Actions Speak Louder Than Goals: Valuing Player Actions in Soccer. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1851–1861, July 2019. doi: 10.1145/3292500.3330758.

Jake Ensum and Samuel Taylor. Estimating the probability of a shot resulting in a goal: The effects of distance, angle and space. *Int. J. Soccer Sci.*, 2, January 2004.

Javier Fernandez, Luke Bornn, and Daniel Cervone. A framework for the fine-grained evaluation of the instantaneous expected value of soccer possessions. *Machine Learning*, 110(6):1389–1427, June 2021. ISSN 0885-6125, 1573-0565. doi: 10.1007/s10994-021-05989-6.

Middle East Forbes. Forbes List: The World's 50 Most Valuable Sports Teams 2024. https://www.forbesmiddleeast.com/lifestyle/sports/the-worlds-50-most-valuable-sports-teams-2024.

Fortune Business Insights. Sports Analytics Market Size, Share, Global Growth Report, 2034. https://www.fortunebusinessinsights.com/sports-analytics-market-102217.

Jess. The true size of the global sports industry, December 2024.

Charbel Merhej, Ryan Beal, Sarvapali Ramchurn, and Tim Matthews. What Happened Next? Using Deep Learning to Value Defensive Actions in Football Event-Data. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3394–3403, August 2021. doi: 10.1145/3447548.3467090.

Thijs Overmeer, Tim Janssen, and Wim P. M. Nuijten. Revisiting Expected Possession Value in Football: Introducing a Benchmark, U-Net Architecture, and Reward and Risk for Passes, February 2025.

Karun Singh. Introducing Expected Threat (xT), December 2018.

Michael Stöckl, Thomas Seidl, Daniel Marley, and Paul Power. Making Offensive Play Predictable - Using a Graph Convolutional Network to Understand Defensive Performance in Soccer.

6.1. Additional Resources Used. I used ChatGPT, GitHub Co-Pilot and Gemini at various points in time to help me with both writing the report and coding.

Appendix A. Tables and Figures.

Table 2: Comparative Model Architectures

Feature	**2D CNN Variants**	**3D Voxel CNN**
Input Shape	$(3, 12, 8)$	$(3, 4, 12, 8)$
Conv Block 1	16 filters, 3×3, pad 1	16 filters, $3 \times 3 \times 3$, pad 1
Normalization	BatchNorm2d	BatchNorm3d
Pooling 1	MaxPool2d (2×2)	MaxPool3d ($2 \times 2 \times 2$)
Conv Block 2	32 filters, 3×3, pad 1	32 filters, $3 \times 3 \times 3$, pad 1
Pooling 2	MaxPool2d (2×2)	MaxPool3d ($2 \times 2 \times 2$)
Flatten Size	192 units	192 units
Shared FC	128 units	128 units
Event Head	Linear ($128 \rightarrow 3$)	Linear ($128 \rightarrow 3$)
Goal Head	Linear ($128 \rightarrow 1$)	Linear ($128 \rightarrow 1$)

Table 3: Hyperparameter Specifications across Model Architectures

Parameter	**2D Baseline**	**Situational / Kinetic**	**3D Voxel**
Batch Size	64	64	64
Epochs	5	5	5
Optimizer	Adam	Adam	Adam
Learning Rate	1×10^{-3}	1×10^{-3}	1×10^{-4}
Event Weights $[W_0, W_1, W_2]$	$[0.45, 0.74, 1.8]$	$[0.45, 0.74, 1.8]$	$[1.0, 1.0, 1.0]^*$
Goal Pos Weight	3.0	3.0	3.0
Activation	LeakyReLU (0.1)	LeakyReLU (0.1)	LeakyReLU (0.1)
Dropout	0.3	0.3	0.3
Initialisation	Default	Default	Kaiming (He)
Global Seed	42	42	42

*The 3D Voxel model utilizes a WeightedRandomSampler for data-level balancing in lieu of loss-weighting, while the 2D models use inverse frequency weights.

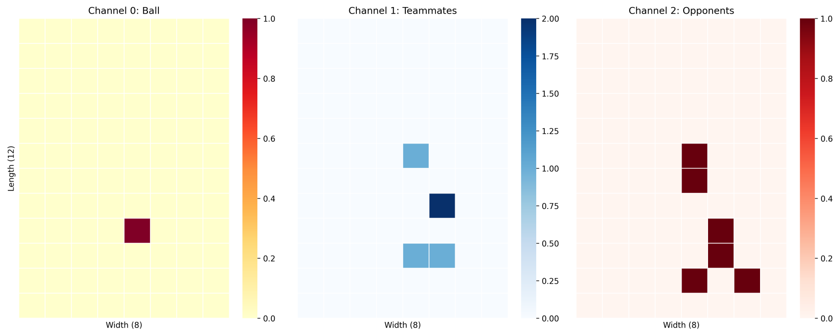
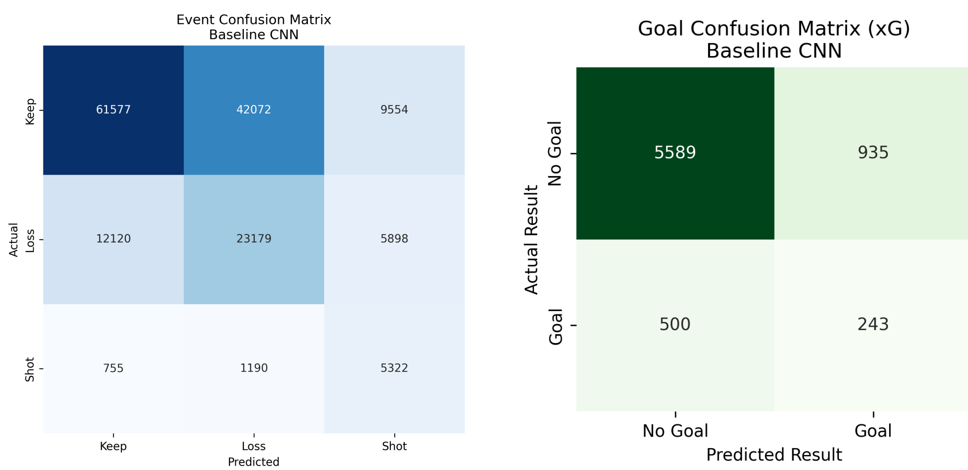


