

Analytics, have some humility: a statistical view of fourth-down decision making

Ryan S. Brill* and Abraham J. Wyner[†]

November 8, 2023

Abstract

Expected points (EP) and win probability (WP) are value functions fundamental to strategic in-game decision making in American football, particularly for fourth down decision making. The EP and WP functions which are widely used today are statistical models fit from historical data. These models, however, are subject to serious statistical flaws: selection bias, overfitting, ignoring autocorrelation, and ignoring uncertainty quantification. We develop a machine learning framework that accounts for these issues and extracts our analysis into a decision-making inference. Along the way, we introduce a novel methodological approach to mitigate overfitting in machine learning models. Specifically, we extend the catalytic prior, initially developed in the context of linear models, to smooth our tree machine learning models. Our final product is a major advance in fourth-down strategic decision making: far fewer fourth-down decisions are as obvious as analysts claim.

1 Introduction

In-game strategic decision making is one of the fundamental objectives of sports analytics. To mathematically compare strategies, analysts need a value function that measures the value of each game-state. The optimal decision maximizes the value of the next game-state. Across sports, however, value functions are not observable quantities; they are defined by models. It is the sports analysts' task to infer the value of each game-state from the massive dataset of all plays in the recent history of a given sport.

*Graduate Group in Applied Mathematics and Computational Science, University of Pennsylvania. Correspondence to: ryguy123@sas.upenn.edu

[†]Dept. of Statistics and Data Science, The Wharton School, University of Pennsylvania

The two most widely used value functions by analysts of American football are expected points and win probability. *Win probability* (WP) measures the probability that the team with possession at the current game-state wins the game. *Expected points* (EP) measures the expected value of the net number of points of the next scoring event in the half given the current game-state.¹ The most prominent example of analysts using these value functions to dictate in-game strategy is fourth-down decision making. On fourth down, a football coach has three choices: go for it (Go), attempt a field goal (FG), or punt the ball (Punt). Initial attempts from [Romer \(2006\)](#) and [Burke \(2009b\)](#) suggest making the decision that maximizes expected points. Modern attempts from [Baldwin \(2021a\)](#) and Burke² suggest making the decision that maximizes win probability.³ These analyses found that NFL coaches are too conservative on fourth down; they often settle for kicks even when they should go for it.

These value functions arise broadly from one of two classes of models, probabilistic state-space models or statistical models. State-space models simplify the game of football into a series of transitions between game-states. Transition probabilities are estimated from play-level data and are then propagated into WP or EP by simulating games. When done right, these models are a good way to estimate value functions, but they are difficult to do: they require a careful encoding of the convoluted rules of football into a set of states and the actions between those states, careful estimation of transition probabilities, and enough computing power to run enough simulated games to achieve desired granularity. Each of these are nontrivial.

On the other hand, statistical models are fit entirely from historical data. Given the results of a set of observed football plays, statistical models fit the relationship between certain game-state variables using data-driven regression or machine learning approaches. These models are widely used today in the football analytics community because rich publicly available play-by-play data (e.g., nflFastR ([Carl and Baldwin, 2022](#))) and powerful off-the-shelf machine learning models (e.g., XGBoost ([Chen and Guestrin, 2016](#))) have become widely accessible. Additionally, due to a perceived abundance of data, these machine learning models are viewed as more trustworthy than previous mathematical models that make more simplifying assumptions. For these reasons, we focus on statistical EP and WP models in this paper.

These statistical models, however, are subject to several serious statistical issues. By not adjusting for team quality, EP models are victims of selection bias.⁴ Hence we must adjust for team quality.

¹Net points is relative to the team with possession (negative if the opposing team is expected to score more points next).

²Burke's method is proprietary.

³Burke (Twitter: @bburkeESPN) and Baldwin (Twitter: @ben_bot_baldwin) post their fourth down recommendations on Twitter during and after real football games.

⁴Because good teams have more plays and good teams score more points, EP for randomly drawn teams is different from EP for average teams, *ceteris paribus*.

The task is then to fit a complicated function of many variables that interact and have nonlinear relationships. Additive regression models underfit, but blackbox machine learning models like XGBoost aggressively overfit. WP models, moreover, do not account for the auto-correlated nature of football play-by-play data.⁵ Although a dataset of 511,264 plays from 2006 to 2021 may seem large enough to fit an accurate statistical WP model, it actually is not. In particular, auto-correlation reduces the effective sample size and inflates WP standard errors. This uncertainty should percolate into the fourth-down decision procedure, which currently treats EP and WP as known quantities rather than estimates.

We address the aforesaid issues with these statistical models. To mitigate selection bias, we create measures of team quality and adjust for them. Including these additional covariates exacerbates overfitting of our machine learning models. To reduce overfitting and smooth these models, we introduce a novel methodological approach: we extend the catalytic prior to our machine learning framework. Initially developed in the context of linear models (Huang et al., 2020, 2022), the catalytic prior involves using synthetic data imputed from a simpler smoother model as a prior for a more complex model with interactions. We find that a catalytic prior from a simpler additive regression model effectively “Laplace smooths” our tree machine learning models. Then, we construct bootstrapped WP confidence intervals which account for auto-correlation, and validate our methods on a simulated random walk version of football in which true win probabilities are known. Finally, we percolate uncertainty estimates through the fourth-down decision procedure using bootstrap voting. Our final product is a major advance in fourth-down strategic decision making: far fewer fourth-down decisions are as obvious as analysts claim. Thus, we ask football analysts to have some *humility*: for many game-states, there is simply *not enough data* to know which decision is optimal.

The remainder of this paper is organized as follows. In Sections 2 and 3, respectively, we discuss statistical EP and WP models, including model specifications, issues, and how we address those issues. In Section 2 we also detail our extension of catalytic priors to our machine learning context. We discuss our improved fourth down decision procedure in Section 4 and conclude in Section 5.

2 Expected points models

The mathematical approach to in-game strategic decision making is to make the decision which maximizes the value of the next game-state. The first value function used for fourth-down decision making in American football (by Romer (2006)) was *expected points* (EP), the expected value of

⁵Concisely, *every game has only one winner*. The binary win/loss response values are not independent, as all plays from the same game share the exact same draw of the win/loss outcome.

the net number of points of the next score in the half. In this Section, we discuss statistical expected points models in detail. We begin with an overview of existing open-source expected points models. These models are fit from historical data but don't adjust for team quality, leading to selection bias. Hence we need to adjust for team quality. This task is not easy: we need to fit expected points as a function of team quality, yardline, down, yards to go, time remaining, timeouts, and more game-state variables. We want to capture the nonlinear and interacting relationships between these variables, but we don't want to overfit. Moreover, we find that widely used XGBoost EP models overfit even without adjusting for team quality. To mitigate overfitting, we use a catalytic prior to shrink XGBoost towards a simpler smoother model. We find that our catalytic machine learning model indeed performs best.

2.1 Problems with statistical expected points models

In Table 1 we summarize well known open-source statistical expected points models. The covariates \mathbf{x} encode the game-state. There are seven potential outcomes of the next scoring event in the half after the current play,

$$\left\{ \begin{array}{l} \text{Touchdown (7), Opp. Team Touchdown (-7),} \\ \text{Field Goal (3), Opp. Team Field Goal (-3),} \\ \text{Safety (2), Opp. Team Safety (-2),} \\ \text{No Score (0)} \end{array} \right\}. \quad (2.1)$$

Each event has an associated value, the net points scored from the event. Earlier approaches from [Romer \(2006\)](#) and [Burke \(2009a\)](#) treat EP as a regression problem in which the outcome variable is a real number, the net points of the next score. More recent approaches from [Yurko et al. \(2018\)](#) and [Baldwin \(2021b\)](#) treat EP as a classification problem in which the outcome variable is categorical. Given estimated outcome probabilities, the expected points at game-state \mathbf{x} is given by the weighted sum

$$\widehat{\text{EP}}(\mathbf{x}) = \sum_k k \cdot \widehat{\mathbb{P}}(y = k | \mathbf{x}). \quad (2.2)$$

We include detailed specifications of these models in Appendix A.1.

None of these expected points models adjust for team quality. There are two reasons for this. First, these models are viewed as representing EP for an average offense facing an average defense, and so imply decision making for average teams. For instance, [Romer \(2006\)](#) writes that his EP model represents the “expected long-run value ... of the difference between the points scored by the team with the ball and its opponent when the two teams are evenly matched, average NFL teams.” Also,

modeler	model	game-state variables	training set	outcome variable
Romer (2006)	instrumental variables regression	yardline	all plays	points of the next score (a real number)
Burke (2009a)	linear regression with a spline	yardline	first down plays	\wedge
Yurko et al. (2018)	multinomial logistic regression	transformations of yardline, down, yards to go, time remaining	all plays	outcome of the next score (categorical)
Baldwin (2021b)	XGBoost	yardline, down, yards to go, time remaining, home, era, roof type, timeouts	all plays	\wedge

Table 1: Summary of well known open-source statistical expected points models.

it is not easy to adjust for team quality alongside all the other game-state variables, as they have nonlinear and interacting relationships.

To illustrate why not adjusting for team quality causes problems, we conduct a thought experiment via three questions.

1. What is the probability that an “*average*” *NFL kicker* sinks a 70 yard field goal in neutral weather conditions?
2. What is the probability that *Justin Tucker*⁶ sinks a 70 yard field goal in neutral weather conditions?
3. What is the probability that a *randomly drawn kicker*⁷ sinks a 70 yard field goal in neutral weather conditions?

Justin Tucker made a 66 yard field goal in 2021, the longest field goal in NFL history. Just one

⁶Justin Tucker is widely considered the best NFL kicker of all time. In Appendix C.3 we show that Tucker has the highest career mean “kicker quality” of all kickers in our dataset.

⁷Randomly drawn from our dataset of all field goal attempts from 2006 to 2021.

kicker has made a 64 yard field goal, just 6 kickers have made a 63 yard field goal, and each of these kickers were above average. So, for purposes of this thought experiment, suppose an “average” kicker has 0 probability of sinking a 70 yard field goal. On the other hand, since Justin Tucker is the best kicker in the NFL, suppose he has some $\varepsilon > 0$ probability of sinking a 70 yard field goal. Then a randomly drawn kicker also has a positive probability, since we can randomly draw Justin Tucker.

This thought experiment highlights the difference between an average player and a randomly drawn player. Expected points models, which don’t adjust for team quality, report EP for randomly drawn teams, not for average teams. This causes problems for several reasons. First, no team wants an EP value for a randomly drawn team, and there is no such thing as a decision made by a random team. The 2022 Philadelphia Eagles have a great offense and want it’s team’s EP. Failing to adjust for team quality also introduces selection bias. Specifically, good teams have more plays and good teams score more points, which we visualize in Figure 1. As a result, expected points for a team from a randomly drawn play from our historical dataset is not the same as expected points for an average team, all else equal. So, these statistical EP models are biased, and a better model should adjust for team quality.⁸

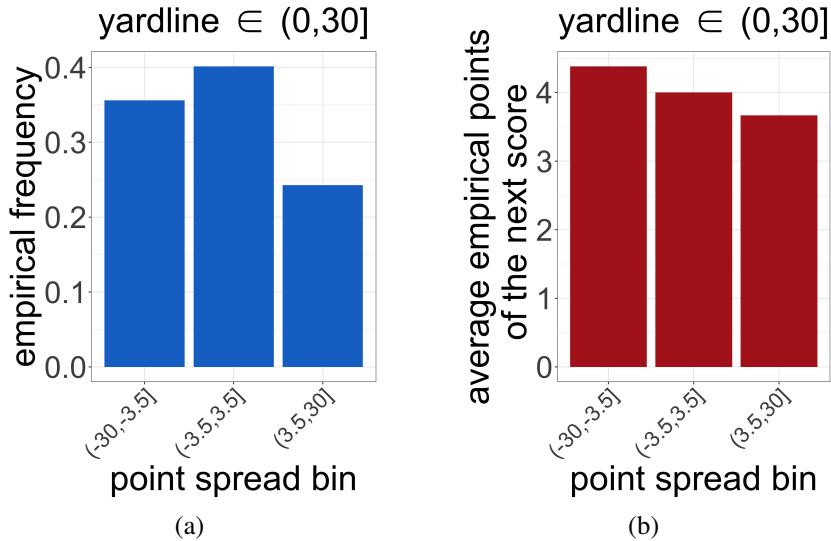


Figure 1: Good teams have more plays (a) and good teams score more points (b) conditional on being near the opposing team’s endzone. Specifically, on the left: a higher proportion of plays feature good teams (large negative point spreads) than bad teams (large positive point spreads). On the right: the average empirical net points of the next score is higher for good teams.

These expected points models also don’t adjust for score differential, leading to further bias, which

⁸Today, some proprietary expected points models do adjust for team quality. For instance, Burke’s EP model at ESPN accounts for team strength using FPI (football power index).

we discuss in detail in Appendix A.3. To mitigate selection bias and score differential bias, we include score differential and measures of offensive and defensive quality as covariates. Point spread is the easiest way to adjust for the difference in strength between the offensive and defensive teams. Additionally, in Appendix A.4, we create our own measures of offensive and defensive quality which leverage information from each of a team’s previous plays while carefully controlling for data bleed.

An interesting football lesson that arises from modeling EP using our own team quality metrics is that offensive quality is more predictive of points than defensive quality (see Appendix A.5). Further, using a modified version of the knockoffs procedure from [Candes et al. \(2016\)](#) and [Ren et al. \(2020\)](#) to fit the autocorrelated nature of our football play-by-play dataset, we find that each of the offensive quality metrics and none of the defensive quality metrics are significant (see Appendix A.6). In other words, our defensive quality metrics are not significantly distinguishable from noise in predicting the points of the next score. This provides further evidence that offense matters more than defense for scoring points.

Back to the task at hand, we want to fit an EP model as a function of many parameters – yardline, yards to go, time remaining, team quality, score differential, etc. – replete with nonlinear relationships and interactions. We want to fit a flexible enough model to capture a sufficiently complex relationship, but we don’t want to overfit. An additive model like multinomial logistic regression underfits and lack interactions. For example, expected points via multinomial logistic regression doesn’t capture the right trajectory of EP at the end of the half.⁹ A flexible machine learning model like XGBoost has the capacity to fit such a complicated function given enough data, but in practice we find that it overfits. For example, expected points via XGBoost classification isn’t monotonic in certain variables such as yardline even though it should be.¹⁰ Whereas expected points via XGBoost regression is monotonic in these variables, XGBoost regression also has trouble fitting EP at the end of each half.¹¹ We visualize the underfitting and overfitting of these models in Figure 2. We find further evidence that XGBoost overfits in its relatively poor out-of-sample predictive performance in our EP model comparison (Section 2.3).

⁹At the end of the half, EP should converge to 0 far from the opponent’s endzone (large yardlines). EP from multinomial logistic regression wrongly converges to a positive number (Figure 2a), whereas EP from XGBoost classification correctly converges to 0 (Figure 2c).

¹⁰EP should be increasing in yardline even though it isn’t as shown in Figure 2d (yardlines 92-94). We can’t use monotonicity constraints in XGBoost classification because the probability that the next score is a field goal, for instance, is not monotonic in yardline.

¹¹We can leverage monotonicity constraints in XGBoost regression because expected points itself is monotonic in yardline. But, in Figure 2b we see that EP via XGBoost regression, like that from multinomial logistic regression, doesn’t converge to 0 at the end of the half. XGBoost classification fits EP at the end of the half better than XGBoost regression because it explicitly fits the probability that there is no next score in the half, which increases as time remaining decreases.

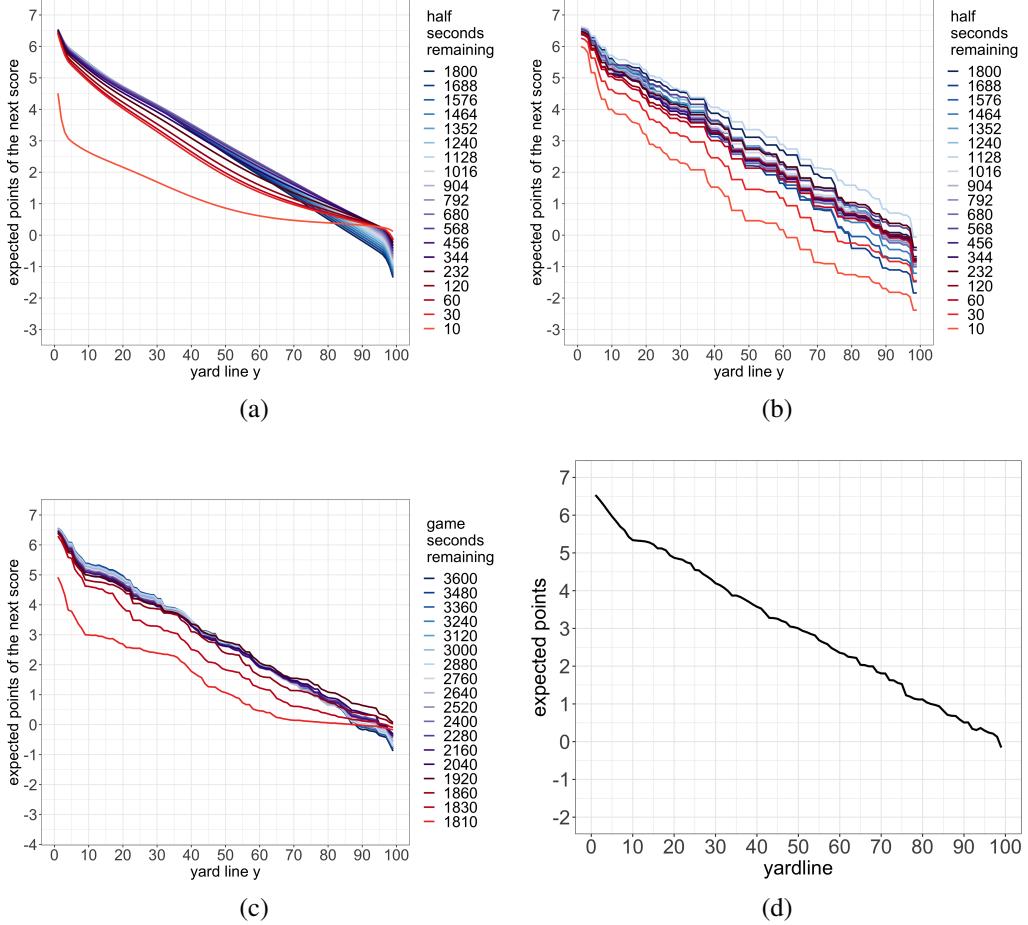


Figure 2: Predicted EP vs. yardline for various values of half seconds remaining according to multinomial logistic regression (Figure (a)), XGBoost regression (Figure (b)), and XGBoost classification (Figure (c)), holding other variables constant. According to multinomial logistic regression and XGBoost regression, \widehat{EP} doesn't converge to 0 far from the oponent's endzone at the end of the half, which is wrong. Although \widehat{EP} correctly converges to 0 far from the oponent's endzone at the end of the half according to XGBoost classification, it is not monotonic in yardline (Figure (d), yardlines 92-94).

