Grids to Graphs: GNNs improve NFL defensive coverage classification over CNNs

Abstract:

This paper investigates the application of predictive neural network models to classify NFL defensive coverages using player tracking data. I propose a novel solution using a graph neural network (GNN) as an alternative to the conventional use of convolutional neural networks (CNNs). Both models were compared based on their test accuracy, F1 score, and AUC. While the CNN reported a greater test accuracy (84.3%) than the GNN (83.7%), this was primarily due to its bias towards zone coverage, the majority class. In contrast, the GNN yielded more balanced predictions with a greater F1 score (0.737 vs. 0.730) and AUC (0.920 vs. 0.904) than the CNN. The GNN was also more effective than the CNN when classifying less common coverage scheme variations. These findings suggest that GNNs are better suited for classifying NFL defensive coverages. This is because their unique architecture represents information as graphs, helping them better understand player-player interactions.

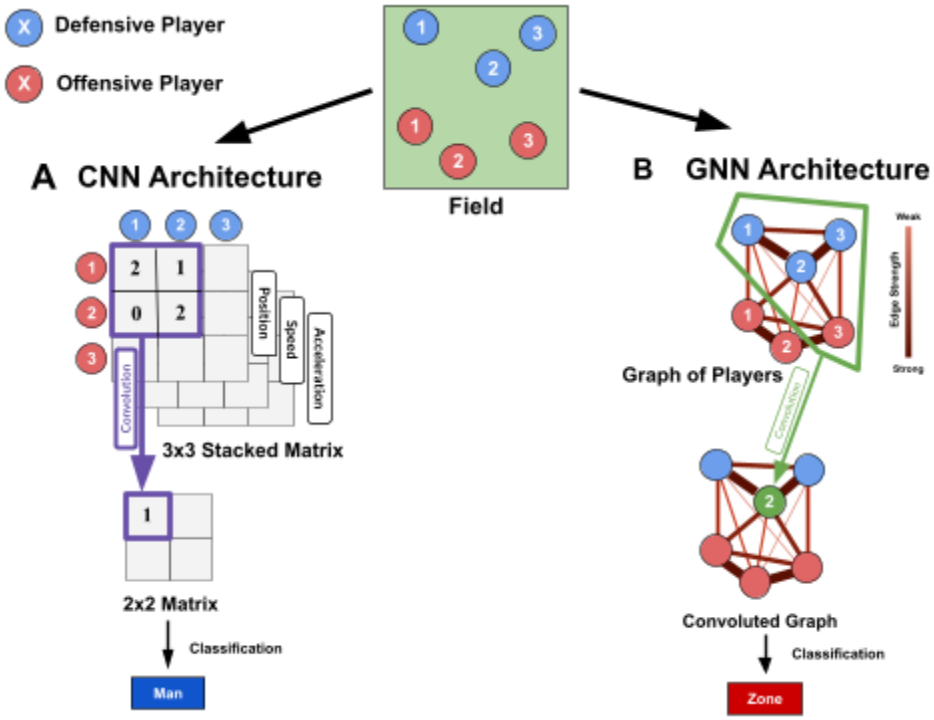
Introduction:

The National Football League (NFL) has long focused its analysis and changes towards offense to make the viewing experience more engaging. Per the NFL's Competition Committee, "if someone wants to accuse the National Football League of promoting offense to make the game more exciting, [the committee believes the league should plead guilty]" (NFL Football Operations, 2025). Hence, the recent rise in machine learning (ML)-powered football analytics has predominantly focused on offensive playcalling and performance (McManus, 2025; Goyal, 2020; Walsh, 2021; Fernandes et al., 2019; Taylor, 2020). On the other hand, considerably less analysis has been performed regarding the defensive end of the floor.

On defense, the coaching staff is responsible for positioning their players in specific formations, known as coverage schemes. Defensive coverage schemes determine the responsibilities of each player on the defensive team. Coverage schemes are categorized into two broad types: man coverage and zone coverage. In man coverage, each defender is responsible for a specific offensive player throughout the play. In contrast, zone coverage requires defensive players to defend any players that enter a particular area on the field known as their "zone" (Kingston, 2023). Both coverage schemes are employed for vastly different use cases, with each having its own benefits. While zone coverage often minimizes big plays, man coverage leads to a tighter defense, placing greater pressure on the opposing quarterback to make an accurate throw (Capital QB's, 2023).

Determining which coverage scheme the defensive team will use helps the coaching staff of the offensive team choose plays to increase their chances of scoring. As a result, the coaches of the defensive team often disguise their coverage scheme by making their players rotate following the snap—the start of a play in football. This can make play calling difficult for even the most skilled coaches. Hence, having a method of classifying coverage schemes pre-snap would enable the offense to anticipate and plan against the defensive strategies used by the opposing team.

One way of classifying defensive coverage is by using predictive models. Currently, the most successful method of classifying defensive coverage schemes is convolutional neural networks (CNNs) (Song et al., 2023). CNNs are a form of artificial neural network that process data structured in grid-like matrices. They use convolutional layers (which can be thought of as a sliding window) to capture local patterns. Throughout the training process, the weights of the window are learned (Figure 1). This process of convolution is why CNNs are often used when processing images. Images have strong spatial locality (pixels within a proximity are highly correlated), meaning the ability to capture local patterns effectively allows CNNs to learn the complex structures of objects within images without becoming overwhelmed by the large quantity of pixels (Purwono et al., 2022).



[Figure 1: Panel A shows diagrams explaining the architecture of the CNN. The football field (green square) has three offensive players (blue) and three defensive players (red). The first

illustration in panel A illustrates how the field is represented as a stacked matrix, with the values representing the distinct pairwise relative features. The purple square is then convoluted into one number in the following illustration before ultimately becoming a binary output. Panel B presents an illustration of the players in the form of a graph. The lines between the circles (nodes) represent the connections between players (edges). The thickness and color of the edges represent the strength of these connections. Similar to panel A, the values of the graph are convoluted into a new graph. Ultimately, the GNN classifies into a binary output.

Modern methods of applying CNNs to coverage scheme classification have found success in using relative data in conjunction with static data. Song et al. (2023) demonstrate that relying solely on static features significantly limits the predictive capabilities of a CNN. However, by conducting feature engineering to construct a matrix of pairwise relative features between each member of the offensive team and the players on the defensive team, they were able to model complex player interactions, thereby maintaining greater spatial and relational context (Figure 1) (Song et al., 2023).

While Song et al. (2023) found success through the use of relative features, it must be noted that they increase the dimensionality of the data significantly due to the many static values that are repeated throughout the matrix. This introduces various concerns, including greater computational demands, a high risk of overfitting, and an unintentional emphasis placed on static features, as they are constantly repeated within each pairwise comparison. Additionally, CNNs likely struggle to anticipate shifts that could occur to players' positions throughout a play due to their inability to understand how each player relates to the others on the field.

In contrast, the use of a graph neural network (GNN) should theoretically be more effective due to its unique architecture. GNNs operate on graph-structured data, describing data through nodes, edges, and a global context. The graph is passed through a differentiable function, such as a multilayer perceptron (MLP), to process the data. In the case of binary classification, such as predicting coverage schemes, the prediction target is the global context. As such, the information within the nodes and edges must be passed through a pooling layer, creating a concatenated matrix that can be used for classification (Sanchez-Lengeling et al., 2021). This process of pooling information from the nodes and edges to make predictions about the global context makes GNNs particularly robust, as it allows the GNN to learn from the intricate relationships between players on the field. Hence, the use of a GNN should enhance the ability to classify defensive coverages.

Due to the GNN's network structure, it can capture the relational context between players, making it better suited for modeling NFL plays. As CNN's interpretation of rigid, grid-like

matrices relies heavily on spatial adjacencies, it is unable to effectively capture relational context when modeling the alignment of players on the field. Thus, the GNN should produce superior results when classifying coverage schemes compared to the CNN.

Previous works, including those of Song et al. (2023), have primarily focused on optimizing the effectiveness of the CNN model by incorporating methods such as temporal learning to maximize accuracy. However, this paper will instead focus on comparing the performance of CNN models to that of GNN models in an attempt to develop a novel classification method for NFL coverage schemes. Consequently, this paper discusses the hypothesis that a GNN is more effective at NFL defensive coverage classification than a CNN.

Methods:

In football, man and zone are the two primary labels used to classify defensive coverages. Therefore, differentiating between man and zone coverages is both relevant and sufficient to identify NFL defensive coverages. As there are two primary outputs, this paper will view coverage scheme classification as a binary classification problem between man and zone coverage. The study will be performed on the 2025 Big Data Bowl, a public dataset released annually by the NFL. The dataset includes tracking data for all 22 players on the field (11 offense and 11 defense) during passing plays, along with detailed play-level descriptions. Furthermore, the dataset includes contextual information for each game as well as player-specific attributes such as build and composition (Lopez et al., 2024). The data is drawn from the first nine weeks of the 2022-23 NFL season and includes 3,023 plays.

Initially, the data is pre-processed, with any plays not labeled as either man or zone being filtered out. These plays are labeled as “other” in the dataset and constitute plays where either the line of scrimmage is within the endzone, meaning traditional forms of pass coverage are not employed, or during a quarterback kneel, which is a special form of play that occurs when the knee of a quarterback touches the ground directly after the snap and thereby immediately ends the play for strategic reasons. Subsequently, the dataset is filtered such that only the 31st frame of each play remains. The data is recorded at 10 frames per second, meaning the 31st frame represents three seconds into the play. As no snaps occur before the 31st frame, it ensures that the data is pre-snap and hence applies to offenses scheming around the defensive coverage employed. Furthermore, given that the frame is three seconds into the play, it is presumed that most players are already in their coverage positions. Though ideally, a model should use all pre-snap frames, a singular frame was chosen due to computational limitations.

CNN (normalized)	GNN (normalized)
Distance from the line of scrimmage	Distance from the line of scrimmage
Position on the y-axis	Position on the y-axis
Speed in the x-direction	Speed in the x-direction
Speed in the y-direction	Speed in the y-direction
Speed	Speed
Acceleration in the x-direction	Acceleration in the x-direction
Acceleration in the y-direction	Acceleration in the y-direction
Acceleration	Acceleration
Orientation	Orientation
Relative speed in the x-direction	Nearest opponent distance
Relative speed in the y-direction	Distance to team centroid
Relative acceleration in the x-direction	Position (represented as 10 distinct binary channels representing the quarterback, running back, wide receiver, tight end, offensive lineman, defensive lineman, linebacker, cornerback, safety, and nickel)
Relative acceleration in the y-direction	
Relative position in the x-direction	
Relative position in the y-direction	

[Table 1: The list of all the features used when training the CNN and GNN models. All listed features were either taken directly from or calculated with data found on the 2025 Big Data Bowl dataset and were all normalized for stability.

The decision to provide speed along with the x and y components of speed as separate features was made intentionally. CNNs and GNNs are both linear models and are hence unable to derive the speed of a player based on the directional components. By providing features such as speed and acceleration in both formats, the model can choose the

representation it finds most effective, ensuring flexibility between scalar and directional values, and can therefore identify new patterns to better classify coverage schemes.

The CNN was built following the architecture described in Song et al. (2023). The model was trained on a 15x11x11 stacked matrix—with 15 features, 11 offensive players, and 11 defensive players (Table 1). The data were then reshaped into batches of 64 plays to enhance the training time and generalization of the model (Ofeidis et al., 2024). The architecture of the CNN utilizes two convolutional layers, designed to extract high-level representations while preserving the spatial structure of the matrix. The first layer consists of three successive 1x1 convolutional layers. The use of a unit kernel enables the combination of all 15 feature values into a single value without blurring or merging with information from other cells. Hence, the output maintains its 11x11 spatial identity and arrangement while reducing the computational load (Ajit et al., 2020). Subsequently, a dropout layer is introduced to drop 30% of the data, thereby minimizing overfitting and regularizing the data. Following the dropout, the features are processed through a pooling layer that consists of average and max pooling. The purpose of implementing the pooling layer is to down-sample the feature maps following the convolutional layers to reduce computational cost and control overfitting (Gholamalinezhad & Khosravi, 2020). Specifically, the pooling layer collapses the dimension of offensive players, transforming the data into 128-d vectors for each defensive player. Average pooling captures the overall trend, while max pooling identifies the strongest signals. A weighted sum of both pooling values is taken (Equation 1).

$$x = (mean \times 0.7) + (max \times 0.3) \qquad [1]$$

Following the pooling layer, the features are input into another convolutional layer consisting of three 1x1 convolutional layers, each followed by batch normalization to normalize along the channel dimension. Once more, the unit kernel allows for linear transformations of the channels, increasing the channels to 160 before ultimately outputting 96 channels. After the second convolutional layer, the data is transformed through a second pooling layer that collapses the defensive-player axis while implementing the same weighted sum of average and max pooling. Hence, the output is a 96-d vector describing each batch. Finally, the data is transformed through a set of dense layers. The first dense layer employs a ReLU activation that maps the 96 features to 96 output dimensions, thereby introducing non-linearity for more complex feature representations. Subsequently, batch normalization is applied to ensure a standard distribution of the activations throughout the batch. Thereafter, a second dense layer is employed, expanding the feature space from 96 to 256 dimensions, thereby utilizing a higher-dimensional space to capture more complex patterns and linearly separable relationships. After another batch normalization is applied to enhance stability and convergence. Subsequently, layer

normalization is applied, which stabilizes the dynamics of the hidden state and improves training time (Ba et al., 2016). Finally, the last dense layer reduces the 256-dimensional normalized vector to two outputs, being the two coverage schemes, and applies a softmax activation to transform the logits into a probability distribution between the coverage schemes. This serves as the classifier, matching each sample to a class based on the learned feature representation from the previous stages of the model. Ultimately, the class with the greater probability is output as the model's prediction of the class (Figure 2).

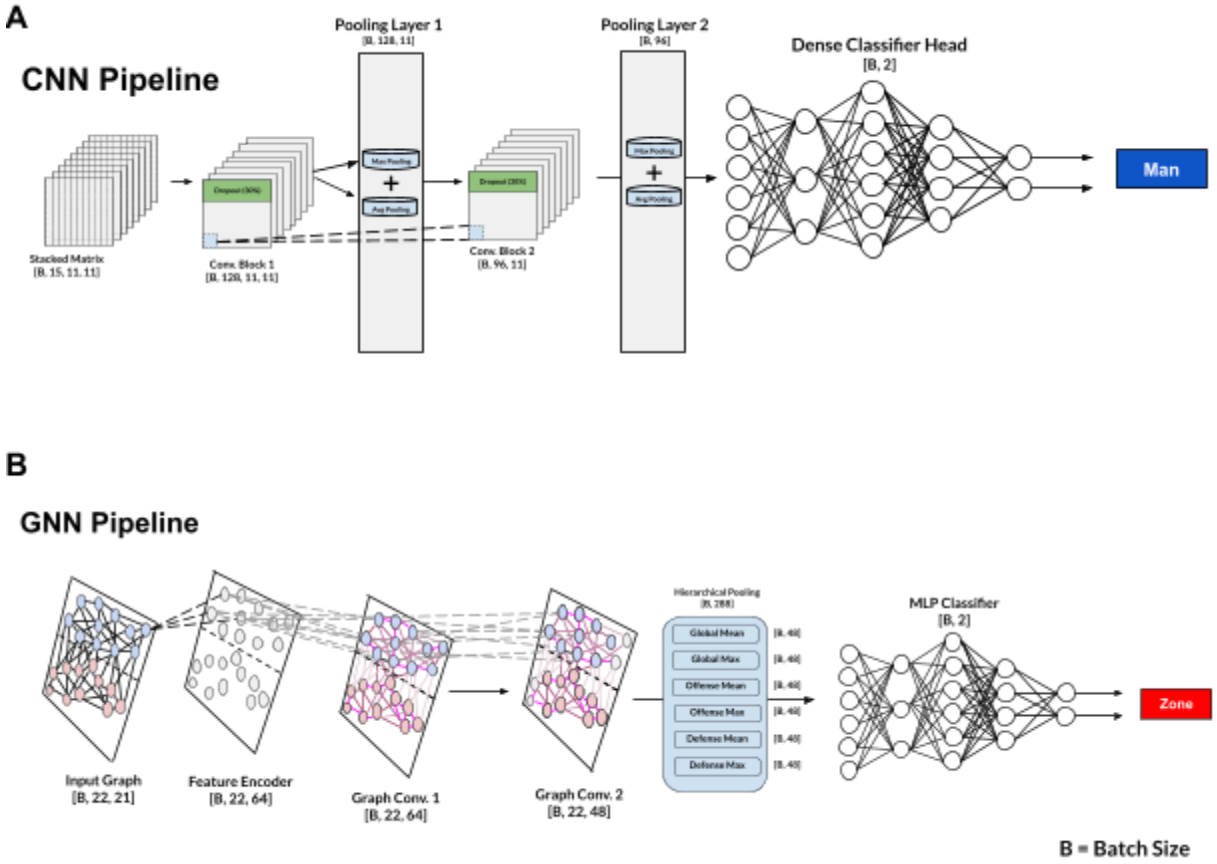
The GNN was trained on a graph consisting of 22 nodes (11 offensive players and 11 defensive players). Each of these nodes contains a 21-dimensional feature vector. To construct the graph, edges are defined by both Euclidean distance and team membership. Edges are placed between all players within 12 yards of each other, with weights between players on the same team having a multiplier. The value of 12 yards between players and the multiplier for same-team allegiance were determined empirically to optimize the model's accuracy. The graph is then transformed by an encoder that maps the 21-dimensional feature vectors onto a 64-dimensional space. This layer uses L2 regularization with a regularization parameter of 10^{-4}. Then, the data is stabilized through batch normalization, before introducing non-linearity through a ReLU activation, and employing a dropout layer to minimize overfitting.

The projected features are then processed through two convolutional layers. Node representations, the feature vectors for each player, are then updated through weighted message passing, which gathers aggregate features from all adjacent nodes (weighted by the strength of that edge) and performs a linear transformation followed by a ReLU activation. The first graph layer outputs 64 channels, followed by batch normalization and a dropout layer. In comparison, the second outputs 48 channels and is also followed by batch normalization and a dropout layer.

After the second convolutional layer, the node embeddings, which are 22 x 48 in size, are transformed through a pooling layer. The pooling layer first performs average and max pooling on the 22 players in the plays, followed by applying the same operations to defensive and offensive subgraphs separately. These processes yield six distinct 48-dimensional pooled vectors, which are concatenated into a 288-dimensional representation of the play. This 288-dimensional graph vector is then passed through a three-layer multi-layer perceptron serving as the classifier. Each hidden layer employs L2 regularization, batch normalization, ReLU activation, and dropout. The final layer applies a softmax activation function identical to the one used in the CNN to generate class probabilities, with the model ultimately predicting the class with the greater probability (Figure 2).

Both the CNN and GNN are trained with 5-fold cross-validation for 50 epochs, using early stopping and a learning rate decay to optimize the training. The criteria for the early stopping of both models are set such that if the validation accuracy does not increase for 10 epochs, the training stops early. Similarly, if the validation accuracy does not increase for three epochs, the learning rate is halved until it reaches below 0.000001, at which point the training is stopped early.

To compare model performance, the same 20% of the dataset (3023 plays) was held out during model training. To evaluate the performance, I looked at the AUC, F1 score, and test accuracy of both models.



[Figure 2: The above diagram includes two panels illustrating the pipeline of the CNN and GNN models, respectively. Panel A depicts the CNN pipeline, starting with a stacked matrix as the input. The data are then transformed through two convolutional blocks, each followed by a pooling layer consisting of both max and average pooling. Finally, the data goes through the dense classifier head, which outputs a binary result indicating either man or zone coverage. Panel B shows the GNN pipeline. The initial input graph is transformed by the feature encoder that maps the 21-dimensional feature vector onto a 64-dimensional space. Subsequently, the graph is transformed through two convolutional layers before

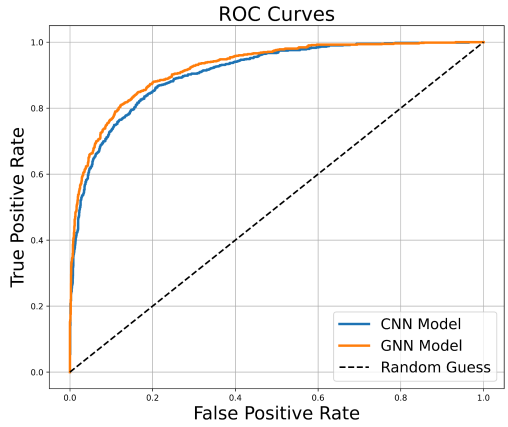
ultimately being input into a hierarchical pooling layer that concatenates six different matrices that have been pooled. Finally, the data goes through the MLP classifier before also outputting a binary value indicating either man or zone coverage.

Results:

To evaluate the performance of the models, I analyzed the test accuracy, F1 score, and AUC-ROC (area under the Receiver Operating Characteristic curve). Ultimately, the GNN had a greater AUC and F1 score; however, the CNN had a greater test accuracy. The values for the AUC, test accuracy, and F1 score are displayed in Table 2, while the AUC-ROC is illustrated in Figure 3:

	Test Accuracy	F1 Score	AUC
GNN Model	83.7%	0.737	0.920
CNN Model	84.3%	0.730	0.904

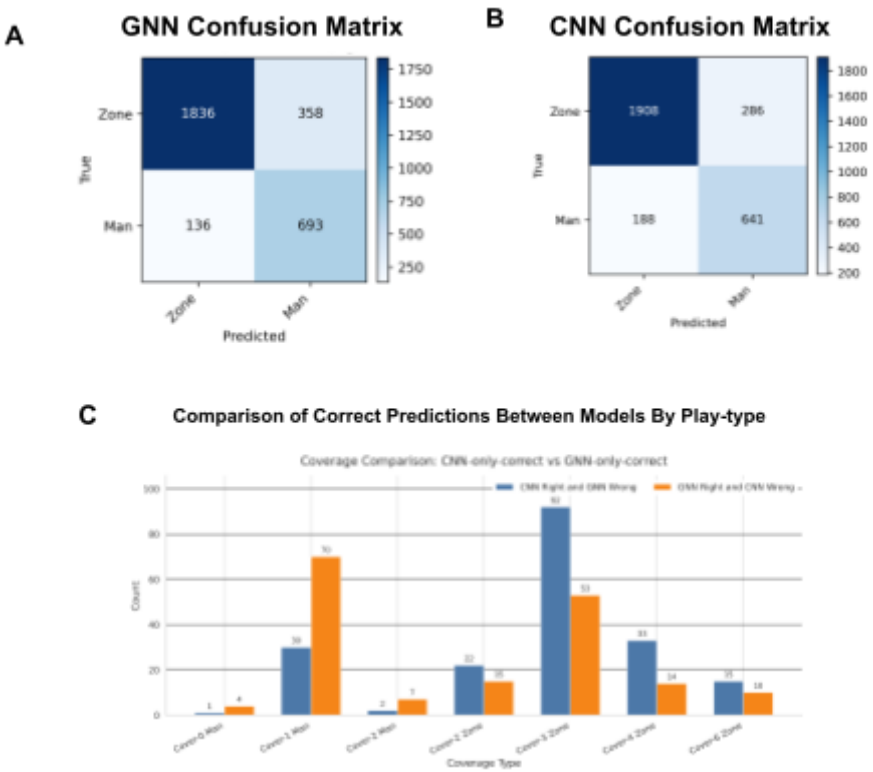
[Table 2: Table of the test accuracy, AUC, and F1 score for the GNN and CNN models. Each value is listed to three significant figures.]



[Figure 3: AUC-ROC of the CNN and GNN models. The blue line represents the CNN model, while the orange line represents the GNN. The dotted black line is a baseline of 50% accuracy.]

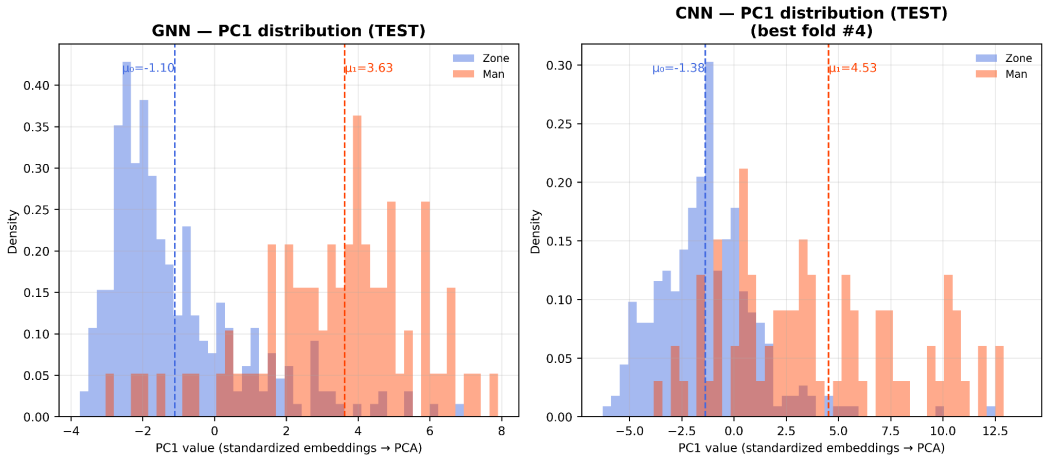
Test accuracy measures the proportion of correct predictions to the total number of predictions. Having a higher test accuracy implies that the model made fewer mistakes overall. However, in the case of class imbalance, accuracy alone can be misleading. F1-score

is the harmonic mean of the precision and recall of a model. Precision measures the number of times a model predicted a particular class correctly compared to the total number of times it predicted that class. Recall measures the number of instances of a class the model successfully predicted. Taking the harmonic mean incentivizes a balance between the two metrics, ensuring that a high score in one doesn't overshadow a poor score in the other. A model with a greater F1-score provides more balanced predictions, particularly for imbalanced datasets. Finally, AUC (Area Under Curve) is the area under an ROC curve. An ROC (Receiver Operating Characteristic) curve is a graph used to evaluate the performance of a binary classification model. It plots the relationship between true positives and false positives in order to assess the effectiveness of a model at separating the two classes at different thresholds. Consequently, the AUC reflects a model's ability to distinguish between classes across different thresholds, illustrating how well the model would generalize to unseen data.



[Figure 4: Panel A is the confusion matrix of the GNN labeled with the true values and the predicted values by the mode, while Panel B illustrates the confusion matrix of the CNN model. The color represents the frequency of that prediction. Panel C shows the differences in correct predictions between the models depending on the specific type of zone or man coverage.]

