Prescribing Defensive Strategies from Offensive Pre-Snap Behavior

15.905: Machine Learning Under a Modern Optimization Lens

Fall 2024

Data Source: 2025 NFL Big Data Bowl

Azfal Peermohammed & James Pinter

Abstract

This paper presents a novel framework for optimizing defensive strategies in the National Football League using pre-snap offensive behavior data. By combining player tracking data, counterfactual estimation, and Optimal Policy Trees, we develop a method that prescribes defensive schemes to minimize offensive yards gained. We introduce a new player performance metric that evaluates all offensive positions fairly, addressing limitations in traditional fantasy football metrics. Our multi-modal approach incorporates both tabular data and CNN-extracted embeddings from pre-snap animations, achieving a mean absolute error of 11.82 yards in predicting play outcomes. The resulting tree provides actionable defensive recommendations based on offensive formations and player movements, particularly excelling at identifying motion plays. While our model slightly underperforms expert defensive coaches (6.28 versus 5.73 average yards gained by the offense), it demonstrates the potential of machine learning and modern optimization techniques in sports analytics by providing a systematic framework for evaluating "what-if" scenarios in football strategy.

Introduction

With an average of 21 million viewers per week, the National Football League (NFL) contributes significantly to American culture and entertainment. The annual revenue of the NFL increased to $20.2 billion in 2023 as American football continues to gain popularity [Sta24]. What makes the NFL so interesting? Well, the seemingly simple game of progressing an oblong ball down a 120-yard field is surprisingly complex. Offenses must develop strategies to out-maneuver the opposing team while 11 muscular men attempt to prevent the offense from progressing the ball into their endzone. Offensive strategies begin with a pre-snap formation and behavior, providing hints as to what players might do during the play. The 2025 NFL Big Data Bowl centers around these hints and challenges data scientists to use pre-snap information to predict what will happen after the snap. We propose a framework for choosing a defensive strategy that implements counterfactual estimation and Optimal Policy Trees (OPTs), therefore showcasing how machine learning under a modern optimization lens can be applied to the field of sports analytics.

Problem Background & Motivation

Unlike football (also called soccer) in other parts of the world, American football is not a game that is played with a running clock. Instead, the game is a collection of "plays" where the offense iteratively attempts to progress a ball down a 120-yard field. Before each play, there is a "pre-snap" period where the offense plans what they will do to progress the ball down the field. The defense visually analyzes the offense's behavior and quickly develops its strategy to prevent the offense from progressing down the field in the next play. We attempt to increase the defense's ability to minimize the offense's yards gained with machine learning and optimization under the assumption that we can predict how offenses will perform based on information from a 40-second pre-snap interval. Furthermore, we propose a framework that leverages these predictions to provide prescriptions, or defensive strategies, providing a mechanism to analyze "what if" scenarios in football. With a better understanding of optimal strategies to counter specific offensive behavior, defensive coaches can increase the competitiveness of the game, make American football safer, and boost the entertainment of 410 million NFL fans [Gen23].

Data

The NFL provided comprehensive data from weeks 1-9 of the 2022 regular season through their open-ended Big Data Bowl 2025 competition hosted through Kaggle [Kag24]. The dataset includes detailed player tracking information, play-by-play statistics, and game-level data. Table 1 summarizes the primary data sources used in this analysis.

Dataset	**Description**	**Records**
tracking_week_*.csv	Player position coordinates (x,y), speed, acceleration, direction and other features captured at 10Hz	9 files
plays.csv	Play-level information including formations, schemes, and outcomes	24,072 plays
players.csv	Player biographical and physical attributes	1,697 players
games.csv	Game-level information and final scores	144 games

Table 1: Primary data sources from NFL Big Data Bowl 2025

The tracking data provides the foundation for analyzing pre-snap behavior, capturing player movements at a frequency of 10Hz (10 frames per second). Each frame contains the precise location of all 22 players on the field, along with their instantaneous speed, acceleration, and direction of movement. Figure 1 illustrates the coordinate system used for player tracking.

The plays dataset contains detailed information, including offensive and defensive formations, play types (run/pass), and yardage gained for each play. Player biographical data includes physical attributes (height, weight), experience level, and position information, which we used to develop position-specific performance metrics. Game-level data provides contextual information such as venue and final scores.