2.2 Catalytic prior to mitigate overfitting

We want to use machine learning to capture a complex high-dimensional function, but flexible machine learning models like XGBoost tend to aggressively overfit football play-by-play data. The statistics community typically deals with overfitting using regularization or shrinkage towards simpler models. These methods are well known for parametric models in a Bayesian context, but are much more difficult in the context of blackbox machine learning. Burke aptly summarized this problem on the September 19, 2023 episode of the Wharton Moneyball podcast,¹²

¹²<https://businessradio.wharton.upenn.edu/wharton-moneyball/>

“The in-game models are not Bayesian. Congratulations to you if you can figure out how to do that. Most publicly available models are ... XGBoost models.”

To mitigate the overfitting of machine learning EP models, we extend the idea of a catalytic prior from Huang et al. (2020, 2022) to shrink a complex XGBoost classification model towards a simpler multinomial regression model. The idea is to generate synthetic prior data from a simpler model, the *catalytic prior* model, and then use this synthetic data to represent the prior distribution. The catalytic prior serves as a prior on functions or models, and the goal is to “pull” the complex target model towards the simpler model in order to reduce overfitting. This process is catalytic in that a catalyst in chemistry stimulates a reaction to take place. Here, the reaction is fitting our model. Although catalytic priors were introduced in the context of linear models (Huang et al., 2020, 2022), we are the first, to our knowledge, to use catalytic priors in a machine learning context.

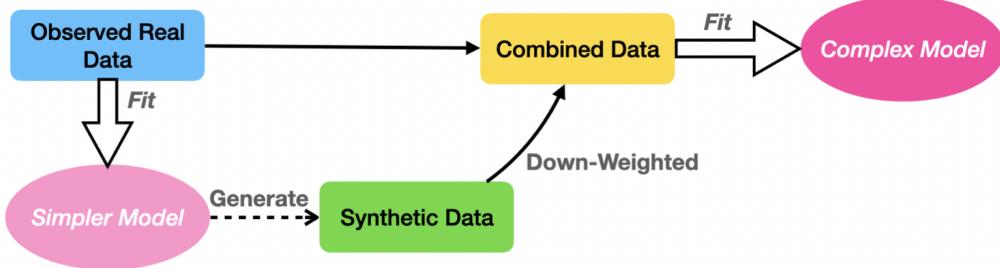


Figure 3: The catalytic prior modeling process from Huang et al. (2022).

We visualize the catalytic modeling process in Figure 3. In our context of expected points models for American football, the catalytic modeling process begins with fitting a catalytic prior model (e.g., a multinomial logistic regression model) to observed data. The observed data (\mathbf{X}, \mathbf{y}) consists of real football plays, where \mathbf{X} encodes game-states and \mathbf{y} encodes the response (either the outcome or points of the next score). Next, we generate fake football plays $(\mathbf{X}_*, \mathbf{y}_*)$. Specifically, we generate a matrix \mathbf{X}_* of synthetic game-states by sampling with replacement from each column of \mathbf{X} and jittering the continuous-valued variables using Gaussian noise. Then, for each synthetic game-state in \mathbf{X}_* we generate a synthetic response variable using our fitted catalytic prior model, yielding synthetic response vector \mathbf{y}_* . Next, we combine the observed football plays with the synthetic football plays, potentially downweighting the synthetic plays since they are fake. Finally, we fit a more complicated model (e.g., a XGBoost model) on this combined dataset $(\{\mathbf{X}, \mathbf{X}_*\}, \{\mathbf{y}, \mathbf{y}_*\})$.

More specifically, to fit a catalytic XGBoost classification model, we use a multinomial logistic regression catalytic prior model to generate fake outcomes of the next score. The number of synthetic datapoints and the weight of each synthetic datapoint are treated as hyperparameters in

the XGBoost tuning process. On the other hand, to fit a catalytic XGBoost regression model, we use a linear regression or multinomial logistic regression catalytic prior model to generate a vector of synthetic expected points numbers. We use fake *expected* points of the next score, rather than fake points of the next score in $\{7, 3, 2, 0, -2, -3, -7\}$, because it allows the complex target model to digest the information of the simpler prior model using fewer synthetic datapoints.

2.3 Model comparison

Now, we compare the out-of-sample predictive performance of various EP models. We use data from from nflFastR, an R package created to efficiently scrape NFL play-by-play data going back to 1999 ([Carl and Baldwin, 2022](#)), to train and test our models.

We begin with a dataset of 511,264 non-special teams football plays (and 216,149 first down plays) from 2006 to 2021. We use an alternative data truncation to Burke’s that removes garbage time plays but keeps plays in the second and fourth quarters, allowing us to model EP as a function of score differential and time remaining.¹³ This yields a dataset 346,400 plays (and 146,845 first down plays). Our dataset is clustered into *epochs*, groups of plays which share the same outcome of the next score. To keep the clustered nature of our dataset intact and to avoid data bleed, we split our dataset in half by randomly sampling 50% of all epochs. The first down plays from the first 50% of these epochs form the hold-out test set. We test on first down plays because fourth-down decision making relies on the value of having a first down (see Appendix C.1 for details). The plays from the other 50% of these epochs form the training set. To tune XGBoost models, we again split the training set in half, preserving the epoch correlation structure, to form a hyperparameter validation set and a training set. We then tune our XGBoost models in a similar fashion as in [Baldwin \(2021a\)](#).¹⁴

We visualize the results of our model comparison in Table 2. In Appendix A.1 we include detailed specifications of the well known statistical EP models and in Appendix A.2 we include detailed specifications of the remaining models.¹⁵ We denote the best version of XGBoost classification (resp., multinomial logistic regression) we could find by Baldwin+ (resp., Yurko+). The best catalytic machine learning model uses Baldwin+ as the complex target model and Yurko+ as the catalytic prior. We use $M = 3$ million synthetic outcomes generated from the catalytic prior. Only $M_U = 100$ of these synthetic outcomes come from under $U = 30$ half seconds remaining; we shrink less at the end of each half because XGBoost is better than MLR at the end of each half.

¹³Details of our data truncation and Burke’s data truncation are included in Appendix A.3.

¹⁴We don’t use the cross validation default in the XGBoost R package to tune XGBoost models because it doesn’t preserve the epoch correlation structure.

¹⁵Note that we fit Yurko et al. and Baldwin’s models without their row weighting procedure in order to judge the underlying structure of their models. See Appendix A.3 for more details.

We tune the catalytic hyperparameters (M, M_U, U) in a similar manner as the standard XGBoost hyperparameters.

model name	model type	team quality	out-of-sample MAE
Catalytic	Catalytic	point spread	3.744
Yurko+	Multinomial logistic regression	point spread	3.749
Baldwin+	XGBoost classification	point spread	3.753
	XGBoost regression	point spread	3.757
	Multinomial logistic regression	ours ¹⁶	3.768
Baldwin (2021b)	XGBoost classification	none	3.803
Yurko et al. (2018)	Multinomial logistic regression	none	3.808
Burke (2009a)	Linear regression	none	3.833
Romer (2006)	Instrumental variables regression	none	3.864

Table 2: Predictive performance of various EP models.

Romer and Burke’s methods perform worst because they are univariate functions of yardline. Romer’s method performs worse than Burke’s because it averages across all downs, whereas Burke’s is fit on just first downs. Yurko et al. and Baldwin’s models perform better because they adjust for other confounders. The models which adjust for team quality significantly outperform those that don’t. Models which adjust for team quality using point spread outperform those which use our hand-crafted measures because point spread is more up-to-date (e.g., it accounts for injury information). XGBoost classification outperforms XGBoost regression because leveraging the knowledge that there are seven potential outcomes of the next score yields a better EP model near the end of each half. That the best multinomial logistic regression model we could find outperforms the best XGBoost model we could find provides evidence that XGBoost overfits the data. Finally, the catalytic model, which shrinks the complex XGBoost model towards the simpler multinomial logistic regression model, performs best. This validates our hypothesis that expected points models fit using XGBoost need to be regularized or shrunk to create a better model subject to less overfitting.

3 Win probability models

Romer (2006) uses expected points as the basis of fourth-down decision making, but it is the wrong objective function. A team’s goal is to win the game, not score more points on average, so *win probability* (WP) is the right objective function.¹⁷ In Figure 4 we visualize an example

¹⁶See Appendix A.4 for details of our 8 measures of offensive and defensive quality.

¹⁷Expected points is only the first moment of win probability, which is a combination of the expected value and the variance of the points of the next score.

play in which the decision which maximizes expected points is clearly suboptimal. Even though attempting a field goal produces more points over average, scoring just 3 points when down by 6 at the end of the game all but guarantees a loss.

Down 6, 4th & 7, 10 yards from opponent endzone					
Qtr 4, 1:00 Timeouts: Off 1, Def 3 Point Spread: 0					
decision	EP	success prob	EP if fail	EP if succeed	SD of EP
Field goal	2.857	0.952	0.014	3.000	0.637
Go for it	2.382	0.392	0.014	6.052	2.948
Punt	-0.300				

Figure 4: An example of EP-based fourth-down decision making which shows that EP is the wrong objective function. Kicking a field goal yields more points on average, but the offense is down by 6 points with 1 minute to go, so it needs to go for a touchdown.

Hence in this Section we discuss statistical win probability models, which form the basis of the fourth-down decision procedure used today. These models, however, are fit from highly auto-correlated observational football data, so they produce highly uncertain win probability estimates. To understand just how difficult it is to accurately fit a win probability model from historical data, we conduct a simulation study using a simplified random walk version of football in which the true win probabilities are known. Although statistical win probability models find the general trend of true win probability (i.e., they are unbiased), they are subject to substantial uncertainty (to obtain good coverage, WP confidence intervals need to be substantially wide).

3.1 Problems with statistical win probability models

Statistical win probability models are fit from historical play-by-play data in a manner similar to expected points models. As before, the covariates \mathbf{x} encode the game-state. The game-state for WP models includes pre-game point spread and score differential, so WP isn't as biased as EP. The binary outcome variable y is 1 if the team with possession wins the game, else 0. In Table 3 we provide a summary of well known open-source win probability models. We include detailed specifications of these models in Appendix B.1. Further, in Appendix B.3, we conduct a model selection. A catalytic WP model, where XGBoost is the complex target model and a GAM is the catalytic prior, outperforms other models.

Football analysts see a dataset of 511,264 plays from 2006 to 2021 and think this is enough data to fit accurate statistical WP models. This is not true because the binary win/loss response variable is noisy and highly auto-correlated: *every game has only one winner*. Formally, the binary response

modeler	model	game-state variables	training set	outcome variable
Yurko et al. (2018)	GAM	transformations of score differential, time remaining, timeouts remaining, expected points	all plays	binary win/loss
Baldwin (2021a)	XGBoost	transformations of score differential, time remaining, timeouts remaining, yardline, down, yards to go, home, receive 2 nd half kickoff	all plays	binary win/loss

Table 3: Summary of well known open-source statistical win probability models.

variable y_i of the i^{th} play indicates whether the team with possession won the game.¹⁸ The response values are not independent, as all plays from the same game share the *same draw* of the response column. Thus the effective sample size is somewhere between the number of plays (511,264) and the number of non-tied games (4,101) from 2006 to 2021.¹⁹ This is not enough data to experience the full variability of the nonlinear and interacting variables of score differential, time remaining, point spread, yardline, yards to go, timeouts, etc. In fitting win probability models, we are in a limited-data context, and as such we expect wide confidence intervals for WP point estimates.

3.2 Simulation study: random walk football

To illustrate just how difficult it is to accurately fit a statistical win probability model from highly auto-correlated observational data, we conduct a simulation study. Specifically, we create a simplified random walk version of football in which the true win probability at each game-state is known. Then, we see how well statistical WP models recover the true win probability. These models find the general WP trend: they are unbiased, with a mean absolute error of less than 2% WP. But, they are subject to substantial uncertainty: to obtain 90% coverage, bootstrapped WP confidence intervals have a substantially wide average width of 8% WP. As real football is exponentially more complex, its confidence intervals should be far wider.

Rules of random walk football. Random walk football begins at midfield. Each play, the ball

¹⁸The response variable for fitting EP models from observational data is also auto-correlated, as plays are clustered into *epochs* (plays which share the same next score outcome). But our dataset contains 47,874 epochs and we find that auto-correlation impacts EP models significantly less than it affects WP models.

¹⁹The effective sample size is larger for plays closer to the end of the game in that uncertainty in win probability and the fourth-down decision diminishes towards the end of the game (see Section 4 for details).

moves left or right by one yardline with equal probability. If the ball reaches the left (right) end of the field, team one (two) scores a touchdown, worth +1 (−1) point. The ball resets to midfield after each touchdown. After N plays, the game ends. If the game is still tied after N plays, a fair coin is flipped to determine the winner. We include the formal mathematical specification of the game in Appendix B.4. We also explicitly compute true win probability as a function of time, field position, and score differential using dynamic programming in Appendix B.4.

Simulation methodology. $M = 25$ times, we simulate G games, each with N plays per game. We use $L = 4$ yardlines so that the average number of first down plays between each score is similar to that of a real football game. This yields M simulated datasets of simplified football plays, each of the form

$$\mathcal{D} = \{(n, X_{gn}, S_{gn}, y_{gn}) : n = 1, \dots, N \text{ and } g = 1, \dots, G\}. \quad (3.1)$$

For each play of game g we record the timestep n , the field position X_{gn} , the score differential S_{gn} , and a binary variable y_{gn} indicating whether the team with possession wins the game. The response variable y is auto-correlated, as each play within the same game shares the same random draw of y .

On each simulated dataset, we use machine learning to estimate win probability as a function of timestep n , field position x , and score differential s ,

$$\widehat{\text{WP}}(n, x, s) = \text{XGBoost}(\mathcal{D})(y|n, x, s). \quad (3.2)$$

We then compute the mean absolute error between the true and estimated win probabilities averaged over the M simulations. We also compare the coverage and lengths of the WP confidence intervals produced by various bootstraps, discussed below, averaged over the M simulations.

Bootstrap confidence interval methodology. We compare the coverage and lengths of WP confidence intervals produced by the standard bootstrap, cluster bootstrap, and randomized cluster bootstrap, averaged over the M simulations. In the standard bootstrap, which assumes each row (play) of the dataset is independently drawn, each of B bootstrapped datasets are formed by resampling N plays with replacement. In the cluster bootstrap, each of B bootstrapped datasets are formed by resampling G' games with replacement, keeping each observed row within each resampled game. Finally, in the randomized cluster bootstrap, each of B bootstrapped datasets are formed by resampling G' games with replacement, and within each game resampling plays with replacement. To achieve better coverage, we re-sample half as many games as in the original dataset, $G' = G/2$. Then, for each bootstrap method, we fit a WP model WP_b to each bootstrapped dataset b . The confidence interval for the WP estimate at game-state \mathbf{x} is defined by the 2.5th and 97.5th quantiles of $\{\text{WP}_1(\mathbf{x}), \dots, \text{WP}_B(\mathbf{x})\}$. As we want our WP confidence intervals to be quickly

evaluable for fourth-down decision making, we use $B = 26$.²⁰

Simulation results. We report the results of our simulation study in Table 4. For the first row, each simulated dataset consists of $G = 4,101$ games and $N = 53$ plays per game, which matches the number of games and the average number of first down plays in our dataset of real football plays. Each game in each of these datasets consists of $K = 53$ auto-correlated plays per game. For the second row, each simulated dataset consists of $G = 4,101 \cdot 53$ games and $N = 53$ plays per game. Then, we remove all but $K = 1$ play per game so that each timestep n has exactly 4,101 corresponding i.i.d. rows. In other words, those datasets consist of 217,353 plays with an i.i.d. response column.

G	N	K	MAE bt WP and \widehat{WP}	CI covg.			CI length		
				SB	CB	RCB	SB	CB	RCB
4,101	53	53	0.0179	0.73	0.85	0.90	0.048	0.067	0.079
$4,101 \cdot 53$	53	1	0.0164	0.78	0.78	0.78	0.049	0.049	0.049

Table 4: Simulation study results. SB means standard bootstrap, CB means cluster bootstrap, and RCB means randomized cluster bootstrap.

