

A Game Theoretical Approach to Optimal Pitch Sequencing

William Melville, Jesse Melville, Theodore Dawson, Delma Nieves-Rivera,
Christopher Archibald, David Grimsman

1. Introduction

In baseball, the pitcher's primary goal is to prevent their opponent from scoring runs. They have a variety of tools at their disposal to achieve this goal. They have seven fielders behind them ready to deal with any batted balls in play. They have a catcher who works to prevent any base stealing and who uses framing to make pitches look more like strikes [2]. They also have a repertoire of different pitch types with a variety of velocities and trajectories, and with the help of their catcher they get to decide when and where to throw each of the pitches in their repertoire. This strategic decision-making process is often referred to as pitch sequencing.

Currently, most sequencing strategies are based on simple heuristics. A pitcher may use his fastball to get back into strikeout counts and his slider only late in counts to strike hitters out [11]. They may adopt the "hard in, soft away" strategy where they throw their fastball inside to set up a breaking ball away on the next pitch [14]. Of course, smart hitters are aware of these simple pitching strategies. A smart hitter may take advantage of a pitcher's predictability by crowding the plate and moving up in the batter's box after seeing an inside fastball because that puts him in a better position to hit the outside breaking ball that is likely coming next. Likewise, if a pitcher always throws a fastball down the middle when he is behind in a 3-0 count, a smart hitter can take advantage of that by swinging more frequently because he knows he is going to get a good pitch to hit. Of course, a smart pitcher may respond to the hitter swinging more frequently by throwing breaking balls in the dirt and inducing more swings and misses, which would then incentivize the hitter to swing less frequently. Theoretically, there is some swing rate for the hitter and breaking ball in the dirt rate for the pitcher such that the pitcher can no longer adjust their breaking ball rate to take advantage of the hitter, and the hitter can no longer adjust their swing rate to take advantage of the pitcher. There is no longer any incentive for either player to adjust their strategy. This combination of strategies is known as an *equilibrium* in game theory.

In this paper, we present a novel approach to optimizing pitch sequencing by modeling the matchup between a batter and a pitcher as a zero-sum game and finding the equilibrium, game-theoretically optimal strategies for each player in this game. We explore three different specific game models, which differ in the information available to the batter before making their decision to swing or not. The first game assumes the batter observes nothing about the pitch before making their decision. The second assumes the batter observes the full pitch before making their decision. The final game we investigate assumes that the batter can observe everything up to a decision point, and then must make their decision. We solve for the equilibria in all three games, comparing the resulting pitch recommendations and the insight they give for optimal pitch sequencing.

The remainder of the paper proceeds as follows: First, in section 2 we discuss previous work that is relevant to our presentation. Then, in section 3 we give an overview of our proposed game-

theoretic approach and use a specific at-bat as an example to demonstrate the different game models, their assumptions, and their recommendations. The later sections provide the specific details and components that are necessary to specify the different game models. In section 4 we present our novel recurrent neural network architecture that estimates the distribution over pitch outcomes for a pitch given the batter, the previous pitches in the plate appearance, the previous pitches' outcomes, and the current pitch's characteristics. We use this outcome distribution in the calculation of expected run value, which forms the basis of the utility of the game models. In section 5 we present a novel approach to modeling and estimating pitcher command, or accuracy, which allows the game models to take into consideration a pitcher's ability to hit their target when recommending the next pitch's location. Given all of the required components, in sections 6, 7, and 8 we introduce the details of the three game models, discuss how an equilibrium for each is computed, and provide additional examples of recommendations based on these equilibria. We conclude with a summary and discussion in section 9.

2. Related Work

Much research has already been done on the challenge of pitch sequencing. Roegle concluded that throwing back to back pitches of different pitch types on similar trajectories up to a decision point can help pitchers induce swings and misses [25]. Bonney found evidence that throwing a lower velocity pitch after a fastball can benefit the pitcher [12]. Carleton modeled the effect that a change in velocity in a two pitch sequence has on swing and miss rate for a set of 302 batters, and not surprisingly he found that different batters respond differently to changes in velocity [13], which suggests that it is important to make batter specific pitch sequencing recommendations. None of these analyses consider sequences longer than two pitches, whereas our method considers the full sequence regardless of length. Our method also considers all relevant characteristics of the pitch, not just velocity or trajectory.