Figure 1: NFL field coordinate system for player tracking.

Extracting Player Scores

Although the NFL data had x-y coordinates of player positions and basic player statistics, there was no feature that represented how good a particular player was. A defense would obviously want its best players to defend the offense's best players, and x-y coordinates are only half of that evaluation. A common practice uses Fantasy Football metrics to gain an understanding of individual player performance, but we felt this was a biased metric lacking fair evaluation for all positions. For example, 3 of the 7 offensive positions (the linemen), and all of the defensive positions are not measured using Fantasy Football metrics. We therefore developed our own method for determining how good a player is, which we termed as a `player_score`. Our initial scoring process began by merging play-by-play data with individual player statistics. This combined dataset allowed for a comprehensive view of each player's performance across multiple games. Player performance metrics were then aggregated by NFL player ID `nflId`, creating separate offensive and defensive statistical profiles for each player. We developed a weighting system that emphasized the most relevant statistics for each position.

For example, in the table below, the `player_score` for a particular Quarterback (QB) is a weighted sum of his passing yards, running yards, and expected points added (EPA, which measures how well a player performs based on their current field position, down, and other situational factors on each play).

Position	**Primary Metrics**	**Weights**
QB	Passing Yards, Running Yards, EPA	0.3, 0.3, 0.4
WR	Receiving Yards, Receptions, Targets	0.3, 0.3, 0.4
C/G/T	Pressure Allowed, Time to Pressure, EPA	-0.3, 0.3, 0.4

Table 2: Example position-specific metrics & weights.

To calculate every player's `player_score`, each metric was first normalized to a scale between 0 and 1. The normalized values were then multiplied by their position-specific weights and summed to create a composite score. This approach allowed for fair comparisons between positions while still accounting for the unique contributions of each role.

Robust Score Enhancement

To refine the initial scores and account for potential outliers or inconsistencies, we implemented a robust linear regression model for five key positions: Center (C), Guard (G), Tackle (T), Wide Receiver (WR), and Tight End (TE). We decided to only enhance the `player_score` of these 5 positions due to the additional uncertainty in performance specific to their position. Weather conditions, field type, and mild injuries all disproportionally affect these positions due to the nature of their offensive roles [Bur10]. This approach aimed to create a more statistically robust and reliable `player_score` metric for C, G, T, WR, & TE. The feature set for each position was carefully curated to include the most relevant statistics. These features were then standardized to prevent any single metric from dominating the model.

We employed Huber Regression, a robust form of linear regression that is less sensitive to outliers than ordinary least squares regression. The Huber loss function for each of the 5 positions is given by [FW02]:

$$\min_{\beta} \sum_{i=1}^{n} L_\delta(y_i - f(x_i)) + \alpha\|\beta\|_2^2$$

$$L_\delta(y - f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for } |y - f(x)| \leq \delta, \\ \delta(|y - f(x)| - \frac{1}{2}\delta) & \text{otherwise.} \end{cases}$$

where:

- β: vector of regression coefficients to be optimized
- n: number of players in a specific position
- y_i: actual `player_score` for player i
- $f(x_i)$: predicted `player_score` for player i
- δ: threshold parameter (1.5 in our models) for switching between squared and linear loss
- α: regularization parameter (0.5 in our implementation) that controls the strength of the L2 penalty
- $\|\beta\|_2^2$: L2 norm of the coefficient vector, which helps prevent overfitting
- L_δ: Huber loss function that combines squared loss for small residuals and linear loss for large residuals

The model was implemented with an increased δ value of 1.5 for better outlier handling, and an α value of 0.5 for regularization to prevent overfitting. We also employed adaptive cross-validation based on sample size (referring to the number of players in each position group), using leave-one-out cross-validation for very small samples ($n < 10$) and k-fold cross-validation with increasing k for larger sample sizes. For positions with 10-30 players, we used 3-fold cross-validation, while positions with 30-50 players used 5-fold cross-validation, and positions with more than 50 players used 10-fold cross-validation. This adaptive approach ensured robust validation while maintaining sufficient training data for each position group, which was particularly important given the varying roster sizes across positions.