In the simulation study with auto-correlation ($K = 53$), the mean absolute error (MAE) between the true and estimated WP is less than 2% over average. Also, the MAE is smaller than about 3.5% WP across all values of true win probability (see Figure 15a of Appendix B.4). So, XGBoost recovers the general trend of true WP, which we visualize in Figure 5a. In the simulation study without auto-correlation ($K = 1$), the MAE is similar but slightly smaller. This suggests that much of the bias induced by fitting WP from observational data is the result of having limited data and a noisy binary response column, not from auto-correlation.

The length and coverage of WP confidence intervals, on the other hand, are significantly impacted by auto-correlation. In the simulation study with auto-correlation ($K = 53$), the standard bootstrap, which ignores auto-correlation, produces confidence intervals which are too narrow at an average width of about 5% WP, leading to a subpar 73% coverage. The cluster bootstrap produces wider confidence intervals at an average width of about 7% WP, leading to a higher 85% coverage. The randomized cluster bootstrap produces even wider confidence intervals at an average width of about 8% WP, leading to a satisfactory frequentist coverage of 90% over average. In other words, to achieve satisfactory coverage, WP confidence intervals need to be substantially wide, which we visualize in Figure 5b.

Coverage is similar across all values of true win probability except near 0 and 1 (see Figure 15b

²⁰Specifically, we use the original dataset together with 25 bootstrapped datasets, forming 26 total datasets.

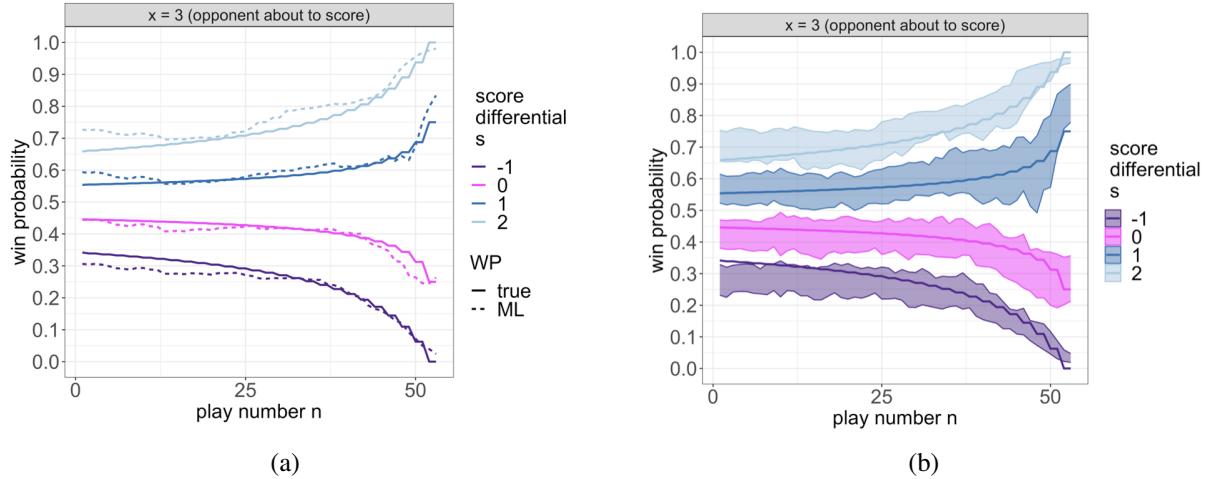


Figure 5: On the left: WP estimates (dotted line) get the general trend right. On the right: bootstrapped WP confidence intervals need to be substantially wide to achieve good coverage. In both figures: true WP is given by the solid line, and we display the results from one simulation and at yardline $x = 3$.

of Appendix B.4). To increase coverage at the extremes, we widen our confidence intervals when $\widehat{WP} < 0.025$ to have a lower bound of 0 and when $\widehat{WP} > 0.975$ to have an upper bound of 1. Also, average confidence interval length from the randomized cluster bootstrap is at most 12% for some values of true WP, and C.I. length decreases as true WP moves towards the extremes (see Figure 15c of Appendix B.4).

In the simulation study without auto-correlation ($K = 1$), each bootstrap method is identical and yields an average confidence interval length of about 5% WP (similar to the average C.I. length from the standard bootstrap on auto-correlated data). The frequentist coverage is 78%; to increase coverage we could widen the confidence intervals by resampling fewer than N plays per game in the standard bootstrap.

4 Fourth-down decision making

Current popular fourth-down decision procedures involve making the decision $\alpha \in \{\text{Go, FG, Punt}\}$ which maximizes estimated win probability.²¹ These existing decision procedures, based solely on effect size, ignore the uncertainty inherent in estimating win probability from highly auto-correlated historical data. Thus, in this Section, we modify the decision procedure to account for uncertainty. We find that decision making changes substantially. In particular, far fewer fourth-

²¹In Appendix C.1 we detail how we estimate win probability if a team goes for it, attempts a field goal, or punts as a function of win probability if a team has possession and a first down at a certain game-state.

down decision are as obvious as analysts claim.

4.1 Uncertainty in the estimated optimal decision

We use bootstrapping to incorporate uncertainty quantification into the fourth-down decision procedure. We use the same bootstrapping structure from the simulation study from Section 3.2 since it achieved adequate coverage. Specifically, we use a randomized cluster bootstrap to create B bootstrapped datasets. Each bootstrapped dataset arises from resampling games with replacement and then within each game resampling plays with replacement. We use just $B = 26$ bootstrapped datasets, a fairly small number for bootstrapping, because we want to be able to quickly evaluate decision uncertainty during a football game.²²

Then, to each bootstrapped dataset, we fit a win probability model using catalytic XGBoost.²³ Each bootstrapped model produces an estimated optimal decision, the decision which maximizes estimated win probability. To quantify uncertainty of this estimate, we use *bootstrap percentage*, the percentage of bootstrapped models which report decision α as optimal. If most of the bootstrapped models say α is optimal, we are more confident in the point estimate of the optimal decision. If the bootstrapped models are split across multiple decisions, we cannot rely on our point estimate.

Bootstrap percentage is a measure of *data reliability*. At each game-state the model produces a point estimate of the fourth-down decision; bootstrap percentage tells us how reliable this estimate is, or how much the data trusts its own estimate. To understand, think of the outcome (winning team) of each row (play) in the dataset as a random draw. If some of these draws resulted in different outcomes, our fitted win probability functions would be different. The less data we have access to, the more sensitive models are to the random idiosyncrasies of any particular training dataset. The bootstrap quantifies this sensitivity: given the amount of data we have, it quantifies the spectrum of variability in potential resulting fitted models. Specifically, it measures: given the amount of data we have, in what proportion of draws of the training set would α be the optimal decision according to win probability point estimates?

Bootstrap percentage is the right way to bring uncertainty into the decision procedure because it quantifies uncertainty of the *decision itself* (Friedman et al., 1999). This is distinct from uncertainty in the win probability estimates of each individual decision (Go, FG, or Punt) because these individual estimates are correlated across different draws of the training dataset. For example, across different draws of the training set, for some game-states Go is always better than FG

²²Note that in each of the $B = 26$ bootstrapped datasets we sample with replacement half as many games as in the original observed dataset. We validated this approach in the simulation study from Section 3.2.

²³In Appendix B.3 we conduct a WP model selection. Our catalytic WP model, where XGBoost is the complex target model and a GAM is the catalytic prior, outperforms other models.

even though individual WP estimates are highly variable. Another measure of decision confidence which accounts for this correlation is the bootstrapped confidence interval²⁴ of the win probability *gain* of a decision (e.g., the *difference* in WP between Go and FG).

4.2 How fourth-down decision making changes

When we account for uncertainty in win probability estimates, fourth-down decision making changes substantially. We illustrate this using example plays.

Example play 1. First we compare Baldwin’s fourth-down decision making procedure to ours. Baldwin suggests making the decision which maximizes estimated win probability. Further, he measures the strength of a decision by the estimated gain in win probability by making that decision. Figures 6a and 6b illustrate Baldwin’s decision making²⁵ for a play from the 2023 AFC Championship game. Baldwin views Go as a “strong” decision because he estimates that going for it provides a 3.8% gain in win probability over attempting a field goal. In Figure 6c we add uncertainty quantification to Baldwin’s decision making. Although our point estimate (the blue column) suggests that Go provides a 1.5% gain in win probability over FG, our confidence interval $[-3.7\%, 4.5\%]$ suggests that Go could either be a terrible or a great decision. There is not enough data to estimate win probability with enough granularity to know which decision is best. Further, about half of the bootstrapped models say Go is better than FG (the orange column), reflecting considerable uncertainty in the optimal fourth-down decision.

Example play 2. Next we compare Burke’s fourth-down decision making procedure to ours. Burke, whose win probability model is proprietary, also suggests making the decision which maximizes estimated win probability. Figure 7a illustrates Burke’s fourth-down decision boundary chart²⁶ for a play from the 2023 NFC Championship game. The chart visualizes the estimated optimal decision (color) as a function of yardline (x -axis) and yards to go (y -axis), holding the other game-state variables constant. Burke views Go as the right decision because the yellow dot (representing the actual play’s yardline and yards to go) lies squarely in the red region and is far from the decision boundary. The chart, however, says nothing about the estimated strength of making the optimal decision or about uncertainty quantification. Hence in Figure 7b we show our version of Burke’s chart²⁷ which uses color intensity to visualize the estimated gain in win probability by making a decision (darker colors indicate larger values). The pink dot (representing the actual

²⁴This confidence interval is the 2.5th and 97.5th quantiles of the estimated gain in win probability across all the bootstrapped samples.

²⁵These figures were taken from Baldwin’s fourth down Twitter bot @ben.bot.baldwin.

²⁶This figure was taken from Burke’s Twitter @bburkeESPN.

²⁷We use green for Go, yellow for FG, and red for Punt because we liked Burke’s Twitter thought that fourth-down decision making is like a traffic light.

Up 3, 4th & 1, 14 yards from opponent end zone Qtr 2, 03:58 Timeouts: Off 0, Def 3 <hr/> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th style="text-align: center;">Win %</th><th style="text-align: center;">Success %¹</th><th style="text-align: center;">Fail</th><th style="text-align: center;">Succeed</th></tr> </thead> <tbody> <tr> <td>Go for it</td><td style="text-align: center;">72</td><td style="text-align: center;">68</td><td style="text-align: center;">64</td><td style="text-align: center;">76</td></tr> <tr> <td>Field goal attempt</td><td style="text-align: center;">68</td><td style="text-align: center;">94</td><td style="text-align: center;">62</td><td style="text-align: center;">69</td></tr> </tbody> </table> <p>¹ Likelihood of converting on 4th down or of making field goal Source: @ben_bot_baldwin</p>		Win %	Success % ¹	Fail	Succeed	Go for it	72	68	64	76	Field goal attempt	68	94	62	69	 4th down decision bot @ben_bot_baldwin · Jan 29 Automated ---> CIN (3) @ KC (6) <--- KC has 4th & 1 at the CIN 14 Recommendation (STRONG): ⚡ Go for it (+3.8 WP) Actual play: ⚡ (Shotgun) P.Mahomes pass short right to T.Kelce for 14 yards, TOUCHDOWN. H.Butker extra point is GOO
	Win %	Success % ¹	Fail	Succeed												
Go for it	72	68	64	76												
Field goal attempt	68	94	62	69												

(a)

(b)

Up 3, 4th & 1, 14 yards from opponent endzone

Qtr 2, 4:00 | Timeouts: Off 0, Def 3 | Point Spread: -2

decision	WP	WP gain CI	boot %	success prob	WP if fail	WP if succeed	SD of WP
Go for it	73.7%	[-3.7%, 4.5%]	53.8%	66.7%	66.7%	77.2%	4.9%
Field goal	72.2%		46.2%	93.2%	66.7%	72.6%	1.5%
Punt	65.5%		0.0%				

(c)

Figure 6: For example play 1, Figures (a) and (b) illustrate Baldwin’s decision making and Figure (c) shows our decision making.

play’s yardline and yards to go) lies in a light green region, indicating a smaller estimated gain in win probability by going for it. Being far from the decision boundary, however, does not imply it the best decision with certainty. Hence in Figure 7c we provide an additional chart which illustrates uncertainty in the estimated optimal decision. Here, the color intensity indicates the proportion of bootstrapped models which make the estimated optimal decision. Aside from some darker patches in the lower left and upper right, most of the figure features lighter colors, indicating high uncertainty. For these game-states, including the actual play, we don’t have enough data to know which decision is best.

4.3 Our improved fourth-down decision procedure

By accounting for uncertainty in win probability estimates, our fourth-down decision making procedure recognizes when we don’t know the best decision. We illustrate our improved decision procedure through more example plays.

Example play 3. In Figure 8 we visualize our decision procedure²⁸ for a fourth-down play in

²⁸To compare our decision making procedure to the decisions that actual football coaches tend to make, we model the probability that a coach chooses a decision in {Go, FG, Punt} as a function of game-state. We discuss this *baseline coach model* in detail in Appendix C.5 and include the model’s predictions in our decision figures.

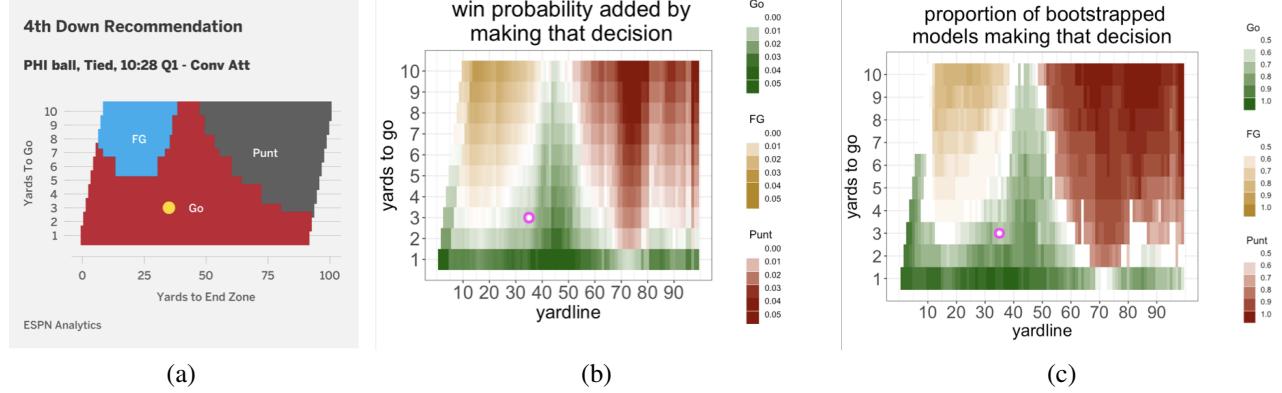


Figure 7: For example play 2, Figure (a) illustrates Burke’s decision making and Figures (b) and (c) show our decision making.

which the Bears had the ball against the Jets in Week 12 of 2022. FG provides a solid edge over Go according to the WP point estimate (+2.1% WP). But our confidence interval of the estimated gain in win probability by attempting a field goal is $[-3.7\%, 4.4\%]$, indicating that FG could either be a great or a terrible decision. Also, about half of our bootstrapped models say Go is better. In other words, we do not have enough data to be confident in our win probability point estimates, and we don’t know the optimal fourth-down decision at this game-state. So, we recommend leaving the fourth-down decision to the coach’s discretion. Further, in the bottom right plot, notice how most of the colors are light. This indicates that the optimal decision is uncertain at most other combinations of yardline and yards to go at this game-state.

Example play 4. In Figure 9 we visualize our decision procedure for a fourth-down play in which the Commanders had the ball against the Colts in Week 8 of 2022. Punt provides a slight edge over Go according to the WP point estimate (+0.5% WP). But, nearly all of the bootstrapped models say Punt is better and our confidence interval of the estimated gain in win probability by punting is $[0\%, 4.8\%]$, which is positive. So, even if the edge is small, we are confident in this edge and recommend that the Commanders should Punt. Further, in the bottom right plot, notice how most of the colors are dark outside of a large white boundary region. This indicates that we have higher certainty in our estimated optimal decision at this game-state.

Example play 5. In Figure 10 we visualize our decision procedure for an infamous fourth-down play in which the Raiders had the ball against the Rams in Week 14 of 2022. Go provides a strong edge over Punt according to the WP point estimate (+3.5% WP). Further, 100% of the bootstrapped models say Go is better and our confidence interval of the estimated gain in win probability by going for it is $[0.30\%, 5.23\%]$, which is strictly positive. Thus, we are confident in

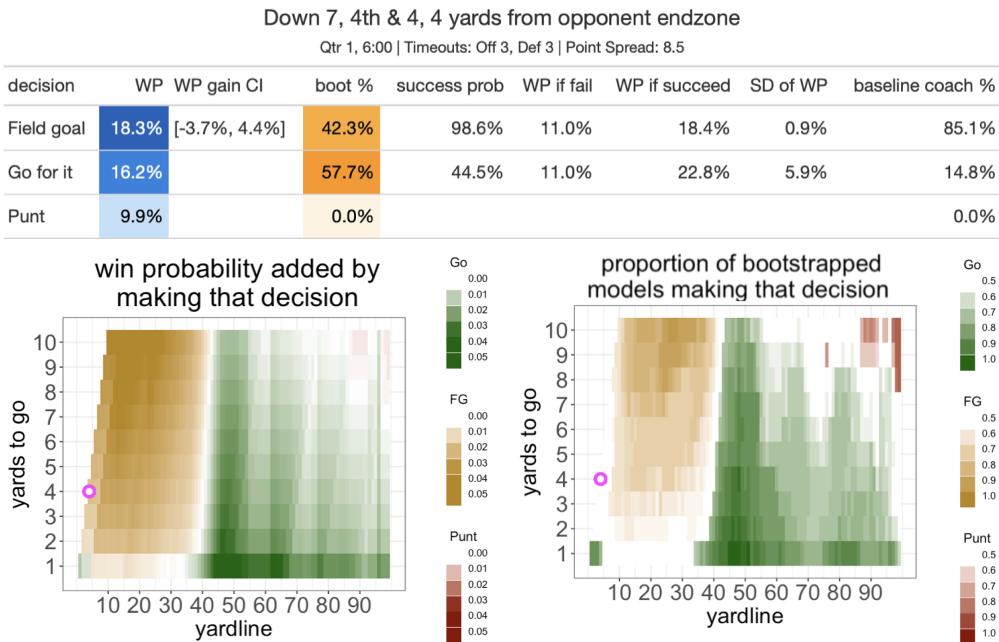


Figure 8: Our decision making for example play 3.