As test accuracy can often be misleading in cases of class imbalance, further analysis was conducted to examine how the models differed in their predictions. The confusion matrices show that while the CNN outperformed the GNN on zone plays, the GNN was more accurate on man plays. Diving deeper, the performance of the two models was compared based on more specific representations of man and zone plays. While the labels of man and zone are sufficient to describe defensive coverage, each has various subtypes that provide a more detailed description of how the defense is structured. When comparing the performance of the models on these class subtypes, a clear trend emerged. The CNN had correctly predicted many more plays of the majority class, cover-3 zone—constituting 37.4% of all plays—compared to the GNN, along with a better performance on all other types of zone coverage; however, the CNN performed worse than the GNN on all types of man coverage (Figure 4). Overall, the GNN achieved a greater AUC and F1 score while the CNN had a higher test accuracy. However, both outperformed the random classifier, with the GNN having an AUC of 0.920 compared to the CNN's 0.904.



[Figure 5: A plot of the distribution of the first principal component (PC1) of the standardized penultimate layer of the GNN (left) and CNN (right) models, illustrating the separation of the true labels zone (blue) and man (red). The dotted lines, annotated with the symbol μ_0 for zone plays and μ_1 for man plays, show the mean PC1 value of all plays for that respective coverage type for each respective model.]

While both models were able to distinguish between man and zone plays, the penultimate layer of the CNN showed greater linear separation along the first principal component (PC1). In contrast, the penultimate layer of the GNN had a more compact and structured shape. Hence, while the CNN learns spatial filters that examine localized, rigid features,

creating a decision boundary in a feature space that is often linear, the GNN can view coverages using relational context, allowing it to classify coverages using more distributed, contextual representations, allowing it to capture more nuanced patterns compared to the CNN model (Figure 5).

Discussion:

The results of this study support the hypothesis that the use of a GNN is more effective than a CNN in classifying NFL defensive coverage schemes when using player tracking data. While the CNN achieved a marginally greater test accuracy, further analysis reveals that this was primarily driven by the CNN's bias towards the majority class (zone coverage), resulting in significantly worse performance than the GNN on man plays (Figure 4). In contrast, the GNN achieved a greater AUC and F1 score, indicating that it outperformed the CNN with stronger generalization and a more balanced classification performance.

The superior performance of the GNN can be explained by the fundamental differences between the two models' architectures. CNNs process features as fixed, grid-like matrices. This makes them well-equipped when analyzing spatially consistent data such as images. However, football plays cannot be evaluated by simply analyzing each player on the field independently. Instead, it is essential to also understand how players on the field interact with each other. Hence, the architecture in this paper adapts the CNN to consider relational context, inputting the relative distance, speed, and acceleration of each player in relation to each player on the opposing team. However, despite these efforts to improve the effectiveness of the CNN, the GNN is still better suited for this task. A typical CNN performs its convolutions on a fixed, grid-like window (Figure 1). Thus, the model gives greater weight to local neighbourhoods as it assumes data adjacent in the input are likely structured similarly in the field. However, given that players often change how they line up, their alignment on the field will rarely match the structure of the matrix. Consequently, the CNNs in this paper use a 1x1 kernel to avoid the incorrect assumption that adjacency on the grid and the field are related. This significantly limits the model's predictive power by forcing the CNN to process each player-player pair in isolation.

In contrast, the GNN represents each player as a node in a graph and forms relationships between nodes in the form of edges. This allows the GNN to describe the complex and often arbitrary shape of the players on a football field. Additionally, the unique architecture of the GNN enables it to be further tuned by adjusting the weight of specific edges, placing greater weight on certain connections. In particular, the GNN employed in this paper placed greater weights based on proximity and a multiplier on within-team connections, which was determined to enhance the effectiveness of the model through empirical testing. The use of edge weights impacts the message-passing process, wherein the edge

weight scales node embeddings of neighbours. Hence, neighbours with stronger edge weights influence the update of the node's embeddings more. By contrast, this would be much more difficult to implement into the CNN due to the lack of edges to demonstrate connections between players. Instead, interactions between players are interpreted through fixed arrangements, making player-player relationships much more difficult to adjust. The GNN's ability to model and optimize a relational graph allows it to retain contextual information across layers that would have been lost through transformations to a CNN's input matrix. Due to its better relational reasoning ability, the GNN can better adapt to shifts in alignment and generalize to new plays more effectively.

Although this paper categorizes coverage schemes into two labels, man and zone coverage, there are many variations of each that are employed by defensive coaching staff, which significantly affect defender alignment. Most coverage schemes can be classified into one of seven distinct classes: cover-0 man, cover-1 man, cover-2 man, cover-2 zone, cover-3 zone, cover-4 zone, and cover-6 zone. To understand why the CNN and GNN models both exhibited certain tendencies in their predictions, it is essential to first understand the distinctions between each coverage type.

The coverage types each include a number indicating the number of deep defenders present during that play. Zone coverage, to have their defenders cover different zones throughout the field, will often include more deep defenders who are responsible for covering deeper passes in zones further downfield. There are three types of man coverage, with the first being Cover-0 Man. Cover-0 involves significant risk as no deep defenders are employed. Instead, five defenders are all responsible for marking their respective eligible receiver while the remaining six players on the defensive team rush the quarterback in what is called a blitz. Similarly, Cover-1 Man involves only one deep defender, with the only difference being that five players will now rush the quarterback, also known as a dog. However, at times, defenses will instead sacrifice one rusher and rather have two defenders cover a receiver to ensure they have little chance of catching the ball. The final man coverage titled Cover-2 Man is a hybrid between man and zone coverage, making it often difficult to classify. In Cover-2 Man, five defenders are once more responsible for covering the eligible receivers through man coverage. However, unlike the other forms of man coverage, two safeties now cover the deep half of the field, providing help to the man defenders in the case of a deep ball.

In contrast, Cover-2 Zone employs two deep safeties with all underneath defenders playing zone coverage. While Cover-2 Zone appears similar to Cover-2 Man at snap, the key difference is the lack of man coverage for close balls and hence the absence of help defense for receivers downfield. The remaining zone coverages follow a similar pattern, with Cover-3 Zone employing three deep defenders and four underneath defenders, Cover-4

Zone (also referred to as quarters) using four deep defenders and three underneath defenders. Finally, Cover-6 Zone appears slightly different from the other forms of zone coverage, with the defense essentially splitting the field into two halves. On one half, it plays a Cover-4 Zone, employing four deep defenders, while on the other, it plays a Cover-2 Zone, utilizing two deep defenders. This split coverage allows the defense to scheme around specific receivers and to prevent deep balls from being thrown.

From analyzing the Big Data Bowl dataset, the majority class appears to be Cover-3 Zone. Hence, given the apparent overfitting in the CNN, it is reasonable that it predicts Cover-3 Zone at a significantly higher rate than the GNN (Figure 4). With 92 more correct predictions than the GNN on Cover-3 Zone plays, the results highlight that the CNN heavily relies on this class for its high test accuracy. Furthermore, the CNN outperforms the GNN on all other zone coverages, with the GNN rarely predicting zone coverage plays correctly that the CNN failed to classify. However, the GNN significantly outperforms the CNN on man coverage, with 70 correct classifications that the CNN was unable to predict. Even more interestingly, on Cover-2 Man plays, a coverage that is typically difficult to predict due to its similarity to Cover-2 Zone plays, particularly pre-snap, the CNN was rarely able to classify any plays that the GNN incorrectly labeled correctly. These results highlight the CNN's failure to classify rarer plays, often resulting in the model incorrectly classifying the play as zone coverage. In contrast, the GNN's architecture allows it to process much more complex patterns, making it more reliable for real-world defensive coverage recognition.

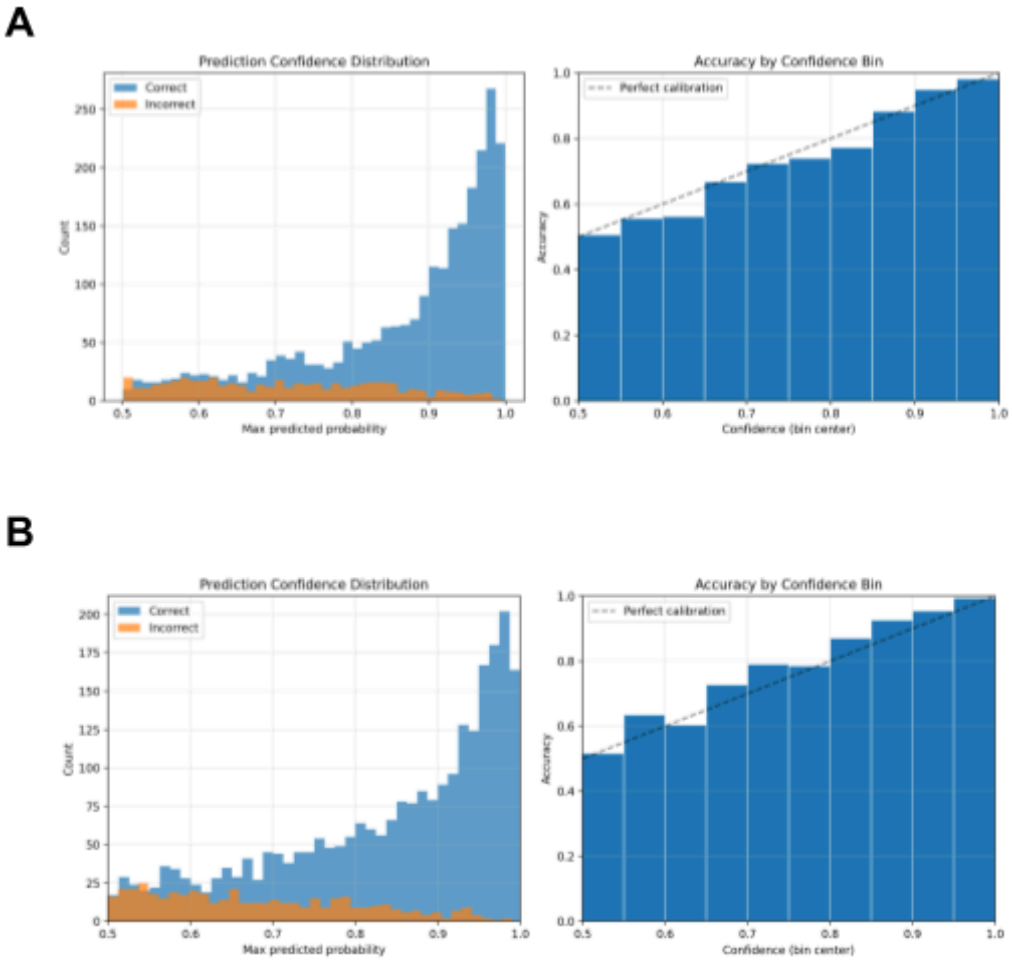
Conclusion and Future Directions:

This study evaluated the performance of a CNN and GNN for the pre-snap classification of NFL defensive coverage schemes using player tracking data from the 2025 Big Data Bowl. Overall, the CNN achieved a marginally higher test accuracy due to its bias towards predicting the major class, while the GNN outperformed the CNN with both a greater F1 and AUC score. The GNN's ability to model the field as a graph allowed it to maintain greater relational context than the CNN. This lack of signal decreased the confidence of the CNN in its predictions and led it to primarily predict zone coverage. These findings underscore the GNN's superior ability to analyze dynamic, non-grid environments such as American football, taking advantage of its graph-based architecture. Ultimately, the GNN proved to be more effective in making balanced predictions, supporting the idea that it is better suited for coverage scheme classification due to its ability to analyze connections and contextual relationships when making predictions (Khemani et al., 2024).

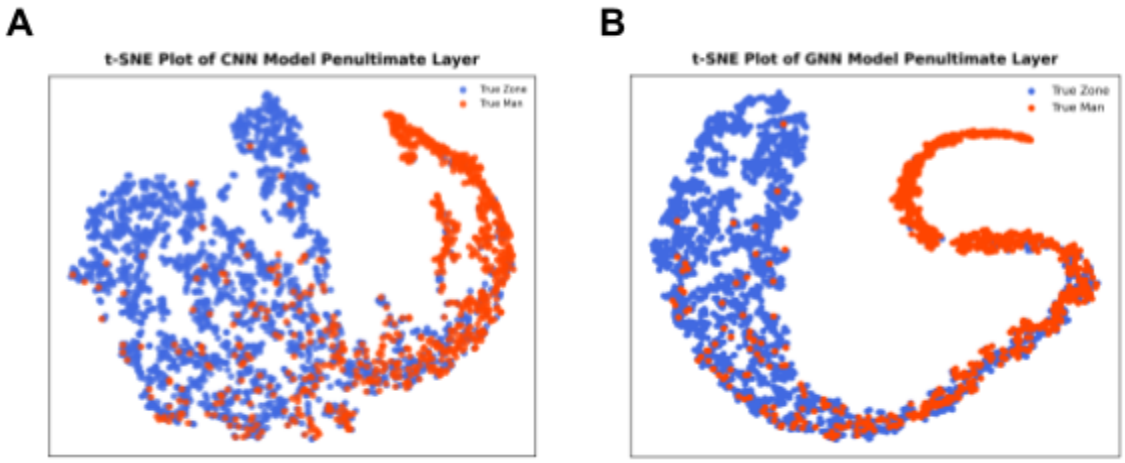
In future work, I plan to explore how incorporating temporal learning can impact both models. A limitation of the models used in this paper was their reliance on a single frame.

Defensive coverages are dynamic, with players' alignment, motion, and position constantly changing before the snap. Hence, the models used in this paper were limited by their inability to capture patterns introduced by the movement of players prior to the snap. Temporal learning achieves this by allowing the models to analyze multiple frames and predict based on how the data changes over time. Specifically, the use of recurrent or temporal GNNs could address the limitations caused by static frames by modeling how the relationships between players evolve across frames, rather than being constrained to singular snapshots. As a result, the model capture a more nuanced representation of what is occurring pre-snap and potentially detect new patterns by analyzing the sequential shifts in player positions. As our knowledge of football evolves, we must explore how data science can complement human coaching to enhance decision-making, enabling fans to experience sports at the highest level both physically and tactically.

Appendix:



[Figure 6: Panel A shows the count of correct and incorrect guesses relative to the model's predicted probability of success, along with the model's accuracy relative to their confidence in that guess for the GNN. Panel A shows the same diagrams for the CNN.]



[Figure 7: Panel A shows the t-SNE plot of the penultimate layer of the CNN model, while Panel B shows the t-SNE plot of the penultimate layer of the GNN model. In both models, red points illustrate man plays while blue points represent zone plays.]

References

Ajit, A., Acharya, K., & Samanta, A. (2020). A Review of Convolutional Neural Networks. *IEEE*, 1-5. https://doi.org/10.1109/ic-ETITE47903.2020.049

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016, July 21). Layer Normalization. *arXiv*. https://doi.org/10.48550/arXiv.1607.06450 (Non-peer-reviewed)

Fernandes, C. J., Yakubov, R., Li, Y., Prasad, A. K., & Chan, T. C.Y. (2019, October 26). Predicting plays in the National Football League. *Journal of Sports Analytics*, 6(1), 35-43. https://doi.org/10.3233/JSA-190348

Gholamalinezhad, H., & Khosravi, H. (2020, September 16). Pooling Methods in Deep Neural Networks, a Review. *arXiv*. https://doi.org/10.48550/arXiv.2009.07485 (Non-peer-reviewed)

Goyal, U. (2020, February). Leveraging Machine Learning to Predict Playcalling Tendencies in the NFL. *DSpace@MIT*. Retrieved August 14, 2025, from https://dspace.mit.edu/handle/1721.1/129909

The History of the Rules. (2025). NFL Football Operations. Retrieved August 8, 2025, from https://operations.nfl.com/the-rules/evolution-of-the-nfl-rules/ (Non-peer-reviewed)

Identifying significant features for Player Evaluation in NFL comparing ANNs and Traditional Models. (2021). *Arrow@TU Dublin*. https://doi.org/10.21427/EAC6-4R95

Khemani, B., Patil, S., Kotecha, K., & Tanwar, S. (2024, January 16). A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions. *Journal of Big Data*, *11*(18). https://doi.org/10.1186/s40537-023-00876-4

Kingston, C. (2023, May 12). Measuring Grit in NFL Cornerbacks using Statistical Analysis. https://dspace.mit.edu/bitstream/handle/1721.1/151676/kingston-ctk-meng-eecs-2023-thesis.pdf?sequence=1&isAllowed=y

Lopez, M., Bliss, T., Blake, A., Mooney, P., & Howard, A. (2024). *NFL Big Data Bowl 2025*. Kaggle. Retrieved August 14, 2025, from https://kaggle.com/competitions/nfl-big-data-bowl-2025 (Non-peer-reviewed)

McManus, C. (2025, March 17). Artificial Intelligence for better in-game NFL performance. *Scholarworks at WMU*. Retrieved August 14, 2025, from https://scholarworks.wmich.edu/cgi/viewcontent.cgi?article=4965&context=honors_theses

Ofeidis, I., Kiedanski, D., & Tassiulas, L. (2025, January 16). An Overview of the Data-Loader Landscape: Comparative Performance Analysis. *IEEE*, pp. 360–367. https://doi.org/10.1109/BigData62323.2024.10825421

Purwono, P., Ma'arif, A., Rahmaniar, W., Fathurrahman, H. I. K., Frisky, A. Z. K., & Haq, Q. M. u. (2023). Understanding of Convolutional Neural Network (CNN): A Review. *International Journal of Robotics and Control Systems*, 2(4), 739-748. https://doi.org/10.31763/ijrcs.v2i4.888

Quarterbacks: Finding Weakness and Strength in Zone vs. Man Coverage. (2023, August 1). Capital QB's. Retrieved August 14, 2025, from https://www.capitalqbs.com/quarterbacks-finding-weakness-and-strength-in-zone-vs-man-coverage/ (Non-peer-reviewed)

Sanchez-Lengeling, B., Reif, E., Pearce, A., & Wiltschko, A. B. (2021, September 2). A Gentle Introduction to Graph Neural Networks. *Distill*. https://doi.org/10.23915/distill.00033 (Non-peer-reviewed)

Song, H., Jazaery, M. A., Ding, H., Cheong, L. L., Jung, J., Band, M., Chi, M., & Bliss, T. (2023). Explainable Defense Coverage Classification in NFLGames using Deep Neural Networks. *Amazon ML Solutions Lab*. https://www.amazon.science/publications/explainable-defense-coverage-classification-in-nfl-games-using-deep-neural-networks

Taylor, C. (2020). Deep Learning for In-Game NFL Predictions. https://cs230.stanford.edu/projects_winter_2020/reports/32263160.pdf#page=1.33 (Non-peer-reviewed)

Grids to Graphs: GNNs improve NFL defensive coverage classification over CNNs

Abstract:

This paper investigates the application of predictive neural network models to classify NFL defensive coverages using player tracking data. I propose a novel solution using a graph neural network (GNN) as an alternative to the conventional use of convolutional neural networks (CNNs). Both models were compared based on their test accuracy, F1 score, and AUC. While the CNN reported a greater test accuracy (84.3%) than the GNN (83.7%), this was primarily due to its bias towards zone coverage, the majority class. In contrast, the GNN yielded more balanced predictions with a greater F1 score (0.737 vs. 0.730) and AUC (0.920 vs. 0.904) than the CNN. The GNN was also more effective than the CNN when classifying less common coverage scheme variations. These findings suggest that GNNs are better suited for classifying NFL defensive coverages. This is because their unique architecture represents information as graphs, helping them better understand player-player interactions.

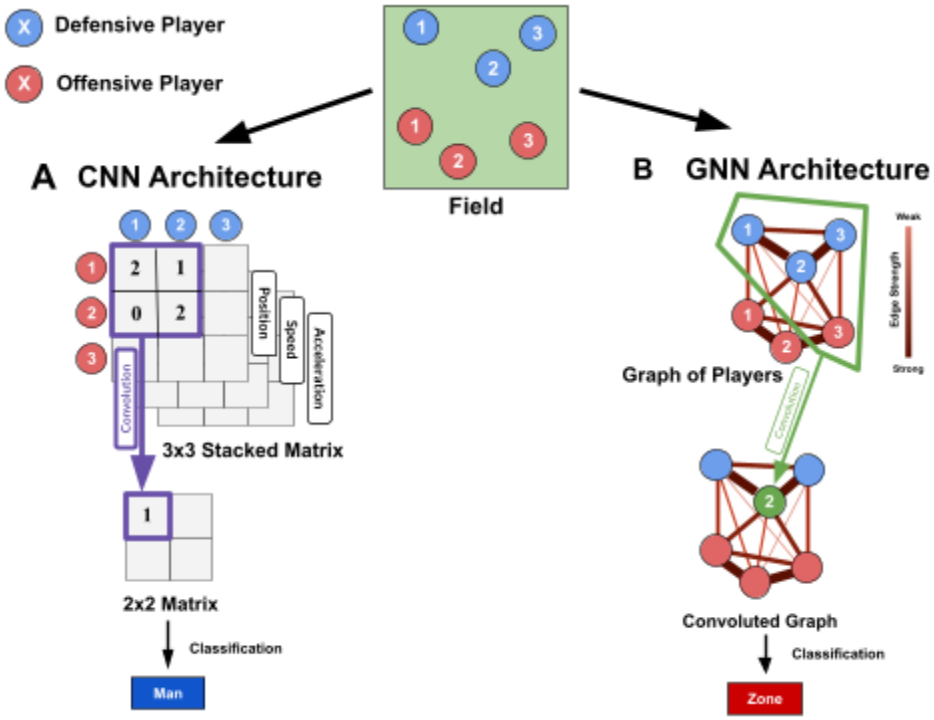
Introduction:

The National Football League (NFL) has long focused its analysis and changes towards offense to make the viewing experience more engaging. Per the NFL's Competition Committee, "if someone wants to accuse the National Football League of promoting offense to make the game more exciting, [the committee believes the league should plead guilty]" (NFL Football Operations, 2025). Hence, the recent rise in machine learning (ML)-powered football analytics has predominantly focused on offensive playcalling and performance (McManus, 2025; Goyal, 2020; Walsh, 2021; Fernandes et al., 2019; Taylor, 2020). On the other hand, considerably less analysis has been performed regarding the defensive end of the floor.

On defense, the coaching staff is responsible for positioning their players in specific formations, known as coverage schemes. Defensive coverage schemes determine the responsibilities of each player on the defensive team. Coverage schemes are categorized into two broad types: man coverage and zone coverage. In man coverage, each defender is responsible for a specific offensive player throughout the play. In contrast, zone coverage requires defensive players to defend any players that enter a particular area on the field known as their "zone" (Kingston, 2023). Both coverage schemes are employed for vastly different use cases, with each having its own benefits. While zone coverage often minimizes big plays, man coverage leads to a tighter defense, placing greater pressure on the opposing quarterback to make an accurate throw (Capital QB's, 2023).

Determining which coverage scheme the defensive team will use helps the coaching staff of the offensive team choose plays to increase their chances of scoring. As a result, the coaches of the defensive team often disguise their coverage scheme by making their players rotate following the snap—the start of a play in football. This can make play calling difficult for even the most skilled coaches. Hence, having a method of classifying coverage schemes pre-snap would enable the offense to anticipate and plan against the defensive strategies used by the opposing team.

One way of classifying defensive coverage is by using predictive models. Currently, the most successful method of classifying defensive coverage schemes is convolutional neural networks (CNNs) (Song et al., 2023). CNNs are a form of artificial neural network that process data structured in grid-like matrices. They use convolutional layers (which can be thought of as a sliding window) to capture local patterns. Throughout the training process, the weights of the window are learned (Figure 1). This process of convolution is why CNNs are often used when processing images. Images have strong spatial locality (pixels within a proximity are highly correlated), meaning the ability to capture local patterns effectively allows CNNs to learn the complex structures of objects within images without becoming overwhelmed by the large quantity of pixels (Purwono et al., 2022).



[Figure 1: Panel A shows diagrams explaining the architecture of the CNN. The football field (green square) has three offensive players (blue) and three defensive players (red). The first

illustration in panel A illustrates how the field is represented as a stacked matrix, with the values representing the distinct pairwise relative features. The purple square is then convoluted into one number in the following illustration before ultimately becoming a binary output. Panel B presents an illustration of the players in the form of a graph. The lines between the circles (nodes) represent the connections between players (edges). The thickness and color of the edges represent the strength of these connections. Similar to panel A, the values of the graph are convoluted into a new graph. Ultimately, the GNN classifies into a binary output.

Modern methods of applying CNNs to coverage scheme classification have found success in using relative data in conjunction with static data. Song et al. (2023) demonstrate that relying solely on static features significantly limits the predictive capabilities of a CNN. However, by conducting feature engineering to construct a matrix of pairwise relative features between each member of the offensive team and the players on the defensive team, they were able to model complex player interactions, thereby maintaining greater spatial and relational context (Figure 1) (Song et al., 2023).

While Song et al. (2023) found success through the use of relative features, it must be noted that they increase the dimensionality of the data significantly due to the many static values that are repeated throughout the matrix. This introduces various concerns, including greater computational demands, a high risk of overfitting, and an unintentional emphasis placed on static features, as they are constantly repeated within each pairwise comparison. Additionally, CNNs likely struggle to anticipate shifts that could occur to players' positions throughout a play due to their inability to understand how each player relates to the others on the field.

In contrast, the use of a graph neural network (GNN) should theoretically be more effective due to its unique architecture. GNNs operate on graph-structured data, describing data through nodes, edges, and a global context. The graph is passed through a differentiable function, such as a multilayer perceptron (MLP), to process the data. In the case of binary classification, such as predicting coverage schemes, the prediction target is the global context. As such, the information within the nodes and edges must be passed through a pooling layer, creating a concatenated matrix that can be used for classification (Sanchez-Lengeling et al., 2021). This process of pooling information from the nodes and edges to make predictions about the global context makes GNNs particularly robust, as it allows the GNN to learn from the intricate relationships between players on the field. Hence, the use of a GNN should enhance the ability to classify defensive coverages.

Due to the GNN's network structure, it can capture the relational context between players, making it better suited for modeling NFL plays. As CNN's interpretation of rigid, grid-like

matrices relies heavily on spatial adjacencies, it is unable to effectively capture relational context when modeling the alignment of players on the field. Thus, the GNN should produce superior results when classifying coverage schemes compared to the CNN.

Previous works, including those of Song et al. (2023), have primarily focused on optimizing the effectiveness of the CNN model by incorporating methods such as temporal learning to maximize accuracy. However, this paper will instead focus on comparing the performance of CNN models to that of GNN models in an attempt to develop a novel classification method for NFL coverage schemes. Consequently, this paper discusses the hypothesis that a GNN is more effective at NFL defensive coverage classification than a CNN.

Methods:

In football, man and zone are the two primary labels used to classify defensive coverages. Therefore, differentiating between man and zone coverages is both relevant and sufficient to identify NFL defensive coverages. As there are two primary outputs, this paper will view coverage scheme classification as a binary classification problem between man and zone coverage. The study will be performed on the 2025 Big Data Bowl, a public dataset released annually by the NFL. The dataset includes tracking data for all 22 players on the field (11 offense and 11 defense) during passing plays, along with detailed play-level descriptions. Furthermore, the dataset includes contextual information for each game as well as player-specific attributes such as build and composition (Lopez et al., 2024). The data is drawn from the first nine weeks of the 2022-23 NFL season and includes 3,023 plays.