Score Refinement

The robust regression models were used to predict updated scores for players in the five selected positions. These predicted scores replaced the original scores in the final dataset, while scores for players in other positions remained unchanged. This selective refinement allowed us to improve the reliability of scores in positions where uncertainty in performance was most prominent, while preserving the original scores for positions where the initial heuristic approach was deemed sufficient. The result of this methodology is a set of player performance scores that combine domain expertise in football statistics with advanced statistical techniques, providing a more nuanced and robust evaluation of player performance.

Aggregating Tabular Data

After extracting the `player_score` feature, we conducted aggregation on our tabular data. Although the NFL tracking data was incredibly granular (10 rows per second per player), we instead needed 1 row per play so that each row had an independent value for `yardsGained`, our dependent variable. The general structure of the initial dataframe is depicted in Table 3, below.

playId	frameId	nflId	x	y	dis	position	defense	yards gained
1	1	01234	0.84	23.12	26.78	T	Cover 2	7
1	2	01234	0.84	23.73	26.86	T	Cover 2	7
1	1	56789	0.76	22.97	12.52	WR	Cover 4	7
1	2	56789	0.76	22.83	12.64	WR	Cover 4	7
...	...	...	...	...	...	...	...	...
2	1	01234	0.84	26.88	28.36	T	Nickle	4
2	1	56789	0.76	25.61	14.15	WR	Nickle	4

Table 3: Original player tracking data for frames of plays.

Thus, for each player, we extracted aggregated features such as total distance traveled before the snap along with details like his final orientation, speed, acceleration, and location (in x-y coordinates) at the snap. We only included details about offensive players to limit the number of features in the aggregated dataframe. These player features were added to a dataframe with additional details about the play, including yards needed for a first down and direction of the play.

An example of the first three rows of our aggregated dataset (with only a few of the features included) is depicted in Table 4.

playId	club	yardsToGo	x_1	x_11	dis_1	dis_11	position_1	position_11	defense	yardsGained
1	BOS	15	24.38	27.34	0.31	0.06	RB	T	Man	7
2	BOS	10	52.26	52.07	0.01	0.01	G	WR	Zone	4
3	SEA	8	54.28	53.54	0.4	0.02	TE	RB	spec	-2

Table 4: Aggregated play data with a subset of features.

The columns in our dataset follow a specific naming convention to represent features for each of the 11 offensive players on the field. For any given feature (like x-coordinate, distance, or speed), there are 11 corresponding columns, one for each player, denoted by `<feature>_1` through `<feature>_11`. For example, `x_1` represents the final x-coordinate of player 1, while `x_11` represents the final x-coordinate of player 11. The integer suffix (1-11) serves as a player identifier within each play, though this identifier may not consistently represent the same player across different plays.

Each player has multiple features describing their pre-snap behavior: `x_n, y_n` (final coordinates at snap); `s_n` (speed at snap, yards/second); `a_n` (acceleration at snap, yards/second^2); `dis_n` (total distance traveled during pre-snap); `o_n` (orientation at snap, degrees); `dir_n` (direction of movement at snap, degrees); `total_distance_n` (cumulative distance traveled before snap); `position_n` (player's position, e.g., RB, WR, T); `player_score_n` (performance score based on pre-snap movement).

Play-level features like `yardsToGo`, `club`, and `defense` apply to the entire play rather than individual players. The target variable `yardsGained` represents the outcome of the play.

Additionally, we aggregated defensive strategies to serve as our treatment variable, `defense`. There were 36 defensive schemes present in the initial dataset, far too many to use for counterfactual estimation and OPT development. It is widely accepted that there are three main categories of defensive strategy: man coverage, zone coverage, and a special hybrid of the two [Dai23]. We therefore categorized each of the 36 defensive schemes as `Man`, `Zone`, or `spec` using our knowledge of American football, stemming from 16 years of playing and watching the sport.

Aggregating our tabular data helped us properly define the following:

- **Data X:** row-wise pre-snap information by play, including individual player and general details;
- **Treatment `defense`:** the category of defensive scheme used to counter the offense on a play;
- **Outcome `yardsGained`:** the number of yards gained by the offense on a play. The defense's goal is to minimize this dependent variable.