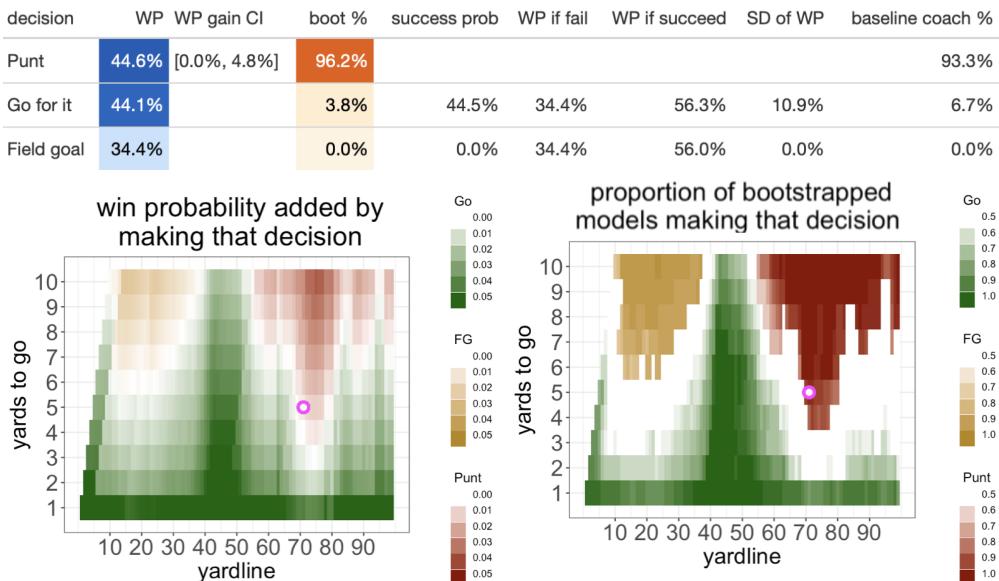


Figure 9: Our decision making for example play 4.

this edge, and we strongly recommend that the Raiders should Go.²⁹

²⁹In real life, the Raiders punted. Then, Rams quarterback Baker Mayfield countered with a successful 98 yard drive to win the game.

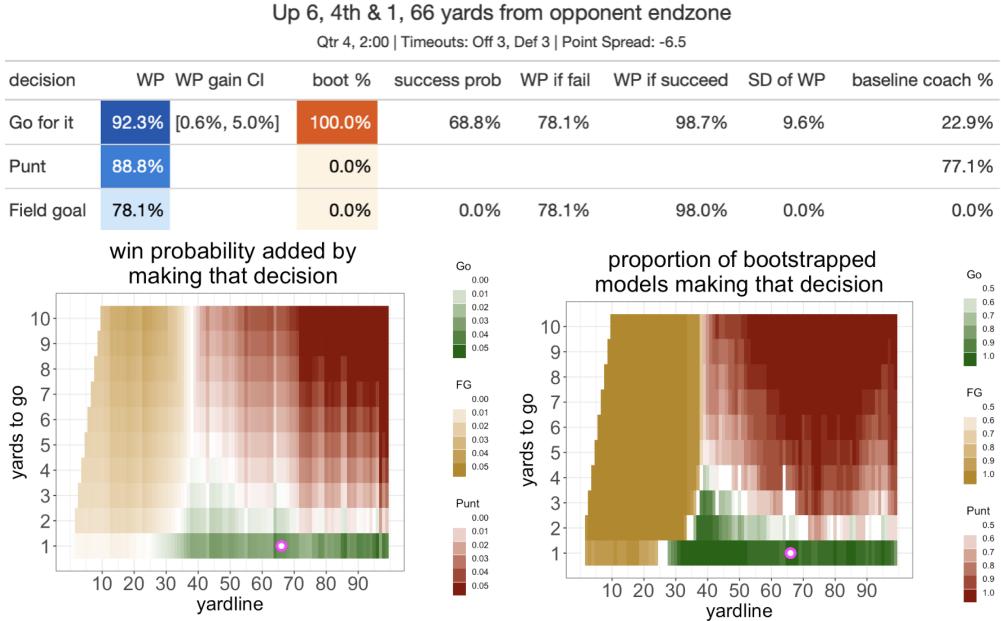


Figure 10: Our decision making for example play 5.

4.4 Analytics, have some humility

The practical football lesson arising from our new decision procedure is that far fewer fourth-down decisions are as obvious as analysts claim. In Figure 11 we quantify the extent to which fourth-down decisions are less obvious than before. We define a decision as “obvious” according to our decision procedure if the percentage of bootstrapped models making that decision is above, say, 85%. We define a decision as “obvious” according to the previous decision procedure if the estimated gain in WP by making that decision is above, say, 0.015. We call non-obvious decisions “nebulous.” Of the 1,968 fourth-down decisions from 2018-2022 that were previously considered obvious, we consider 8,139 (24%) to be nebulous. There is an asymmetry: 32% of previously obvious Go decisions are nebulous and 19% of previously obvious kicks are nebulous. This reflects substantial overconfidence in WP point estimates.

Thus, we ask football analysts to have some *humility*: for many game-states there is simply *not enough data* to use statistical win probability point estimates to make fourth-down decisions. In these cases of high uncertainty, regardless of the point estimate it is arrogant for analysts to recommend decisions to play calling experts and coaches. These experts spend a significant amount of time with their players and have access to information which doesn’t show up in the data. For instance, Eagles coach Nick Sirianni may notice that his usually dominant offensive line is missing a key player today, which does not currently appear in the data. When the estimated optimal fourth-down decision has too much uncertainty, we suggest leaving the decision to the coach’s discretion. Similarly, a football analyst should evaluate a coach’s fourth-down decision making only on plays

		Decision from bootstrap percentage			
Decision from estimated WP gain		obvious go	nebulous	obvious kick	total
obvious go		0.68	0.32	0.00	3,242
nebulous		0.08	0.76	0.15	11,036
obvious kick		0.00	0.19	0.81	4,897
total		3,145	10,400	5,630	19,175

threshold for "obvious" defined by point estimates: WP gain > 0.015
 threshold for "obvious" defined by uncertainty quantification: boot % > 0.85

Figure 11: Quantifying the extent to which fourth-down decisions are less obvious than before using all fourth-down plays from 2018-2022.

where uncertainty is low.

Although football analysts have been overconfident in their point estimates, the football analytics community was largely correct that NFL coaches do not go for it enough on fourth down. Across all fourth-down plays from 2018-2022 that we consider obvious, the coach made the right decision for 91% plays where he should have kicked but just 55% of plays where he should have gone for it. Play calling in the NFL is still far too conservative: coaches consistently make wrong decisions, particularly when they should go for it.

5 Discussion

In-game strategic decision making in sports, and in particular the fourth-down decision problem in American football, is a classic example of a rich applied statistics problem. Statistical expected points and win probability models fit from historical data form the backbone of the decision procedure: make the decision which maximizes the value of the next game-state. In developing these models, however, we encounter a series of complex statistical notions. By not adjusting for team quality, well known open-source statistical EP models suffer from selection bias.³⁰ Adding additional covariates to adjust for this exacerbates overfitting. Open-source statistical WP models, moreover, don't account for the auto-correlated nature of play-by-play football data. These issues are not just unique to our specific data problem, but appear in applied statistics problems across

³⁰Today, some proprietary expected points models do adjust for team quality (e.g., Burke's EP model at ESPN accounts for team strength using FPI (football power index)).

many domains.³¹

In this paper we discussed these issues in detail and devised ways to mitigate them. To adjust for selection bias, we created and included measures of team quality as covariates. To reduce overfitting and smooth XGBoost, we extended the catalytic prior to our machine learning framework. To quantify uncertainty in win probability models, and thus in the fourth-down decision itself, we used a version of the bootstrap which accounts for auto-correlation and is tuned on a simplified random walk version of football.

Our main contribution to the statistics community is our extension of the catalytic prior, initially developed in the context of linear models (Huang et al., 2020, 2022), to a machine learning context. We found that the catalytic prior, which used synthetic data imputed from a simpler smoother model as a prior for a more complex model with interactions, effectively smoothed our tree machine learning models. Our other contribution to the statistics community is our framing of this paper as an exemplary case study of how to conduct a real-world data analysis. We take the reader through formulating a problem, dissecting a classic, massive, rich dataset, identifying and facing a series of complex statistical obstacles (including selection bias, overfitting, and auto-correlation), incorporating uncertainty quantification, and synthesizing our analysis into a final decision making inference.

Our contribution to the football analytics community is a major advance in fourth-down strategic decision making. We devised a better expected points model, as well as quantified uncertainty in win probability and in the fourth-down decision itself. The practical football lesson arising from our new decision procedure is that far fewer fourth-down decisions are as obvious as analysts claim. In particular, for a huge proportion of game-states, there is too much uncertainty in the estimated optimal decision. Thus, we ask football analysts to have some humility: there is simply not enough data to use statistical win probability point estimates to make fourth-down decisions in many cases. For game-states in which uncertainty is high, we suggest leaving the decision to the coach, an on-field expert who has access to information that doesn't show up in the data. Nonetheless, NFL coaches still skew too conservatively: they still don't go for it enough on fourth down even when it is mathematically obvious.

Although our analysis improves the state of the art, it is not without limitations. Even though the randomized cluster bootstrap produces substantially wide confidence intervals for win probability estimates, it underestimates uncertainty because it quantifies sampling uncertainty³² but not model

³¹For instance, auto-correlation appears in climate statistics (e.g., McShane and Wyner (2011)) and finance (e.g., Yang et al. (2013)), selection bias arises in epidemiology (e.g., Tripepi et al. (2010); Hernán et al. (2004)), and overfitting is prevalent throughout the literature (e.g., Peng and Nagata (2020); Subramanian and Simon (2013)).

³²Uncertainty in our point estimates resulting from fitting a model on a finite dataset, also known as “variance.”

uncertainty.³³ In our simulation study from Section 3.2, there is no model uncertainty because we know true win probability is indeed a function of time, score differential, and field position. Win probability in real football, however, is highly likely a function of unobserved confounders. For example, how well Tom Brady slept the previous night could affect his team’s win probability. Additionally, our analysis doesn’t capture uncertainty in the conversion probability model, field goal probability model, and expected next yardline after punting model. These models are fit from thousands of play-level i.i.d. observations, and so are subject to little sampling uncertainty, but are subject to nontrivial model uncertainty as they make simplifying assumptions. Conversion probability in particular is a delicate concept, as it depends on the offensive play call, the defensive play call, and the individual characteristics of each of the players on the field. A more fine-grained analysis would account for this additional uncertainty.

Also, because statistical win probability models produce estimates that are too uncertain at many game-states, we suggest in future work exploring probabilistic state-space models to estimate win probability. Probabilistic models simplify the game of football into a series of transitions between game-states. Transition probabilities are estimated from play-level data and win probability is calculated by simulating games. The effective sample size of transition probability models is the number of plays because they are fit from independent play-level observations. Some analysts in private industry have created proprietary probabilistic win probability models, which they believe are more accurate than statistical models because they have a larger effective sample size. Through the lens of the bias-variance tradeoff, proprietors of these models believe that introducing bias in order to reduce variance improves the overall accuracy of the resulting win probability estimator. Nevertheless, these models are subject to their own set of issues, and we believe they aren’t as low-variance as some analysts claim. Specifically, properly modeling the distribution of the outcome of a play is nontrivial. In contrast to the simple binary win/loss outcome of statistical win probability models, the outcome variable of a transition probability model is the next game-state, which could include a change in yardline, score, or time. This distribution is quite complex: there is a spike at gaining 0 yards for incompletions, a spike for a touchdown, spikes for penalties, and other smooth possibly multimodal distributions for the outcome of run or pass plays, each of which change as a function of team quality and other confounders. Uncertainty in these transition probabilities may, after being properly propagated through a state-space model, result in just as much (if not more) uncertainty in estimated win probability than estimates from statistical models. Additionally, one must take great care to carefully encode all the subtle rules of football into her model, and one needs sufficient computing power to simulate enough games to estimate win probability with enough granularity. We look forward to a public facing exploration of probabilistic win probability models

³³Uncertainty caused by our model being wrong or biased, also known as “bias.”

in the future.

Acknowledgements

The authors thank the many football analysts who contributed to the development of expected points models, win probability models, and the fourth down problem. In particular, we thank Brian Burke for providing helpful feedback. The authors acknowledge the High Performance Computing Center (HPCC) at The Wharton School, University of Pennsylvania for providing computational resources that have contributed to the research results reported within this paper.

References

- Baldwin, B. (2021a). NFL win probability from scratch using xgboost in R.
Baldwin, B. (2021b). nflfastR EP, WP, CP xYAC, and xPass models.
<https://www.opensourcefootball.com/posts/2020-09-28-nflfastr-ep-wp-and-cp-models>.
- Burke, B. (2009a). *The 4th Down Study - Part 1*.
<http://www.advancedfootballanalytics.com/2009/09/4th-down-study-part-1.html>.
- Burke, B. (2009b). *The 4th Down Study - Part 3*.
<http://www.advancedfootballanalytics.com/2009/09/4th-down-study-part-3.html>.
- Burke, B. (2014). Expected Points (EP) and Expected Points Added (EPA) Explained.
<http://www.advancedfootballanalytics.com/2010/01/expected-points-ep-and-expected-points.html>.
- Candes, E., Fan, Y., Janson, L., and Lv, J. (2016). Panning for gold: Model-free knockoffs for high-dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80.
- Carl, S. and Baldwin, B. (2022). *nflfastR: Functions to Efficiently Access NFL Play by Play Data*.
<https://www.nflfastr.com/>.
- Carroll, B., Palmer, P., and Thorn, J. (1989). *The Hidden Game of Football*. A Football ink book.
Grand Central Pub.

- Carter, V. and Machol, R. E. (1971). Technical Note—Operations Research on Football. *Operations Research*, 19(2):541–544.
- Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’16*, pages 785–794, New York, NY, USA. ACM.
- Eager, E. (2020). PFF WAR: Modeling Player Value in American Football.
- Friedman, N., Goldszmidt, M., and Wyner, A. (1999). Data analysis with bayesian networks: A bootstrap approach. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, UAI’99*, page 196–205, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Hastie, T. and Tibshirani, R. (1986). Generalized Additive Models. *Statistical Science*, 1(3):297 – 310.
- Hernán, M. A., Hernández-Díaz, S., and Robins, J. M. (2004). A structural approach to selection bias. *Epidemiology*, 15(5):615–625.
- Huang, D., Stein, N., Rubin, D. B., and Kou, S. C. (2020). Catalytic prior distributions with application to generalized linear models.
- Huang, D., Wang, F., Rubin, D. B., and Kou, S. C. (2022). Catalytic priors: Using synthetic data to specify prior distributions in bayesian analysis.
- Lock, D. and Nettleton, D. (2014). Using random forests to estimate win probability before each play of an nfl game. *Journal of Quantitative Analysis in Sports*, 10.
- Lopez, M. (2017). *All win probability models are wrong — Some are useful.*
<https://statsbylopez.com/2017/03/08/all-win-probability-models-are-wrong-some-are-useful>.
- McShane, B. B. and Wyner, A. J. (2011). A statistical analysis of multiple temperature proxies: Are reconstructions of surface temperatures over the last 1000 years reliable? *The Annals of Applied Statistics*, 5(1):5 – 44.
- Medvedovsky, K. and Patton, A. (2022). *Daily Adjusted and Regressed Kalman Optimized projections — DARKO*. <https://apanalytics.shinyapps.io/DARKO/>.
- Peng, Y. and Nagata, M. H. (2020). An empirical overview of nonlinearity and overfitting in machine learning using covid-19 data. *Chaos, Solitons & Fractals*, 139:110055.

- Pro Football Reference (2023). *The P-F-R Win Probability Model*.
https://www.pro-football-reference.com/about/win_prob.htm.
- Ren, Z., Wei, Y., and Candès, E. (2020). Derandomizing knockoffs.
- Romer, D. (2006). Do Firms Maximize? Evidence from Professional Football. *Journal of Political Economy*, pages 340–365.
- Schneider, T. (2023). *Gambletron 2000*.
<https://www.gambletron2000.com/nfl/30295/new-england-patriots-at-atlanta-falcons>.
- Subramanian, J. and Simon, R. (2013). Overfitting in prediction models – is it a problem only in high dimensions? *Contemporary Clinical Trials*, 36(2):636–641.
- Tripepi, G., Jager, K. J., Dekker, F. W., and Zoccali, C. (2010). Selection Bias and Information Bias in Clinical Research. *Nephron Clinical Practice*, 115(2):c94–c99.
- Yang, H., Wan, H., and Zha, Y. (2013). Autocorrelation type, timescale and statistical property in financial time series. *Physica A: Statistical Mechanics and its Applications*, 392(7):1681–1693.
- Yurko, R., Ventura, S., and Horowitz, M. (2018). nflWAR: A Reproducible Method for Offensive Player Evaluation in Football.