Arthur, Bock, and Healey and Zhao each separately developed a measure of the predictability of a pitch sequence, and each found that higher predictability was related to worse pitching performance [9][10][17]. None of these methods are able to make specific recommendations about which pitch to throw next in a sequence, other than suggesting not to be too predictable. Additionally, Arthur and Healey and Zhao only consider two pitch sequences, whereas our methods consider the full sequence, and Arthur and Bock only consider the pitch type, whereas our methods consider all other relevant information, particularly the pitch locations.

There is existing research that analyzes the full sequence of pitches. Prasad models full pitch sequences using directed graph embeddings. They then perform exploratory analysis on these embeddings [24]. Yoshihara and Takahashi use a probabilistic topic model to identify pitch sequence trends in Nippon Professional Baseball [32]. Although both of these analyses provide interesting and valuable insights into existing sequencing strategies, neither are able to make sequencing recommendations for a specific batter and pitcher and game state.

A large amount of work has already been done to assign a value to a pitch. Healey created a Bayes classifier to define what he called the intrinsic value of a pitch [16]. Moore uses a set of random forest regression models to estimate the expected run value of each pitch [20]. Max Bay created a model to predict pitch outcomes, which he then uses to assign pitches a value called stuff+ [26]. Unlike our recurrent neural network, none of these models consider the effect that the batter may have on the value of a pitch. Some pitches may be better to some batters than they are to other

batters. A fastball down the middle is a much worse pitch to Aaron Judge than it is to Billy Hamilton. Additionally, all of these models consider each pitch by themselves, without taking into consideration the effect that the prior pitches in the sequence may have on the value of the next pitch. Our model does consider that effect.

Although our pitch outcome model is the first model we know of to use a recurrent neural network to predict the outcome of the next pitch, it is not the first model to apply a recurrent neural network to pitch sequences. [31][33][21] each created a RNN that predicts the next pitch type, rather than pitch outcome, in a pitch sequence.

A key element of our pitch outcome model is the batter embedding. This is what makes our pitch values batter specific, and the idea for this came from Alcorn’s work on the (batter|pitcher)2vec model [6].

Douglas, Witt, Bendy, and Vorobeychik attempted to solve the pitch sequencing problem using a very similar method that we present here. They also model the batter and pitcher matchup as a zero sum game and apply game theoretical methods to solve for an equilibrium[15]. However, in their game they assume that the batter and pitcher make their decisions simultaneously, which means the batter decides whether he will swing or take the pitch before he has seen the pitch. That is not a reasonable assumption, and we will explore why in section 6. They also use a different structure for their pitch outcome model, a different method for evaluating pitcher command, and a different utility function. Their utility function is on-base percentage, which means a home run is as valuable in their game as a single. Our utility function is based on the expected run value of pitch outcomes, so it more appropriately weights home runs relative to singles.

3. Description of the Batter/Pitcher Matchup Game

The batter/pitcher interaction is highly complex when one considers the strategic, physical, and mental factors that contribute to each player’s decision. In this section, we present a model for this interaction to be used throughout the paper. To begin assume that each pitch is partitioned into some properties that are observable and some that are unobservable to the batter. Denote the set of possible observable properties as P_o and the set of possible unobservable properties as P_u . Since the batter is not able to observe the parts of the pitch in P_u , it would be foolish for the pitcher to deterministically choose an element from P_u , i.e., if the batter knows everything about the incoming pitch, it is to the batter’s advantage. Therefore, in our model of the game, we allow the pitcher to choose a probability distribution across P_u . This allows the pitcher to leverage the fact that some pitch properties are not revealed to the batter before the batter must choose whether to swing. The pitcher’s decision is thus to select an element from the *pitching strategy set* $P = P_o \times \Delta P_u$, where ΔP_u is the set of all probability distributions across P_u .

In like manner, it is generally not advantageous for the batter to determine a priori whether to swing (for instance, if a batter committed to not swinging before seeing the pitch, the pitcher could react by throwing a strike right down the middle). Therefore, the batter’s decision is to choose a strategy from ΔS , where $S = \{\text{swing}, \text{take}\}$.