Extracting Embeddings from Animations

The aggregation of the tabular data made it possible to predict the `yardsGained` per play, but it was disappointing to lose the granularity of the original data. We endeavored to regain information by developing animations that visualized pre-snap behavior of both the offense and defense.

Additionally, we wanted our model a capture a crucial component of pre-snap player behavior: "motion," an offensive strategy that enables a player to perform a running start upon the snap instead of having to begin stationary. Typically, a motion is a strong indicator of who will be getting the ball during the play. Though this feature is partially captured by the `dis_n` (total distance) and `s_n` (speed at the snap) columns, we lose spatial and temporal information with respect to defensive players. To account for this loss of information, we decided to incorporate multi-modal machine learning: a process which leverages a variety of data sources to resemble real-world phenomena.

We used the tabular data to make animations for each play by plotting all 22 players on the field for every frame of each unique play. We removed the end zone and yard lines of the football field to let our model focus on the player positions rather the profound edges. Refer to Figure 2 for one frame of an animation.

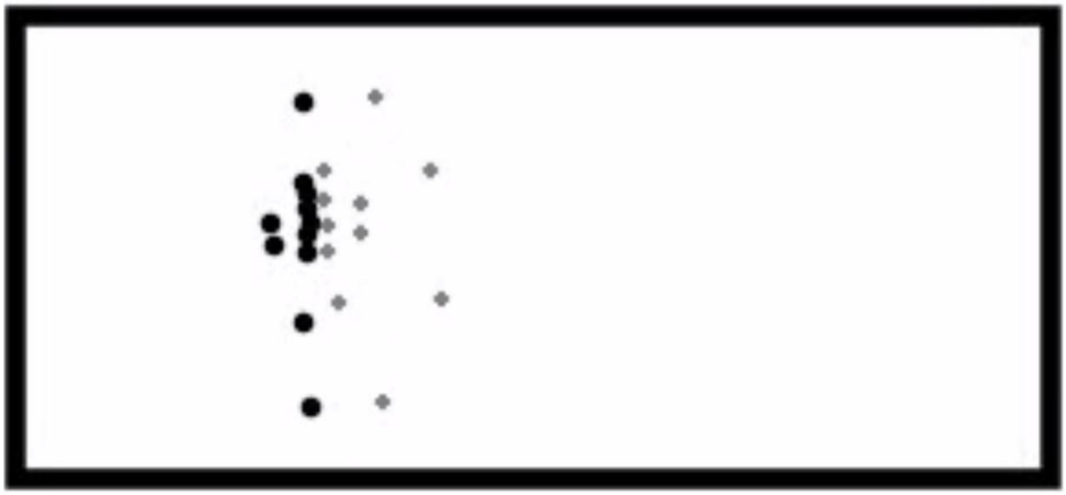

Figure 2: One frame depicting the pre-snap location the offense (black circles) and defense (gray pluses).

The frames were then encoded into 4 dimensional tensors represented by (batch size, num frames, pixel height, pixel width) or gray scale images. The frames were also downsamples for computational reasons.

Once the videos were batched into tensors, we utilized a convolutional neural network (CNN) to run a supervised learning problem regressing on `yardsGained` for meaningful embedding extraction to be used in conjunction with our tabular data. 3-D convolutional blocks were used to capture the temporal aspect of the animation frames. The CNN was trained from scratch with the belief that a smaller network would extract the patterns within the animations due to the simplicity of the images. We adjusted the kernel sizes, number of filters, and number of convolutional layers. Our final CNN achieved a loss of roughly 8 yards.

We then passed each play through our trained CNN to extract a ten-dimensional vector. We also applied principle component analysis (PCA) to restrict the embeddings to one column, encouraging our OPT to not choose this column when we viewed our OPT. Our data engineering produced four possibilities of input data for our model: solely tabular data, solely animation embeddings, tabular data paired with animation embeddings, and tabular data paired with PCA animation embeddings.