A Expected points details

A.1 Statistical expected points models

Early models. Expected points models for American football have a long history, beginning with former Cincinnati Bengals quarterback Virgil Carter who created expected points ([Carter and Ma-chol, 1971](#)). Later, [Carroll et al. \(1989\)](#) developed the first linear expected points model, a linear version of Burke’s Method (below). [Romer \(2006\)](#) developed an expected points model using instrumental variable regression and used expected points in the context of fourth-down decision making.

Burke’s method. [Burke \(2009a\)](#) uses linear regression to model expected points as a cubic spline in yardline. In particular, Burke models

$$P = \mathbf{x}^\top \beta + \varepsilon, \quad (\text{A.1})$$

where P is the points of the next score (a real number in $\{7, 3, 2, 0, -2, -3, -7\}$, which is positive if the team with possession scores next and negative if the opposing team scores next), \mathbf{x} is a yardline spline basis, and ε is mean zero noise. Then, the estimated expected points of the next score is $\mathbf{x}^\top \hat{\beta}$. Burke estimates the spline coefficients $\hat{\beta}$ from a historical dataset of first down plays. Burke estimates EP from just first down plays because fourth-down decision making relies on the value of a first down (see Appendix C.1 for details).

Yurko’s method. [Yurko et al. \(2018\)](#) model the probability of each potential outcome of the next score and compute expected points as a function of these outcome probabilities. Specifically, they use a multinomial logistic regression (MLR) to model the log-odds of the next scoring event as a function of the game-state,

$$\log \left(\frac{\mathbb{P}(\mathbf{y} = k | \mathbf{x})}{\mathbb{P}(\mathbf{y} = 0 | \mathbf{x})} \right) = \mathbf{x}^\top \beta_k, \quad (\text{A.2})$$

where \mathbf{y} denotes the outcome of the next scoring event in the half after the current play,

$$\mathbf{y} \in \left\{ \begin{array}{l} \text{Touchdown (7), Opp. Team Touchdown (-7),} \\ \text{Field Goal (3), Opp. Team Field Goal (-3),} \\ \text{Safety (2), Opp. Team Safety (-2),} \\ \text{No Score (0)} \end{array} \right\}. \quad (\text{A.3})$$

\mathbf{x} encodes the game-state: yardline, down (categorical), half seconds remaining, log yards to go to achieve a first down, goal-to-go, and an under-two-minutes indicator. Then, the expected points at

game-state \mathbf{x} is

$$\widehat{\text{EP}}(\mathbf{x}) = \sum_k k \cdot \widehat{\mathbb{P}}(y = k | \mathbf{x}), \quad (\text{A.4})$$

where

$$\widehat{\mathbb{P}}(y = k | \mathbf{x}) = \frac{\mathbb{I}(k = 0) + \mathbb{I}(k \neq 0) \cdot \exp(\mathbf{x}^\top \widehat{\beta}_k)}{1 + \sum_{j \neq 0} \exp(\mathbf{x}^\top \widehat{\beta}_j)}. \quad (\text{A.5})$$

Yurko et al. estimate the coefficients $\widehat{\beta}_k$ from historical data.

Baldwin’s method. Baldwin (2021b) uses XGBoost (Chen and Guestrin, 2016) to estimate the probability that the next score results in each of the seven outcomes from Formula (A.3) as a function of game-state. Baldwin uses a larger feature set than previous expected points models, codifying the game-state using yardline, down, yards to go, whether the team with possession is at home, half seconds remaining, roof type (retractable, dome, or outdoors), timeouts remaining for each team, and era (1999-2001, 2002-2005, 2006-2013, 2014-2017, and 2018-present).

A.2 EP model specification details

Yurko+. We denote the best multinomial logistic regression EP model we could find by Yurko+. As before, the response variable is the outcome of the next score (\mathbf{y} from Formula (A.3)), and the model structure is the same from Formula (A.2) except with different game-state covariates \mathbf{x} . This model is trained only on first down plays. We include the R code for Yurko+ below.

```
multinom(label ~
  bs(yardline, knots=c(7,33,67,93)) +
  bs(half_seconds_remaining, degree=1, knots=c(30)) +
  utm:as.numeric(posteam_timeouts_remaining==0) +
  factor(era) +
  posteam_spread + posteam_spread:yardline +
  I((score_differential <= -11)) + ### way behind
  I((score_differential >= 11)) + ### way ahead
  I((score_differential <= -4)*(game_seconds_remaining <= 900)) + ### need TD
  I((-3 <= score_differential & score_differential <= 0)*(game_seconds_remaining <= 900)) +
  I((1 <= score_differential & score_differential <= 3)*(game_seconds_remaining <= 900)) +
  I((4 <= score_differential & score_differential <= 10)*(game_seconds_remaining <= 900))
).
```

The best multinomial logistic regression which uses our eight team quality metrics. The best multinomial logistic regression which uses our eight hand-crafted team quality measures from Appendix A.4 (four each for the offensive and defensive teams) is similar to the Yurko+ model, except we replace the point spread terms with eight linear terms, one for each metric.

Baldwin+. We denote the best XGBoost EP model we could find by Baldwin+. The best XGBoost model is a XGBoost classification model which predicts the probability of each of the seven potential outcomes of the next score as a function of game-state: yardline, half seconds remaining, half, score differential, point spread, timeouts remaining for the offensive team, and era (2006-2013, 2014-2017, 2018-present). This model is trained only on first down plays. The hyperparameters of Baldwin+ are

```
{
  booster: gbtree
  objective: multi:softprob
  eval_metric: mlogloss
  num_class: 7.0
  eta: 0.087408
  gamma: 2.2223731e-05
  subsample: 0.38
  colsample_bytree: 0.8571429
  max_depth: 3
  min_child_weight: 17
  nrounds: 146.0
}.
}
```

The best XGBoost regression. The best XGBoost regression is similar to Baldwin+ except it directly predict the expected points of the next score rather than the probability of each potential outcome of the next score. It uses the same covariates as Baldwin+ and is also trained on first down plays. The hyperparameters of this model are

```
{
  booster: gbtree
  objective: reg:logistic
  eval_metric: mae
  eta: 0.0544011
  gamma: 1.3450771e-06
  subsample: 0.12
  colsample_bytree: 0.7142857
  max_depth: 7
  min_child_weight: 24
  monotone_constraints:
  - -1.0 (for yardline)
}.
}
```

```

    - 0.0 (for half seconds remaining)
    - 1.0 (for era)
    - 1.0 (for offensive team's timeouts remaining)
    - 0.0 (for half)
    - 0.0 (for score differential)
    - -1.0 (for point spread)
nrounds: 234.0
}.

```

EP⁽⁰⁾ model. The simple EP⁽⁰⁾ model which we fit on hold-out data from 1999-2005 in order to craft our team quality metrics is a linear regression in which the points of the next score is a linear function of yardline, log yards to go, and one combined indicator for third and fourth down.

A.3 Score differential bias details

Score differential influences the expected points of the next score. For instance, when a team is leading by a large number of points at the end of a game, it will sacrifice scoring points for letting time run off the clock (Yurko et al., 2018). Burke (2014) fits his EP model on all plays from the first and third quarters featuring a score differential within 10 points, allowing him to ignore score differential as a covariate without incurring much bias. His model, however, isn't suitable for decision making in the second and fourth quarters, where EP substantially differs from the first and third quarters. Yurko et al. (2018) and Baldwin (2021b), on the other hand, use a row weighting procedure to adjust for score differential. Specifically, they weight each row (play) such that the closer a play is to having a score differential magnitude of zero (tie game), the more weight that play is given. As team play style changes substantially depending on the interaction between score differential and time remaining, a more fine-grained EP model would instead include both score differential and time remaining as covariates. In this Section we discuss the details and issues with Burke's data truncation and Yurko et al. and Baldwin's row weighting procedure.

Burke's data truncation. (Burke, 2014) truncates his dataset by only including plays in the first and third quarters where the score differential is within 10 points. By restricting the range of the score differential, Burke removes garbage time plays. It is important to remove garbage time plays because fourth-down decision models are useful when the game is still winnable. Additionally, by removing all second and fourth quarter plays, Burke can exclude score differential as a covariate without incurring much bias. Expected points, however, is substantially different at the end of a half than at the beginning of a half. In particular, the less time remaining in a half, the more likely that neither team scores before the half ends due to the clock running out. Therefore, a more fine-grained EP model would include score differential and time remaining as covariates and train on

plays from all four quarters. In other words, the fundamental problem with Burke’s data truncation is that it removes too many plays from the dataset.

Alternative data truncation. In order to fit an EP model which allows us to make better fourth-down decisions in the second and fourth quarters, we use an alternative data truncation to Burke’s. Our data truncation keeps plays in the second and fourth quarters, allowing us to model EP as a function of score differential and time remaining, but eliminates garbage time plays. In particular, we construct a simple win probability model $WP^{(0)}$ as a function of score differential and time remaining, using held-out data from 1999 to 2005 to avoid data bleed. Then, we keep all plays with $WP^{(0)} \in [0.15, 0.85]$. We discuss the details of our $WP^{(0)}$ model in Appendix B.2.

Yurko et al. and Baldwin’s row weighting procedure. Yurko et al. (2018) and Baldwin (2021b), on the other hand, use a row weighting procedure to address the score differential problem. Because they primarily use expected points for player valuation rather than decision making, their row weighting procedure isn’t as much of an issue for them as it is for us. They weight each row (play) such that the closer a play is to having a score differential magnitude of zero (tie game), the more weight that play is given. Specifically, they weight the i^{th} play in their dataset using the score differential S_i by

$$w_i(S_i) = \frac{\max_j |S_j| - |S_i|}{\max_j |S_j| - \min_j |S_j|}. \quad (\text{A.6})$$

The primary problem with this row weighting procedure is that it emphasizes plays where the game is closer to being tied, even though a coach may want to use an EP-based decision making procedure for other values of the score differential (e.g., being down by seven). Plays in which the team with possession is down by, say, seven points should have different EP values (depending on the time remaining) than if the game were tied, so using such a weighted EP model for decision making at all score differentials is wrong. Moreover, even though garbage time plays are downweighted, they still influence the resulting EP model.

In addition to weighting plays by their score differential, Yurko et al. and Baldwin also weight plays according to their “distance” to the next score in terms of the number of drives. For each play i , they find the difference in the number of drives from the next score, $D_i = d_{\text{next score}} - d_i$, where $d_{\text{next score}}$ and d_i are the drive numbers for the next score and play i , respectively. This difference is then scaled from zero to one in the same way as the score differential in Formula (A.6). The score differential and drive score difference weights are then added together and again rescaled from zero to one in the same manner, resulting in a combined weighting scheme. By combining the two weights, they are placing equal emphasis on both the score differential and the number of drives until the next score.

The primary problem with weighting plays by the number of drives until the next score is that it introduces bias. Possessions at the beginning of an epoch (which are downweighted) feature punts and turnovers, and possessions at the end of an epoch (which are upweighted) feature scores, and more generally plays at yardlines closer to scoring. Thus, for instance, an EP model using this row weighting scheme may overestimate EP at yardlines close to scoring.

In this paper, we do not use either of these row weighting procedures in fitting our EP models. Additionally, in the EP model comparison of Section 2.3, we fit Yurko et al. and Baldwin’s EP models without these row weights to test the quality of their underlying model structures.

A.4 Adjusting for team quality

In this Section we create our own measures of offensive and defensive quality. In crafting our player and team quality metrics, we use the result of each play, rather than each possession or game, because it leads to better predictive performance. A common and good way of quantifying play success is *expected points added* (EPA) (Yurko et al., 2018). The EPA of play n is the difference in expected points between the end and the beginning of the play,

$$\text{EPA}_n = \text{EP}_n - \text{EP}_{n-1}. \quad (\text{A.7})$$

Recall, however, that our goal is to develop team quality metrics to mitigate selection bias in expected points models. This creates a paradox: we want to use EP in order to fit EP models. If we were not careful, we would first fit an EP model that ignores team quality, denoted $\text{EP}^{(0)}$. Then we would use $\text{EP}^{(0)}$ to create EPA based team quality metrics (e.g., EPA per play). Finally, we would fit an EP model which adjusts for team quality. This, however, would introduce data bleed into our analysis. Specifically, we would use the response column \mathbf{y} to fit $\text{EP}^{(0)}$, which we would then transform into features used to fit the team quality adjusted EP model, which is to use \mathbf{y} to predict \mathbf{y} . So, to avoid data bleed, we remove all plays from 1999 to 2005 from our play-by-play dataset and fit $\text{EP}^{(0)}$ to this held-out data. Now, we can use $\text{EPA}^{(0)}$ to fit EP models which adjust for team quality. The specification of our simple EP⁽⁰⁾ model is detailed in Appendix A.2.

Now, we use $\text{EPA}^{(0)}$ (EPA derived from $\text{EP}^{(0)}$) to craft our team and player quality metrics. For concreteness, consider deriving the quarterback quality of Patrick Mahomes. Index all the plays from 2006 to 2021 in which Mahomes passes or runs the ball by n . We define Mahomes’ quarterback quality prior to play n by a weighted sum of the $\text{EPA}^{(0)}$ of his previous plays,

$$q_n := \gamma_n \cdot \alpha \cdot q_{n-1} + \text{EPA}^{(0)}_{n-1}, \quad (\text{A.8})$$

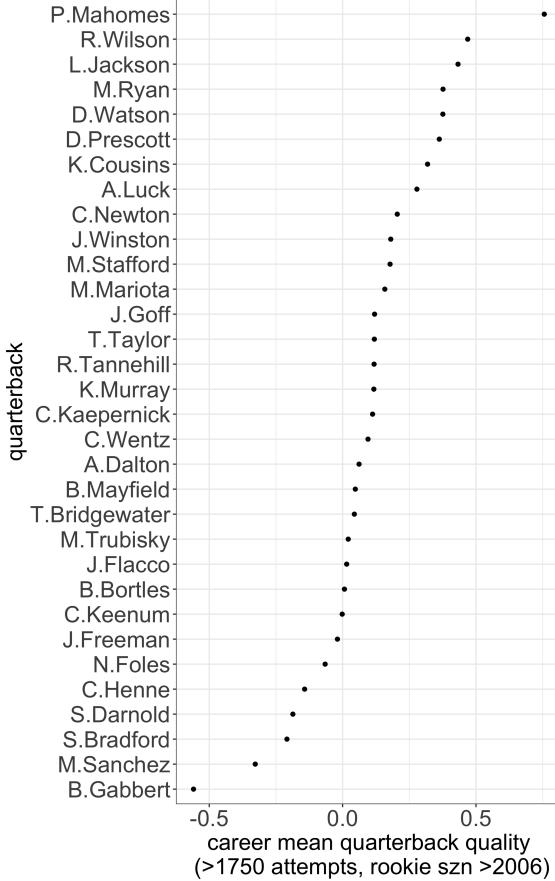


Figure 12: The career mean quarterback quality of each quarterback with over 1750 attempts and whose rookie season came after 2006.

where $q_0 := 0$ and $\text{EPA}^{(0)}_0 := 0$. We use an exponential decay weight α to upweight more recent plays (Medvedovsky and Patton, 2022). We set $\alpha = 0.995$ for each of our team and player quality metrics. For instance, a play which occurred 138 plays ago is weighted half as much as the previous play since $0.995^{138} = 0.5$. Also, a play which occurred 459 plays ago is weighted one-tenth as much as the previous play since $0.995^{459} = 0.1$. Additionally, at the start of each season, we shrink by a multiplicative factor γ . So, the shrinkage weight γ_n of play n is γ if this is Mahomes' first play of the season, otherwise it is 1. We use different shrinkage values γ for each team and player quality metric, shown in Table 5. We tuned these γ values to optimize predictive performance on a held-out validation set. A smaller γ reflects a quantity that is noisier from year to year. Finally, we standardize each offensive and defensive quality metric to have mean zero and standard deviation 1/2. In Figure 12 we plot the career mean quarterback quality of each quarterback with over 1750 attempts and whose rookie season came after 2006. As expected, Patrick Mahomes has by far the highest quarterback quality. We construct the other team quality metrics described in Table 5 in a similar fashion, via Formula (A.8).

variable	γ
quarterback quality	3/4
non-quarterback offensive quality	1/2
defensive quality against the pass	1/3
defensive quality against the run	1/3

Table 5: Summary of team quality metrics computed via Formula (A.8) and their start-of-season shrinkage parameters γ . Each of these metrics apply to both the offensive and defensive teams in a play, yielding eight total metrics.

A.5 Offense is more important than defense for scoring points

In Figure 13 we visualize the best multinomial logistic regression EP model³⁴ which uses our eight hand-crafted team quality measures³⁵ (four each for the offensive and defensive teams). Each metric is standardized to have mean 0 and s.d. 1/2. For each team quality metric, we plot expected points as a function of yardline for various values of that metric, holding each of the other seven team quality values fixed at 0 (representing the average). The offensive team’s quarterback quality and the defensive team’s quarterback quality have the largest and second largest impact on expected points, respectively. We visualize this “impact” by the width of the lines in the spaghetti plot. The offensive team’s remaining offensive quality and the defensive team’s remaining offensive quality have the third and fourth largest impact on expected points, respectively. Finally, each defensive quality metric has minimal impact on expected points. This provides further evidence for the notion that offense is more important than defense in scoring points, and hence in constructing a successful football team.³⁶

A.6 Team quality knockoffs

Defensive quality has a minimal impact on scoring points, and we wonder whether this impact is significant. To investigate, we modify the knockoffs procedure from [Candes et al. \(2016\)](#) and [Ren et al. \(2020\)](#) to fit the autocorrelated nature of our football play-by-play dataset. In particular, the original model-X knockoffs framework assumes each predictor \mathbf{X}_j consists of i.i.d. observations, which does not hold for our team quality metrics. Across 25 runs of the knockoffs procedure, each offensive quality metric is selected each time, indicating that offensive quality matters in predicting the points of the next score. The defensive quality metrics, on the other hand, are not selected most of the time. In other words, our defensive quality metrics are not significantly better at predicting points than random noise.