Initially, the data is pre-processed, with any plays not labeled as either man or zone being filtered out. These plays are labeled as “other” in the dataset and constitute plays where either the line of scrimmage is within the endzone, meaning traditional forms of pass coverage are not employed, or during a quarterback kneel, which is a special form of play that occurs when the knee of a quarterback touches the ground directly after the snap and thereby immediately ends the play for strategic reasons. Subsequently, the dataset is filtered such that only the 31st frame of each play remains. The data is recorded at 10 frames per second, meaning the 31st frame represents three seconds into the play. As no snaps occur before the 31st frame, it ensures that the data is pre-snap and hence applies to offenses scheming around the defensive coverage employed. Furthermore, given that the frame is three seconds into the play, it is presumed that most players are already in their coverage positions. Though ideally, a model should use all pre-snap frames, a singular frame was chosen due to computational limitations.

CNN (normalized)	GNN (normalized)
Distance from the line of scrimmage	Distance from the line of scrimmage
Position on the y-axis	Position on the y-axis
Speed in the x-direction	Speed in the x-direction
Speed in the y-direction	Speed in the y-direction
Speed	Speed
Acceleration in the x-direction	Acceleration in the x-direction
Acceleration in the y-direction	Acceleration in the y-direction
Acceleration	Acceleration
Orientation	Orientation
Relative speed in the x-direction	Nearest opponent distance
Relative speed in the y-direction	Distance to team centroid
Relative acceleration in the x-direction	Position (represented as 10 distinct binary channels representing the quarterback, running back, wide receiver, tight end, offensive lineman, defensive lineman, linebacker, cornerback, safety, and nickel)
Relative acceleration in the y-direction	
Relative position in the x-direction	
Relative position in the y-direction	

[Table 1: The list of all the features used when training the CNN and GNN models. All listed features were either taken directly from or calculated with data found on the 2025 Big Data Bowl dataset and were all normalized for stability.

The decision to provide speed along with the x and y components of speed as separate features was made intentionally. CNNs and GNNs are both linear models and are hence unable to derive the speed of a player based on the directional components. By providing features such as speed and acceleration in both formats, the model can choose the

representation it finds most effective, ensuring flexibility between scalar and directional values, and can therefore identify new patterns to better classify coverage schemes.

The CNN was built following the architecture described in Song et al. (2023). The model was trained on a 15x11x11 stacked matrix—with 15 features, 11 offensive players, and 11 defensive players (Table 1). The data were then reshaped into batches of 64 plays to enhance the training time and generalization of the model (Ofeidis et al., 2024). The architecture of the CNN utilizes two convolutional layers, designed to extract high-level representations while preserving the spatial structure of the matrix. The first layer consists of three successive 1x1 convolutional layers. The use of a unit kernel enables the combination of all 15 feature values into a single value without blurring or merging with information from other cells. Hence, the output maintains its 11x11 spatial identity and arrangement while reducing the computational load (Ajit et al., 2020). Subsequently, a dropout layer is introduced to drop 30% of the data, thereby minimizing overfitting and regularizing the data. Following the dropout, the features are processed through a pooling layer that consists of average and max pooling. The purpose of implementing the pooling layer is to down-sample the feature maps following the convolutional layers to reduce computational cost and control overfitting (Gholamalinezhad & Khosravi, 2020). Specifically, the pooling layer collapses the dimension of offensive players, transforming the data into 128-d vectors for each defensive player. Average pooling captures the overall trend, while max pooling identifies the strongest signals. A weighted sum of both pooling values is taken (Equation 1).

$$x = (mean \times 0.7) + (max \times 0.3) \qquad [1]$$

Following the pooling layer, the features are input into another convolutional layer consisting of three 1x1 convolutional layers, each followed by batch normalization to normalize along the channel dimension. Once more, the unit kernel allows for linear transformations of the channels, increasing the channels to 160 before ultimately outputting 96 channels. After the second convolutional layer, the data is transformed through a second pooling layer that collapses the defensive-player axis while implementing the same weighted sum of average and max pooling. Hence, the output is a 96-d vector describing each batch. Finally, the data is transformed through a set of dense layers. The first dense layer employs a ReLU activation that maps the 96 features to 96 output dimensions, thereby introducing non-linearity for more complex feature representations. Subsequently, batch normalization is applied to ensure a standard distribution of the activations throughout the batch. Thereafter, a second dense layer is employed, expanding the feature space from 96 to 256 dimensions, thereby utilizing a higher-dimensional space to capture more complex patterns and linearly separable relationships. After another batch normalization is applied to enhance stability and convergence. Subsequently, layer

normalization is applied, which stabilizes the dynamics of the hidden state and improves training time (Ba et al., 2016). Finally, the last dense layer reduces the 256-dimensional normalized vector to two outputs, being the two coverage schemes, and applies a softmax activation to transform the logits into a probability distribution between the coverage schemes. This serves as the classifier, matching each sample to a class based on the learned feature representation from the previous stages of the model. Ultimately, the class with the greater probability is output as the model's prediction of the class (Figure 2).

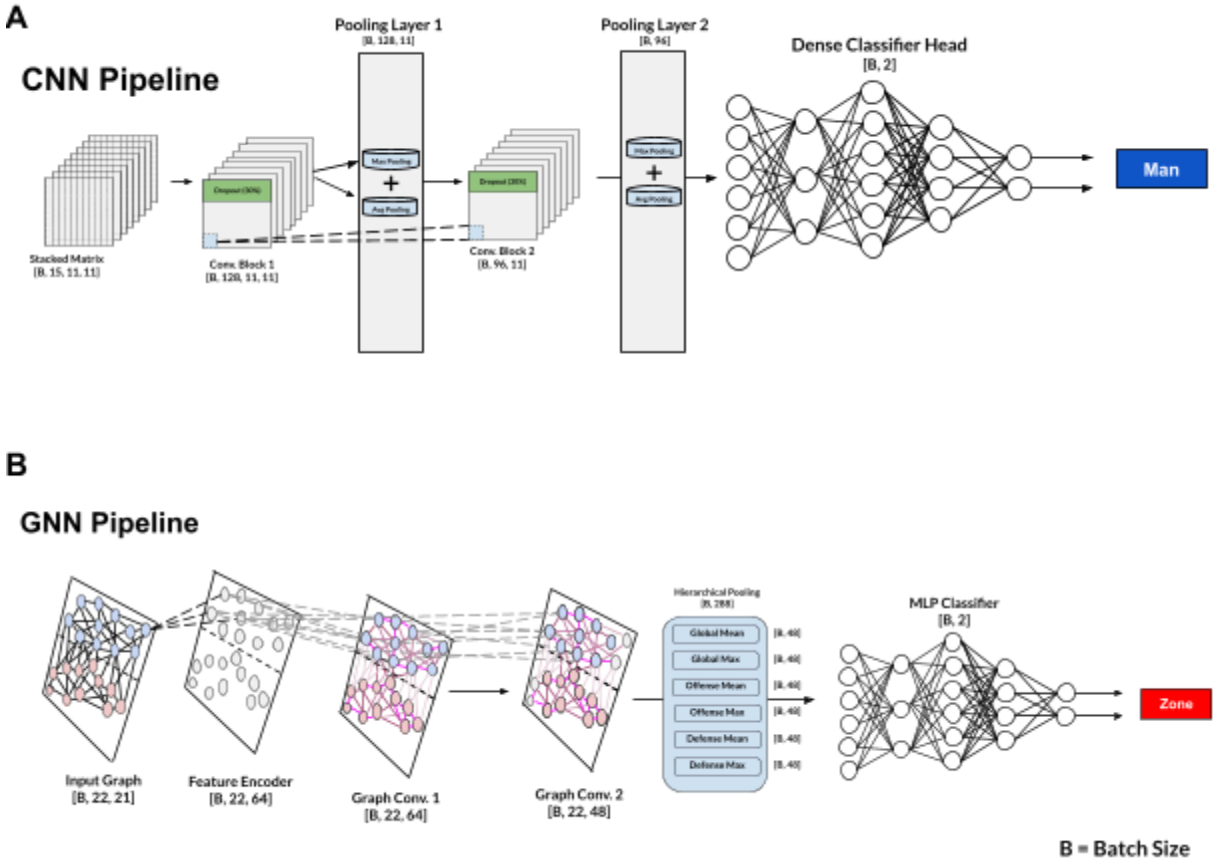
The GNN was trained on a graph consisting of 22 nodes (11 offensive players and 11 defensive players). Each of these nodes contains a 21-dimensional feature vector. To construct the graph, edges are defined by both Euclidean distance and team membership. Edges are placed between all players within 12 yards of each other, with weights between players on the same team having a multiplier. The value of 12 yards between players and the multiplier for same-team allegiance were determined empirically to optimize the model's accuracy. The graph is then transformed by an encoder that maps the 21-dimensional feature vectors onto a 64-dimensional space. This layer uses L2 regularization with a regularization parameter of 10^{-4}. Then, the data is stabilized through batch normalization, before introducing non-linearity through a ReLU activation, and employing a dropout layer to minimize overfitting.

The projected features are then processed through two convolutional layers. Node representations, the feature vectors for each player, are then updated through weighted message passing, which gathers aggregate features from all adjacent nodes (weighted by the strength of that edge) and performs a linear transformation followed by a ReLU activation. The first graph layer outputs 64 channels, followed by batch normalization and a dropout layer. In comparison, the second outputs 48 channels and is also followed by batch normalization and a dropout layer.

After the second convolutional layer, the node embeddings, which are 22 x 48 in size, are transformed through a pooling layer. The pooling layer first performs average and max pooling on the 22 players in the plays, followed by applying the same operations to defensive and offensive subgraphs separately. These processes yield six distinct 48-dimensional pooled vectors, which are concatenated into a 288-dimensional representation of the play. This 288-dimensional graph vector is then passed through a three-layer multi-layer perceptron serving as the classifier. Each hidden layer employs L2 regularization, batch normalization, ReLU activation, and dropout. The final layer applies a softmax activation function identical to the one used in the CNN to generate class probabilities, with the model ultimately predicting the class with the greater probability (Figure 2).

Both the CNN and GNN are trained with 5-fold cross-validation for 50 epochs, using early stopping and a learning rate decay to optimize the training. The criteria for the early stopping of both models are set such that if the validation accuracy does not increase for 10 epochs, the training stops early. Similarly, if the validation accuracy does not increase for three epochs, the learning rate is halved until it reaches below 0.000001, at which point the training is stopped early.

To compare model performance, the same 20% of the dataset (3023 plays) was held out during model training. To evaluate the performance, I looked at the AUC, F1 score, and test accuracy of both models.



[Figure 2: The above diagram includes two panels illustrating the pipeline of the CNN and GNN models, respectively. Panel A depicts the CNN pipeline, starting with a stacked matrix as the input. The data are then transformed through two convolutional blocks, each followed by a pooling layer consisting of both max and average pooling. Finally, the data goes through the dense classifier head, which outputs a binary result indicating either man or zone coverage. Panel B shows the GNN pipeline. The initial input graph is transformed by the feature encoder that maps the 21-dimensional feature vector onto a 64-dimensional space. Subsequently, the graph is transformed through two convolutional layers before

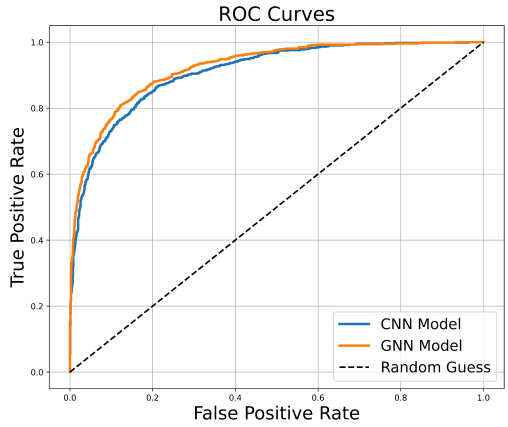
ultimately being input into a hierarchical pooling layer that concatenates six different matrices that have been pooled. Finally, the data goes through the MLP classifier before also outputting a binary value indicating either man or zone coverage.

Results:

To evaluate the performance of the models, I analyzed the test accuracy, F1 score, and AUC-ROC (area under the Receiver Operating Characteristic curve). Ultimately, the GNN had a greater AUC and F1 score; however, the CNN had a greater test accuracy. The values for the AUC, test accuracy, and F1 score are displayed in Table 2, while the AUC-ROC is illustrated in Figure 3:

	Test Accuracy	F1 Score	AUC
GNN Model	83.7%	0.737	0.920
CNN Model	84.3%	0.730	0.904

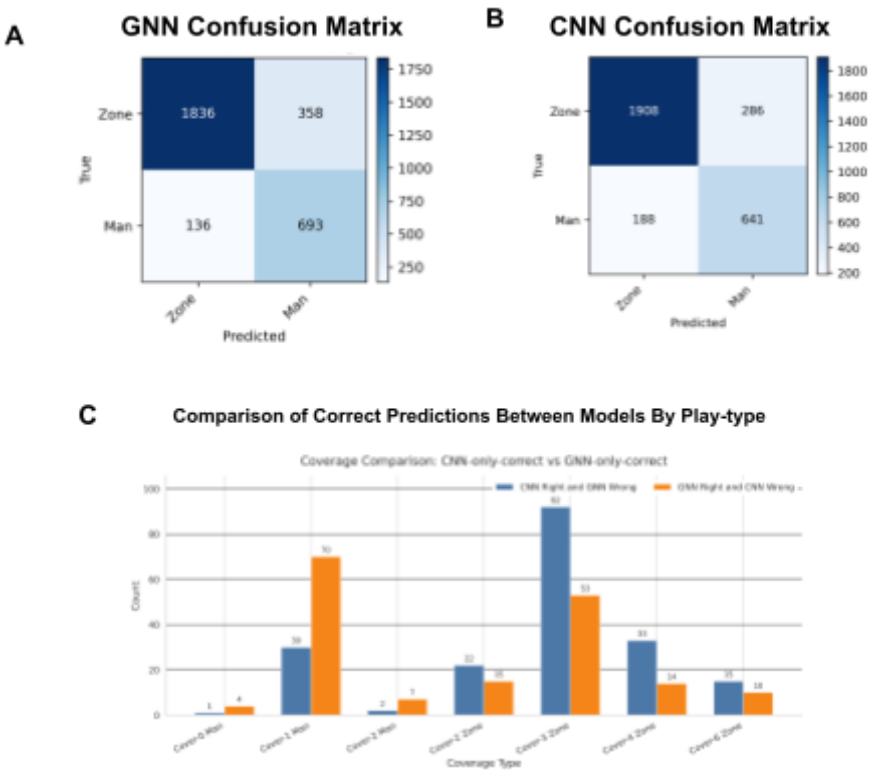
[Table 2: Table of the test accuracy, AUC, and F1 score for the GNN and CNN models. Each value is listed to three significant figures.]



[Figure 3: AUC-ROC of the CNN and GNN models. The blue line represents the CNN model, while the orange line represents the GNN. The dotted black line is a baseline of 50% accuracy.]

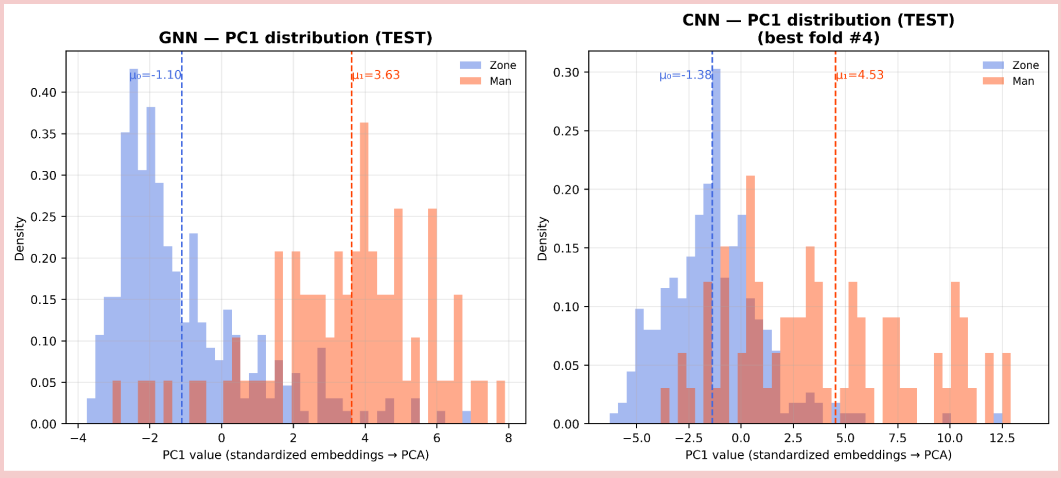
Test accuracy measures the proportion of correct predictions to the total number of predictions. Having a higher test accuracy implies that the model made fewer mistakes overall. However, in the case of class imbalance, accuracy alone can be misleading. F1-score

is the harmonic mean of the precision and recall of a model. Precision measures the number of times a model predicted a particular class correctly compared to the total number of times it predicted that class. Recall measures the number of instances of a class the model successfully predicted. Taking the harmonic mean incentivizes a balance between the two metrics, ensuring that a high score in one doesn't overshadow a poor score in the other. A model with a greater F1-score provides more balanced predictions, particularly for imbalanced datasets. Finally, AUC (Area Under Curve) is the area under an ROC curve. An ROC (Receiver Operating Characteristic) curve is a graph used to evaluate the performance of a binary classification model. It plots the relationship between true positives and false positives in order to assess the effectiveness of a model at separating the two classes at different thresholds. Consequently, the AUC reflects a model's ability to distinguish between classes across different thresholds, illustrating how well the model would generalize to unseen data.



[Figure 4: Panel A is the confusion matrix of the GNN labeled with the true values and the predicted values by the mode, while Panel B illustrates the confusion matrix of the CNN model. The color represents the frequency of that prediction. Panel C shows the differences in correct predictions between the models depending on the specific type of zone or man coverage.]

As test accuracy can often be misleading in cases of class imbalance, further analysis was conducted to examine how the models differed in their predictions. The confusion matrices show that while the CNN outperformed the GNN on zone plays, the GNN was more accurate on man plays. Diving deeper, the performance of the two models was compared based on more specific representations of man and zone plays. While the labels of man and zone are sufficient to describe defensive coverage, each has various subtypes that provide a more detailed description of how the defense is structured. When comparing the performance of the models on these class subtypes, a clear trend emerged. The CNN had correctly predicted many more plays of the majority class, cover-3 zone—constituting 37.4% of all plays—compared to the GNN, along with a better performance on all other types of zone coverage; however, the CNN performed worse than the GNN on all types of man coverage (Figure 4). Overall, the GNN achieved a greater AUC and F1 score while the CNN had a higher test accuracy. However, both outperformed the random classifier, with the GNN having an AUC of 0.920 compared to the CNN's 0.904.



[Figure 5: A plot of the distribution of the first principal component (PC1) of the standardized penultimate layer of the GNN (left) and CNN (right) models, illustrating the separation of the true labels zone (blue) and man (red). The dotted lines, annotated with the symbol μ_0 for zone plays and μ_1 for man plays, show the mean PC1 value of all plays for that respective coverage type for each respective model.]

While both models were able to distinguish between man and zone plays, the penultimate layer of the CNN showed greater linear separation along the first principal component (PC1). In contrast, the penultimate layer of the GNN had a more compact and structured shape. Hence, while the CNN learns spatial filters that examine localized, rigid features,

creating a decision boundary in a feature space that is often linear, the GNN can view coverages using relational context, allowing it to classify coverages using more distributed, contextual representations, allowing it to capture more nuanced patterns compared to the CNN model (Figure 5).

Discussion:

The results of this study support the hypothesis that the use of a GNN is more effective than a CNN in classifying NFL defensive coverage schemes when using player tracking data. While the CNN achieved a marginally greater test accuracy, further analysis reveals that this was primarily driven by the CNN's bias towards the majority class (zone coverage), resulting in significantly worse performance than the GNN on man plays (Figure 4). In contrast, the GNN achieved a greater AUC and F1 score, indicating that it outperformed the CNN with stronger generalization and a more balanced classification performance.

The superior performance of the GNN can be explained by the fundamental differences between the two models' architectures. CNNs process features as fixed, grid-like matrices. This makes them well-equipped when analyzing spatially consistent data such as images. However, football plays cannot be evaluated by simply analyzing each player on the field independently. Instead, it is essential to also understand how players on the field interact with each other. Hence, the architecture in this paper adapts the CNN to consider relational context, inputting the relative distance, speed, and acceleration of each player in relation to each player on the opposing team. However, despite these efforts to improve the effectiveness of the CNN, the GNN is still better suited for this task. A typical CNN performs its convolutions on a fixed, grid-like window (Figure 1). Thus, the model gives greater weight to local neighbourhoods as it assumes data adjacent in the input are likely structured similarly in the field. However, given that players often change how they line up, their alignment on the field will rarely match the structure of the matrix. Consequently, the CNNs in this paper use a 1x1 kernel to avoid the incorrect assumption that adjacency on the grid and the field are related. This significantly limits the model's predictive power by forcing the CNN to process each player-player pair in isolation.

In contrast, the GNN represents each player as a node in a graph and forms relationships between nodes in the form of edges. This allows the GNN to describe the complex and often arbitrary shape of the players on a football field. Additionally, the unique architecture of the GNN enables it to be further tuned by adjusting the weight of specific edges, placing greater weight on certain connections. In particular, the GNN employed in this paper placed greater weights based on proximity and a multiplier on within-team connections, which was determined to enhance the effectiveness of the model through empirical testing. The use of edge weights impacts the message-passing process, wherein the edge

weight scales node embeddings of neighbours. Hence, neighbours with stronger edge weights influence the update of the node's embeddings more. By contrast, this would be much more difficult to implement into the CNN due to the lack of edges to demonstrate connections between players. Instead, interactions between players are interpreted through fixed arrangements, making player-player relationships much more difficult to adjust. The GNN's ability to model and optimize a relational graph allows it to retain contextual information across layers that would have been lost through transformations to a CNN's input matrix. Due to its better relational reasoning ability, the GNN can better adapt to shifts in alignment and generalize to new plays more effectively.

Although this paper categorizes coverage schemes into two labels, man and zone coverage, there are many variations of each that are employed by defensive coaching staff, which significantly affect defender alignment. Most coverage schemes can be classified into one of seven distinct classes: cover-0 man, cover-1 man, cover-2 man, cover-2 zone, cover-3 zone, cover-4 zone, and cover-6 zone. To understand why the CNN and GNN models both exhibited certain tendencies in their predictions, it is essential to first understand the distinctions between each coverage type.

The coverage types each include a number indicating the number of deep defenders present during that play. Zone coverage, to have their defenders cover different zones throughout the field, will often include more deep defenders who are responsible for covering deeper passes in zones further downfield. There are three types of man coverage, with the first being Cover-0 Man. Cover-0 involves significant risk as no deep defenders are employed. Instead, five defenders are all responsible for marking their respective eligible receiver while the remaining six players on the defensive team rush the quarterback in what is called a blitz. Similarly, Cover-1 Man involves only one deep defender, with the only difference being that five players will now rush the quarterback, also known as a dog. However, at times, defenses will instead sacrifice one rusher and rather have two defenders cover a receiver to ensure they have little chance of catching the ball. The final man coverage titled Cover-2 Man is a hybrid between man and zone coverage, making it often difficult to classify. In Cover-2 Man, five defenders are once more responsible for covering the eligible receivers through man coverage. However, unlike the other forms of man coverage, two safeties now cover the deep half of the field, providing help to the man defenders in the case of a deep ball.

In contrast, Cover-2 Zone employs two deep safeties with all underneath defenders playing zone coverage. While Cover-2 Zone appears similar to Cover-2 Man at snap, the key difference is the lack of man coverage for close balls and hence the absence of help defense for receivers downfield. The remaining zone coverages follow a similar pattern, with Cover-3 Zone employing three deep defenders and four underneath defenders, Cover-4

Zone (also referred to as quarters) using four deep defenders and three underneath defenders. Finally, Cover-6 Zone appears slightly different from the other forms of zone coverage, with the defense essentially splitting the field into two halves. On one half, it plays a Cover-4 Zone, employing four deep defenders, while on the other, it plays a Cover-2 Zone, utilizing two deep defenders. This split coverage allows the defense to scheme around specific receivers and to prevent deep balls from being thrown.

From analyzing the Big Data Bowl dataset, the majority class appears to be Cover-3 Zone. Hence, given the apparent overfitting in the CNN, it is reasonable that it predicts Cover-3 Zone at a significantly higher rate than the GNN (Figure 4). With 92 more correct predictions than the GNN on Cover-3 Zone plays, the results highlight that the CNN heavily relies on this class for its high test accuracy. Furthermore, the CNN outperforms the GNN on all other zone coverages, with the GNN rarely predicting zone coverage plays correctly that the CNN failed to classify. However, the GNN significantly outperforms the CNN on man coverage, with 70 correct classifications that the CNN was unable to predict. Even more interestingly, on Cover-2 Man plays, a coverage that is typically difficult to predict due to its similarity to Cover-2 Zone plays, particularly pre-snap, the CNN was rarely able to classify any plays that the GNN incorrectly labeled correctly. These results highlight the CNN's failure to classify rarer plays, often resulting in the model incorrectly classifying the play as zone coverage. In contrast, the GNN's architecture allows it to process much more complex patterns, making it more reliable for real-world defensive coverage recognition.