Layer Type	**Output Shape**	**Parameters**
Input	(None, 20, 100, 200, 1)	0
Convolutional Block 1		
Conv2Plus1D	(None, 14, 81, 196, 2)	232
BatchNorm + ReLU	(None, 14, 81, 196, 2)	8
Convolutional Block 2		
Conv2Plus1D	(None, 8, 72, 192, 4)	520
BatchNorm + ReLU	(None, 8, 72, 192, 4)	16
Pooling and Flattening		
GlobalAvgPool3D	(None, 4)	0
Flatten	(None, 4)	0
Dense Layers		
Dense + ReLU	(None, 100)	500
Dense + ReLU	(None, 50)	5,050
Dense + ReLU	(None, 10)	510
Dense + ReLU	(None, 5)	55
Dense	(None, 1)	6
	Total Parameters	**6,897**

Table 5: CNN Architecture for Feature Extraction.

Methods

Counterfactual Estimation

Before prescribing defensive strategy, we estimate a rewards matrix for our counterfactuals. We refer to causal inference literature and employ the doubly robust method, which is a balance of both the direct and the inverse propensity weighting methods. We use Interpretable AI's categorical regression rewards estimator since our outcome `yardsGained` is a continuous variable. The model of choice for both the direct method and the inverse propensity weighting is a random forest classifier to estimate the probability distribution of each play getting assigned a defensive treatment and a random forest regressor to predict `yardsGained` with the direct method. The doubly robust method combines the strengths of both approaches using the formula:

$$\Gamma_{it} = f_t(x_i) + 1\{T_i = t\} \cdot \frac{1}{p_{it}}(y_i - f_t(x_i))$$

Where $f_t(x_i)$ represents the direct method prediction for treatment `defense` (either `Man`, `Zone`, or `spec`), and $\frac{1}{p_{it}}$ is the inverse propensity weighting adjustment. This method maintains high-quality estimates if either the propensity scores or outcome estimates are accurate, making it particularly suitable to our football application where defensive assignments may have inherent biases. For example, certain defensive schemes might be more commonly used against specific offensive formations, regardless of their effectiveness. The structure of the reward estimation process is depicted below.

$$\underset{\mathbf{X}}{\begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \end{bmatrix}} \quad \underset{\texttt{defense}}{\begin{bmatrix} \texttt{Man} \\ \texttt{Zone} \\ \texttt{spec} \end{bmatrix}} \quad \underset{\texttt{yardsGained}}{\begin{bmatrix} y_{11} & ? & ? \\ ? & y_{22} & ? \\ ? & ? & y_{33} \end{bmatrix}} \rightarrow \underset{\boldsymbol{\Gamma}}{\begin{bmatrix} \Gamma_{11} & \Gamma_{12} & \Gamma_{13} \\ \Gamma_{21} & \Gamma_{22} & \Gamma_{23} \\ \Gamma_{31} & \Gamma_{32} & \Gamma_{33} \end{bmatrix}}$$

Evaluating the Rewards Matrix

We then evaluated our rewards matrix to ensure that counterfactual outcomes (`yardsGained`) were accurately predicted. We computed the mean absolute error (MAE) for `yardsGained` between the estimated counterfactuals and the ground truth scenarios (the defensive strategy that was actually chosen). Our best model (marked with $\star$), trained using both tabular & animation embeddings with PCA, obtained an MAE of 11.82 yards. A model performance summary with various data inputs can be referenced in Table 6.

Data used to train prediction model	**Test MAE**
Tabular Data	12.71
Animation Embeddings	16.80
Tabular + Embeddings (No PCA)	12.14
Tabular + Embeddings (with PCA)	11.82 $\star$

Table 6: Model performance comparison with different data inputs.

Obtaining Prescriptions with an OPT

Once the accuracy from the evaluation step of the rewards matrix was deemed satisfactory, the rewards were fed into an OPT. We tuned the `max_depth` parameter and determined that a value of 5 struck the best balance between performance and interpretability. Refer to the resulting OPT in Figure 3.

The remarkable aspect of our OPT is its ability to generate intuitive policies. However, we were unable to devise a method to ensure that the identifying integer suffix (`_n`) would consistently represent the same position across different plays. This limitation arises because teams may use different player combinations for various offensive plays. To address this limitation, we analyzed the position distribution associated with each identifying integer suffix. We found that WRs represent roughly a quarter of the distributions for suffixes 7 and 9, making it considerably more prominent for those two suffixes than the other 6 offensive positions.