³⁴See Appendix A.2 for details.

³⁵See Appendix A.4 for details.

³⁶[Eager \(2020\)](#) also provides evidence that offense is more valuable than defense.

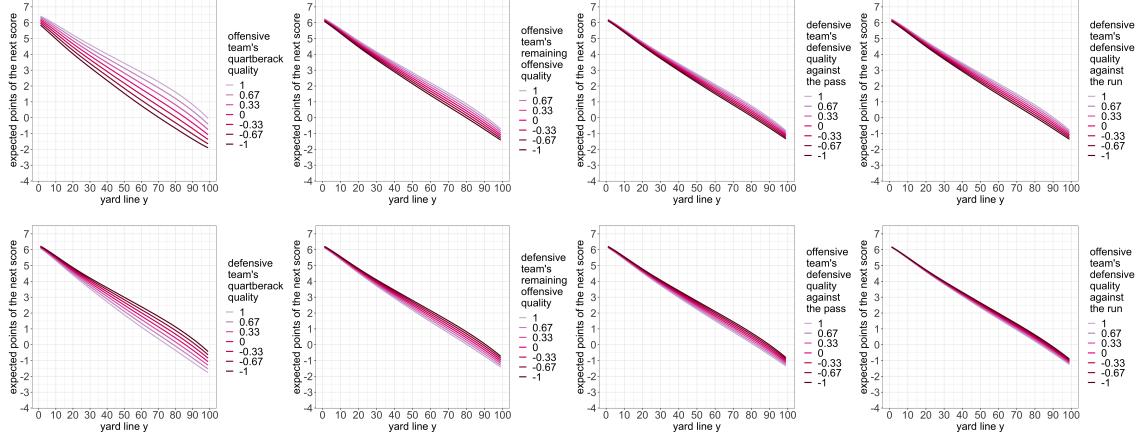


Figure 13: Expected points as a function of yardline for various values of our eight hand-crafted team quality measures, according to a multinomial logistic regression model, holding other confounders constant. Quarterback quality has the largest impact on EP. Defensive quality has a minimal impact on EP.

Each team quality metric has a similar form. For instance, Mahomes’ quarterback quality prior to play n from Formula (A.8) is equivalently expressed as

$$q_n := \sum_{k=1}^n \left[\alpha^{n-k} \prod_{j=k+1}^n \gamma_j \right] \text{EPA}^{(0)}_{k-1}. \quad (\text{A.9})$$

We think of the result of play $k-1$, and hence $\text{EPA}^{(0)}_{k-1}$, as the realization of a random variable. As the rolling sum of these play results, each resulting team quality vector (q_n) does not consist of independent observations. But, due to Formula (A.9), to construct a valid knockoff of each team quality metric we need only construct a valid knockoff of the vector $\text{EPA}^{(0)}$.

To construct a knockoff of the observed $\text{EPA}^{(0)}$ vector, we need to model the distribution of the $\text{EP}^{(0)}$ vector. We model $\text{EP}^{(0)}$ using ridge regression³⁷ and use Gaussian noise with mean given by predictions from the ridge regression³⁸ to generate a synthetic $\text{EP}^{(0)}$ vector. Then, we construct a synthetic $\text{EPA}^{(0)}$ vector from this synthetic $\text{EP}^{(0)}$ vector, recalling that $\text{EPA}^{(0)}_{k-1} = \text{EP}^{(0)}_k - \text{EPA}^{(0)}_{k-1}$. Finally, we keep just every 5th observation, which makes the vector $\text{EPA}^{(0)}$ nearly uncorrelated and its generated synthetic counterpart nearly uncorrelated. Thus, we generate a knockoff $\widetilde{\text{EPA}}^{(0)}$ of $\text{EPA}^{(0)}$ such that they are approximately equal in distribution, visualized in Figure 14.

³⁷Specifically, we use ridge regression to model $\text{EP}^{(0)}$ as a function of yardline, an indicator for 3rd or 4th down, and log yards to go to match the $\text{EP}^{(0)}$ model from Appendix A.2, and indicators for offensive team-season and defensive team-season since the result of a play depends on the teams playing.

³⁸and with s.d. 1/20

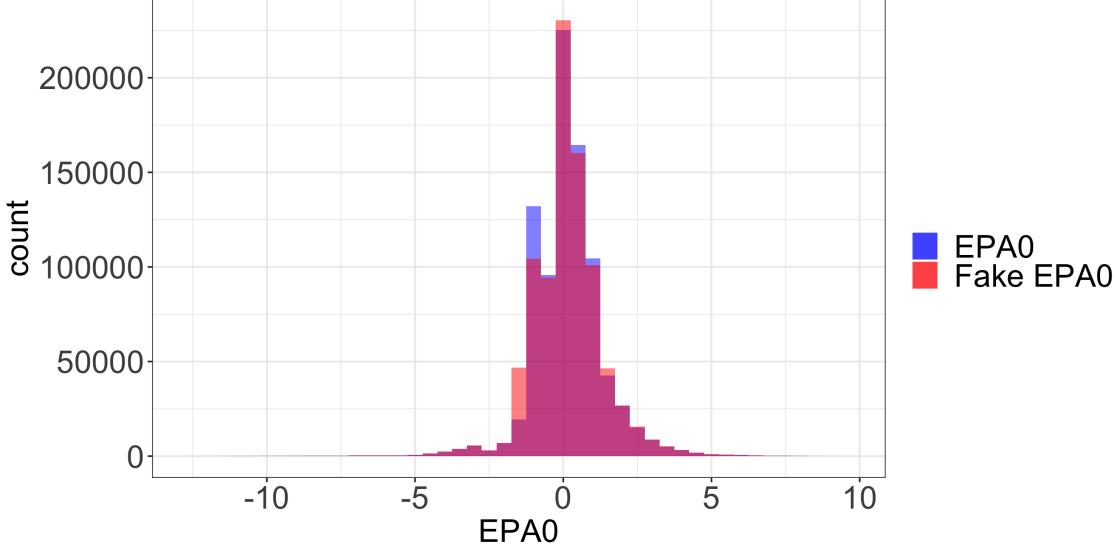


Figure 14: Distribution of $\text{EPA}^{(0)}$ and a knockoff $\tilde{\text{EPA}}^{(0)}$.

With our knockoff of each team quality metric constructed from our knockoff of $\text{EPA}^{(0)}$, the procedure from [Candes et al. \(2016\)](#) holds. In particular, for any subset $S \subset \{1, \dots, p\}$, denote the original predictor matrix by \mathbf{X} and its knockoff by $\tilde{\mathbf{X}}$. Let $(\mathbf{X}, \tilde{\mathbf{X}})_{\text{swap}(S)}$ refer to concatenating \mathbf{X} with $\tilde{\mathbf{X}}$ and swapping each predictor in \mathbf{X} with its knockoff. Then, we still approximately have

$$(\mathbf{X}, \tilde{\mathbf{X}})_{\text{swap}(S)} \stackrel{d}{=} (\mathbf{X}, \tilde{\mathbf{X}}). \quad (\text{A.10})$$

Also, we construct the knockoffs $\tilde{\mathbf{X}}$ without looking at the response variable (e.g., points of the next score). Therefore, Lemma 3.2, Lemma 3.3, and Theorem 3.4 from [Candes et al. \(2016\)](#) still hold.

Now, using a knockoff $\tilde{\mathbf{X}}$ of \mathbf{X} , we fit a lasso regression where the response variable is the points of the next score and the predictor variables are the game-state, the eight real team quality metrics, and the eight knockoff team quality metrics. As in [Candes et al. \(2016\)](#), we select the team quality variables j where the statistics

$$W_j = |\hat{b}_j(\lambda)| - |\hat{b}'_j(\lambda)| \quad (\text{A.11})$$

are larger than a smartly chosen threshold. Here, λ is the lasso penalty hyperparameter, \hat{b}_j is the fitted lasso coefficient of the real team quality metric, and \hat{b}'_j is the fitted lasso coefficient of the knockoff team quality metric. Moreover, as in [Ren et al. \(2020\)](#), we de-randomize the knockoffs procedure by repeating the procedure 25 times.

In Table 6 we show the proportion of the 25 knockoffs procedures in which each team quality vari-

able is not selected. Each offensive quality metric is selected each time, indicating that offensive quality matters in predicting the points of the next score. The defensive quality metrics, on the other hand, are not selected much of the time. This provides evidence that our defensive quality metrics are not significantly better than noise at predicting the points of the next score.

variable	not selected proportion
offensive team's quarterback quality	0
defensive team's quarterback quality	0
offensive team's non-quarterback offensive quality	0
defensive team's non-quarterback offensive quality	0
defensive team's defensive quality against the pass	0.88
offensive team's defensive quality against the pass	0.84
defensive team's defensive quality against the run	0.40
offensive team's defensive quality against the run	0.76

Table 6: The proportion of knockoffs procedures in which each team quality variable is not selected.

B Win probability details

B.1 Statistical win probability models

Yurko’s method. Yurko et al. (2018) use a generalized additive model (GAM) (Hastie and Tibshirani, 1986) to estimate win probability from historical data. A GAM allows win probability to vary according to the sum of smooth nonlinear functions of the dependent variables. Formally, they model

$$\log\left(\frac{\mathbb{P}(y=1)}{\mathbb{P}(y=0)}\right) = s(\mathbb{E}[S]) + s\left(\frac{\mathbb{E}[S]}{s_g+1}\right) + \sum_{j=1}^2 h_j \cdot (s(s_h) + \beta_2 \cdot u \cdot t_{\text{off}} + \beta_1 \cdot u \cdot t_{\text{def}}), \quad (\text{B.1})$$

where s is a smooth function while the other variables are defined in Table 7.

Baldwin’s method. Baldwin (2021a) uses XGBoost (Chen and Guestrin, 2016) to estimate win probability from historical data. The response variable is again a binary variable indicating whether the team with possession wins the game. Baldwin models win probability as a function of score differential, game seconds remaining, half seconds remaining, yardline, down, yards to go, whether the team with possession is at home, whether the team with possession receives the second half kickoff, and the number of timeouts remaining for each team. Baldwin uses two additional features

³⁹We use EP⁽⁰⁾, an EP model fit on hold-out data from 1999-2005, to avoid data bleed in implementing Yurko et al.’s win probability model (see Appendix A.2 for details).

variable	variable description
y	1 if the team with possession wins the game, else 0
u	1 if time remaining in half is under two minutes, else 0
h_1	1 if first half, else 0
h_2	1 if second half, else 0
s_h	half seconds remaining
s_g	game seconds remaining
t_{off}	timeouts remaining for offensive team
t_{def}	timeouts remaining for defensive team
$\mathbb{E}[S]$	expected score differential = EP + S where EP = expected points ³⁹ and S = score differential

Table 7: Variables in Yurko et al.’s win probability model.

to capture the change in impact of point spread and score differential over the course of a game,

$$\text{spread-time} = (\text{point spread}) \cdot \exp \left(-4 \cdot \left(1 - \frac{3600}{\text{game seconds remaining}} \right) \right) \quad (\text{B.2})$$

and

$$\text{diff-time-ratio} = (\text{score differential}) \cdot \exp \left(-4 \cdot \left(1 - \frac{3600}{\text{game seconds remaining}} \right) \right). \quad (\text{B.3})$$

Baldwin includes monotonic constraints for many covariates,⁴⁰ which reduces overfitting. It makes sense to include monotonic constraints in WP classification models but not in EP classification models because, for instance, the probability that the next score is a field goal is not monotonic in yardline but win probability is.

Other methods. Lock and Nettleton (2014) and ESPN (via Burke and Gargiulo) also estimate win probability from historical data – the former use a random forest and the latter an ensemble of machine learning methods (Lopez, 2017). On the other hand, Schneider (2023) uses live betting market data and Pro Football Reference (2023) uses a normal approximation to estimate win probability. In this paper, we give detailed descriptions of Baldwin’s method since it underlies his popular public fourth-down decision framework,⁴¹ and of Yurko’s method since we use it as a catalytic prior. Although Burke’s method is also used for fourth-down decision making in the public sphere, his method is proprietary.

⁴⁰ Specifically, he includes monotonic constraints for yardline, yards to go, down, score differential, timeouts remaining for each team, spread-time, and diff-time-ratio.

⁴¹ via Baldwin’s Twitter @ben_bot_baldwin.

B.2 WP model specification details

Yurko+. We denote the best GAM WP model we could find by Yurko+. As before, the response variable is a binary variable indicating whether the team with possession won the game, and the model structure is the same from Formula (B.1) except with different game-state covariates. This model is trained only on first down plays. Formally, the model is

$$\begin{aligned} \log\left(\frac{\mathbb{P}(y=1)}{\mathbb{P}(y=0)}\right) = & s(S) + \beta_1 \cdot \left(\frac{\mathbb{E}[S]}{s_g + 1} \right) + \beta_2 \cdot \text{spline}(\text{yardline}, \text{knots} = (7, 33, 67, 93)) \\ & + \beta_3 \cdot (\text{point spread}) + \beta_4 \cdot (\text{point spread}) \cdot (\text{yardline}) \\ & + \sum_{j=1}^2 h_j \cdot (s(s_h) + \beta_{5j} \cdot u \cdot \mathbb{I}(t_{\text{off}} = 0) + \beta_{6j} \cdot u \cdot \mathbb{I}(t_{\text{def}} = 0)) \\ & + \sum_{k=1}^3 \beta_{7k} \cdot \mathbb{I}(\text{era} = k), \end{aligned} \quad (\text{B.4})$$

where era is a categorical variable denoting whether the season is in 2006-2012, 2013-2017, or 2018-present, and the other variables are detailed in Table 7.

Baldwin+. We denote the best XGBoost WP model we could find by Baldwin+. The best XGBoost model is a XGBoost classification model which predicts win probability as a function of game-state: score differential, game seconds remaining, point spread, yardline, receive 2nd half kickoff indicator, timeouts, and

$$\text{scoreTimeRatio} = \frac{\text{score_differential}}{0.01 + \text{game_seconds_remaining}} \quad (\text{B.5})$$

to help fit win probability at the very end of the game. This model is trained only on first down plays. This model uses fewer covariates than Baldwin's original model and includes a few new ones. The hyperparameters of Baldwin+ are

```
{
  eta: 0.0658986
  gamma: 0.0079786
  subsample: 0.98
  colsample_bytree: 0.875
  max_depth: 4
  min_child_weight: 4
  monotone_constraints:
  - 1.0 (for score differential)
```

```

    - 0.0 (for game seconds remaining)
    - -1.0 (for point spread)
    - -1.0 (for yardline)
    - 1.0 (for score-time-ratio)
    - 0.0 (for receive 2nd half kickoff)
    - 1.0 (for offensive team's timeouts remaining)
    - -1.0 (for defensive team's timeouts remaining)
nrounds: 189.0
test_loss: 0.4413572
}.

```

WP⁽⁰⁾ model. The simple WP⁽⁰⁾ model which we fit on hold-out data from 1999-2005 in order to truncate our dataset in a way that removes garbage time plays but keeps plays in the second and fourth quarters to train our EP models is a logistic regression as a function of time remaining, score differential, and their interaction.

B.3 WP model comparison

In this Section, we compare the out-of-sample predictive performance of various WP models. Our full dataset consists of all football plays from 2006 to 2021. The dataset is clustered into games, as plays from each game share the same winning team. To keep the clustered nature of our dataset intact and to avoid data bleed, we split our dataset in half by randomly sampling 50% of all games. The first down plays from the first 50% of these games form the hold-out test set. We test on first down plays because, as discussed in Appendix C.1, fourth-down decision making relies on the value of having a first down. The plays from the other 50% of these games form the training set. To tune XGBoost models, we split the training set in half by randomly sampling 50% of the games from the training set. The plays from the first 50% of these games form the XGBoost training set, and the remaining plays form the validation set for hyperparameter tuning. We then tune our XGBoost models in a similar fashion as [Baldwin \(2021a\)](#).

We visualize the results of our model comparison in Table 8. We discussed Yurko et al. and Baldwin’s models in Section B.1. We give detailed descriptions of the best GAM (Yurko+) and XGBoost (Baldwin+) models in Appendix B.2.

An improved GAM outperforms Baldwin’s XGBoost because the latter overfits. An improved XGBoost model outperforms an improved GAM because win probability is a highly nonlinear function of the interacting fundamental variables of score differential and time remaining. The best catalytic machine learning model only slightly edges out the best XGBoost model because XGBoost

model name	model type	team quality	out-of-sample logloss
Catalytic	Catalytic	point spread	0.438
Baldwin+	XGBoost	point spread	0.439
Yurko+	GAM	point spread	0.442
Baldwin	XGBoost	spread-time	0.476
Yurko	GAM		0.480

Table 8: Predictive performance of various WP models.

classification for win probability takes advantage of monotone constraints (e.g., because win probability is monotone in yardline), whereas XGBoost classification for expected points couldn't.⁴² The best catalytic model used the Baldwin+ XGBoost classification model as the complex target model and the Yurko+ GAM model as the catalytic prior. We used $M = 25,000$ synthetic win probability estimates generated from the catalytic prior. We tuned the catalytic hyperparameter M in a similar manner as the standard XGBoost hyperparameters. There are significantly fewer generated datapoints from the WP catalytic prior than from the EP catalytic prior because we generate fake win probability estimates $\widehat{WP} \in [0, 1]$ rather than synthetic outcomes of the next score $y \in \{\text{TD}, \text{FG}, \dots\}$. Win probability estimates \widehat{WP} carry a similar amount of information as, say, 100 win outcomes in $\{0, 1\}$: for instance, a win probability estimate of 0.45 can be equivalently represented by 45 generated ones and 55 zeros.