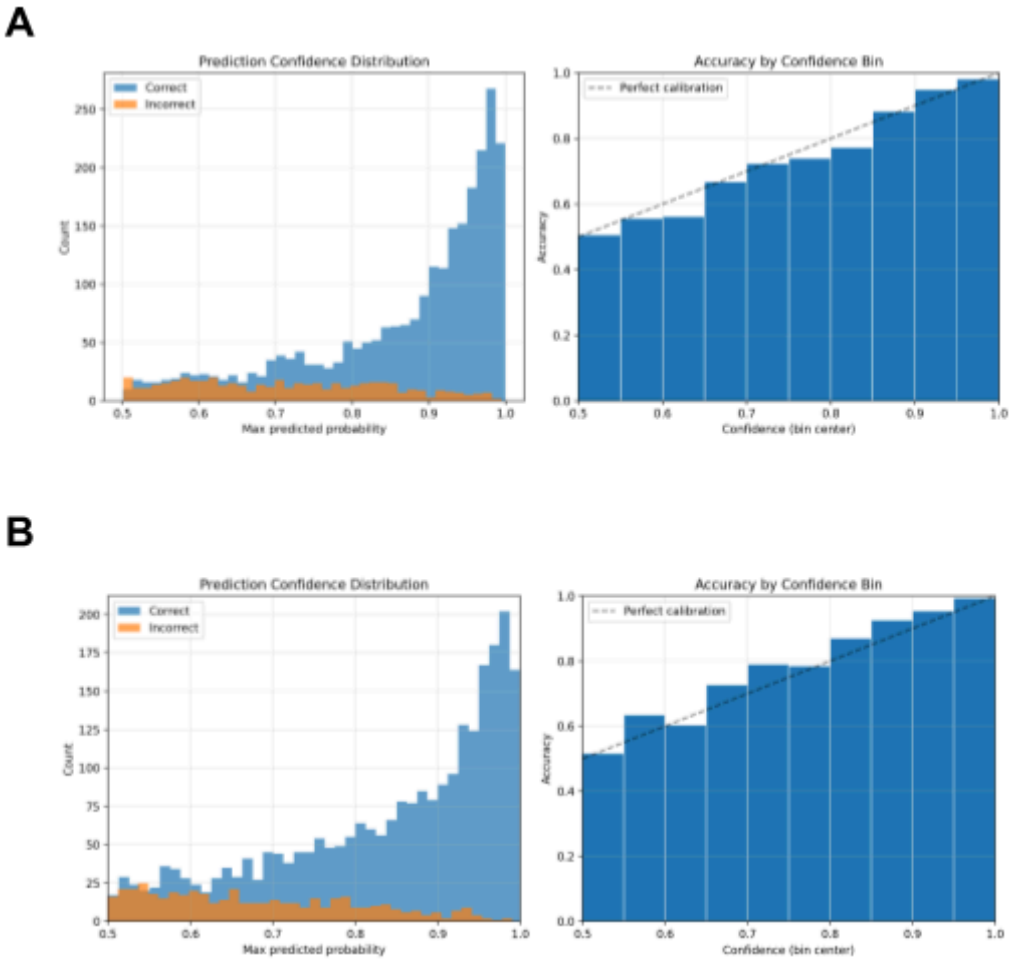
Conclusion and Future Directions:

This study evaluated the performance of a CNN and GNN for the pre-snap classification of NFL defensive coverage schemes using player tracking data from the 2025 Big Data Bowl. Overall, the CNN achieved a marginally higher test accuracy due to its bias towards predicting the major class, while the GNN outperformed the CNN with both a greater F1 and AUC score. The GNN's ability to model the field as a graph allowed it to maintain greater relational context than the CNN. This lack of signal decreased the confidence of the CNN in its predictions and led it to primarily predict zone coverage. These findings underscore the GNN's superior ability to analyze dynamic, non-grid environments such as American football, taking advantage of its graph-based architecture. Ultimately, the GNN proved to be more effective in making balanced predictions, supporting the idea that it is better suited for coverage scheme classification due to its ability to analyze connections and contextual relationships when making predictions (Khemani et al., 2024).

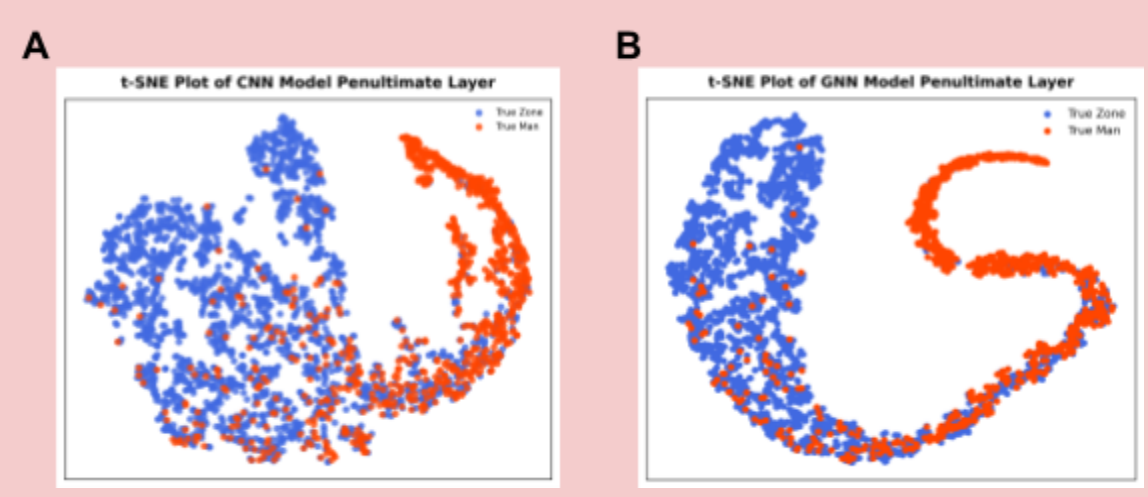
In future work, I plan to explore how incorporating temporal learning can impact both models. A limitation of the models used in this paper was their reliance on a single frame.

Defensive coverages are dynamic, with players' alignment, motion, and position constantly changing before the snap. Hence, the models used in this paper were limited by their inability to capture patterns introduced by the movement of players prior to the snap. Temporal learning achieves this by allowing the models to analyze multiple frames and predict based on how the data changes over time. Specifically, the use of recurrent or temporal GNNs could address the limitations caused by static frames by modeling how the relationships between players evolve across frames, rather than being constrained to singular snapshots. As a result, the model capture a more nuanced representation of what is occurring pre-snap and potentially detect new patterns by analyzing the sequential shifts in player positions. As our knowledge of football evolves, we must explore how data science can complement human coaching to enhance decision-making, enabling fans to experience sports at the highest level both physically and tactically.

Appendix:



[Figure 6: Panel A shows the count of correct and incorrect guesses relative to the model's predicted probability of success, along with the model's accuracy relative to their confidence in that guess for the GNN. Panel A shows the same diagrams for the CNN.]



[Figure 7: Panel A shows the t-SNE plot of the penultimate layer of the CNN model, while Panel B shows the t-SNE plot of the penultimate layer of the GNN model. In both models, red points illustrate man plays while blue points represent zone plays.]

References

Ajit, A., Acharya, K., & Samanta, A. (2020). A Review of Convolutional Neural Networks. *IEEE*, 1-5. https://doi.org/10.1109/ic-ETITE47903.2020.049

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016, July 21). Layer Normalization. *arXiv*. https://doi.org/10.48550/arXiv.1607.06450 (Non-peer-reviewed)

Fernandes, C. J., Yakubov, R., Li, Y., Prasad, A. K., & Chan, T. C.Y. (2019, October 26). Predicting plays in the National Football League. *Journal of Sports Analytics*, 6(1), 35-43. https://doi.org/10.3233/JSA-190348

Gholamalinezhad, H., & Khosravi, H. (2020, September 16). Pooling Methods in Deep Neural Networks, a Review. *arXiv*. https://doi.org/10.48550/arXiv.2009.07485 (Non-peer-reviewed)

Goyal, U. (2020, February). Leveraging Machine Learning to Predict Playcalling Tendencies in the NFL. *DSpace@MIT*. Retrieved August 14, 2025, from https://dspace.mit.edu/handle/1721.1/129909

The History of the Rules. (2025). NFL Football Operations. Retrieved August 8, 2025, from https://operations.nfl.com/the-rules/evolution-of-the-nfl-rules/ (Non-peer-reviewed)

Identifying significant features for Player Evaluation in NFL comparing ANNs and Traditional Models. (2021). *Arrow@TU Dublin*. https://doi.org/10.21427/EAC6-4R95

Khemani, B., Patil, S., Kotecha, K., & Tanwar, S. (2024, January 16). A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions. *Journal of Big Data*, *11*(18). https://doi.org/10.1186/s40537-023-00876-4

Kingston, C. (2023, May 12). Measuring Grit in NFL Cornerbacks using Statistical Analysis. https://dspace.mit.edu/bitstream/handle/1721.1/151676/kingston-ctk-meng-eecs-2023-thesis.pdf?sequence=1&isAllowed=y

Lopez, M., Bliss, T., Blake, A., Mooney, P., & Howard, A. (2024). *NFL Big Data Bowl 2025*. Kaggle. Retrieved August 14, 2025, from https://kaggle.com/competitions/nfl-big-data-bowl-2025 (Non-peer-reviewed)

McManus, C. (2025, March 17). Artificial Intelligence for better in-game NFL performance. *Scholarworks at WMU*. Retrieved August 14, 2025, from https://scholarworks.wmich.edu/cgi/viewcontent.cgi?article=4965&context=honors_theses

Ofeidis, I., Kiedanski, D., & Tassiulas, L. (2025, January 16). An Overview of the Data-Loader Landscape: Comparative Performance Analysis. *IEEE*, pp. 360–367. https://doi.org/10.1109/BigData62323.2024.10825421

Purwono, P., Ma'arif, A., Rahmaniar, W., Fathurrahman, H. I. K., Frisky, A. Z. K., & Haq, Q. M. u. (2023). Understanding of Convolutional Neural Network (CNN): A Review. *International Journal of Robotics and Control Systems*, 2(4), 739-748. https://doi.org/10.31763/ijrcs.v2i4.888

Quarterbacks: Finding Weakness and Strength in Zone vs. Man Coverage. (2023, August 1). Capital QB's. Retrieved August 14, 2025, from https://www.capitalqbs.com/quarterbacks-finding-weakness-and-strength-in-zone-vs-man-coverage/ (Non-peer-reviewed)

Sanchez-Lengeling, B., Reif, E., Pearce, A., & Wiltschko, A. B. (2021, September 2). A Gentle Introduction to Graph Neural Networks. *Distill*. https://doi.org/10.23915/distill.00033 (Non-peer-reviewed)

Song, H., Jazaery, M. A., Ding, H., Cheong, L. L., Jung, J., Band, M., Chi, M., & Bliss, T. (2023). Explainable Defense Coverage Classification in NFLGames using Deep Neural Networks. *Amazon ML Solutions Lab*. https://www.amazon.science/publications/explainable-defense-coverage-classification-in-nfl-games-using-deep-neural-networks

Taylor, C. (2020). Deep Learning for In-Game NFL Predictions. https://cs230.stanford.edu/projects_winter_2020/reports/32263160.pdf#page=1.33 (Non-peer-reviewed)

Grids to Graphs: GNNs improve NFL defensive coverage classification over CNNs

Abstract:

This paper investigates the use of predictive neural network models to classify NFL defensive coverages using player tracking data. I propose a novel solution using a graph neural network (GNN) as an alternative to the conventional use of convolutional neural networks (CNNs). Both model were compared based on their test accuracy, F1 score, and AUC. While the CNN reported a greater test accuracy (84.3%) than the GNN (83.7%), this was primarily due to its bias towards zone coverage, the majority class. In contrast, the GNN gave more balanced predictions with a greater F1 score (0.737 vs. 0.730) and AUC (0.920 vs. 0.904) than the CNN. The GNN was also more effective than the CNN when classifying less common coverage scheme variations. These findings suggest that GNNs are more suitable for NFL defensive coverage classification. This is because their unique architecture represents information as graphs, helping them understand player-player interactions.

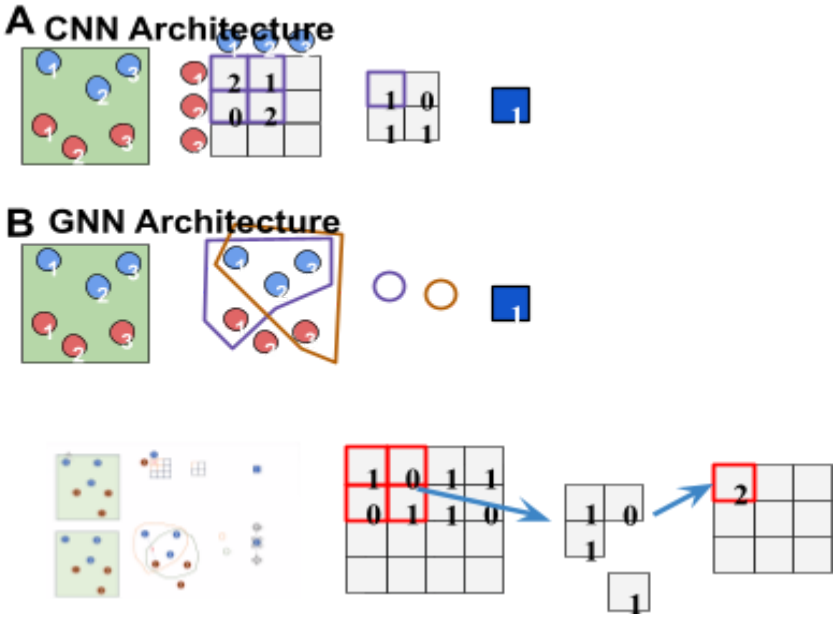
Introduction:

The National Football League (NFL) has long focused its analysis and changes on the offense to make the viewing experience more engaging. Per the NFL's Competition Committee, "if someone wants to accuse the National Football League of promoting offense to make the game more exciting, [the committee believes the league should plead guilty]" (NFL Football Operations, 2025). Hence, the recent rise in machine learning (ML)-powered football analytics has predominantly focused on offensive playcalling and performance (McManus, 2025; Goyal, 2020; Walsh, 2021; Fernandes et al., 2019; Taylor, 2020). On the other hand, considerably less analysis has been performed regarding the defensive end of the floor.

On defense, the coaching staff is responsible for positioning their players in specific formations, known as coverage schemes. Defensive coverage schemes determine the responsibilities of each player on the defensive team. Coverage schemes are generally one of two broad categories: man coverage and zone coverage. In man coverage, each defender is responsible for a certain offensive player throughout the play. In contrast, zone coverage requires defensive players to defend any players that enter a specific area on the field known as their "zone" (Kingston, 2023). Both coverage schemes are employed for vastly different use cases, with each having its own benefits. While zone coverage often minimizes big plays, man coverage leads to a tighter defense, placing greater pressure on the opposing quarterback to make an accurate throw (Capital QB's, 2023).

Determining which coverage scheme the defensive team will use helps coaches choose plays to increase their chances of scoring. As a result, the coaches of the defensive team often disguise their coverage scheme by making their players rotate. This can make play calling difficult for even the most skilled coaches. Hence, having a method of classifying coverage schemes pre-snap would let the offense anticipate and plan against defensive strategies used by the opposing team.

One way of classifying defensive coverage is by using predictive models. Currently, the most successful method of classifying defensive coverage schemes is convolutional neural networks (CNNs) (Song et al., 2023). CNNs are a form of artificial neural network that process data structured in grid-like matrices. They use convolutional layers (which can be thought of as a sliding window) to capture local patterns. Throughout the training process, the weights of the window are learned (Figure 1). This process of convolution is why CNNs are often used when processing images. Images have strong spatial locality (pixels within a proximity are highly correlated), meaning the ability to capture local patterns effectively allows CNNs to learn the complex structures of objects within images without becoming overwhelmed by the large quantity of pixels (Purwono et al., 2022).



[Figure 1: Panel A shows diagrams explaining the architecture of the CNN. The football field (green square) has three offensive players (blue) and three defensive players (red). The second illustration shows how the field is illustrated as a grid, with the values representing the distinct pairwise relative features. The purple square is then convoluted into one number in the following illustration before ultimately becoming a binary output. Panel B

shows a similar illustration of the football field, which then becomes represented as a graph. The graph is then convoluted in the following image, resulting in a binary output.]

Modern methods of applying CNNs to coverage scheme classification have found success in using relative data in conjunction with static data. Song et al. (2023) show that relying on static features alone significantly limits the predictive capabilities of a CNN. However, through conducting feature engineering to construct a matrix of pair-wise relative features between each member of the offensive team and the players on the defensive team, they were able to model complex player interactions to maintain greater spatial and relational context (Figure 1) (Song et al., 2023).

While Song et al. (2023) found success through the use of relative features, it must be noted that they increase the dimensionality of the data significantly due to the many static values that are repeated throughout the matrix. This introduces various concerns, including greater computational demands, a high risk of overfitting, and an unintentional emphasis placed on static features, as they are constantly repeated within each pairwise comparison. Additionally, CNNs likely struggle to anticipate shifts that could occur to players' positions later on in a play due to their inability to understand how each player relates to the others on the field.
In contrast, the use of a graph neural network (GNN) should theoretically be more effective due to its unique architecture. GNNs operate on graph-structured data, describing data through nodes, edges, and a global context. The graph is passed through a differentiable function, such as a multilayer perceptron (MLP), to process the data. In the case of binary classification, such as predicting coverage schemes, the prediction target is the global context. As such, the information within the nodes and edges must be passed through a pooling layer, creating a concatenated matrix that can be used for classification (Sanchez-Lengeling et al., 2021). This process of pooling information from the nodes and edges to make predictions about the global context makes GNNs particularly robust, as it allows the GNN to learn from the intricate relationships between players on the field. Hence, the use of a GNN should enhance the ability to classify defensive coverages. Consequently, this paper discusses the hypothesis that a GNN is more effective at NFL defensive coverage classification than a CNN.

Methods:

This paper will view coverage scheme classification as a binary classification problem between man and zone coverage. The study will be performed on the 2025 Big Data Bowl, a public dataset released annually by the NFL. The dataset includes tracking data for all 22 players on the field (11 offense and 11 defense) during passing plays, along with detailed play-level descriptions. Furthermore, the dataset includes contextual information for each

game as well as player-specific attributes such as build and composition (Lopez et al., 2024). The data is drawn from the first nine weeks of the 2022-23 NFL season and includes 3,023 plays.

Initially, the data is pre-processed, with any plays not labeled as either man or zone being filtered out. These plays are labeled as “other” in the dataset and constitute plays where either the line of scrimmage is within the endzone, meaning traditional forms of pass coverage are not employed, or during a quarterback kneel, which occurs when the knee of a quarterback touches the ground directly after the snap and thereby ends the play. Subsequently, the dataset is filtered such that only the 31st frame of each play remains. The data is recorded at 10 frames per second, meaning the 31st frame represents three seconds into the play. As no snaps occur before the 31st frame, it ensures that the data is pre-snap and hence applies to offenses scheming around the defensive coverage employed. Furthermore, given that the frame is three seconds into the play, it is presumed that most players are already in their coverage positions. Though ideally, a model should use all pre-snap frames, a singular frame was chosen due to computational limitations.

CNN (normalized)	GNN (normalized)
Distance from the line of scrimmage	Distance from the line of scrimmage
Position on the y-axis	Position on the y-axis
Speed in the x-direction	Speed in the x-direction
Speed in the y-direction	Speed in the y-direction
Speed	Speed
Acceleration in the x-direction	Acceleration in the x-direction
Acceleration in the y-direction	Acceleration in the y-direction
Acceleration	Acceleration
Orientation	Orientation
Relative speed in the x-direction	Nearest opponent distance
Relative speed in the y-direction	Distance to team centroid

Relative acceleration in the x-direction	Position (represented as 10 distinct binary channels representing the quarterback, running back, wide receiver, tight end, offensive lineman, defensive lineman, linebacker, cornerback, safety, and nickel)
Relative acceleration in the y-direction	
Relative position in the x-direction	
Relative position in the y-direction	

[Figure 2: The list of all the features used when training the CNN and GNN models. All listed features were either taken directly from or calculated with data found on the 2025 Big Data Bowl dataset and were all normalized for stability.]

The CNN was built following the architecture in Song et al. 2023. The model was trained on a 15x11x11 stacked matrix—with 15 features, 11 offensive players, and 11 defensive players (Figure 2). The data were then reshaped into batches of 64 plays to enhance the training time and generalization of the model (Ofeidis et al., 2024). The architecture of the CNN uses two convolutional layers developed to extract high-level representations while maintaining the spatial structure of the matrix. The first layer comprises three successive 1x1 convolutional layers. The use of a unit kernel enables the combination of all 15 feature values into a single value without blurring or merging with information from other cells. Hence, the output maintains its 11x11 spatial identity and arrangement while reducing the computational load (Ajit et al., 2020). Subsequently, there is a dropout layer that drops 30% of the data to minimize overfitting and help regularize the data. Following the dropout, the features are processed through a pooling layer that consists of average and max pooling. The purpose of implementing the pooling layer is to down-sample the feature maps following the convolutional layers to reduce computational cost and control overfitting (Gholamalinezhad & Khosravi, 2020). Specifically, the pooling layer collapses the dimension of offensive players, transforming the data into 128-d vectors for each defensive player. The average pooling captures the overall trend, while the max pooling finds the strongest signals. A weighted sum of both pooling values is taken (Figure 3).

$$x = (mean \times 0.7) + (max \times 0.3)$$

[Figure 3: Equation used when finding the weighted sum of the average and max pooling]

Following the pooling layer, the features are input into another convolutional layer consisting of three 1x1 convolutional layers, each followed by batch normalization to normalize along the channel dimension. Once more, the unit kernel allows for linear transformations of the channels, increasing the channels to 160 before ultimately outputting 96 channels. After the second convolutional layer, the data is transformed through a second pooling layer that collapses the defensive-player axis while implementing

the same weighted sum of average and max pooling. Hence, the output is a 96-d vector describing each batch. Finally, the data is transformed through a set of dense layers. The first dense layer employs a ReLU activation that maps the 96 features to 96 output dimensions, thereby introducing non-linearity for more complex feature representations. Subsequently, batch normalization is applied to ensure a standard distribution of the activations throughout the batch. Thereafter, a second dense layer is used, expanding the feature space from 96 to 256 dimensions, using a higher-dimensional space to capture more complex patterns and linearly separable relationships. After another batch normalization is applied to enhance stability and convergence. Subsequently, layer normalization is applied, stabilizing hidden state dynamics and improving training time (Ba et al., 2016). Finally, the last dense layer reduces the 256-dimensional normalized vector to two outputs, being the two coverage schemes, and applies a softmax activation to transform the logits into a probability distribution between the coverage schemes. This serves as the classifier, matching each sample to a class based on the learned feature representation from the previous stages of the model. Ultimately, the class with the greater probability is output as the model's prediction of the class.

The GNN was trained on a graph consisting of 22 nodes (11 offensive players and 11 defensive players). Each of these nodes contains a 21-dimensional feature vector. To construct the graph, edges are defined by both Euclidean distance and team membership. Edges are placed between all players within 12 yards of each other, with weights between players on the same team having a multiplier. The value of 12 yards between players and the multiplier for same-team allegiance were determined empirically to optimize the accuracy of the model. The graph is then transformed by an encoder that maps the 21-dimensional feature vectors onto a 64-dimensional space. This layer uses L2 regularization with a regularization parameter of 10^{-4}. Then, the data is stabilized through batch normalization, before introducing non-linearity through a ReLU activation, and employing a dropout layer to minimize overfitting.

The projected features are then processed through two convolutional layers. Node representations, the feature vector for each player, are then updated through weighted message passing, which gathers aggregate features from all adjacent nodes (weighted by the strength of that edge) and performs a linear transformation and a ReLU activation. The first graph layer outputs 64 channels, followed by batch normalization and a dropout layer. In comparison, the second outputs 48 channels and is also followed by batch normalization and a dropout layer.

After the second convolutional layer, the node embeddings of size 22 x 48 are transformed through a pooling layer. The pooling layer first performs average and max pooling on the 22 players in the plays, followed by applying the same operations to defensive and offensive

subgraphs separately. These processes yield six distinct 48-dimensional pooled vectors, which are concatenated into a 288-dimensional representation of the play. This 288-dimensional graph vector is then passed through a three-layer multi-layer perceptron serving as the classifier. Each hidden layer employs L2 regularization, batch normalization, ReLU activation, and dropout. The final layer applies a softmax activation function identical to the one used in the CNN to generate class probabilities, with the model ultimately predicting the class with the greater probability.

Both the CNN and GNN are trained with 5-fold cross-validation for 50 epochs, using early stopping and a learning rate decay to optimize the training. The criteria for the early stopping of both models are set such that if the validation accuracy does not increase for 10 epochs, the training stops early. Similarly, if the validation accuracy does not increase for three epochs, the learning rate is halved until it reaches below 0.000001, at which point the training is stopped early.

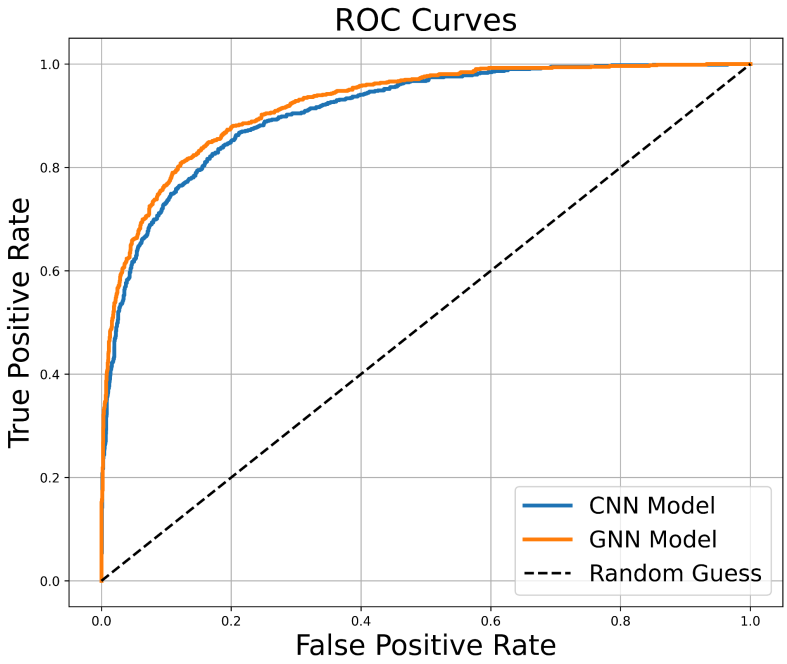
To compare model performance, the same 20% of the dataset (3023 plays) was held out during model training. To evaluate the performance, I looked at the AUC, F1 score, and test accuracy of both models.

Results:

To evaluate the performance of the models, I analyzed the test accuracy, F1 score, and AUC-ROC (area under the Receiver Operating Characteristic curve). Ultimately, the GNN had a greater AUC and F1 score; however, the CNN had a greater test accuracy. The values for the AUC, test accuracy, and F1 score are displayed in Figure 4, while the AUC-ROC is illustrated in both Figures 5:

	Test Accuracy	F1 Score	AUC
GNN Model	83.7%	0.737	0.920
CNN Model	84.3%	0.730	0.904

[Figure 4: Table of the test accuracy, AUC, and F1 score for the GNN and CNN models. Each value is listed to three significant figures.]