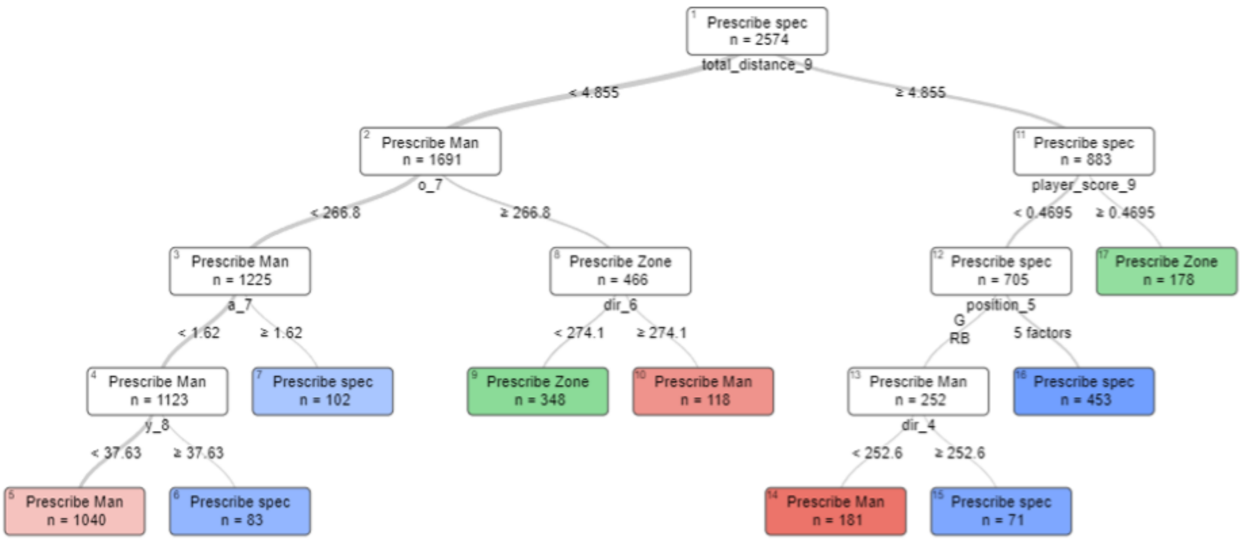

Figure 3: Decision tree showing prescribed defensive strategy based on pre-snap features. Each node shows the prescribed defensive strategy and number of samples (n).

Consider the OPT above. One of its rules states that when the total distance covered by player 9 — a position most commonly associated with a WR — is greater than 4.655 yards, and his `player_score` exceeds 0.4695, the defense should adopt a `Zone` strategy. This rule effectively captures the offensive strategy of motion described earlier, as WRs are the position that use the motion most commonly [Fro23].

We evaluated our OPT by calculating the mean of the counterfactual outcome variable `yardsGained` and comparing to the actual outcomes. Our policy tree achieved an average of 6.28 yards gained by the offense. NFL coaches, whose defensive strategies were actually implemented and lead to the ground truth `yardsGained` per play, outperformed our model by 0.55 yards, averaging 5.73 yards gained by the offense.

Discussion

Our framework demonstrates a successful application of modern optimization and machine learning techniques to sports analytics, achieving defensive play predictions that nearly match expert human performance. While our policies performed slightly worse than those devised by coaches (6.28 vs 5.73 average yards gained by the offense), the model's ability to compete with experienced NFL strategists is remarkable, especially considering it lacked defensive player data and contextual information.

The project's strength lies in its practical utility for defensive strategy optimization. By combining player tracking data, sophisticated counterfactual estimation, and OPTs, we've created a tool that enhances strategic decision-making in three key ways. First, it provides coaches with the ability to evaluate "what-if" scenarios, offering an invaluable resource for game preparation and player development. Second, the model's ability to identify meaningful patterns in pre-snap behavior, particularly in detecting motion plays through WR movement patterns, demonstrates its potential for increasing game competitiveness. Third, our framework's emphasis on zone coverage recommendations when detecting high-motion plays aligns with modern defensive trends that prioritize player safety while maintaining defensive effectiveness.