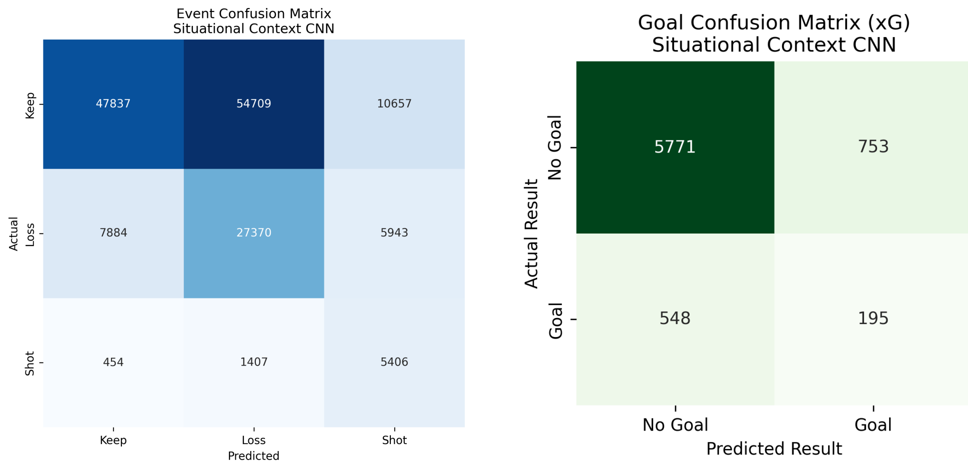
Fig. 1: Decomposed 2D Spatial Input: The three 12×8 channels representing Ball location (Left), Teammate positioning (Center), and Opponent positioning (Right).



(a) Event Outcome Classification

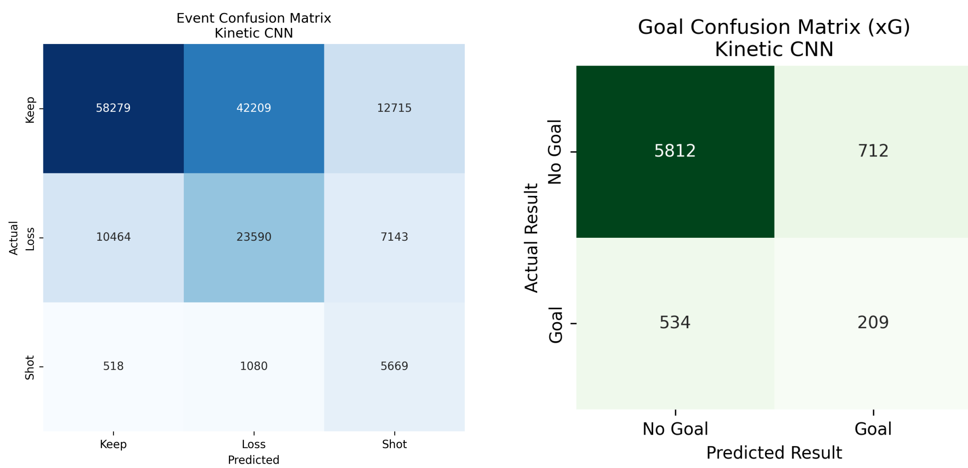
(b) Goal Probability (xG)

Fig. 2: Baseline CNN Confusion Matrices.



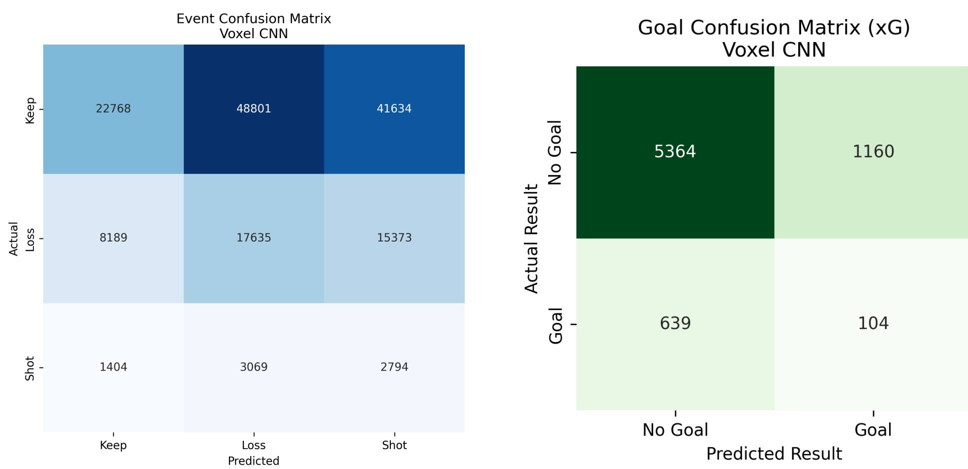
(a) Event Outcome Classification (b) Goal Probability (xG)

Fig. 3: Context CNN Confusion Matrices.



(a) Event Outcome Classification (b) Goal Probability (xG)

Fig. 4: Kinetic CNN Confusion Matrices.



(a) Event Outcome Classification

(b) Goal Probability (xG)

Fig. 5: 3D Voxel CNN Confusion Matrices.