Denote C to be the set of all game contexts, where a game context is characterized by a batter, pitcher, ball and strike count, and the previous pitches in the current plate appearance. Both the pitcher and batter have access to a utility function $U : P \times \Delta S \times C \rightarrow \mathbb{R}$, which measures expected run

value given the parameters. Negative values of U favor the pitcher, and positive values favor the batter. We note that much of this paper is dedicated to ensuring that the computed U is as realistic as possible to actual in-game utilities and incorporates pitcher and batter accuracy (see Section 4 and Section 5). One complication that arises in real-world settings is that pitchers have execution error: while a pitcher may intend to use pitching strategy $\pi \in P$, the actual pitching strategy may turn out to be $\tilde{\pi} = (\tilde{\pi}_o, \tilde{p}_u)$. Given the observation $\tilde{\pi}_o$, the batter may attempt to infer p_u , therefore, it may be to the pitcher's advantage to obscure his intention by randomly selecting the pitching strategy from some distribution. Therefore we allow the pitcher to ultimately choose $p \in \Delta P$.

Since each player knows each other's possible set of decisions and has access to the same context $c \in C$ a natural question arises as to how each player ought to make their decision: if both players play optimally, what is the resulting interaction? Here we leverage the game-theoretic notion of an *equilibrium*, a set of decisions wherein neither player has a unilateral incentive to deviate. Formally, given a game context c and observed portion of the pitch $\pi_o \in P_o$ we

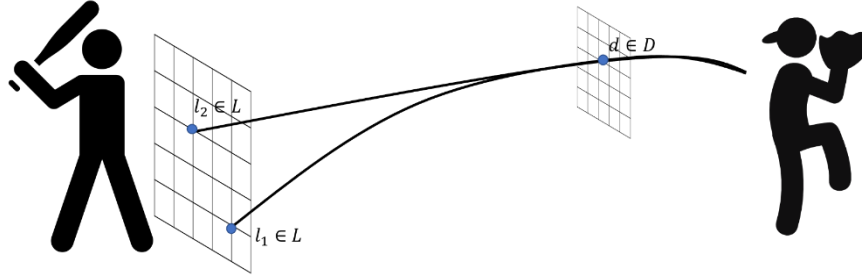


Figure 1: A comparison of the sets D and L . The set L is the set of location points in the strike zone. The set D , on the other hand, is a set of decision points, which represent where the ball is when the batter needs to make a decision whether to swing. Note that a single point in D could correspond to several points in L , depending on the pitch type. One way to model the interaction is to say P_o is every property about the pitch that can be observed before d and P_u is everything that happens after d .

define (p^*, s^*) to be an equilibrium if

$$p^* \in \operatorname{argmin}_{p \in \Delta P} \max_{s \in \Delta S} \mathbb{E}_{\pi} [U(\pi, s, c)] \quad (1a)$$

$$s^* \in \operatorname{argmax}_{s \in \Delta S} \min_{p_u \in \Delta P_u} U(\pi_o^*, p_u, s, c), \quad (1b)$$

where we abuse notation of U , since $(\pi_o, p_u) \in P$. Note the asymmetry between (1a) and (1b): the pitcher makes a pitching decision based on the expected value of U , whereas the batter is able to choose s^* based on the actual observed portion of the pitch. Given that the pitcher chooses p^* , the batter is incentivized to choose s^* , and vice versa. For the settings in this paper, such an equilibrium always exists, and while there may be multiple equilibria, each is equally valuable to both pitcher and batter.

As one varies what is represented by P_o and P_u , the equilibrium in (1) has different interpretations. In this paper we explore three possibilities:

- **Game 1: Simultaneous Game.** Here assume that $P_o = \emptyset$ and $P_u = T \times L$, where T is the set of *pitch types* (fastball, curveball, etc.) that the pitcher can throw and L is the set of possible locations across the plate. In this simplification of the game, the batter observes nothing about the pitch, and must choose a batting strategy purely on the idea that the pitcher wants to minimize utility. Then the equilibrium as defined in (1) corresponds to the well-known Nash equilibrium for zero-sum games.
- **Game 2: Sequential Game.** Here assume that $P_o = T \times L$ and $P_u = \emptyset$. In this simplification of the game, the batter observes the entirety of the pitch and queries the model of himself to determine whether to swing. Note that in this case, since there is no unobserved aspect of the pitch, there is no need for the batter to choose a probability distribution; the batter can choose deterministically whether to swing based on which choice gives the higher utility. Then the equilibrium in (1) corresponds to the well-studied Stackelberg equilibrium.
- **Game 3: Decision Point Game.** Here assume that $P_o = D$ and $P_u = T$, where D is the set of possible decision points, i.e., ball locations where the batter can observe the properties of the pitch up to that point, but beyond which the batter does not have time to react to new information (see Figure 1) ¹. For instance, a pitcher can disguise a curveball as a fastball by assuring that they look the same to the batter before the decision point $d \in D$. After that, when the batter does not have time to react, the true pitch type is revealed. We call this type of equilibrium a *decision-point equilibrium* and it is the focus of much of this work.