Moreover, our project exemplifies the power of machine learning under a modern optimization lens in sports analytics. Our doubly robust approach to counterfactual estimation, combined with optimal policy trees, provides interpretable and actionable insights while handling the inherent complexities of football strategy. While our analysis used data beyond the pre-snap period—unlike the real-time constraints faced by coaches—the framework's value lies not in replacing human decision-making but in augmenting it. The model serves as a powerful tool for strategy evaluation, player development, and understanding the relationship between offensive behavior and optimal defensive responses.

These results support our initial claims regarding enhanced game competitiveness, player safety, and entertainment value through data-driven defensive strategies. The framework's ability to nearly match expert performance while providing interpretable, actionable insights demonstrates its potential for practical application in NFL defensive coaching and strategy development.

Conclusion

Our project successfully demonstrates the application of modern machine learning and optimization techniques to NFL defensive strategy. By combining player tracking data, counterfactual estimation, and optimal policy trees, we developed a framework that nearly matches expert human performance, achieving an average of 6.28 yards gained compared to coaches' 5.73 yards.

Limitations & Future Work

While our results are promising, several limitations present opportunities for future research. The exclusion of defensive player data, though necessary to manage dimensionality, limited our model's contextual understanding. Future work could explore dimensionality reduction techniques that preserve defensive information while maintaining computational feasibility. Additionally, our inability to consistently associate player numbers with specific positions affected the interpretability of our optimal policy tree.

Future research directions include:

- Developing methods to incorporate defensive player data efficiently
- Creating position-consistent player identification systems
- Expanding the model to include game-specific contexts like score and time
- Investigating real-time implementation possibilities for in-game decision support

Final Thoughts

This project represents a significant step forward in applying data science to sports strategy. Our framework not only provides valuable insights for defensive play-calling but also demonstrates the potential for machine learning to enhance decision-making in complex, real-world scenarios. While our model slightly underperformed compared to NFL coaches, its ability to provide interpretable, data-driven recommendations and evaluate hypothetical scenarios makes it a valuable tool for strategic analysis and coaching support in professional football.

Partner Contributions

We collaborated equally throughout all aspects of the project. James focused on the player score development, including the robust enhancement methodology and position-specific metrics. Azfal led the development of the animations and CNN architecture, including the extraction of embeddings and implementation of PCA for dimensionality reduction. We jointly worked on data engineering, counterfactual estimation, and OPT development. The final analysis, interpretation of results, and paper writing were shared responsibilities between both team members.

References

[FW02] John Fox and Sanford Weisberg. "Robust regression". In: *An R and S-Plus companion to applied regression* 91 (2002), p. 6.

[Bur10] Brian Burke. *Valuing Offensive Line Performance*. Published on November 2, 2010. Advanced Football Analytics. 2010. URL: `https://www.advancedfootballanalytics.com/2010/11/valuing-offensive-line-performance.html` (visited on 12/06/2024).

[Dai23] Daily Planet DC. *Game Changers: The Evolutionary Saga of NFL Offense and Defense Strategies*. Daily Planet DC. Sept. 2023. URL: `https://dailyplanetdc.com/2023/09/10/game-changers-the-evolutionary-saga-of-nfl-offense-and-defense-strategies/` (visited on 12/06/2024).

[Fro23] Front Proof Media. *The Revolution of Motion Offense in Today's NFL*. Front Proof Media. 2023. URL: `https://www.frontproofmedia.com/football/the-revolution-of-motion-offense-in-todays-nfl` (visited on 12/06/2024).

[Gen23] Genius Sports. *In Focus — Who are NFL fans and how can brands engage them?* Genius Sports Content Hub. 2023. URL: `https://www.geniussports.com/content-hub/in-focus-nfl-fans-how-can-brands-engage/` (visited on 12/06/2024).

[Kag24] Kaggle. *NFL Big Data Bowl 2025*. Competition dataset. Kaggle. 2024. URL: `https://www.kaggle.com/competitions/nfl-big-data-bowl-2025/data` (visited on 12/06/2024).

[Sta24] Statista Research Department. *Total revenue of all National Football League teams from 2001 to 2023*. Accessed on December 6, 2024. Statista. 2024. URL: `https://www.statista.com/statistics/193457/total-league-revenue-of-the-nfl-since-2005/` (visited on 12/06/2024).