B.4 Simulation study details

Generating plays. Formally, the outcome of the n^{th} play of the g^{th} game is

$$\xi_{gn} \stackrel{iid}{\sim} \pm 1. \quad (\text{B.6})$$

The game starts at midfield, $X_{g0} = L/2$, and the game begins tied, $S_{g0} = 0$. The field position at the start of play n is

$$X_{g,n+1} := \begin{cases} X_{gn} + \xi_{gn} & \text{if } 0 < X_{gn} + \xi_{gn} < L \text{ (not a TD)} \\ L/2 & \text{else,} \end{cases} \quad (\text{B.7})$$

⁴²See Section 2.1 for details.

and the score differential at the start of play n is

$$S_{g,n+1} := \begin{cases} S_{gn} + 1 & \text{if } X_{gn} + \xi_{gn} = 0 \text{ (TD)} \\ S_{gn} - 1 & \text{if } X_{gn} + \xi_{gn} = L \text{ (opp. TD)} \\ S_{gn} & \text{else.} \end{cases} \quad (\text{B.8})$$

The response column win is

$$y_{gn} \equiv y_{g,N+1} := \begin{cases} 1 & \text{if } S_{g,N+1} > 0 \\ 0 & \text{if } S_{g,N+1} < 0 \\ \text{Bernoulli}(1/2) & \text{else (overtime).} \end{cases} \quad (\text{B.9})$$

As in our dataset of real football plays, this response column is highly autocorrelated – plays from the same game share the same draw of the winner of the game.

Generating observational data. We create a dataset of plays from G games. Each game consists of N plays, and the field consists of L yardlines. The results from each game yield a simulated dataset

$$\mathcal{D} = \{(n, X_{gn}, S_{gn}, y_{gn}) : n = 1, \dots, N \text{ and } g = 1, \dots, G\}. \quad (\text{B.10})$$

True win probability. The true win probability

$$\text{WP}(n, x, s) := \mathbb{P}(S_{g,N+1} > 0 | X_{gn} = x, S_{gn} = s) \quad (\text{B.11})$$

of our simplified version of football is computed explicitly using dynamic programming,

$$\text{WP}(N+1, x, s) = \begin{cases} 1 & \text{if } s > 0 \\ 1/2 & \text{if } s = 0 \\ 0 & \text{if } s < 0, \end{cases} \quad (\text{B.12})$$

and

$$\text{WP}(n-1, x, s) = \begin{cases} \frac{1}{2}\text{WP}(n, \frac{L}{2}, s+1) + \frac{1}{2}\text{WP}(n, x+1, s) & \text{if } x = 1 \\ \frac{1}{2}\text{WP}(n, x-1, s) + \frac{1}{2}\text{WP}(n, \frac{L}{2}, s-1) & \text{if } x = L-1 \\ \frac{1}{2}\text{WP}(n, x-1, s) + \frac{1}{2}\text{WP}(n, x+1, s) & \text{else.} \end{cases} \quad (\text{B.13})$$

Visualizing the simulation study results. In Figure 15 we visualize the MAE of WP estimates and the confidence interval lengths and coverages, averaged over all of the simulations.

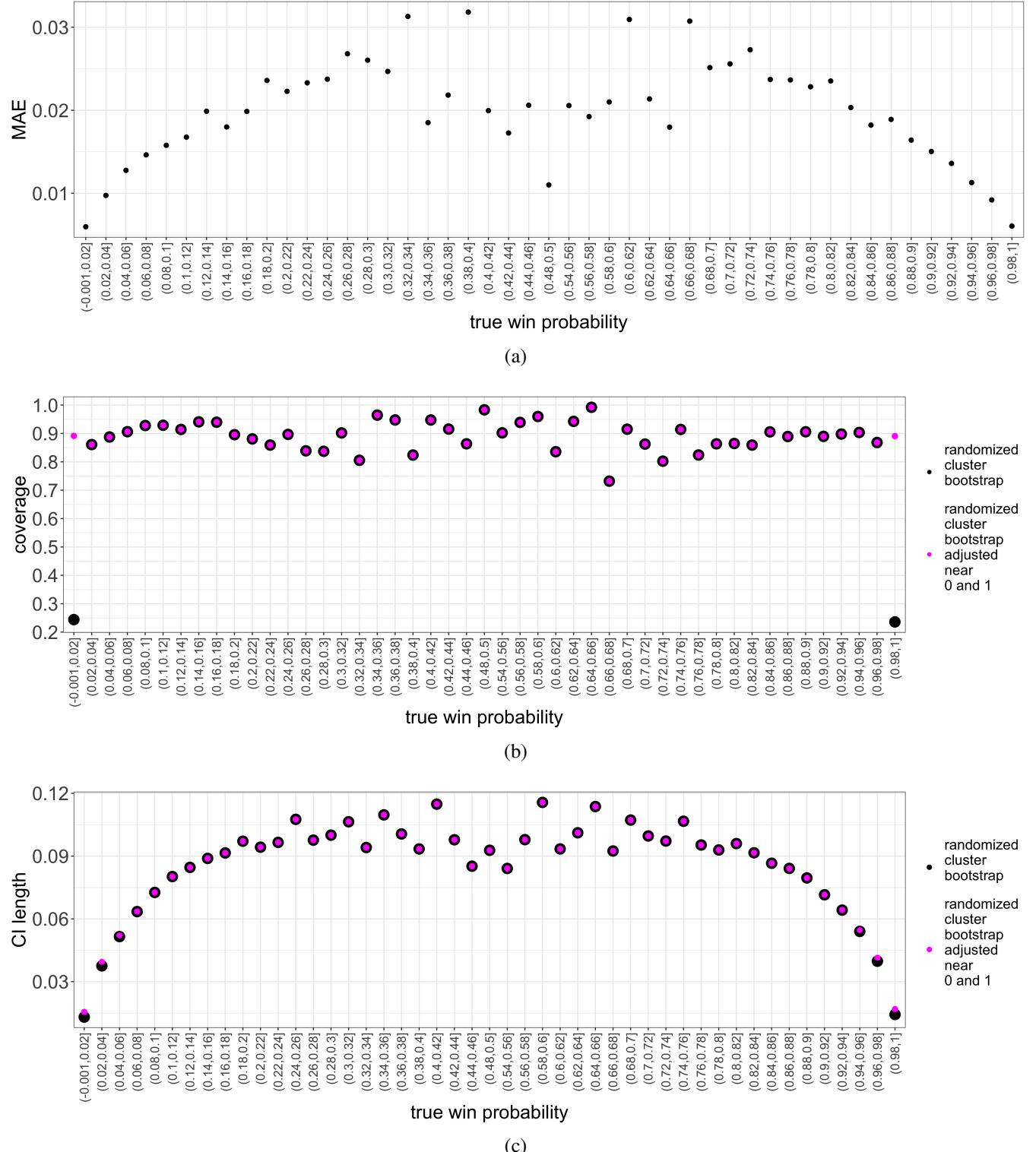


Figure 15: As a function of true WP, MAE of true and estimated WP (Figure (a)), coverage of true WP by randomized cluster bootstrap (Figure (b)), and confidence interval length of randomized cluster bootstrap (Figure (c)).

C Fourth-down decision making details

C.1 Fourth-down decision framework

In this Section we compute the value (e.g., expected points or win probability) of the next game-state if a team punts, kicks a field goal, or attempts a conversion.

The value of a punt. Suppose the offensive team has a fourth down at yardline y . If the offensive team punts, the opposing team has a first down at the next yardline, which we think of as a random variable. Hence the value of punting is negative the opponent's expected value of having a first down at the next yardline y' ,

$$\begin{aligned} & - \sum_{y'} V_1(\text{yardline } y') \cdot \mathbb{P}(\text{yardline after punting is } y') \\ &= -\mathbb{E}_{\text{punt}}[V_1(\text{yardline } y')]. \end{aligned} \tag{C.1}$$

Here, V_1 is the value of having a first down (according to EP or WP). For simplicity, we instead compute the value of having a first down at the expected next yardline after punting,

$$-V_1(\text{yardline } \mathbb{E}_{\text{punt}}[y']). \tag{C.2}$$

Because V_1 is linear in yardline for many game-states, this simplification is reasonable. We discuss the details of our model of the expected next yardline after punting, which is a function of yardline and punter quality, in Appendix C.2. Note that in computing V_1 , we flip the game-state variables which are relative to the team with possession (e.g., score differential, team quality metrics, timeouts remaining, etc.) and we don't alter the game-state variables which apply to both teams (e.g., time remaining, etc.).

The value of a field goal. Suppose the offensive team has a fourth down at yardline y and attempts a field goal. If the offensive team misses the field goal, the opposing team has a first down at yardline $100 - y$. If it makes the field goal, the offensive team scores three points and the opposing team has a first down after a kickoff. Now, the expected value of kicking a field goal is

$$V(\text{make FG}) \cdot \mathbb{P}(\text{make FG}) - V_1(\text{yardline } 100 - y) \cdot (1 - \mathbb{P}(\text{make FG})). \tag{C.3}$$

Here, letting s denote score differential,

$$V(\text{make FG}) := \begin{cases} 3 & \text{if } V = \text{EP}, \\ -\mathbb{E}_{\text{kickoff}}[V_1(\text{yardline } y', s \leftarrow -s - 3)] & \text{if } V = \text{WP}, \end{cases} \tag{C.4}$$

and V_1 is the value of having a first down (according to EP or WP). As before, in computing V_1 and V (make FG), we flip the game-state variables which are relative to the team with possession (e.g., score differential, team quality metrics, timeouts remaining, etc.) and we don't alter the game-state variables which apply to both teams (e.g., time remaining, etc.). We discuss the details of our field goal probability model, which is a function of yardline and kicker quality, in Appendix C.3. Similar to the simplification we made in computing the value of a punt, here we simplify by

$$\begin{aligned} & -\mathbb{E}_{\text{kickoff}}[V_1(\text{yardline } y')] \\ & \approx -V_1(\text{yardline } \mathbb{E}_{\text{kickoff}}[y']) \\ & \approx -V_1(\text{yardline } 75), \end{aligned} \tag{C.5}$$

assuming the kickoff ends in a touchback.

The value of going for it. Suppose the offensive team has a fourth down and z yards-to-go at yardline y . If the offensive team goes for it and gains $\Delta \geq z$ yards, then in the next play it has a first down at yardline $y - \Delta$. Conversely, if the offensive team goes for it and gains $\Delta < z$ yards, then in the next play the opposing team has a first down at yardline $100 - (y - \Delta)$. Hence the expected value of going for it on fourth down is

$$\sum_{\Delta \geq z} V_1(\text{yardline } y - \Delta) \cdot \mathbb{P}(\text{gain } \Delta \text{ yards}) - \sum_{\Delta < z} V_1(\text{yardline } 100 - (y - \Delta)) \cdot \mathbb{P}(\text{gain } \Delta \text{ yards}), \tag{C.6}$$

where V_1 is the value of having a first down (according to EP or WP). As the probability density of the yards gained on a conversion attempt is complex, we employ a standard simplification from Burke (2009b). In particular, if the offensive team converts on fourth down and z yards-to-go at yardline y , we assume they gain z yards on that play. Also, if the offensive team fails to convert, we assume they turn the ball over at the initial yardline of the play, leaving the opposing team with a first down at yardline $100 - y$. Thus the value of going for it becomes

$$V_1(\text{yardline } y - z) \cdot \mathbb{P}(\text{convert}) - V_1(\text{yardline } 100 - y) \cdot (1 - \mathbb{P}(\text{convert})). \tag{C.7}$$

We discuss the details of our conversion probability model, which is a function of yardline, yards-to-go, and offensive and defensive quality, in Appendix C.4. As before, in computing V_1 , we flip the game-state variables which are relative to the team with possession (e.g., score differential, team quality metrics, timeouts remaining, etc.) and we don't alter the game-state variables which apply to both teams (e.g., time remaining, etc.).

C.2 Expected next yardline after a punt model

In this Section, we model the expected next yardline (from the perspective of the receiving team) after a punt as a function of yardline and punter quality (from the perspective of the punting team). Our punter quality adjustment, detailed below, is similar to our offensive and defensive quality adjustments from Section A.4.

Punter quality. We define a punter’s quality by a weighted sum of his punt yards over expected over all his previous punts in his career. To begin, we fit a simple expected next yardline after punting model $\mathbb{E}_{\text{Punt}}^{(0)}$ on a held-out dataset of all punts from 1999 to 2005 to avoid data bleed. We fit $\mathbb{E}_{\text{Punt}}^{(0)}$ using linear regression as a function of yardline (specifically, a cubic polynomial in yardline). Then, we define the *punt yards over expected* (PYOE) of the n^{th} punt by

$$\text{PYOE}_n := \text{actual yardline after the } n^{\text{th}} \text{ punt} - \mathbb{E}_{\text{Punt}}^{(0)}(\text{yardline prior to the } n^{\text{th}} \text{ punt}). \quad (\text{C.8})$$

Now, we define punter quality, using Rams’ punter Johnny Hekker for concreteness. Index all of Hekker’s punts from 2006 to 2021 by n . We define Hekker’s punter quality prior to punt n by a weighted sum of the punt yards over expected from his previous kicks,

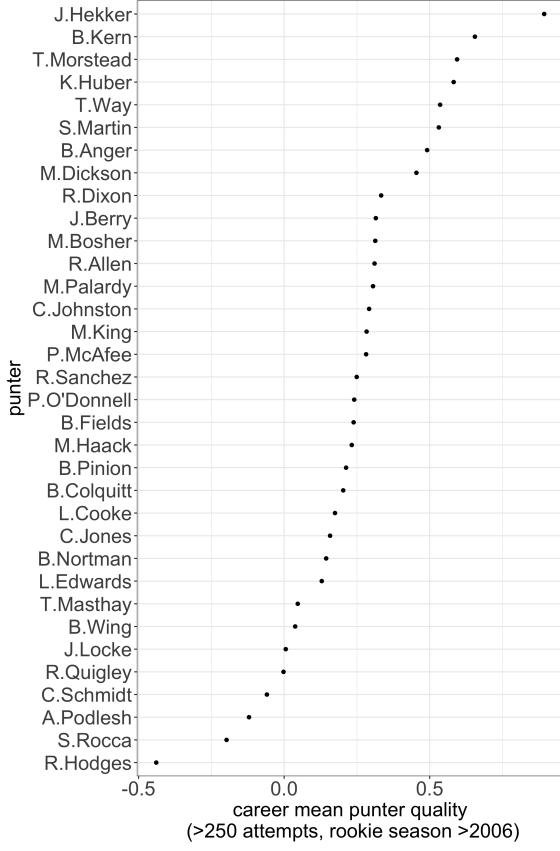
$$\text{pq}_n := \alpha \cdot \text{pq}_{n-1} + \text{PYOE}_{n-1}, \quad (\text{C.9})$$

where $\text{pq}_0 := 0$ and $\text{PYOE}_0 := 0$. As before, we use an exponential decay weight $\alpha = 0.995$ to upweight more recent punts. Finally, we standardize the punter quality column to have mean zero and standard deviation 1/2. In Figure 16a we plot the career mean punter quality of each punter with over 250 punt attempts from 2006 to 2021. As expected, Johnny Hekker has the highest punter quality.

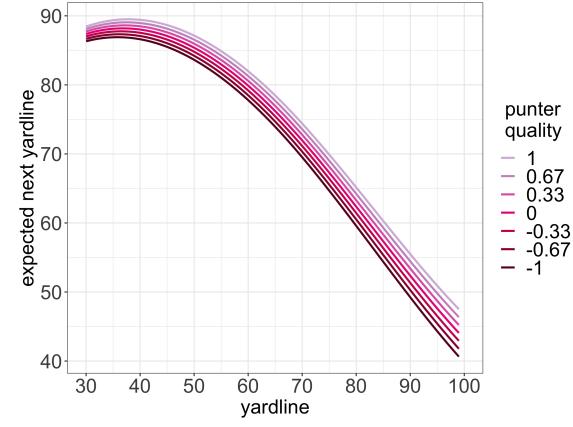
Expected next yardline after a punt model. We use linear regression to model the expected next yardline after a punt as a function of yardline and punter quality (pq). Formally, our best model of the expected next yardline after punting is

$$\begin{aligned} \mathbb{E}_{\text{punt}}[\text{next yardline}] &= \vec{\alpha} \cdot \text{spline}(\text{yardline}, \text{df} = 4) \\ &\quad + \beta_1 \cdot \text{pq} + \beta_2 \cdot \text{pq} \cdot \text{yardline}. \end{aligned} \quad (\text{C.10})$$

The model is trained on a dataset of 36,493 punts from 2006 to 2021, all beyond the 30 yardline. In Figure 16b we plot the expected next yardline after a punt according to our model as a function of yardline for various punter quality values.



(a)



(b)

Figure 16: On the left, the career mean punter quality of each punter with over 250 punts from 2006 to 2021. On the right, the expected next yardline after a punt according to our model as a function of yardline for various punter quality values.

C.3 Field goal probability model

In this Section, we model the probability that a kicker makes a field goal as a function of yardline and kicker quality. It is important to adjust for kicker quality to avoid selection bias, as good kickers attempt more field goals from long distance than bad kickers. Our kicker quality adjustment, detailed below, is similar to our punter quality adjustment from the previous Section.