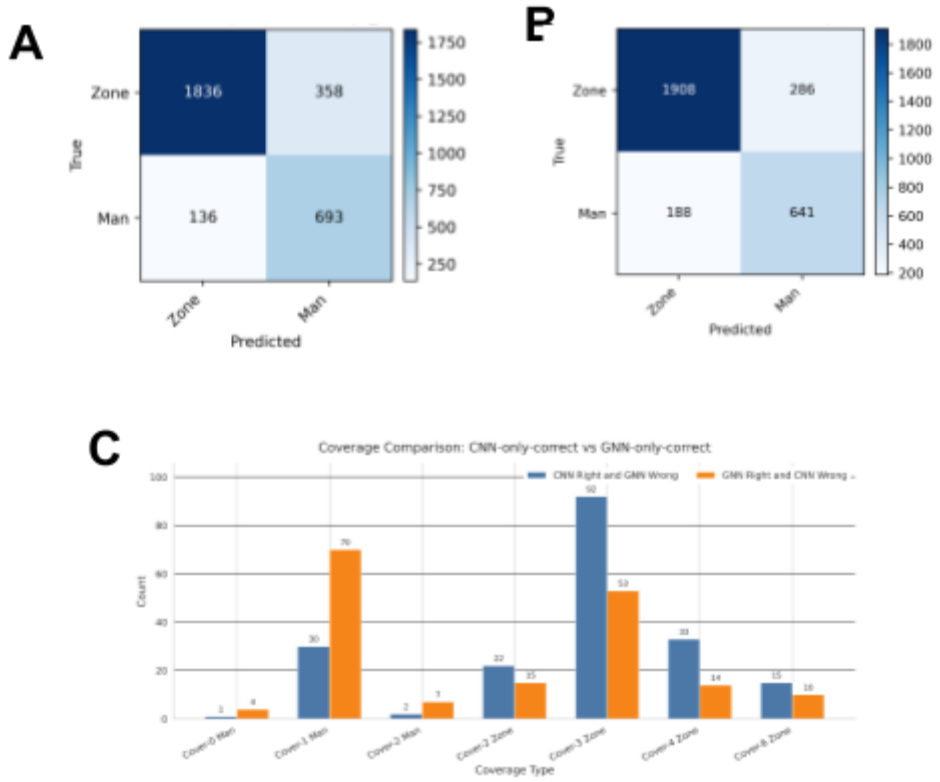


[Figure 5: AUC-ROC of the CNN and GNN models. The blue line represents the CNN model, while the orange represents the GNN. The dotted black line is a baseline of 50% accuracy.]

The GNN showed an improvement over the CNN model, though both outperformed the random classifier, with the GNN having an AUC of 0.923 compared to the CNN's 0.910.

Discussion:

The results of this study support the hypothesis that the use of a GNN is more effective than a CNN in classifying NFL defensive coverage schemes when using player tracking data. While the CNN achieved a marginally greater test accuracy, further analysis shows that this was primarily driven by the CNN's bias towards the majority class (zone coverage), leading to a significantly worse performance than the GNN on man plays (Figure 6). In contrast, the GNN achieved a greater AUC and F1 score, indicating that it outperformed the CNN with stronger generalization and a more balanced classification performance.



[Figure 6: Panel A is the confusion matrix of the GNN labeled with the true values and the predicted values by the mode, while Panel B illustrates the confusion matrix of the CNN model. The color represents the frequency of that prediction. Panel C shows the differences in correct predictions between the models depending on the specific type of zone or man coverage.]

The superior performance of the GNN can be explained by the fundamental differences between the two models' architectures. CNNs process features as fixed, grid-like matrices. This makes them well-equipped when analyzing spatially consistent data such as images. However, football plays cannot be evaluated by simply analyzing each player on the field independently. Rather, it is essential to also understand how players on the field interact with each other. Hence, the architecture in this paper adapted the CNN to consider relational context, inputting the relative distance, speed, and acceleration of each player to that of each player on the opposing team. However, despite these efforts to improve the effectiveness of the CNN, the GNN is still better suited for this task. A typical CNN performs its convolutions on a fixed, grid-like window (Figure 1). Thus, the model gives greater weight to local neighbourhoods as it assumes data adjacent in the input are likely structured similarly on the field. However, given that players often change how they line

up, their alignment on the field will rarely match the structure of the matrix. Consequently, the CNNs in this paper use a 1x1 kernel to avoid the incorrect assumption that adjacency on the grid and the field are related. This significantly limits the model's predictive power by forcing the CNN to process each player-player pair in isolation.

In contrast, the GNN represents each player as a node in a graph and forms relationships between nodes in the form of edges. This allows the GNN to describe the complex and often arbitrary shape of the players on a football field. Additionally, the unique architecture of the GNN enables it to be further tuned by adjusting the weight of specific edges, placing greater weight on certain connections. In particular, the GNN employed in this paper placed greater weights based on proximity and a multiplier on within-team connections, which was determined to enhance the effectiveness of the model through empirical testing. The use of edge weights impacts the message-passing process, wherein the edge weight scales node embeddings of neighbours. Hence, neighbours with stronger edge weights influence the update of the node's embeddings more. By contrast, this would be much more difficult to implement into the CNN due to the lack of edges to demonstrate connections between players. Instead, interactions between players are interpreted through fixed arrangements, making player-player relationships much more difficult to adjust. The GNN's ability to model and optimize a relational graph allows it to retain contextual information across layers that would have been lost through transformations to a CNN's input matrix. Due to its better relational reasoning ability, the GNN can better adapt to shifts in alignment and generalize to new plays more effectively.

Though this paper groups coverage schemes into two labels, man and zone coverage, there are many variations of each that are employed by defensive coaching staff that affect defender alignment significantly. Most coverage schemes can be classified into seven distinct classes: cover-0 man, cover-1 man, cover-2 man, cover-2 zone, cover-3 zone, cover-4 zone, and cover-6 zone. To understand why the CNN and GNN both showed certain tendencies in their predictions, it is essential first to understand the distinctions between each coverage type.

The coverage types each include a number indicating the number of deep defenders present during that play. Zone coverage, to have their defenders cover different zones throughout the field, will often include more deep defenders who are responsible for covering deeper passes in zones further downfield. There are three types of man defense, with the first of which being Cover-0 Man. Cover-0 involves significant risk as no deep defenders are employed. Instead, five defenders are all responsible for marking their respective eligible receiver while the remaining six players on the defensive team rush the quarterback in what is called a blitz, placing pressure with the hope of a sack. Similarly, Cover-1 Man involves only one deep defender, with the only difference being that five

players will now rush the quarterback, also known as a dog. However, at times, defenses will instead sacrifice one rusher and rather have two defenders cover a receiver to ensure they have little chance of receiving the ball. The final man coverage titled Cover-2 Man is a hybrid between man and zone coverage, making it often difficult to classify. In Cover-2 Man, five defenders are once more responsible for covering the eligible receivers through man coverage. However, unlike the other forms of man coverage, two safeties now cover the deep half of the field, providing help to the man defenders in the case of a deep ball.

In contrast, Cover-2 Zone employs two deep safeties with all underneath defenders playing zone coverage. While Cover-2 Zone appears similar to Cover-2 Man at snap, the key difference is the lack of man coverage for close balls and hence the absence of help defense for receivers downfield. The remaining zone coverages follow a similar pattern, with Cover-3 Zone employing three deep defenders and four underneath defenders, Cover-4 Zone (also referred to as quarters) using four deep defenders and three underneath defenders. Finally, Cover-6 Zone appears slightly different from the other forms of zone coverage, with the defense essentially splitting the field into two halves, playing a Cover-4 Zone on one half with four deep defenders, and a Cover-2 Zone on the other, employing two deep defenders. This split coverage allows the defense to scheme around specific receivers and to prevent deep balls from being thrown.

From analyzing the Big Data Bowl dataset, the majority class appears to be Cover-3 Zone. Hence, given the overfitting apparent in the CNN, it makes sense that it predicts Cover-3 Zone at a significantly higher rate than the GNN (Figure 6). With 92 more correct predictions than the GNN on Cover-3 Zone plays, the results highlight that the CNN heavily relies on this class for its high test accuracy. Furthermore, the CNN outperforms the GNN on all other zone coverages, with the GNN rarely predicting zone coverage plays correctly that the CNN failed to classify. However, the GNN severely outperforms the CNN on man coverage, with 70 correct classifications that the CNN was unable to predict. Even more interestingly, on Cover-2 Man plays, a coverage that is typically difficult to predict due to its similarity to Cover-2 Zone plays, particularly pre-snap, the CNN was rarely able to correctly classify any plays that the GNN incorrectly labeled. These results highlight the CNN's failure to classify rarer plays, often resulting in the model incorrectly classifying the play as zone coverage. In contrast, the GNN's architecture allows it to process much more complex patterns, making it more reliable for real-world defensive coverage recognition.

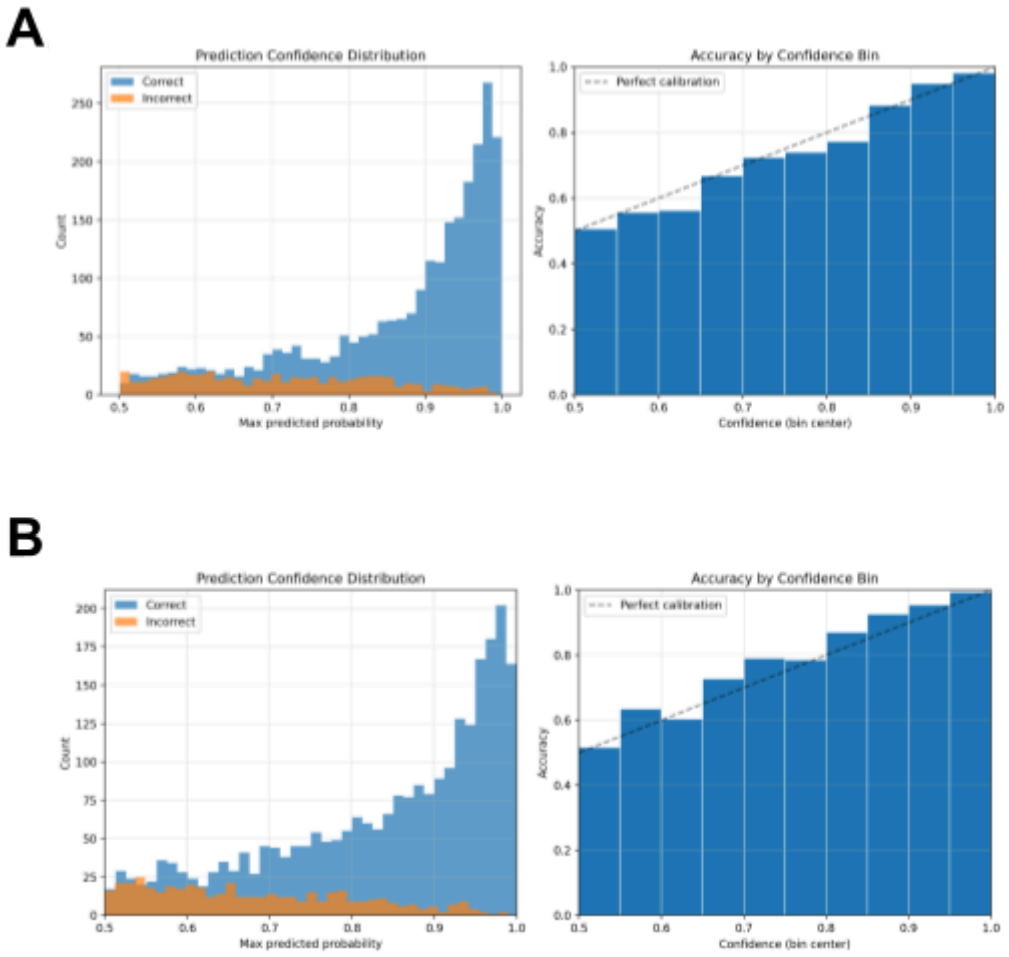
Conclusion and Future Directions:

This study evaluated the performance of a CNN and GNN for the pre-snap classification of NFL defensive coverage schemes using player tracking data from the 2025 Big Data Bowl.

Overall, the CNN achieved a marginally higher test accuracy due to its bias towards predicting the major class, while the GNN outperformed the CNN with both a greater F1 and AUC score. The GNN's ability to model the field as a graph allowed it to maintain greater relational context than the CNN struggled to capture. This lack of signal decreased the confidence of the CNN in its predictions and led to it primarily predicting zone coverage. These findings underscore the GNN's superior ability to analyze dynamic, non-grid environments such as American football, taking advantage of its graph-based architecture. Ultimately, the GNN showed to make more balanced predictions, supporting the idea that it is more suitable for coverage scheme classification due to its ability to analyze connections and contextual relationships when making predictions (Khemani et al., 2024).

In future work, I plan to explore how incorporating temporal learning can impact both models. A limitation of the models used in this paper was their reliance on a single frame. Defensive coverages are dynamic, with players' alignment, motion, and position constantly changing before the snap. Hence, the models used in this paper were both limited by their inability to capture patterns introduced by the movement of players pre-snap. Temporal learning achieves this by allowing the models to analyze multiple frames and predict based on how the data changes over time. As our knowledge of football develops, we must explore how data science can complement human coaching to enrich decision-making, allowing fans to witness sports at the highest level both physically and tactically.

Appendix:



[Figure 7: Panel A shows the count of correct and incorrect guesses relative to the model's predicted probability of success, along with the model's accuracy relative to their confidence in that guess for the GNN. Panel A shows the same diagrams for the CNN.]

References

Ajit, A., Acharya, K., & Samanta, A. (2020). A Review of Convolutional Neural Networks. *IEEE*, 1-5. https://doi.org/10.1109/ic-ETITE47903.2020.049

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016, July 21). Layer Normalization. *arXiv*. https://doi.org/10.48550/arXiv.1607.06450

Fernandes, C. J., Yakubov, R., Li, Y., Prasad, A. K., & Chan, T. C.Y. (2019, October 26). Predicting plays in the National Football League. *Journal of Sports Analytics*, 6(1), 35-43. https://doi.org/10.3233/JSA-190348

Gholamalinezhad, H., & Khosravi, H. (2020, September 16). Pooling Methods in Deep Neural Networks, a Review. *arXiv*. https://doi.org/10.48550/arXiv.2009.07485

Goyal, U. (2020, February). Leveraging Machine Learning to Predict Playcalling Tendencies in the NFL. *DSpace@MIT*. Retrieved August 14, 2025, from https://dspace.mit.edu/handle/1721.1/129909

The History of the Rules. (2025). NFL Football Operations. Retrieved August 8, 2025, from https://operations.nfl.com/the-rules/evolution-of-the-nfl-rules/

Identifying significant features for Player Evaluation in NFL comparing ANNs and Traditional Models. (2021). *Arrow@TU Dublin*. https://doi.org/10.21427/EAC6-4R95

Khemani, B., Patil, S., Kotecha, K., & Tanwar, S. (2024, January 16). A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions. *Journal of Big Data*, 11(18). https://doi.org/10.1186/s40537-023-00876-4

Kingston, C. (2023, May 12). Measuring Grit in NFL Cornerbacks using Statistical Analysis.

https://dspace.mit.edu/bitstream/handle/1721.1/151676/kingston-ctk-meng-eecs-2023-thesis.pdf?sequence=1&isAllowed=y

Lopez, M., Bliss, T., Blake, A., Mooney, P., & Howard, A. (2024). *NFL Big Data Bowl 2025*. Kaggle. Retrieved August 14, 2025, from https://kaggle.com/competitions/nfl-big-data-bowl-2025

McManus, C. (2025, March 17). Artificial Intelligence for better in-game NFL performance. *Scholarworks at WMU*. Retrieved August 14, 2025, from https://scholarworks.wmich.edu/cgi/viewcontent.cgi?article=4965&context=honors_theses

Ofeidis, I., Kiedanski, D., & Tassiulas, L. (2025, January 16). An Overview of the Data-Loader Landscape: Comparative Performance Analysis. *IEEE*, pp. 360–367. https://doi.org/10.1109/BigData62323.2024.10825421

Purwono, P., Ma'arif, A., Rahmaniar, W., Fathurrahman, H. I. K., Frisky, A. Z. K., & Haq, Q. M. u. (2023). Understanding of Convolutional Neural Network (CNN): A Review. *International Journal of Robotics and Control Systems*, 2(4), 739-748. https://doi.org/10.31763/ijrcs.v2i4.888

Quarterbacks: Finding Weakness and Strength in Zone vs. Man Coverage. (2023, August 1). Capital QB's. Retrieved August 14, 2025, from https://www.capitalqbs.com/quarterbacks-finding-weakness-and-strength-in-zone-vs-man-coverage/

Sanchez-Lengeling, B., Reif, E., Pearce, A., & Wiltschko, A. B. (2021, September 2). A Gentle Introduction to Graph Neural Networks. *Distill*. https://doi.org/10.23915/distill.00033

Song, H., Jazaery, M. A., Ding, H., Cheong, L. L., Jung, J., Band, M., Chi, M., & Bliss, T. (2023). Explainable Defense Coverage Classification in NFLGames using Deep Neural Networks. *Amazon ML Solutions Lab*. https://www.amazon.science/publications/explainable-defense-coverage-classification-in-nfl-games-using-deep-neural-networks

Taylor, C. (2020). Deep Learning for In-Game NFL Predictions. https://cs230.stanford.edu/projects_winter_2020/reports/32263160.pdf#page=1.33

Grids to Graphs: GNNs improve NFL defensive coverage classification over CNNs

Abstract:

This paper investigates the application of predictive neural network models to classify NFL defensive coverages using player tracking data. I propose a novel solution using a graph neural network (GNN) as an alternative to the conventional use of convolutional neural networks (CNNs). Both models were compared based on their test accuracy, F1 score, and AUC. While the CNN reported a greater test accuracy (84.3%) than the GNN (83.7%), this was primarily due to its bias towards zone coverage, the majority class. In contrast, the GNN yielded more balanced predictions with a greater F1 score (0.737 vs. 0.730) and AUC (0.920 vs. 0.904) than the CNN. The GNN was also more effective than the CNN when classifying less common coverage scheme variations. These findings suggest that GNNs are better suited for classifying NFL defensive coverages. This is because their unique architecture represents information as graphs, helping them better understand player-player interactions.

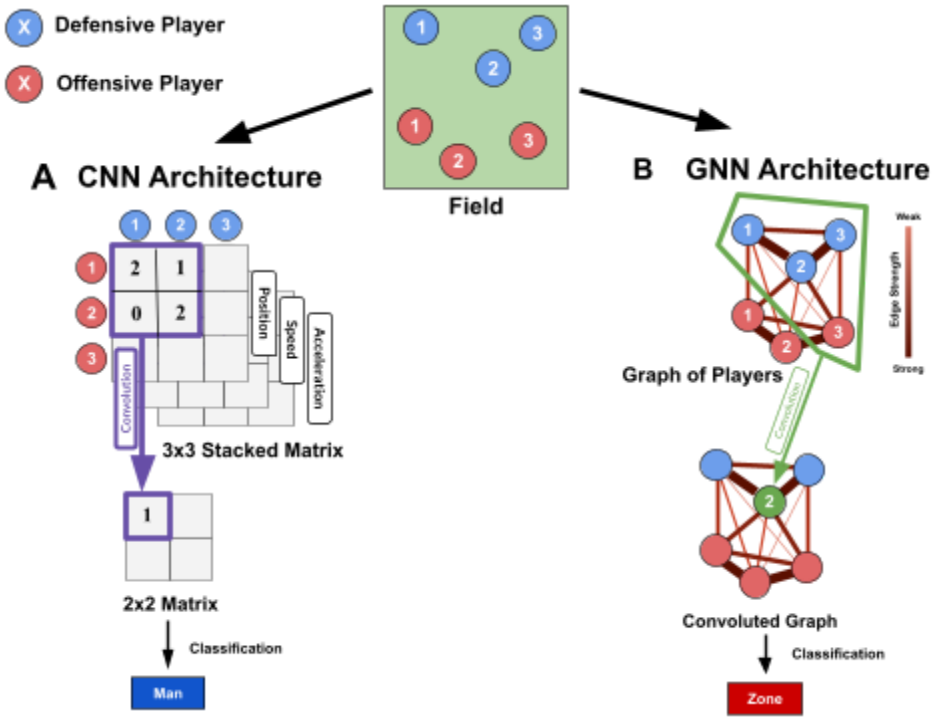
Introduction:

The National Football League (NFL) has long focused its analysis and changes towards offense to make the viewing experience more engaging. Per the NFL's Competition Committee, "if someone wants to accuse the National Football League of promoting offense to make the game more exciting, [the committee believes the league should plead guilty]" (NFL Football Operations, 2025). Hence, the recent rise in machine learning (ML)-powered football analytics has predominantly focused on offensive playcalling and performance (McManus, 2025; Goyal, 2020; Walsh, 2021; Fernandes et al., 2019; Taylor, 2020). On the other hand, considerably less analysis has been performed regarding the defensive end of the floor.

On defense, the coaching staff is responsible for positioning their players in specific formations, known as coverage schemes. Defensive coverage schemes determine the responsibilities of each player on the defensive team. Coverage schemes are categorized into two broad types: man coverage and zone coverage. In man coverage, each defender is responsible for a specific offensive player throughout the play. In contrast, zone coverage requires defensive players to defend any players that enter a particular area on the field known as their "zone" (Kingston, 2023). Both coverage schemes are employed for vastly different use cases, with each having its own benefits. While zone coverage often minimizes big plays, man coverage leads to a tighter defense, placing greater pressure on the opposing quarterback to make an accurate throw (Capital QB's, 2023).

Determining which coverage scheme the defensive team will use helps the coaching staff of the offensive team choose plays to increase their chances of scoring. As a result, the coaches of the defensive team often disguise their coverage scheme by making their players rotate following the snap—the start of a play in football. This can make play calling difficult for even the most skilled coaches. Hence, having a method of classifying coverage schemes pre-snap would enable the offense to anticipate and plan against the defensive strategies used by the opposing team.

One way of classifying defensive coverage is by using predictive models. Currently, the most successful method of classifying defensive coverage schemes is convolutional neural networks (CNNs) (Song et al., 2023). CNNs are a form of artificial neural network that process data structured in grid-like matrices. They use convolutional layers (which can be thought of as a sliding window) to capture local patterns. Throughout the training process, the weights of the window are learned (Figure 1). This process of convolution is why CNNs are often used when processing images. Images have strong spatial locality (pixels within a proximity are highly correlated), meaning the ability to capture local patterns effectively allows CNNs to learn the complex structures of objects within images without becoming overwhelmed by the large quantity of pixels (Purwono et al., 2022).



[Figure 1: Panel A shows diagrams explaining the architecture of the CNN. The football field (green square) has three offensive players (blue) and three defensive players (red). The first

illustration in panel A illustrates how the field is represented as a stacked matrix, with the values representing the distinct pairwise relative features. The purple square is then convoluted into one number in the following illustration before ultimately becoming a binary output. Panel B presents an illustration of the players in the form of a graph. The lines between the circles (nodes) represent the connections between players (edges). The thickness and color of the edges represent the strength of these connections. Similar to panel A, the values of the graph are convoluted into a new graph. Ultimately, the GNN classifies into a binary output.

Modern methods of applying CNNs to coverage scheme classification have found success in using relative data in conjunction with static data. Song et al. (2023) demonstrate that relying solely on static features significantly limits the predictive capabilities of a CNN. However, by conducting feature engineering to construct a matrix of pairwise relative features between each member of the offensive team and the players on the defensive team, they were able to model complex player interactions, thereby maintaining greater spatial and relational context (Figure 1) (Song et al., 2023).

While Song et al. (2023) found success through the use of relative features, it must be noted that they increase the dimensionality of the data significantly due to the many static values that are repeated throughout the matrix. This introduces various concerns, including greater computational demands, a high risk of overfitting, and an unintentional emphasis placed on static features, as they are constantly repeated within each pairwise comparison. Additionally, CNNs likely struggle to anticipate shifts that could occur to players' positions throughout a play due to their inability to understand how each player relates to the others on the field.

In contrast, the use of a graph neural network (GNN) should theoretically be more effective due to its unique architecture. GNNs operate on graph-structured data, describing data through nodes, edges, and a global context. The graph is passed through a differentiable function, such as a multilayer perceptron (MLP), to process the data. In the case of binary classification, such as predicting coverage schemes, the prediction target is the global context. As such, the information within the nodes and edges must be passed through a pooling layer, creating a concatenated matrix that can be used for classification (Sanchez-Lengeling et al., 2021). This process of pooling information from the nodes and edges to make predictions about the global context makes GNNs particularly robust, as it allows the GNN to learn from the intricate relationships between players on the field. Hence, the use of a GNN should enhance the ability to classify defensive coverages.

Previous works, including those of Song et al. (2023), have primarily focused on optimizing the effectiveness of the CNN model by incorporating methods such as temporal learning to

maximize accuracy. However, this paper will instead focus on comparing the performance of CNN models to that of GNN models in an attempt to develop a novel classification method for NFL coverage schemes. Consequently, this paper discusses the hypothesis that a GNN is more effective at NFL defensive coverage classification than a CNN.

Methods:

In football, man and zone are the two primary labels used to classify defensive coverages. Therefore, differentiating between man and zone coverages is both relevant and sufficient to identify NFL defensive coverages. As there are two primary outputs, this paper will view coverage scheme classification as a binary classification problem between man and zone coverage. The study will be performed on the 2025 Big Data Bowl, a public dataset released annually by the NFL. The dataset includes tracking data for all 22 players on the field (11 offense and 11 defense) during passing plays, along with detailed play-level descriptions. Furthermore, the dataset includes contextual information for each game as well as player-specific attributes such as build and composition (Lopez et al., 2024). The data is drawn from the first nine weeks of the 2022-23 NFL season and includes 3,023 plays.

Initially, the data is pre-processed, with any plays not labeled as either man or zone being filtered out. These plays are labeled as “other” in the dataset and constitute plays where either the line of scrimmage is within the endzone, meaning traditional forms of pass coverage are not employed, or during a quarterback kneel, which is a special form of play that occurs when the knee of a quarterback touches the ground directly after the snap and thereby immediately ends the play for strategic reasons. Subsequently, the dataset is filtered such that only the 31st frame of each play remains. The data is recorded at 10 frames per second, meaning the 31st frame represents three seconds into the play. As no snaps occur before the 31st frame, it ensures that the data is pre-snap and hence applies to offenses scheming around the defensive coverage employed. Furthermore, given that the frame is three seconds into the play, it is presumed that most players are already in their coverage positions. Though ideally, a model should use all pre-snap frames, a singular frame was chosen due to computational limitations.

CNN (normalized)	GNN (normalized)
Distance from the line of scrimmage	Distance from the line of scrimmage
Position on the y-axis	Position on the y-axis
Speed in the x-direction	Speed in the x-direction
Speed in the y-direction	Speed in the y-direction
Speed	Speed
Acceleration in the x-direction	Acceleration in the x-direction
Acceleration in the y-direction	Acceleration in the y-direction
Acceleration	Acceleration
Orientation	Orientation
Relative speed in the x-direction	Nearest opponent distance
Relative speed in the y-direction	Distance to team centroid
Relative acceleration in the x-direction	Position (represented as 10 distinct binary channels representing the quarterback, running back, wide receiver, tight end, offensive lineman, defensive lineman, linebacker, cornerback, safety, and nickel)
Relative acceleration in the y-direction	
Relative position in the x-direction	
Relative position in the y-direction	

[Table 1: The list of all the features used when training the CNN and GNN models. All listed features were either taken directly from or calculated with data found on the 2025 Big Data Bowl dataset and were all normalized for stability.