Games 1 and 2 above represent extremes in information available to the batter as they make their decision. We assert that Game 3 gives us the most realistic of the three, since it models the idea that the batter has some but not all of the information.

We note that this formulation assumes the batter can calculate the solution to (1b) and then sample from his chosen probability distribution almost instantaneously. Clearly this is not a realistic assumption, however we argue that this modeling choice is relevant and has the following benefits:

1. For a pitcher, implementing p^* implies that there is a best-case bound on how well the batter can respond even if the batter could perform this instantaneous calculation.
2. For a pitcher/batter, one can look historically and see how close to optimal each is performing.
3. Solving (1) can in many cases yield a deterministic solution for the pitcher, which is a simple strategy to implement and test.

3.1. Calculating the Utility Function

In this section we present an overview of the utility function U , which will give context to Sections 4 and 5. Recall that negative values of U favor the pitcher whereas positive values favor the batter - thus in (1), the pitcher seeks to minimize U and the batter seeks to maximize. Recall also that the pitcher decision is $\pi = (\pi_o, p_u)$, where p_u is a probability distribution on P_u , the batter chooses s ,

¹ Pitcher icon found at <https://flaticon.com>

which is a probability distribution on S . Therefore, U is an expected value across these utility functions:

$$U(\pi_o, p_u, s, c) = \mathbb{E}_{\pi_u, \tau}[R(\pi_o, \pi_u, \tau, c)] = \sum_{\pi_u \in P_u} \sum_{\tau \in S} p_u(\pi_u) s(\tau) R(\pi_o, \pi_u, \tau, c), \quad (2)$$

where $p_u(\pi_u)$ is the probability the pitcher chooses $\pi_u \in P_u$ under strategy p_u , $s(\tau)$ is the probability the batter chooses $\tau \in S$ under strategy s , and $R : P_o \times P_u \times S \times C \rightarrow \mathbb{R}$ is a function that gives value for a particular pitch choice (π_o, π_u) , swing choice τ , and game context c .

In order to learn the function R , we map the outcome of each pitch to a set of 9 categories $X = \{B, CS, SS, F, 1B, 2B, 3B, HR, O\}$, which represent respectively a called ball, a called strike, a swinging strike, a foul ball, a single, a double, a triple, a home run, and an out. The value $v(x, c)$ of each outcome x given a game context c is adapted from [28] shown in Figure 2, where the relevant game context c is simply the ball and strike count. The function R must also account for the pitcher's and batter's inaccuracies. Therefore, we formulate R as the following expected value:

$$R(\pi_o, \pi_u, \tau, c) = \sum_{\tilde{\pi}_o \in P_o, \tilde{\pi}_u \in P_u} P(\tilde{\pi}_o, \tilde{\pi}_u | \pi_o, \pi_u) \sum_{x \in X} P(x | \tilde{\pi}_o, \tilde{\pi}_u, \tau, c) v(x, c) \quad (3)$$

Count	B	CS	SS	F	1B	2B	3B	HR	O
3-0	0.131	-0.07	-0.07	-0.07	0.287	0.583	0.861	1.2	-0.496
3-1	0.201	-0.076	-0.076	-0.076	0.356	0.652	0.93	1.269	-0.426
2-0	0.11	-0.062	-0.062	-0.062	0.397	0.693	0.971	1.31	-0.385
3-2	0.276	-0.351	-0.351	0	0.432	0.728	1.006	1.345	-0.35
2-1	0.103	-0.071	-0.071	-0.071	0.459	0.755	1.033	1.372	-0.323
1-0	0.063	-0.05	-0.05	-0.05	0.46	0.756	1.034	1.373	-0.323
0-0	0.034	-0.043	-0.043	-0.043	0.494	0.79	1.068	1.407	-0.289
1-1	0.05	-0.067	-0.067	-0.067	0.51	0.805	1.083	1.423	-0.273
2-2	0.098	-0.252	-0.252	0	0.53	0.826	1.104	1.443	-0.252
0-1	0.027	-0.062	-0.062	-0.062	0.537	0.832	1.11	1.45	-0.246
1-2	0.046	-0.206	-0.206	0	0.577	0.872	1.15	1.49	-0.206
0-2	0.022	-0.184	-0.184	0	0.598	0.894	1.172	1.511	-0.184

Figure 2: The values for each outcome

Finding the conditional probabilities in (3) is nontrivial. One of the contributions of this work is a novel batter and sequence-specific prediction model that estimates $P(x | \tilde{\pi}_o, \tilde{\pi}_u, \tau, c)$ given in Section 4. Another contribution is an accuracy estimation $P(\tilde{\pi}_o, \tilde{\pi}_u | \pi_o, \pi_u)$ specific to each pitcher, given in Section 5.