Kicker quality. We define a kicker’s quality by a weighted sum of his field goal probability added over all his previous kicks in his career. To begin, we fit a simple field goal probability model $P_{FG}^{(0)}$ on a held-out dataset of all field goals from 1999 to 2005 to avoid data bleed. We fit $P_{FG}^{(0)}$ using logistic regression as a function of yardline (specifically, a cubic polynomial spline with five degrees of freedom on the yardline). Then, we define the *field goal probability added* (FGPA) of

the n^{th} field goal by

$$\text{FGPA}_n := \mathbb{I}\left(n^{th} \text{ field goal is made}\right) - P_{\text{FG}}^{(0)}(\text{yardline of the } n^{th} \text{ field goal}). \quad (\text{C.11})$$

Now, we define kicker quality, using Ravens' kicker Justin Tucker for concreteness. Index all of Tucker's field goals from 2006 to 2021 by n . We define Tucker's kicker quality prior to field goal n by a weighted sum of the field goal probability added in his previous kicks,

$$\text{kq}_n := \alpha \cdot \text{kq}_{n-1} + \text{FGPA}_{n-1}, \quad (\text{C.12})$$

where $\text{kq}_0 := 0$ and $\text{FGPA}_0 := 0$. As before, we use an exponential decay weight $\alpha = 0.995$ to upweight more recent kicks. Finally, we standardize the kicker quality column to have mean zero and standard deviation 1/2. In Figure 17a we plot the career mean kicker quality of each kicker with over 100 field goal attempts from 2006 to 2021. As expected, Justin Tucker has by far the highest kicker quality.

Field goal probability model. We use logistic regression to model the probability that a kicker makes a field goal as a function of yardline and kicker quality (kq). Formally, our best field goal probability model is

$$\log\left(\frac{\mathbb{P}(\text{make FG})}{1 - \mathbb{P}(\text{make FG})}\right) = \vec{\alpha} \cdot \text{spline}(\log(\text{yards to go}), \text{df} = 5) + \beta \cdot \text{kq}. \quad (\text{C.13})$$

Fitting this model on our dataset of 15,472 observed field goals from 2006 to 2021 yields nontrivial probability predictions for extremely long field goals that have never before been made (e.g., nontrivial probability for a 73 yard field goal from the 55 yardline). To shrink these field goal probability predictions to zero, we impute 500 synthetic missed field goals, randomly distributed from the 51 to the 99 yardline, into our dataset. In Figure 17b we plot the probability of making a field goal according to our model as a function of yardline for various kicker quality values.

C.4 Conversion probability model

In this Section, we model fourth down conversion probability as a function of yards to go and team quality. One difficulty with fitting a fourth down conversion probability model is there are only 8,258 fourth down conversion attempts in our dataset from 2006 to 2021. Existing models use third down as a proxy for fourth down, as they are also high pressure situations in which the offensive team attempts to gain a first down on that play (Romer, 2006). There may be, however, a fundamental difference in conversion probability between third and fourth down plays, perhaps due to psychological reasons. Therefore, in our model comparison, we test models on a random

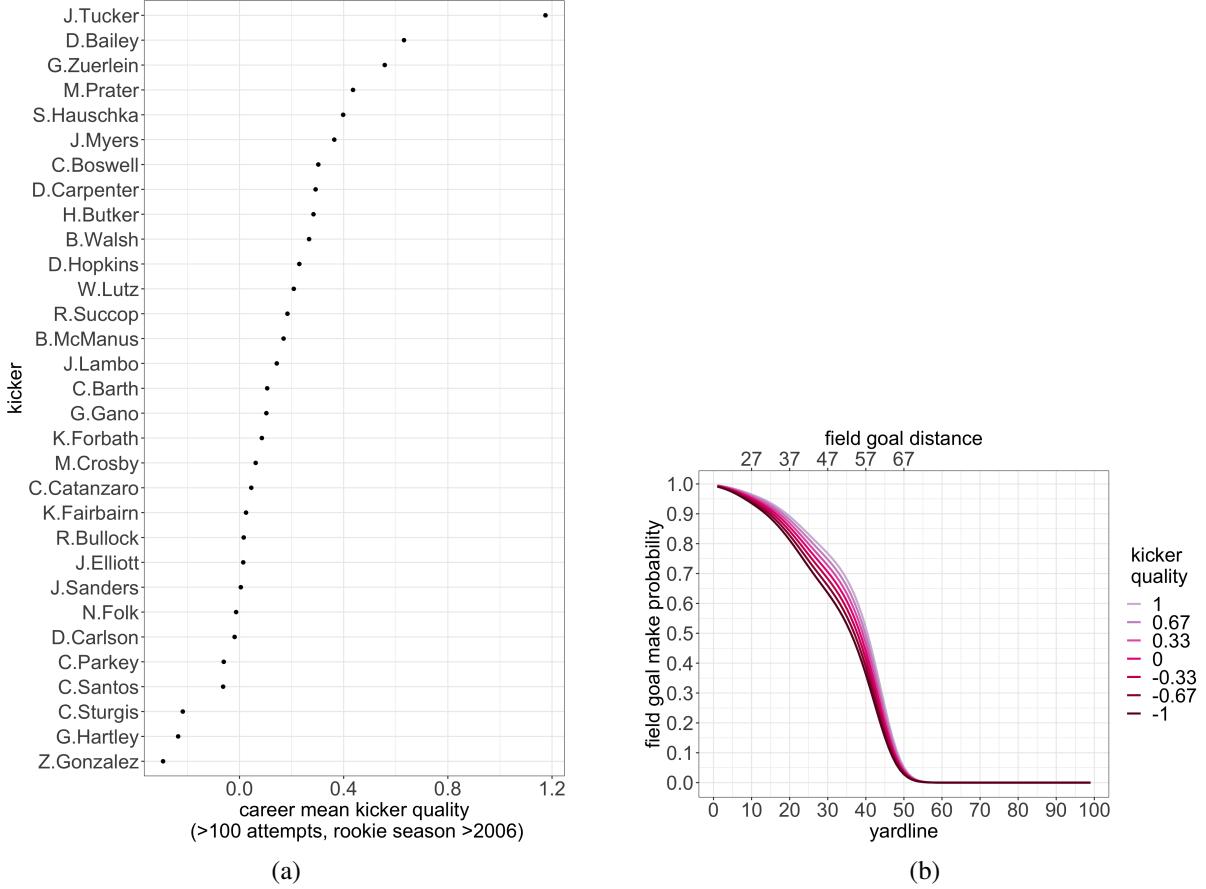


Figure 17: On the left, the career mean kicker quality of each kicker with over 100 field goal attempts from 2006 to 2021. On the right, the probability of making a field goal according to our model as a function of yardline for various kicker quality values.

50% of fourth down plays, and we train some models on a dataset consisting entirely of fourth down plays and other models on a dataset consisting of third and fourth down plays. We find that the parameters of our best conversion probability model, detailed below in Formula (C.14), borrow strength from third down plays.

Our best logistic regression model adjusts for yards to go, down (third vs. fourth down), and our offensive and defensive quality metrics from Section A.4: quarterback quality of the offensive team (qbqot), non-quarterback offensive quality of the offensive team (oqrqt), defensive quality of the defensive team against the pass (dqdtap), and defensive quality of the defensive team against

the run (dqdtar). Formally, our best conversion probability model is

$$\begin{aligned} \log\left(\frac{\mathbb{P}(\text{convert})}{1 - \mathbb{P}(\text{convert})}\right) &= \vec{\alpha}_1 \cdot \mathbb{I}(\text{fourth down}) \cdot \text{spline}(\log(\text{yards to go}), \text{df} = 4) \\ &\quad + \vec{\alpha}_2 \cdot \mathbb{I}(\text{third down}) \cdot \text{spline}(\log(\text{yards to go}), \text{df} = 4) \\ &\quad + \beta_0 + \beta_1 \cdot \text{qbqot} + \beta_2 \cdot \text{oqrrot} + \beta_3 \cdot \text{dqdtap} + \beta_4 \cdot \text{dqdtar}. \end{aligned} \quad (\text{C.14})$$

In Figure 18 we visualize conversion probability as a function of yardline for various values of yards to go. We see a large spike in conversion probability with one yard to go, potentially due to quarterback sneaks. Additionally, in Figure 19 we plot conversion probability as a function of yards to go for various values of team quality. We find that quarterback quality significantly impacts conversion probability, but the other team quality measures have little impact.

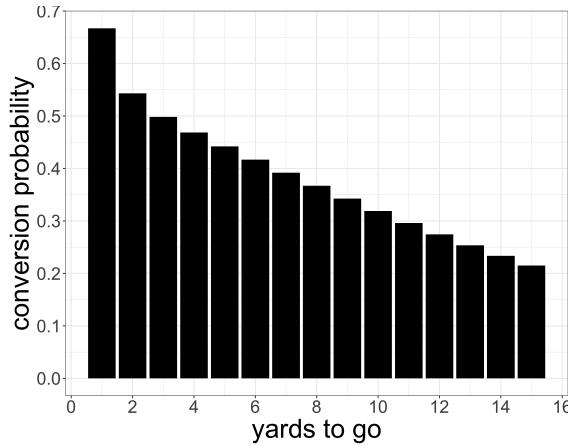


Figure 18: Conversion probability according to our model as a function of yards to go, assuming average team quality.

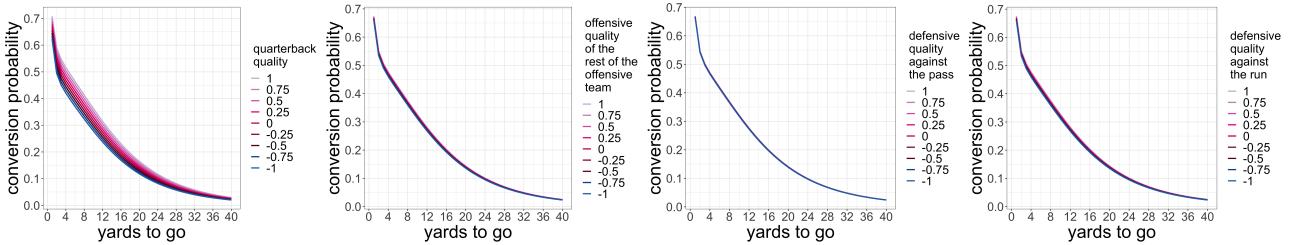


Figure 19: Conversion probability according to our model as a function of yards to go for various values of team quality. Quarterback quality has by far the largest impact on conversion probability.

A more elaborate conversion probability model may adjust for yardline. In particular, it is plausible that it is more difficult to convert near each endzone, where space is more constricted. Additionally, a more fine-grained model would be continuous in yards to go rather than treating it as an

integer. Of course, our model can only be as good as available data, which treats yards to go as an integer. But, anecdotally, 4th down and inches has a significantly higher conversion probability than fourth down and one yard to go. Finally, conversion probability depends on the offensive and defensive play call. On this view, in practice it may be better for teams to input their own conversion probability estimates as they are more informed on their play calling tendencies.

C.5 Baseline coach’s decision model

To compare our decision making procedure to the decisions that actual football coaches tend to make, we model the probability that a coach chooses a decision in {Go, FG, Punt} as a function of game-state. We use XGBoost to fit these coach probabilities. XGBoost works well here because we have 94,786 fourth down plays in our full dataset of plays from 1999 to 2022, and each play is an independent observation of a coach’s decision. In particular, we fit these coach probabilities as a function of yardline, yards to go, game seconds remaining, score differential, point spread, and era (1999-2001, 2002-2005, 2006-2013, 2014-2017, and 2018-present). In Figure 20 we visualize these coach decision models, and the results make intuitive sense. For the most part, coaches punt deep in their own territory and kick field goals near the opponent’s endzone, except for with one and sometimes two yards to go. Also, at the end of the game, coaches’ decision making changes depending on the number of points they need to score to win the game.

In Figure 21 we visualize the variable importance (via gain) of our XGBoost model. Interestingly, point spread has an extremely small impact on coaches’ fourth-down decisions. Perhaps this is because coaches don’t like to admit when their teams are underdogs as some sort of psychological leadership tool. We find, however, that point spread should impact fourth-down decision making. For instance, in certain game-states, it is advantageous for the favorites to be more aggressive (e.g., late in close games).

The parameters of our XGBoost model are

```
{
  booster: gbtree
  objective: multi:softprob
  eval_metric: mlogloss
  num_class: 3.0
  eta: 0.0453573
  gamma: 0.0013761
  subsample: 0.6698295
  colsample_bytree: 0.8924051
```

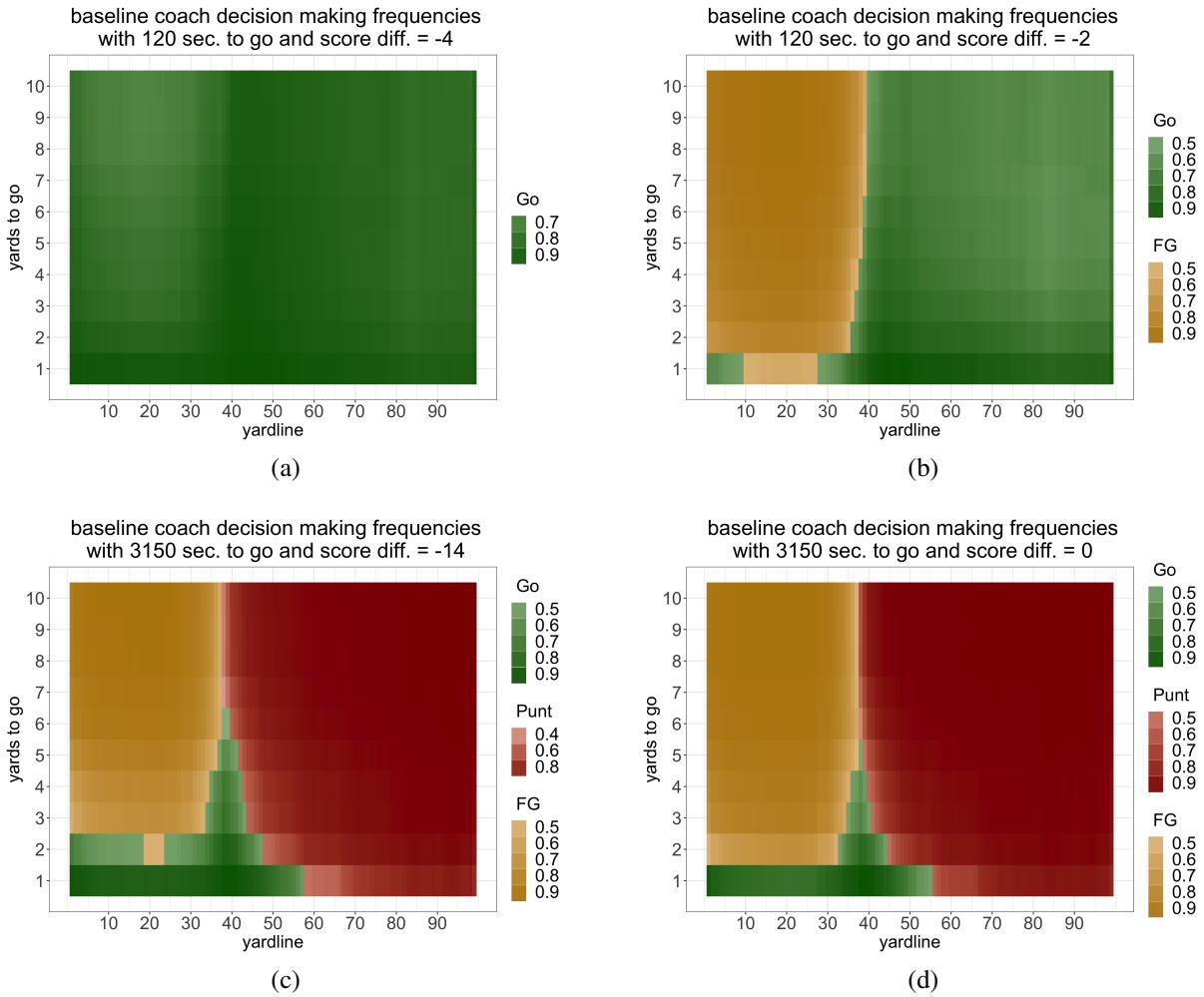


Figure 20: Visualizing our model of the typical coach’s fourth-down decision as a function of yardline and yards to go, for various values of time remaining and score differential. Green, yellow, and red indicate that Go, FG, and Punt is the most likely decision, respectively. The darkness of the color reflects the likelihood that a coach makes that decision.

```

max_depth: 4
min_child_weight: 6
nrounds: 592
}.

```

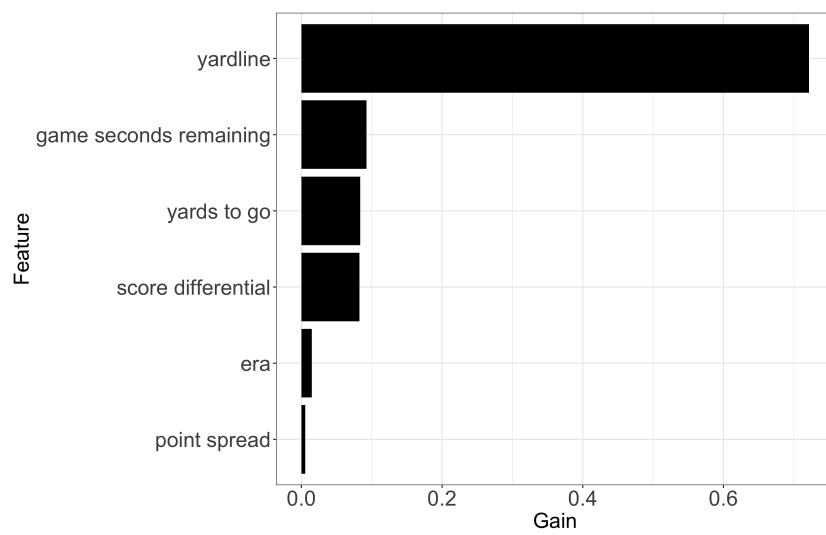


Figure 21: Variable importance (gain) for coach's decision probability model.