Although some of these features may appear redundant, such as having speed, along with the x and y components of speed as separate features, they are provided intentionally. This allows the models to choose the representation that they find most effective, and consequently, which features they will propagate through their layers.

The CNN was built following the architecture described in Song et al. (2023). The model was trained on a 15x11x11 stacked matrix—with 15 features, 11 offensive players, and 11 defensive players (Table 1). The data were then reshaped into batches of 64 plays to enhance the training time and generalization of the model (Ofeidis et al., 2024). The architecture of the CNN utilizes two convolutional layers, designed to extract high-level representations while preserving the spatial structure of the matrix. The first layer consists of three successive 1x1 convolutional layers. The use of a unit kernel enables the combination of all 15 feature values into a single value without blurring or merging with information from other cells. Hence, the output maintains its 11x11 spatial identity and arrangement while reducing the computational load (Ajit et al., 2020). Subsequently, a dropout layer is introduced to drop 30% of the data, thereby minimizing overfitting and regularizing the data. Following the dropout, the features are processed through a pooling layer that consists of average and max pooling. The purpose of implementing the pooling layer is to down-sample the feature maps following the convolutional layers to reduce computational cost and control overfitting (Gholamalinezhad & Khosravi, 2020). Specifically, the pooling layer collapses the dimension of offensive players, transforming the data into 128-d vectors for each defensive player. Average pooling captures the overall trend, while max pooling identifies the strongest signals. A weighted sum of both pooling values is taken (Equation 1).

$$x = (mean \times 0.7) + (max \times 0.3) \qquad [1]$$

Following the pooling layer, the features are input into another convolutional layer consisting of three 1x1 convolutional layers, each followed by batch normalization to normalize along the channel dimension. Once more, the unit kernel allows for linear transformations of the channels, increasing the channels to 160 before ultimately outputting 96 channels. After the second convolutional layer, the data is transformed through a second pooling layer that collapses the defensive-player axis while implementing the same weighted sum of average and max pooling. Hence, the output is a 96-d vector describing each batch. Finally, the data is transformed through a set of dense layers. The first dense layer employs a ReLU activation that maps the 96 features to 96 output dimensions, thereby introducing non-linearity for more complex feature representations. Subsequently, batch normalization is applied to ensure a standard distribution of the activations throughout the batch. Thereafter, a second dense layer is employed, expanding the feature space from 96 to 256 dimensions, thereby utilizing a higher-dimensional space to capture more complex patterns and linearly separable relationships. After another batch normalization is applied to enhance stability and convergence. Subsequently, layer normalization is applied, which stabilizes the dynamics of the hidden state and improves training time (Ba et al., 2016). Finally, the last dense layer reduces the 256-dimensional normalized vector to two outputs, being the two coverage schemes, and applies a softmax

activation to transform the logits into a probability distribution between the coverage schemes. This serves as the classifier, matching each sample to a class based on the learned feature representation from the previous stages of the model. Ultimately, the class with the greater probability is output as the model's prediction of the class (Figure 2).

The GNN was trained on a graph consisting of 22 nodes (11 offensive players and 11 defensive players). Each of these nodes contains a 21-dimensional feature vector. To construct the graph, edges are defined by both Euclidean distance and team membership. Edges are placed between all players within 12 yards of each other, with weights between players on the same team having a multiplier. The value of 12 yards between players and the multiplier for same-team allegiance were determined empirically to optimize the model's accuracy. The graph is then transformed by an encoder that maps the 21-dimensional feature vectors onto a 64-dimensional space. This layer uses L2 regularization with a regularization parameter of 10^{-4}. Then, the data is stabilized through batch normalization, before introducing non-linearity through a ReLU activation, and employing a dropout layer to minimize overfitting.

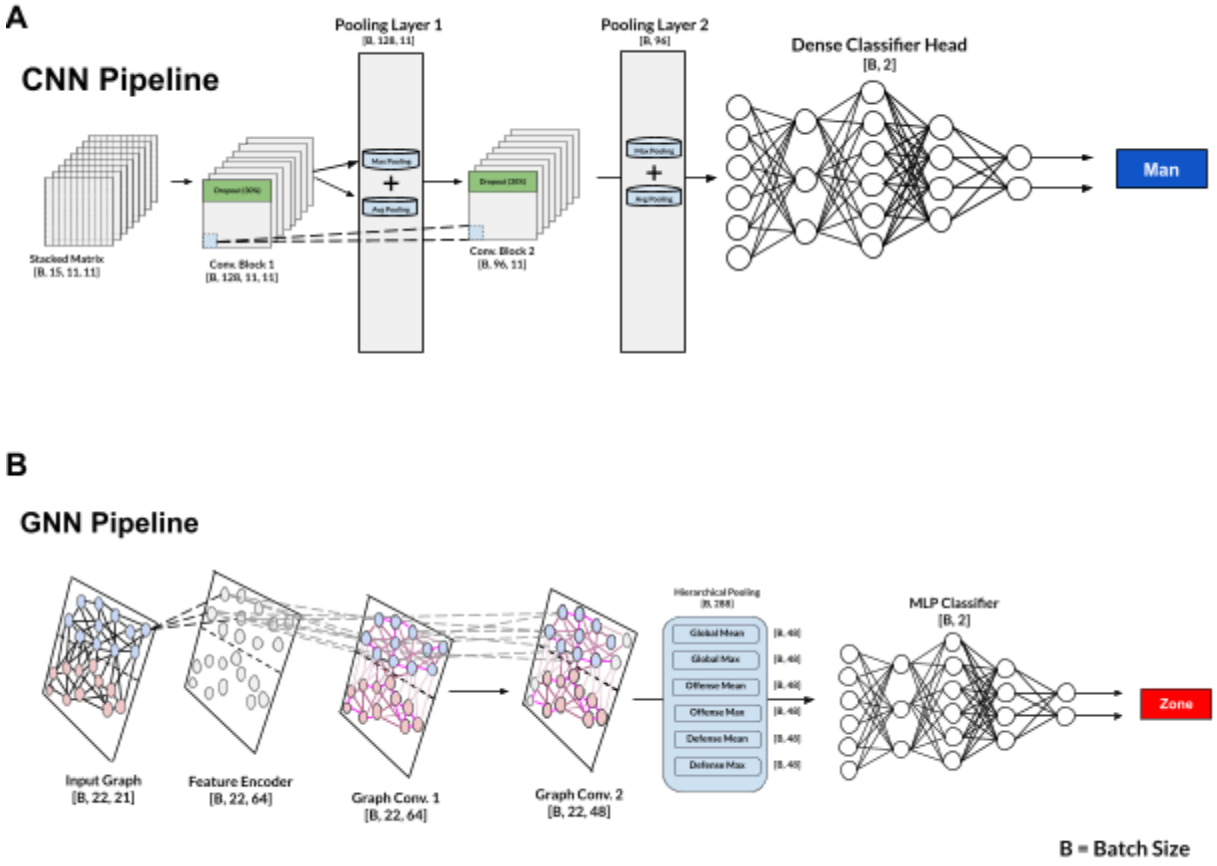
The projected features are then processed through two convolutional layers. Node representations, the feature vectors for each player, are then updated through weighted message passing, which gathers aggregate features from all adjacent nodes (weighted by the strength of that edge) and performs a linear transformation followed by a ReLU activation. The first graph layer outputs 64 channels, followed by batch normalization and a dropout layer. In comparison, the second outputs 48 channels and is also followed by batch normalization and a dropout layer.

After the second convolutional layer, the node embeddings, which are 22 x 48 in size, are transformed through a pooling layer. The pooling layer first performs average and max pooling on the 22 players in the plays, followed by applying the same operations to defensive and offensive subgraphs separately. These processes yield six distinct 48-dimensional pooled vectors, which are concatenated into a 288-dimensional representation of the play. This 288-dimensional graph vector is then passed through a three-layer multi-layer perceptron serving as the classifier. Each hidden layer employs L2 regularization, batch normalization, ReLU activation, and dropout. The final layer applies a softmax activation function identical to the one used in the CNN to generate class probabilities, with the model ultimately predicting the class with the greater probability (Figure 2)

Both the CNN and GNN are trained with 5-fold cross-validation for 50 epochs, using early stopping and a learning rate decay to optimize the training. The criteria for the early stopping of both models are set such that if the validation accuracy does not increase for 10

epochs, the training stops early. Similarly, if the validation accuracy does not increase for three epochs, the learning rate is halved until it reaches below 0.000001, at which point the training is stopped early.

To compare model performance, the same 20% of the dataset (3023 plays) was held out during model training. To evaluate the performance, I looked at the AUC, F1 score, and test accuracy of both models.



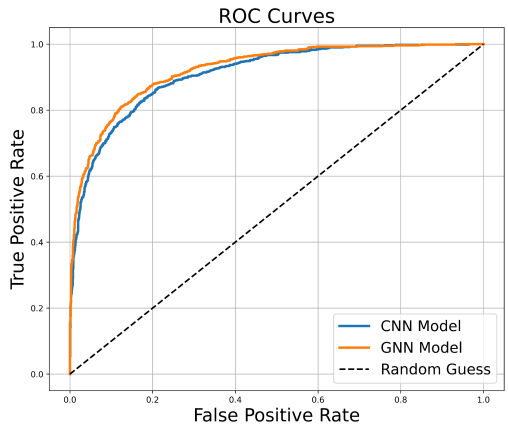
[Figure 2: The above diagram includes two panels illustrating the pipeline of the CNN and GNN models, respectively. Panel A depicts the CNN pipeline, starting with a stacked matrix as the input. The data are then transformed through two convolutional blocks, each followed by a pooling layer consisting of both max and average pooling. Finally, the data goes through the dense classifier head, which outputs a binary result indicating either man or zone coverage. Panel B shows the GNN pipeline. The initial input graph is transformed by the feature encoder that maps the 21-dimensional feature vector onto a 64-dimensional space. Subsequently, the graph is transformed through two convolutional layers before ultimately being input into a hierarchical pooling layer that concatenates six different matrices that have been pooled. Finally, the data goes through the MLP classifier before also outputting a binary value indicating either man or zone coverage.

Results:

To evaluate the performance of the models, I analyzed the test accuracy, F1 score, and AUC-ROC (area under the Receiver Operating Characteristic curve). Ultimately, the GNN had a greater AUC and F1 score; however, the CNN had a greater test accuracy. The values for the AUC, test accuracy, and F1 score are displayed in Table 2, while the AUC-ROC is illustrated in Figure 3:

	Test Accuracy	F1 Score	AUC
GNN Model	83.7%	0.737	0.920
CNN Model	84.3%	0.730	0.904

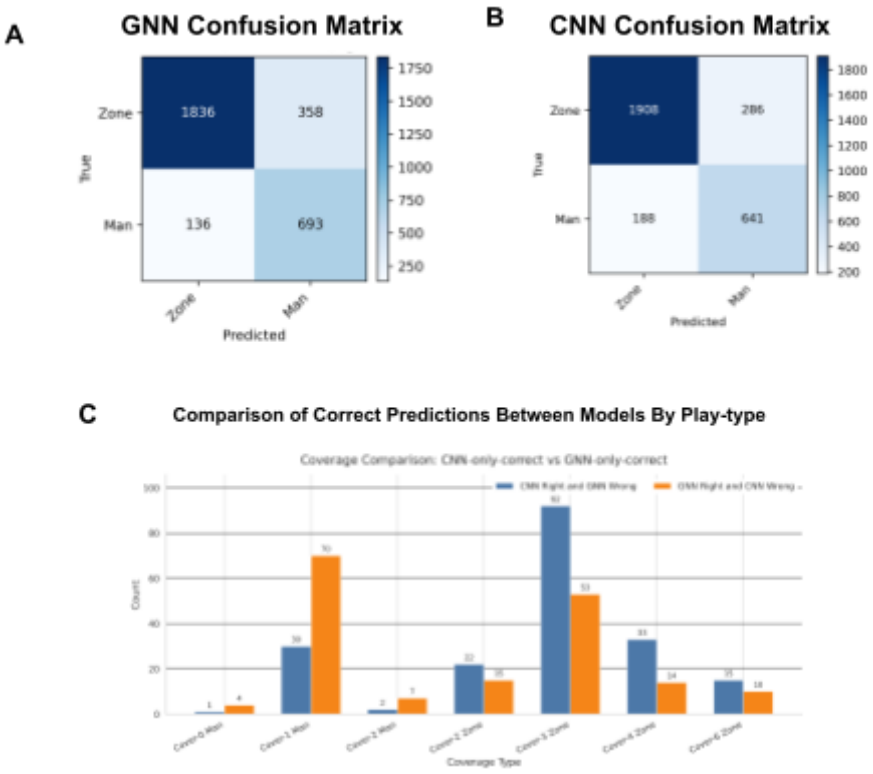
[Table 2: Table of the test accuracy, AUC, and F1 score for the GNN and CNN models. Each value is listed to three significant figures.]



[Figure 3: AUC-ROC of the CNN and GNN models. The blue line represents the CNN model, while the orange line represents the GNN. The dotted black line is a baseline of 50% accuracy.]

Test accuracy measures the proportion of correct predictions to the total number of predictions. Having a higher test accuracy implies that the model made fewer mistakes overall. However, in the case of class imbalance, accuracy alone can be misleading. F1-score is the harmonic mean of the precision and recall of a model. Precision measures the number of times a model predicted a particular class correctly compared to the total number of times it predicted that class. Recall measures the number of instances of a class

the model successfully predicted. Taking the harmonic mean incentivizes a balance between the two metrics, ensuring that a high score in one doesn't overshadow a poor score in the other. A model with a greater F1-score provides more balanced predictions, particularly for imbalanced datasets. Finally, AUC (Area Under Curve) is the area under an ROC curve. An ROC (Receiver Operating Characteristic) curve is a graph used to evaluate the performance of a binary classification model. It plots the relationship between true positives and false positives in order to assess the effectiveness of a model at separating the two classes at different thresholds. Consequently, the AUC reflects a model's ability to distinguish between classes across different thresholds, illustrating how well the model would generalize to unseen data.



[Figure 4: Panel A is the confusion matrix of the GNN labeled with the true values and the predicted values by the mode, while Panel B illustrates the confusion matrix of the CNN model. The color represents the frequency of that prediction. Panel C shows the differences in correct predictions between the models depending on the specific type of zone or man coverage.]

As test accuracy can often be misleading in cases of class imbalance, further analysis was conducted to examine how the models differed in their predictions. The confusion matrices show that while the CNN outperformed the GNN on zone plays, the GNN was more accurate on man plays. Diving deeper, the performance of the two models was compared based on more specific representations of man and zone plays. While the labels of man and zone are sufficient to describe defensive coverage, each has various subtypes that provide a more detailed description of how the defense is structured. When comparing the performance of the models on these class subtypes, a clear trend emerged. The CNN had correctly predicted many more plays of the majority class, cover-3 zone—constituting 37.4% of all plays—compared to the GNN, along with a better performance on all other types of zone coverage; however, the CNN performed worse than the GNN on all types of man coverage (Figure 4). Overall, the GNN achieved a greater AUC and F1 score while the CNN had a higher test accuracy. However, both outperformed the random classifier, with the GNN having an AUC of 0.920 compared to the CNN's 0.904.

Discussion:

The results of this study support the hypothesis that the use of a GNN is more effective than a CNN in classifying NFL defensive coverage schemes when using player tracking data. While the CNN achieved a marginally greater test accuracy, further analysis reveals that this was primarily driven by the CNN's bias towards the majority class (zone coverage), resulting in significantly worse performance than the GNN on man plays (Figure 4). In contrast, the GNN achieved a greater AUC and F1 score, indicating that it outperformed the CNN with stronger generalization and a more balanced classification performance.

The superior performance of the GNN can be explained by the fundamental differences between the two models' architectures. CNNs process features as fixed, grid-like matrices. This makes them well-equipped when analyzing spatially consistent data such as images. However, football plays cannot be evaluated by simply analyzing each player on the field independently. Instead, it is essential to also understand how players on the field interact with each other. Hence, the architecture in this paper adapts the CNN to consider relational context, inputting the relative distance, speed, and acceleration of each player in relation to each player on the opposing team. However, despite these efforts to improve the effectiveness of the CNN, the GNN is still better suited for this task. A typical CNN performs its convolutions on a fixed, grid-like window (Figure 1). Thus, the model gives greater weight to local neighbourhoods as it assumes data adjacent in the input are likely structured similarly in the field. However, given that players often change how they line up, their alignment on the field will rarely match the structure of the matrix. Consequently, the CNNs in this paper use a 1x1 kernel to avoid the incorrect assumption that adjacency on the

grid and the field are related. This significantly limits the model's predictive power by forcing the CNN to process each player-player pair in isolation.

In contrast, the GNN represents each player as a node in a graph and forms relationships between nodes in the form of edges. This allows the GNN to describe the complex and often arbitrary shape of the players on a football field. Additionally, the unique architecture of the GNN enables it to be further tuned by adjusting the weight of specific edges, placing greater weight on certain connections. In particular, the GNN employed in this paper placed greater weights based on proximity and a multiplier on within-team connections, which was determined to enhance the effectiveness of the model through empirical testing. The use of edge weights impacts the message-passing process, wherein the edge weight scales node embeddings of neighbours. Hence, neighbours with stronger edge weights influence the update of the node's embeddings more. By contrast, this would be much more difficult to implement into the CNN due to the lack of edges to demonstrate connections between players. Instead, interactions between players are interpreted through fixed arrangements, making player-player relationships much more difficult to adjust. The GNN's ability to model and optimize a relational graph allows it to retain contextual information across layers that would have been lost through transformations to a CNN's input matrix. Due to its better relational reasoning ability, the GNN can better adapt to shifts in alignment and generalize to new plays more effectively.

Although this paper categorizes coverage schemes into two labels, man and zone coverage, there are many variations of each that are employed by defensive coaching staff, which significantly affect defender alignment. Most coverage schemes can be classified into one of seven distinct classes: cover-0 man, cover-1 man, cover-2 man, cover-2 zone, cover-3 zone, cover-4 zone, and cover-6 zone. To understand why the CNN and GNN models both exhibited certain tendencies in their predictions, it is essential to first understand the distinctions between each coverage type.

The coverage types each include a number indicating the number of deep defenders present during that play. Zone coverage, to have their defenders cover different zones throughout the field, will often include more deep defenders who are responsible for covering deeper passes in zones further downfield. There are three types of man coverage, with the first being Cover-0 Man. Cover-0 involves significant risk as no deep defenders are employed. Instead, five defenders are all responsible for marking their respective eligible receiver while the remaining six players on the defensive team rush the quarterback in what is called a blitz. Similarly, Cover-1 Man involves only one deep defender, with the only difference being that five players will now rush the quarterback, also known as a dog. However, at times, defenses will instead sacrifice one rusher and rather have two defenders cover a receiver to ensure they have little chance of catching

the ball. The final man coverage titled Cover-2 Man is a hybrid between man and zone coverage, making it often difficult to classify. In Cover-2 Man, five defenders are once more responsible for covering the eligible receivers through man coverage. However, unlike the other forms of man coverage, two safeties now cover the deep half of the field, providing help to the man defenders in the case of a deep ball.

In contrast, Cover-2 Zone employs two deep safeties with all underneath defenders playing zone coverage. While Cover-2 Zone appears similar to Cover-2 Man at snap, the key difference is the lack of man coverage for close balls and hence the absence of help defense for receivers downfield. The remaining zone coverages follow a similar pattern, with Cover-3 Zone employing three deep defenders and four underneath defenders, Cover-4 Zone (also referred to as quarters) using four deep defenders and three underneath defenders. Finally, Cover-6 Zone appears slightly different from the other forms of zone coverage, with the defense essentially splitting the field into two halves. On one half, it plays a Cover-4 Zone, employing four deep defenders, while on the other, it plays a Cover-2 Zone, utilizing two deep defenders. This split coverage allows the defense to scheme around specific receivers and to prevent deep balls from being thrown.

From analyzing the Big Data Bowl dataset, the majority class appears to be Cover-3 Zone. Hence, given the apparent overfitting in the CNN, it is reasonable that it predicts Cover-3 Zone at a significantly higher rate than the GNN (Figure 4). With 92 more correct predictions than the GNN on Cover-3 Zone plays, the results highlight that the CNN heavily relies on this class for its high test accuracy. Furthermore, the CNN outperforms the GNN on all other zone coverages, with the GNN rarely predicting zone coverage plays correctly that the CNN failed to classify. However, the GNN significantly outperforms the CNN on man coverage, with 70 correct classifications that the CNN was unable to predict. Even more interestingly, on Cover-2 Man plays, a coverage that is typically difficult to predict due to its similarity to Cover-2 Zone plays, particularly pre-snap, the CNN was rarely able to classify any plays that the GNN incorrectly labeled correctly. These results highlight the CNN's failure to classify rarer plays, often resulting in the model incorrectly classifying the play as zone coverage. In contrast, the GNN's architecture allows it to process much more complex patterns, making it more reliable for real-world defensive coverage recognition.

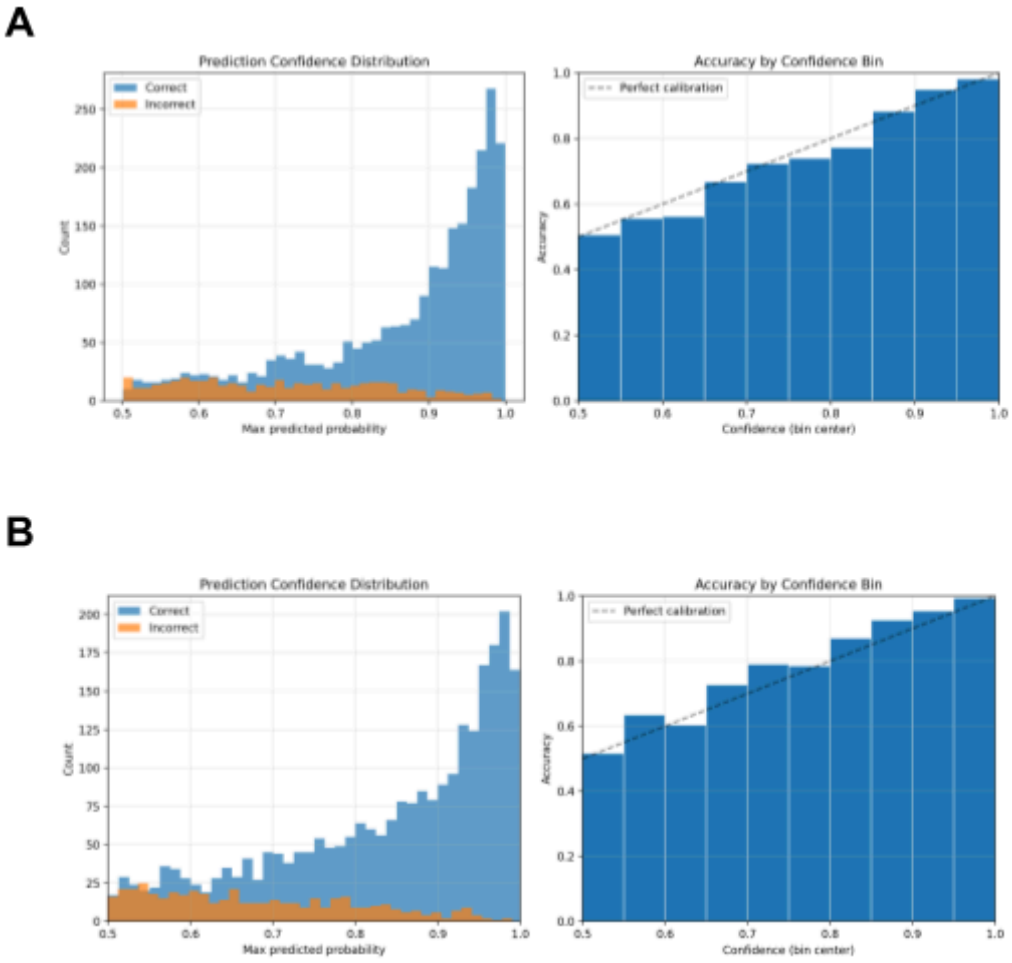
Conclusion and Future Directions:

This study evaluated the performance of a CNN and GNN for the pre-snap classification of NFL defensive coverage schemes using player tracking data from the 2025 Big Data Bowl. Overall, the CNN achieved a marginally higher test accuracy due to its bias towards predicting the major class, while the GNN outperformed the CNN with both a greater F1

and AUC score. The GNN's ability to model the field as a graph allowed it to maintain greater relational context than the CNN. This lack of signal decreased the confidence of the CNN in its predictions and led it to primarily predict zone coverage. These findings underscore the GNN's superior ability to analyze dynamic, non-grid environments such as American football, taking advantage of its graph-based architecture. Ultimately, the GNN proved to be more effective in making balanced predictions, supporting the idea that it is better suited for coverage scheme classification due to its ability to analyze connections and contextual relationships when making predictions (Khemani et al., 2024).

In future work, I plan to explore how incorporating temporal learning can impact both models. A limitation of the models used in this paper was their reliance on a single frame. Defensive coverages are dynamic, with players' alignment, motion, and position constantly changing before the snap. Hence, the models used in this paper were limited by their inability to capture patterns introduced by the movement of players prior to the snap. Temporal learning achieves this by allowing the models to analyze multiple frames and predict based on how the data changes over time. Specifically, the use of recurrent or temporal GNNs could address the limitations caused by static frames by modeling how the relationships between players evolve across frames, rather than being constrained to singular snapshots. As a result, the model capture a more nuanced representation of what is occurring pre-snap and potentially detect new patterns by analyzing the sequential shifts in player positions. As our knowledge of football evolves, we must explore how data science can complement human coaching to enhance decision-making, enabling fans to experience sports at the highest level both physically and tactically.

Appendix:



[Figure 5: Panel A shows the count of correct and incorrect guesses relative to the model's predicted probability of success, along with the model's accuracy relative to their confidence in that guess for the GNN. Panel A shows the same diagrams for the CNN.]

References

Ajit, A., Acharya, K., & Samanta, A. (2020). A Review of Convolutional Neural Networks. *IEEE*, 1-5. https://doi.org/10.1109/ic-ETITE47903.2020.049

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016, July 21). Layer Normalization. *arXiv*. https://doi.org/10.48550/arXiv.1607.06450 (Non-peer-reviewed)

Fernandes, C. J., Yakubov, R., Li, Y., Prasad, A. K., & Chan, T. C.Y. (2019, October 26). Predicting plays in the National Football League. *Journal of Sports Analytics*, *6*(1), 35-43. https://doi.org/10.3233/JSA-190348

Gholamalinezhad, H., & Khosravi, H. (2020, September 16). Pooling Methods in Deep Neural Networks, a Review. *arXiv*. https://doi.org/10.48550/arXiv.2009.07485 (Non-peer-reviewed)

Goyal, U. (2020, February). Leveraging Machine Learning to Predict Playcalling Tendencies in the NFL. *DSpace@MIT*. Retrieved August 14, 2025, from https://dspace.mit.edu/handle/1721.1/129909

The History of the Rules. (2025). NFL Football Operations. Retrieved August 8, 2025, from https://operations.nfl.com/the-rules/evolution-of-the-nfl-rules/ (Non-peer-reviewed)

Identifying significant features for Player Evaluation in NFL comparing ANNs and Traditional Models. (2021). *Arrow@TU Dublin*. https://doi.org/10.21427/EAC6-4R95

Khemani, B., Patil, S., Kotecha, K., & Tanwar, S. (2024, January 16). A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions. *Journal of Big Data*, *11*(18). https://doi.org/10.1186/s40537-023-00876-4

Kingston, C. (2023, May 12). Measuring Grit in NFL Cornerbacks using Statistical Analysis. https://dspace.mit.edu/bitstream/handle/1721.1/151676/kingston-ctk-meng-eecs-2023-thesis.pdf?sequence=1&isAllowed=y

Lopez, M., Bliss, T., Blake, A., Mooney, P., & Howard, A. (2024). *NFL Big Data Bowl 2025*. Kaggle. Retrieved August 14, 2025, from https://kaggle.com/competitions/nfl-big-data-bowl-2025 (Non-peer-reviewed)

McManus, C. (2025, March 17). Artificial Intelligence for better in-game NFL performance. *Scholarworks at WMU*. Retrieved August 14, 2025, from https://scholarworks.wmich.edu/cgi/viewcontent.cgi?article=4965&context=honors_theses

Ofeidis, I., Kiedanski, D., & Tassiulas, L. (2025, January 16). An Overview of the Data-Loader Landscape: Comparative Performance Analysis. *IEEE*, pp. 360–367. https://doi.org/10.1109/BigData62323.2024.10825421

Purwono, P., Ma'arif, A., Rahmaniar, W., Fathurrahman, H. I. K., Frisky, A. Z. K., & Haq, Q. M. u. (2023). Understanding of Convolutional Neural Network (CNN): A Review. *International Journal of Robotics and Control Systems*, 2(4), 739-748. https://doi.org/10.31763/ijrcs.v2i4.888

Quarterbacks: *Finding Weakness and Strength in Zone vs. Man Coverage*. (2023, August 1). Capital QB's. Retrieved August 14, 2025, from https://www.capitalqbs.com/quarterbacks-finding-weakness-and-strength-in-zone-vs-man-coverage/ (Non-peer-reviewed)

Sanchez-Lengeling, B., Reif, E., Pearce, A., & Wiltschko, A. B. (2021, September 2). A Gentle Introduction to Graph Neural Networks. *Distill*. https://doi.org/10.23915/distill.00033 (Non-peer-reviewed)

Song, H., Jazaery, M. A., Ding, H., Cheong, L. L., Jung, J., Band, M., Chi, M., & Bliss, T. (2023). Explainable Defense Coverage Classification in NFLGames using Deep Neural Networks. *Amazon ML Solutions Lab*. https://www.amazon.science/publications/explainable-defense-coverage-classification-in-nfl-games-using-deep-neural-networks

Taylor, C. (2020). Deep Learning for In-Game NFL Predictions. https://cs230.stanford.edu/projects_winter_2020/reports/32263160.pdf#page=1.33 (Non-peer-reviewed)

Grids to Graphs: GNNs improve NFL defensive coverage classification over CNNs

Abstract:

This paper investigates the application of predictive neural network models to classify NFL defensive coverages using player tracking data. I propose a novel solution using a graph neural network (GNN) as an alternative to the conventional use of convolutional neural networks (CNNs). Both models were compared based on their test accuracy, F1 score, and AUC. While the CNN reported a greater test accuracy (84.3%) than the GNN (83.7%), this was primarily due to its bias towards zone coverage, the majority class. In contrast, the GNN yielded more balanced predictions with a greater F1 score (0.737 vs. 0.730) and AUC (0.920 vs. 0.904) than the CNN. The GNN was also more effective than the CNN when classifying less common coverage scheme variations. These findings suggest that GNNs are better suited for classifying NFL defensive coverages. This is because their unique architecture represents information as graphs, helping them better understand player-player interactions.

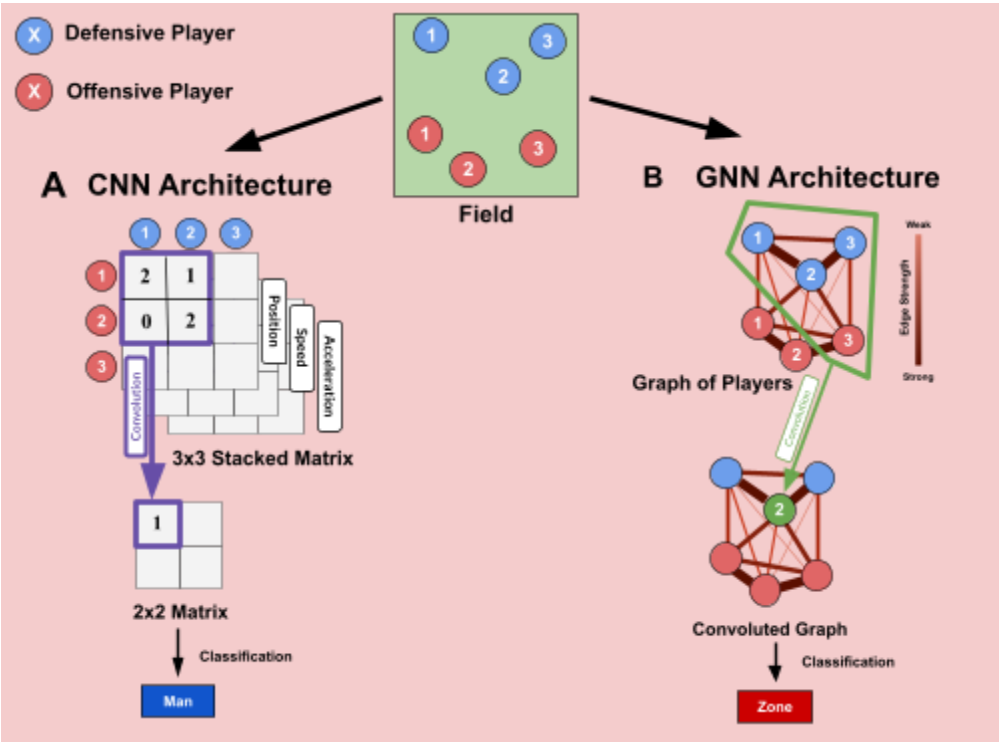
Introduction:

The National Football League (NFL) has long focused its analysis and changes towards offense to make the viewing experience more engaging. Per the NFL's Competition Committee, "if someone wants to accuse the National Football League of promoting offense to make the game more exciting, [the committee believes the league should plead guilty]" (NFL Football Operations, 2025). Hence, the recent rise in machine learning (ML)-powered football analytics has predominantly focused on offensive playcalling and performance (McManus, 2025; Goyal, 2020; Walsh, 2021; Fernandes et al., 2019; Taylor, 2020). On the other hand, considerably less analysis has been performed regarding the defensive end of the floor.

On defense, the coaching staff is responsible for positioning their players in specific formations, known as coverage schemes. Defensive coverage schemes determine the responsibilities of each player on the defensive team. Coverage schemes are categorized into two broad types: man coverage and zone coverage. In man coverage, each defender is responsible for a specific offensive player throughout the play. In contrast, zone coverage requires defensive players to defend any players that enter a particular area on the field known as their "zone" (Kingston, 2023). Both coverage schemes are employed for vastly different use cases, with each having its own benefits. While zone coverage often minimizes big plays, man coverage leads to a tighter defense, placing greater pressure on the opposing quarterback to make an accurate throw (Capital QB's, 2023).

Determining which coverage scheme the defensive team will use helps the coaching staff of the offensive team choose plays to increase their chances of scoring. As a result, the coaches of the defensive team often disguise their coverage scheme by making their players rotate following the snap—the start of a play in football. This can make play calling difficult for even the most skilled coaches. Hence, having a method of classifying coverage schemes pre-snap would enable the offense to anticipate and plan against the defensive strategies used by the opposing team.

One way of classifying defensive coverage is by using predictive models. Currently, the most successful method of classifying defensive coverage schemes is convolutional neural networks (CNNs) (Song et al., 2023). CNNs are a form of artificial neural network that process data structured in grid-like matrices. They use convolutional layers (which can be thought of as a sliding window) to capture local patterns. Throughout the training process, the weights of the window are learned (Figure 1). This process of convolution is why CNNs are often used when processing images. Images have strong spatial locality (pixels within a proximity are highly correlated), meaning the ability to capture local patterns effectively allows CNNs to learn the complex structures of objects within images without becoming overwhelmed by the large quantity of pixels (Purwono et al., 2022).



[Figure 1: Panel A shows diagrams explaining the architecture of the CNN. The football field (green square) has three offensive players (blue) and three defensive players (red). The first

illustration in panel A illustrates how the field is represented as a stacked matrix, with the values representing the distinct pairwise relative features. The purple square is then convoluted into one number in the following illustration before ultimately becoming a binary output. Panel B presents an illustration of the players in the form of a graph. The lines between the circles (nodes) represent the connections between players (edges). The thickness and color of the edges represent the strength of these connections. Similar to panel A, the values of the graph are convoluted into a new graph. Ultimately, the GNN classifies into a binary output.

Modern methods of applying CNNs to coverage scheme classification have found success in using relative data in conjunction with static data. Song et al. (2023) demonstrate that relying solely on static features significantly limits the predictive capabilities of a CNN. However, by conducting feature engineering to construct a matrix of pairwise relative features between each member of the offensive team and the players on the defensive team, they were able to model complex player interactions, thereby maintaining greater spatial and relational context (Figure 1) (Song et al., 2023).

While Song et al. (2023) found success through the use of relative features, it must be noted that they increase the dimensionality of the data significantly due to the many static values that are repeated throughout the matrix. This introduces various concerns, including greater computational demands, a high risk of overfitting, and an unintentional emphasis placed on static features, as they are constantly repeated within each pairwise comparison. Additionally, CNNs likely struggle to anticipate shifts that could occur to players' positions throughout a play due to their inability to understand how each player relates to the others on the field.

In contrast, the use of a graph neural network (GNN) should theoretically be more effective due to its unique architecture. GNNs operate on graph-structured data, describing data through nodes, edges, and a global context. The graph is passed through a differentiable function, such as a multilayer perceptron (MLP), to process the data. In the case of binary classification, such as predicting coverage schemes, the prediction target is the global context. As such, the information within the nodes and edges must be passed through a pooling layer, creating a concatenated matrix that can be used for classification (Sanchez-Lengeling et al., 2021). This process of pooling information from the nodes and edges to make predictions about the global context makes GNNs particularly robust, as it allows the GNN to learn from the intricate relationships between players on the field. Hence, the use of a GNN should enhance the ability to classify defensive coverages.

Previous works, including those of Song et al. (2023), have primarily focused on optimizing the effectiveness of the CNN model by incorporating methods such as temporal learning to

maximize accuracy. However, this paper will instead focus on comparing the performance of CNN models to that of GNN models in an attempt to develop a novel classification method for NFL coverage schemes. Consequently, this paper discusses the hypothesis that a GNN is more effective at NFL defensive coverage classification than a CNN.

Methods:

In football, man and zone are the two primary labels used to classify defensive coverages. Therefore, differentiating between man and zone coverages is both relevant and sufficient to identify NFL defensive coverages. As there are two primary outputs, this paper will view coverage scheme classification as a binary classification problem between man and zone coverage. The study will be performed on the 2025 Big Data Bowl, a public dataset released annually by the NFL. The dataset includes tracking data for all 22 players on the field (11 offense and 11 defense) during passing plays, along with detailed play-level descriptions. Furthermore, the dataset includes contextual information for each game as well as player-specific attributes such as build and composition (Lopez et al., 2024). The data is drawn from the first nine weeks of the 2022-23 NFL season and includes 3,023 plays.

Initially, the data is pre-processed, with any plays not labeled as either man or zone being filtered out. These plays are labeled as "other" in the dataset and constitute plays where either the line of scrimmage is within the endzone, meaning traditional forms of pass coverage are not employed, or during a quarterback kneel, which is a special form of play that occurs when the knee of a quarterback touches the ground directly after the snap and thereby immediately ends the play for strategic reasons. Subsequently, the dataset is filtered such that only the 31st frame of each play remains. The data is recorded at 10 frames per second, meaning the 31st frame represents three seconds into the play. As no snaps occur before the 31st frame, it ensures that the data is pre-snap and hence applies to offenses scheming around the defensive coverage employed. Furthermore, given that the frame is three seconds into the play, it is presumed that most players are already in their coverage positions. Though ideally, a model should use all pre-snap frames, a singular frame was chosen due to computational limitations.

CNN (normalized)	GNN (normalized)
Distance from the line of scrimmage	Distance from the line of scrimmage
Position on the y-axis	Position on the y-axis
Speed in the x-direction	Speed in the x-direction
Speed in the y-direction	Speed in the y-direction
Speed	Speed
Acceleration in the x-direction	Acceleration in the x-direction
Acceleration in the y-direction	Acceleration in the y-direction
Acceleration	Acceleration
Orientation	Orientation
Relative speed in the x-direction	Nearest opponent distance
Relative speed in the y-direction	Distance to team centroid
Relative acceleration in the x-direction	Position (represented as 10 distinct binary channels representing the quarterback, running back, wide receiver, tight end, offensive lineman, defensive lineman, linebacker, cornerback, safety, and nickel)
Relative acceleration in the y-direction	
Relative position in the x-direction	
Relative position in the y-direction	

[Table 1: The list of all the features used when training the CNN and GNN models. All listed features were either taken directly from or calculated with data found on the 2025 Big Data Bowl dataset and were all normalized for stability.

The decision to provide speed along with the x and y components of speed as separate features was made intentionally. CNNs and GNNs are both linear models and are hence unable to derive the speed of a player based on the directional components. By providing features such as speed and acceleration in both formats, the model can choose the representation it finds most effective, ensuring flexibility between scalar and directional values, and can therefore identify new patterns to better classify coverage schemes.

The CNN was built following the architecture described in Song et al. (2023). The model was trained on a 15x11x11 stacked matrix—with 15 features, 11 offensive players, and 11 defensive players (Table 1). The data were then reshaped into batches of 64 plays to enhance the training time and generalization of the model (Ofeidis et al., 2024). The architecture of the CNN utilizes two convolutional layers, designed to extract high-level representations while preserving the spatial structure of the matrix. The first layer consists of three successive 1x1 convolutional layers. The use of a unit kernel enables the combination of all 15 feature values into a single value without blurring or merging with information from other cells. Hence, the output maintains its 11x11 spatial identity and arrangement while reducing the computational load (Ajit et al., 2020). Subsequently, a dropout layer is introduced to drop 30% of the data, thereby minimizing overfitting and regularizing the data. Following the dropout, the features are processed through a pooling layer that consists of average and max pooling. The purpose of implementing the pooling layer is to down-sample the feature maps following the convolutional layers to reduce computational cost and control overfitting (Gholamalinezhad & Khosravi, 2020). Specifically, the pooling layer collapses the dimension of offensive players, transforming the data into 128-d vectors for each defensive player. Average pooling captures the overall trend, while max pooling identifies the strongest signals. A weighted sum of both pooling values is taken (Equation 1).

$$x = (mean \times 0.7) + (max \times 0.3) \qquad [1]$$

Following the pooling layer, the features are input into another convolutional layer consisting of three 1x1 convolutional layers, each followed by batch normalization to normalize along the channel dimension. Once more, the unit kernel allows for linear transformations of the channels, increasing the channels to 160 before ultimately outputting 96 channels. After the second convolutional layer, the data is transformed through a second pooling layer that collapses the defensive-player axis while implementing the same weighted sum of average and max pooling. Hence, the output is a 96-d vector describing each batch. Finally, the data is transformed through a set of dense layers. The first dense layer employs a ReLU activation that maps the 96 features to 96 output dimensions, thereby introducing non-linearity for more complex feature representations. Subsequently, batch normalization is applied to ensure a standard distribution of the activations throughout the batch. Thereafter, a second dense layer is employed, expanding the feature space from 96 to 256 dimensions, thereby utilizing a higher-dimensional space to capture more complex patterns and linearly separable relationships. After another batch normalization is applied to enhance stability and convergence. Subsequently, layer normalization is applied, which stabilizes the dynamics of the hidden state and improves training time (Ba et al., 2016). Finally, the last dense layer reduces the 256-dimensional normalized vector to two outputs, being the two coverage schemes, and applies a softmax

activation to transform the logits into a probability distribution between the coverage schemes. This serves as the classifier, matching each sample to a class based on the learned feature representation from the previous stages of the model. Ultimately, the class with the greater probability is output as the model's prediction of the class (Figure 2).

The GNN was trained on a graph consisting of 22 nodes (11 offensive players and 11 defensive players). Each of these nodes contains a 21-dimensional feature vector. To construct the graph, edges are defined by both Euclidean distance and team membership. Edges are placed between all players within 12 yards of each other, with weights between players on the same team having a multiplier. The value of 12 yards between players and the multiplier for same-team allegiance were determined empirically to optimize the model's accuracy. The graph is then transformed by an encoder that maps the 21-dimensional feature vectors onto a 64-dimensional space. This layer uses L2 regularization with a regularization parameter of 10^{-4}. Then, the data is stabilized through batch normalization, before introducing non-linearity through a ReLU activation, and employing a dropout layer to minimize overfitting.

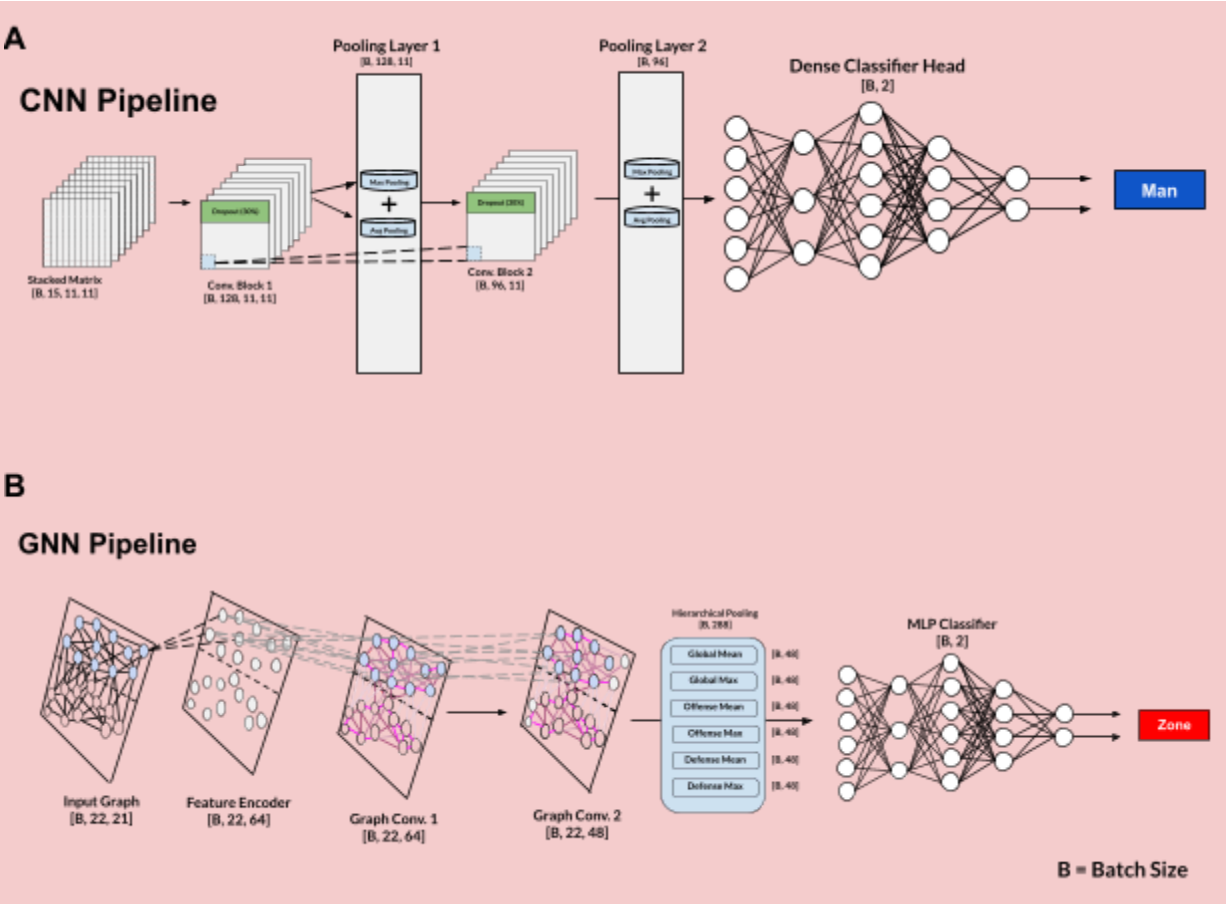
The projected features are then processed through two convolutional layers. Node representations, the feature vectors for each player, are then updated through weighted message passing, which gathers aggregate features from all adjacent nodes (weighted by the strength of that edge) and performs a linear transformation followed by a ReLU activation. The first graph layer outputs 64 channels, followed by batch normalization and a dropout layer. In comparison, the second outputs 48 channels and is also followed by batch normalization and a dropout layer.

After the second convolutional layer, the node embeddings, which are 22 x 48 in size, are transformed through a pooling layer. The pooling layer first performs average and max pooling on the 22 players in the plays, followed by applying the same operations to defensive and offensive subgraphs separately. These processes yield six distinct 48-dimensional pooled vectors, which are concatenated into a 288-dimensional representation of the play. This 288-dimensional graph vector is then passed through a three-layer multi-layer perceptron serving as the classifier. Each hidden layer employs L2 regularization, batch normalization, ReLU activation, and dropout. The final layer applies a softmax activation function identical to the one used in the CNN to generate class probabilities, with the model ultimately predicting the class with the greater probability (Figure 2)

Both the CNN and GNN are trained with 5-fold cross-validation for 50 epochs, using early stopping and a learning rate decay to optimize the training. The criteria for the early stopping of both models are set such that if the validation accuracy does not increase for 10

epochs, the training stops early. Similarly, if the validation accuracy does not increase for three epochs, the learning rate is halved until it reaches below 0.000001, at which point the training is stopped early.

To compare model performance, the same 20% of the dataset (3023 plays) was held out during model training. To evaluate the performance, I looked at the AUC, F1 score, and test accuracy of both models.



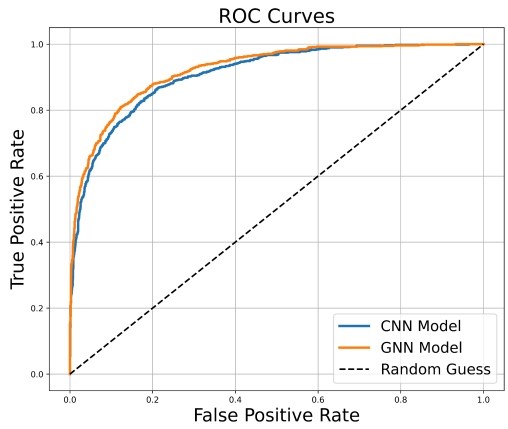
[Figure 2: The above diagram includes two panels illustrating the pipeline of the CNN and GNN models, respectively. Panel A depicts the CNN pipeline, starting with a stacked matrix as the input. The data are then transformed through two convolutional blocks, each followed by a pooling layer consisting of both max and average pooling. Finally, the data goes through the dense classifier head, which outputs a binary result indicating either man or zone coverage. Panel B shows the GNN pipeline. The initial input graph is transformed by the feature encoder that maps the 21-dimensional feature vector onto a 64-dimensional space. Subsequently, the graph is transformed through two convolutional layers before ultimately being input into a hierarchical pooling layer that concatenates six different matrices that have been pooled. Finally, the data goes through the MLP classifier before also outputting a binary value indicating either man or zone coverage.

Results:

To evaluate the performance of the models, I analyzed the test accuracy, F1 score, and AUC-ROC (area under the Receiver Operating Characteristic curve). Ultimately, the GNN had a greater AUC and F1 score; however, the CNN had a greater test accuracy. The values for the AUC, test accuracy, and F1 score are displayed in Table 2, while the AUC-ROC is illustrated in Figure 3:

	Test Accuracy	F1 Score	AUC
GNN Model	83.7%	0.737	0.920
CNN Model	84.3%	0.730	0.904

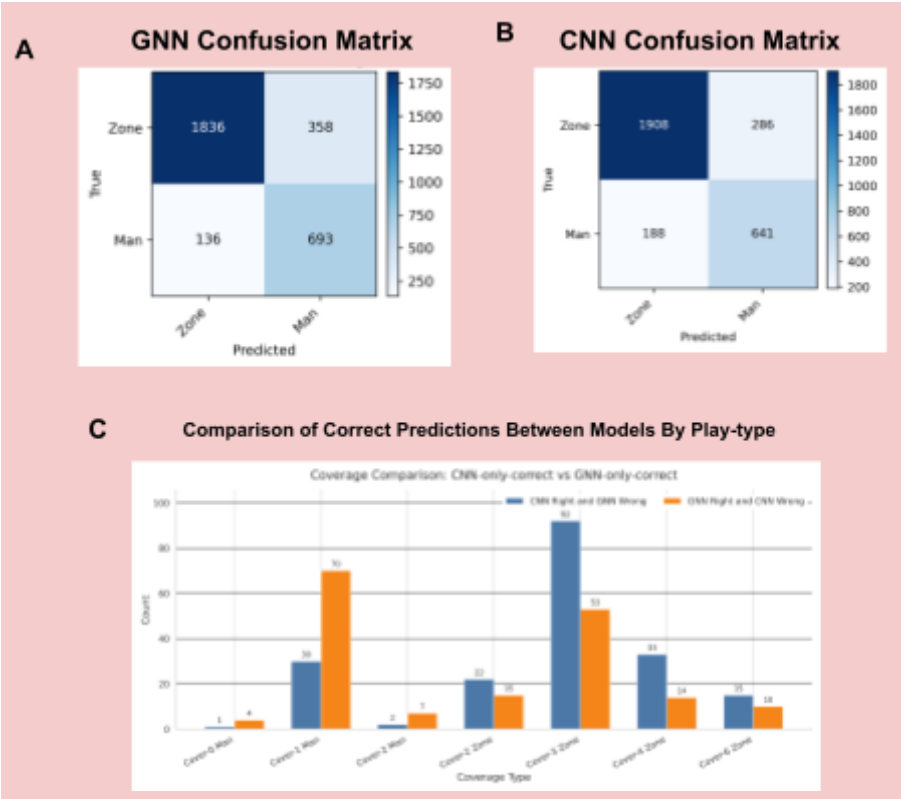
[Table 2: Table of the test accuracy, AUC, and F1 score for the GNN and CNN models. Each value is listed to three significant figures.]



[Figure 3: AUC-ROC of the CNN and GNN models. The blue line represents the CNN model, while the orange line represents the GNN. The dotted black line is a baseline of 50% accuracy.]

Test accuracy measures the proportion of correct predictions to the total number of predictions. Having a higher test accuracy implies that the model made fewer mistakes overall. However, in the case of class imbalance, accuracy alone can be misleading. F1-score is the harmonic mean of the precision and recall of a model. Precision measures the number of times a model predicted a particular class correctly compared to the total number of times it predicted that class. Recall measures the number of instances of a class

the model successfully predicted. Taking the harmonic mean incentivizes a balance between the two metrics, ensuring that a high score in one doesn't overshadow a poor score in the other. A model with a greater F1-score provides more balanced predictions, particularly for imbalanced datasets. Finally, AUC (Area Under Curve) is the area under an ROC curve. An ROC (Receiver Operating Characteristic) curve is a graph used to evaluate the performance of a binary classification model. It plots the relationship between true positives and false positives in order to assess the effectiveness of a model at separating the two classes at different thresholds. Consequently, the AUC reflects a model's ability to distinguish between classes across different thresholds, illustrating how well the model would generalize to unseen data.



[Figure 4: Panel A is the confusion matrix of the GNN labeled with the true values and the predicted values by the mode, while Panel B illustrates the confusion matrix of the CNN model. The color represents the frequency of that prediction. Panel C shows the differences in correct predictions between the models depending on the specific type of zone or man coverage.]

As test accuracy can often be misleading in cases of class imbalance, further analysis was conducted to examine how the models differed in their predictions. The confusion matrices show that while the CNN outperformed the GNN on zone plays, the GNN was more accurate on man plays. Diving deeper, the performance of the two models was compared based on more specific representations of man and zone plays. While the labels of man and zone are sufficient to describe defensive coverage, each has various subtypes that provide a more detailed description of how the defense is structured. When comparing the performance of the models on these class subtypes, a clear trend emerged. The CNN had correctly predicted many more plays of the majority class, cover-3 zone—constituting 37.4% of all plays—compared to the GNN, along with a better performance on all other types of zone coverage; however, the CNN performed worse than the GNN on all types of man coverage (Figure 4). Overall, the GNN achieved a greater AUC and F1 score while the CNN had a higher test accuracy. However, both outperformed the random classifier, with the GNN having an AUC of 0.920 compared to the CNN's 0.904.

Discussion:

The results of this study support the hypothesis that the use of a GNN is more effective than a CNN in classifying NFL defensive coverage schemes when using player tracking data. While the CNN achieved a marginally greater test accuracy, further analysis reveals that this was primarily driven by the CNN's bias towards the majority class (zone coverage), resulting in significantly worse performance than the GNN on man plays (Figure 4). In contrast, the GNN achieved a greater AUC and F1 score, indicating that it outperformed the CNN with stronger generalization and a more balanced classification performance.

The superior performance of the GNN can be explained by the fundamental differences between the two models' architectures. CNNs process features as fixed, grid-like matrices. This makes them well-equipped when analyzing spatially consistent data such as images. However, football plays cannot be evaluated by simply analyzing each player on the field independently. Instead, it is essential to also understand how players on the field interact with each other. Hence, the architecture in this paper adapts the CNN to consider relational context, inputting the relative distance, speed, and acceleration of each player in relation to each player on the opposing team. However, despite these efforts to improve the effectiveness of the CNN, the GNN is still better suited for this task. A typical CNN performs its convolutions on a fixed, grid-like window (Figure 1). Thus, the model gives greater weight to local neighbourhoods as it assumes data adjacent in the input are likely structured similarly in the field. However, given that players often change how they line up, their alignment on the field will rarely match the structure of the matrix. Consequently, the CNNs in this paper use a 1x1 kernel to avoid the incorrect assumption that adjacency on the

grid and the field are related. This significantly limits the model's predictive power by forcing the CNN to process each player-player pair in isolation.

In contrast, the GNN represents each player as a node in a graph and forms relationships between nodes in the form of edges. This allows the GNN to describe the complex and often arbitrary shape of the players on a football field. Additionally, the unique architecture of the GNN enables it to be further tuned by adjusting the weight of specific edges, placing greater weight on certain connections. In particular, the GNN employed in this paper placed greater weights based on proximity and a multiplier on within-team connections, which was determined to enhance the effectiveness of the model through empirical testing. The use of edge weights impacts the message-passing process, wherein the edge weight scales node embeddings of neighbours. Hence, neighbours with stronger edge weights influence the update of the node's embeddings more. By contrast, this would be much more difficult to implement into the CNN due to the lack of edges to demonstrate connections between players. Instead, interactions between players are interpreted through fixed arrangements, making player-player relationships much more difficult to adjust. The GNN's ability to model and optimize a relational graph allows it to retain contextual information across layers that would have been lost through transformations to a CNN's input matrix. Due to its better relational reasoning ability, the GNN can better adapt to shifts in alignment and generalize to new plays more effectively.

Although this paper categorizes coverage schemes into two labels, man and zone coverage, there are many variations of each that are employed by defensive coaching staff, which significantly affect defender alignment. Most coverage schemes can be classified into one of seven distinct classes: cover-0 man, cover-1 man, cover-2 man, cover-2 zone, cover-3 zone, cover-4 zone, and cover-6 zone. To understand why the CNN and GNN models both exhibited certain tendencies in their predictions, it is essential to first understand the distinctions between each coverage type.

The coverage types each include a number indicating the number of deep defenders present during that play. Zone coverage, to have their defenders cover different zones throughout the field, will often include more deep defenders who are responsible for covering deeper passes in zones further downfield. There are three types of man coverage, with the first being Cover-0 Man. Cover-0 involves significant risk as no deep defenders are employed. Instead, five defenders are all responsible for marking their respective eligible receiver while the remaining six players on the defensive team rush the quarterback in what is called a blitz. Similarly, Cover-1 Man involves only one deep defender, with the only difference being that five players will now rush the quarterback, also known as a dog. However, at times, defenses will instead sacrifice one rusher and rather have two defenders cover a receiver to ensure they have little chance of catching

the ball. The final man coverage titled Cover-2 Man is a hybrid between man and zone coverage, making it often difficult to classify. In Cover-2 Man, five defenders are once more responsible for covering the eligible receivers through man coverage. However, unlike the other forms of man coverage, two safeties now cover the deep half of the field, providing help to the man defenders in the case of a deep ball.

In contrast, Cover-2 Zone employs two deep safeties with all underneath defenders playing zone coverage. While Cover-2 Zone appears similar to Cover-2 Man at snap, the key difference is the lack of man coverage for close balls and hence the absence of help defense for receivers downfield. The remaining zone coverages follow a similar pattern, with Cover-3 Zone employing three deep defenders and four underneath defenders, Cover-4 Zone (also referred to as quarters) using four deep defenders and three underneath defenders. Finally, Cover-6 Zone appears slightly different from the other forms of zone coverage, with the defense essentially splitting the field into two halves. On one half, it plays a Cover-4 Zone, employing four deep defenders, while on the other, it plays a Cover-2 Zone, utilizing two deep defenders. This split coverage allows the defense to scheme around specific receivers and to prevent deep balls from being thrown.

From analyzing the Big Data Bowl dataset, the majority class appears to be Cover-3 Zone. Hence, given the apparent overfitting in the CNN, it is reasonable that it predicts Cover-3 Zone at a significantly higher rate than the GNN (Figure 4). With 92 more correct predictions than the GNN on Cover-3 Zone plays, the results highlight that the CNN heavily relies on this class for its high test accuracy. Furthermore, the CNN outperforms the GNN on all other zone coverages, with the GNN rarely predicting zone coverage plays correctly that the CNN failed to classify. However, the GNN significantly outperforms the CNN on man coverage, with 70 correct classifications that the CNN was unable to predict. Even more interestingly, on Cover-2 Man plays, a coverage that is typically difficult to predict due to its similarity to Cover-2 Zone plays, particularly pre-snap, the CNN was rarely able to classify any plays that the GNN incorrectly labeled correctly. These results highlight the CNN's failure to classify rarer plays, often resulting in the model incorrectly classifying the play as zone coverage. In contrast, the GNN's architecture allows it to process much more complex patterns, making it more reliable for real-world defensive coverage recognition.

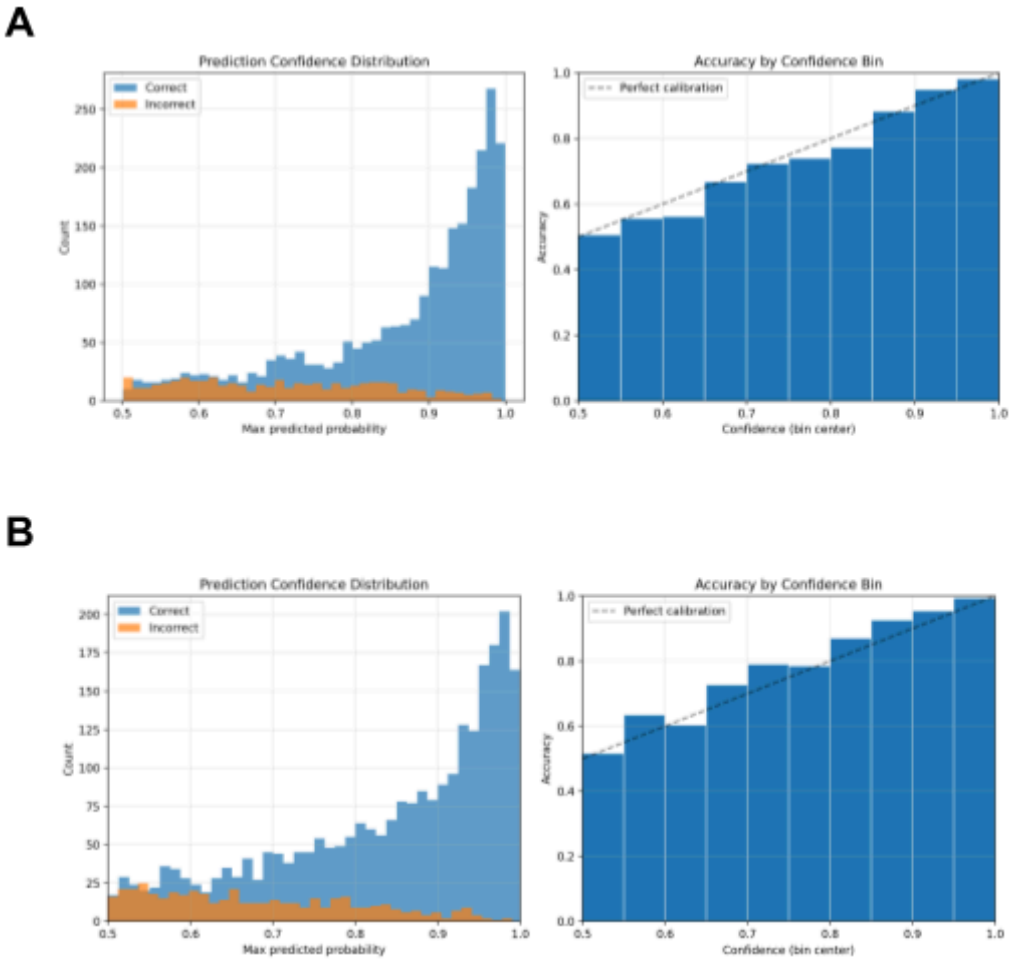
Conclusion and Future Directions:

This study evaluated the performance of a CNN and GNN for the pre-snap classification of NFL defensive coverage schemes using player tracking data from the 2025 Big Data Bowl. Overall, the CNN achieved a marginally higher test accuracy due to its bias towards predicting the major class, while the GNN outperformed the CNN with both a greater F1

and AUC score. The GNN's ability to model the field as a graph allowed it to maintain greater relational context than the CNN. This lack of signal decreased the confidence of the CNN in its predictions and led it to primarily predict zone coverage. These findings underscore the GNN's superior ability to analyze dynamic, non-grid environments such as American football, taking advantage of its graph-based architecture. Ultimately, the GNN proved to be more effective in making balanced predictions, supporting the idea that it is better suited for coverage scheme classification due to its ability to analyze connections and contextual relationships when making predictions (Khemani et al., 2024).

In future work, I plan to explore how incorporating temporal learning can impact both models. A limitation of the models used in this paper was their reliance on a single frame. Defensive coverages are dynamic, with players' alignment, motion, and position constantly changing before the snap. Hence, the models used in this paper were limited by their inability to capture patterns introduced by the movement of players prior to the snap. Temporal learning achieves this by allowing the models to analyze multiple frames and predict based on how the data changes over time. Specifically, the use of recurrent or temporal GNNs could address the limitations caused by static frames by modeling how the relationships between players evolve across frames, rather than being constrained to singular snapshots. As a result, the model capture a more nuanced representation of what is occurring pre-snap and potentially detect new patterns by analyzing the sequential shifts in player positions. As our knowledge of football evolves, we must explore how data science can complement human coaching to enhance decision-making, enabling fans to experience sports at the highest level both physically and tactically.

Appendix:



[Figure 5: Panel A shows the count of correct and incorrect guesses relative to the model's predicted probability of success, along with the model's accuracy relative to their confidence in that guess for the GNN. Panel A shows the same diagrams for the CNN.]

References

Ajit, A., Acharya, K., & Samanta, A. (2020). A Review of Convolutional Neural Networks. *IEEE*, 1-5. https://doi.org/10.1109/ic-ETITE47903.2020.049

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016, July 21). Layer Normalization. *arXiv*. https://doi.org/10.48550/arXiv.1607.06450 (Non-peer-reviewed)

Fernandes, C. J., Yakubov, R., Li, Y., Prasad, A. K., & Chan, T. C.Y. (2019, October 26). Predicting plays in the National Football League. *Journal of Sports Analytics*, *6*(1), 35-43. https://doi.org/10.3233/JSA-190348

Gholamalinezhad, H., & Khosravi, H. (2020, September 16). Pooling Methods in Deep Neural Networks, a Review. *arXiv*. https://doi.org/10.48550/arXiv.2009.07485 (Non-peer-reviewed)

Goyal, U. (2020, February). Leveraging Machine Learning to Predict Playcalling Tendencies in the NFL. *DSpace@MIT*. Retrieved August 14, 2025, from https://dspace.mit.edu/handle/1721.1/129909

The History of the Rules. (2025). NFL Football Operations. Retrieved August 8, 2025, from https://operations.nfl.com/the-rules/evolution-of-the-nfl-rules/ (Non-peer-reviewed)

Identifying significant features for Player Evaluation in NFL comparing ANNs and Traditional Models. (2021). *Arrow@TU Dublin*. https://doi.org/10.21427/EAC6-4R95

Khemani, B., Patil, S., Kotecha, K., & Tanwar, S. (2024, January 16). A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions. *Journal of Big Data*, *11*(18). https://doi.org/10.1186/s40537-023-00876-4

Kingston, C. (2023, May 12). Measuring Grit in NFL Cornerbacks using Statistical Analysis. https://dspace.mit.edu/bitstream/handle/1721.1/151676/kingston-ctk-meng-eecs-2023-thesis.pdf?sequence=1&isAllowed=y

Lopez, M., Bliss, T., Blake, A., Mooney, P., & Howard, A. (2024). *NFL Big Data Bowl 2025*. Kaggle. Retrieved August 14, 2025, from https://kaggle.com/competitions/nfl-big-data-bowl-2025 (Non-peer-reviewed)

McManus, C. (2025, March 17). Artificial Intelligence for better in-game NFL performance. *Scholarworks at WMU*. Retrieved August 14, 2025, from https://scholarworks.wmich.edu/cgi/viewcontent.cgi?article=4965&context=honors_theses

Ofeidis, I., Kiedanski, D., & Tassiulas, L. (2025, January 16). An Overview of the Data-Loader Landscape: Comparative Performance Analysis. *IEEE*, pp. 360–367. https://doi.org/10.1109/BigData62323.2024.10825421

Purwono, P., Ma'arif, A., Rahmaniar, W., Fathurrahman, H. I. K., Frisky, A. Z. K., & Haq, Q. M. u. (2023). Understanding of Convolutional Neural Network (CNN): A Review. *International Journal of Robotics and Control Systems*, 2(4), 739-748. https://doi.org/10.31763/ijrcs.v2i4.888

Quarterbacks: Finding Weakness and Strength in Zone vs. Man Coverage. (2023, August 1). Capital QB's. Retrieved August 14, 2025, from https://www.capitalqbs.com/quarterbacks-finding-weakness-and-strength-in-zone-vs-man-coverage/ (Non-peer-reviewed)

Sanchez-Lengeling, B., Reif, E., Pearce, A., & Wiltschko, A. B. (2021, September 2). A Gentle Introduction to Graph Neural Networks. *Distill*. https://doi.org/10.23915/distill.00033 (Non-peer-reviewed)

Song, H., Jazaery, M. A., Ding, H., Cheong, L. L., Jung, J., Band, M., Chi, M., & Bliss, T. (2023). Explainable Defense Coverage Classification in NFLGames using Deep Neural Networks. *Amazon ML Solutions Lab*. https://www.amazon.science/publications/explainable-defense-coverage-classification-in-nfl-games-using-deep-neural-networks

Taylor, C. (2020). Deep Learning for In-Game NFL Predictions. https://cs230.stanford.edu/projects_winter_2020/reports/32263160.pdf#page=1.33 (Non-peer-reviewed)

Review Report

Manuscript: Grids to Graphs: GNNs Improve NFL Defensive Coverage Classification

Summary

The manuscript explores the application of graph neural networks (GNNs) compared to convolutional neural networks (CNNs) in classifying NFL defensive coverage using player tracking data. The author uses the 2025 NFL Big Data Bowl dataset, compares metrics such as accuracy, F1, and AUC, and argues that GNNs provide more balanced predictions.

Major Comments

1. Novelty and Scope (Page 2, Lines 1–16): The framing is strong but the introduction should state more explicitly how this work differs from prior studies such as Song et al. (2023).

I appreciate the reviewer's feedback. I've added a section at the end of the introduction explaining how this research differs from past works. In particular, I've explicitly stated how it differs from the study by Song et al. (2023), as the paper is frequently mentioned in the introduction.

2. Methods (Page 3, Lines 9–15; Page 4, Lines 65–74): The architecture description is detailed but heavy. A comparative diagram of CNN vs. GNN pipelines would enhance clarity.

I agree with the reviewer's comments. The diagram in Figure 1 has been updated to enhance clarity. Additionally, a figure labeled Figure 2 has been added to the method section, illustrating the different pipelines used by the CNN and GNN models.

3. Results (Page 6, Lines 10–20): Metrics are reported clearly, but slight inconsistencies exist (AUC 0.920 vs. 0.923, 0.904 vs. 0.910). Ensure consistency throughout.

Thank you for pointing out this inconsistency. The metrics have now been corrected to remain consistent throughout the paper.

4. Future Work (Page 7, Lines 22–28): The suggestion of temporal learning is strong. Briefly outline how recurrent or temporal GNNs could address limitations of static frames.

This is a great suggestion. The future work section has now been expanded to briefly explain the advantages of using recurrent or temporal GNNs to address the challenges faced by using static frames.

Minor Comments

- Abstract: Replace 'more suitable' with 'better suited' for smoother expression.

Changes have been made.

- Figures: Verify all figures are numbered sequentially and referenced in the text (e.g., Figure 2 vs. Figure 6).

The tables, equations, and figures are now correctly labeled and in sequential order.

- References (Page 9, Lines 61–75): Some sources are informal (e.g., blogs). Mark clearly as non-peer-reviewed.

Changes have been made.

Recommendation

Accept with minor revisions.

Review for “Grids to Graphs: GNNs improve NFL defensive coverage classification over CNNs”

In this empirical research paper, the author applies a graph neural network to kinematic data captured during football games to evaluate defensive coverage during NFL games. It is an interesting and strong study that proposes a novel approach to defensive coverage classification and compares it with the more commonly used Convolutional Neural Networks. The new approach has been motivated well and a relevant publicly available dataset has been used here. As such, this paper has methodological novelty and offers some useful intuition on the performance improvements and differences between the two approaches.

The literature has been properly cited and the argument for proposing this novel methodology has been developed coherently and convincingly. Simple language has been used to explain complex concepts and methodologies which makes the paper accessible to a larger audience.

- The methodology has been explained well although I would suggest including a figure illustrating GNNs to emphasize how the graph structure enables quantifying interactions and how this feature makes the output more intuitive and interpretable. This is a key point of the new methodology that could be shown graphically.

I appreciate the reviewer's suggestion. Figure 1 has been updated to illustrate the edges between nodes in the GNN, using color and width to quantify their weight.

- Also, why are all these features needed for classification? They appear largely redundant.

This is a valid concern by the reviewer. The methodology now includes a section following Table 1, which explains why the selection of features was intentional and justifies why they aren’t redundant in this context.

- The choice of binary classification could also be justified more. Specifically, why these two coverage schemes are relevant for discrimination should be explained more.

I appreciate this feedback from the reviewer. A section has been added at the beginning of the methodology to justify the choice of binary classification.

- Nice figures have been used to illustrate the performance of the two methodologies. The Results section describing these could be improved though. Some detail on how the results can be understood and interpreted is missing here. The figure captions explain the figures quite clearly – some of this info should find its way in the main text too. For example, what is plotted is the ROC curve could be described more.

This is a valid point presented by the reviewer. Sections have been added to the results that explain all the metrics and their implications within the scope of this paper.

- The Discussion does a good job at interpreting the findings and providing extra intuition as to why one or the other perform better. However, Figure 6 should be

included in the Results and not only here (try to explain what the confusion matrices show and what they are useful for in the Results section).

I agree with the reviewer's suggestions. The confusion matrices have been relocated to the results section, and additional context has been provided to explain what the matrices and the following graph comparing the models reveal, as well as their implications.

Overall, a strong novel study that would benefit from some improvement in structure and presentation.

My overall recommendation is: Accept with major revisions (acceptance conditional on satisfactory major revisions)

Review Report

Manuscript: Grids to Graphs: GNNs Improve NFL Defensive Coverage Classification

Summary

The manuscript explores the application of graph neural networks (GNNs) compared to convolutional neural networks (CNNs) in classifying NFL defensive coverage using player tracking data. The author uses the 2025 NFL Big Data Bowl dataset, compares metrics such as accuracy, F1, and AUC, and argues that GNNs provide more balanced predictions.

Major Comments

1. Novelty and Scope (Page 2, Lines 1–16): The framing is strong but the introduction should state more explicitly how this work differs from prior studies such as Song et al. (2023).

2. Methods (Page 3, Lines 9–15; Page 4, Lines 65–74): The architecture description is detailed but heavy. A comparative diagram of CNN vs. GNN pipelines would enhance clarity.

3. Results (Page 6, Lines 10–20): Metrics are reported clearly, but slight inconsistencies exist (AUC 0.920 vs. 0.923, 0.904 vs. 0.910). Ensure consistency throughout.

4. Future Work (Page 7, Lines 22–28): The suggestion of temporal learning is strong. Briefly outline how recurrent or temporal GNNs could address limitations of static frames.

Minor Comments

- Abstract: Replace 'more suitable' with 'better suited' for smoother expression.

- Figures: Verify all figures are numbered sequentially and referenced in the text (e.g., Figure 2 vs. Figure 6).

- References (Page 9, Lines 61–75): Some sources are informal (e.g., blogs). Mark clearly as non-peer-reviewed.

Recommendation

Accept with minor revisions.

Review for “Grids to Graphs: GNNs improve NFL defensive coverage classification over CNNs”

In this empirical research paper, the author applies a graph neural network to kinematic data captured during football games to evaluate defensive coverage during NFL games. It is an interesting and strong study that proposes a novel approach to defensive coverage classification and compares it with the more commonly used Convolutional Neural Networks. The new approach has been motivated well and a relevant publicly available dataset has been used here. As such, this paper has methodological novelty and offers some useful intuition on the performance improvements and differences between the two approaches.

The literature has been properly cited and the argument for proposing this novel methodology has been developed coherently and convincingly. Simple language has been used to explain complex concepts and methodologies which makes the paper accessible to a larger audience.

- The methodology has been explained well although I would suggest including a figure illustrating GNNs to emphasize how the graph structure enables quantifying interactions and how this feature makes the output more intuitive and interpretable. This is a key point of the new methodology that could be shown graphically.
- Also, why are all these features needed for classification? They appear largely redundant.
- The choice of binary classification could also be justified more. Specifically, why these two coverage schemes are relevant for discrimination should be explained more.
- Nice figures have been used to illustrate the performance of the two methodologies. The Results section describing these could be improved though. Some detail on how the results can be understood and interpreted is missing here. The figure captions explain the figures quite clearly – some of this info should find its way in the main text too. For example, what is plotted is the ROC curve could be described more.
- The Discussion does a good job at interpreting the findings and providing extra intuition as to why one or the other perform better. However, Figure 6 should be included in the Results and not only here (try to explain what the confusion matrices show and what they are useful for in the Results section).

Overall, a strong novel study that would benefit from some improvement in structure and presentation.

My overall recommendation is: Accept with major revisions (acceptance conditional on satisfactory major revisions)

The author has addressed the feedback adequately and thoroughly and the paper can now be accepted for publication.

The paper already had strong content and a solid structure. Now clarity and flow have been improved and some missing information has been provided. Specifically, it now clearer how this study builds on past literature and develops it further to make an original contribution.

The methodology has been illustrated graphically in direct comparison with the more standard approach. **I would suggest adding some more detail on why the proposed methodology is superior in terms of architecture and how this network structure produces better performance.**

Also, the results are now reported clearly and concisely and have a strong flow and a direct link to the figures that include the supporting evidence. A few technical terms have been clarified, and the flow of the main narrative has been enhanced**. It would help with interpretation if for example the learned networks were visualised. This would provide more intuition about the mechanistic advantage of the proposed algorithm.**

The choice of input parameters to the models has been justified and more intuition is provided as to how the different parameters complement each other. Finally, a greater depth in the additive value of temporal learning has strengthened this suggestion for future work.

Overall, this is a strong and relevant research contribution and a great addition to the publications of Convergence, well done!