Note, the last summation in equation (3) can be interpreted as the expected run value of a pitch

$$r_v(\pi_o, \pi_u, \tau, c) = \sum_{x \in X} P(x | \pi_o, \pi_u, \tau, c) v(x, c) \quad (4)$$

where the run value of a pitch is the average change in run potential plus the average runs that score as a result of a pitch outcome [29].

3.2. Example

On October 24, 2020, the Tampa Bay Rays faced off against the Los Angeles Dodgers in game four of the World Series. The Rays were already down 2-1 in the best of seven series, and they were down 7-6 in the game when Brett Phillips stepped up to the plate to face Kenley Jansen with two outs and runners on first and second in the bottom of the ninth. Brett Phillips, who is valued much more for his defense than his ability as a hitter, became the unlikely hero of the night when he concluded a four pitch plate appearance with a single into right field that, with the help of some fielding mistakes, scored both baserunners and gave the Rays a walk-off win, their last win of 2020. In this section, we model the last pitch of that at bat using all three versions of our batter/pitcher matchup game to see what Kenley Jansen could have done differently to get Phillips out.

Kenley Jansen throws three different pitch types: a cutter (FC), a sinker (SI), and a slider (SL), thus $T = \{FC, SI, SL\}$. In this matchup, and in all other matchups for the remainder of this paper, we define L as the grid of 1.5 inch sized squares ranging from -1.5 to 1.5 feet on the horizontal axis and from 1 to 4 feet on the vertical axis where the origin is given by the back tip of home plate. We define our locations from the perspective of the pitcher, so a positive number on the horizontal axis would be inside to a right-handed hitter and a negative number on the horizontal axis would be outside to a right-handed hitter.

3.2.1. Simultaneous Game Analysis

We first analyze game type 1, where $P_o = \emptyset$ and $P_u = T \times L$. In other words, assume that Jansen and Phillips chose their actions simultaneously. The optimal set of strategies (p^*, s^*) is the solution to (1), which can be found using the simplifications described in Section 6. The optimal mixed strategy for Jansen was to throw his cutter with intended location (0.19, 2.69) (measured in feet) with probability 0.63 and to throw his slider with intended location (-1.46, 1.04) with probability 0.37. This strategy, as well as the cutter that Jansen actually ended up throwing, is depicted in Figure 3a.

Phillips' optimal strategy was to swing with probability 0.54. Assuming that Phillips and Jansen both follow their equilibrium strategies, the expected utility of the pitch is -0.07. The actual expected run value of the pitch that Jansen threw, calculated using equation (4), was 0.037. These results suggest that Jansen would have benefitted greatly by randomly sampling a pitch from his optimal mixed strategy distribution. Of course, it is possible that he did do that, and he intended to throw his cutter in the intended location depicted in figure 3a, he just missed his target badly.

It looks like part of Jansen's strategy is to take advantage of Phillips' fixed swing rate by occasionally choosing to throw a slider as far outside of the strike zone as he possibly can. Phillips is assumed to be more likely than not to swing at that pitch, which would be bad for him as he would most likely miss. In reality, Phillips probably would not swing 54% of the time at a pitch that bad.

We can estimate the likelihood that Phillips swings at that pitch by taking the sum of $P(x/p,c)$ for all outcomes x (details in next section) from our pitch outcome model. For that particular slider, Phillips' actual swing likelihood is estimated to be less than 13%, which would incentivize Jansen to throw it much less often. If Jansen is incentivized to change his strategy, then we haven't found a true equilibrium point.

3.2.2. Sequential Game Analysis

That leads to our next version of the batter/pitcher matchup game where we assume that the batter can observe the full pitch before making their decision. We have $P_o = T \times L$ and $P_u = \emptyset$. We assume that Phillips deterministically chooses the $\tau \in S$ that maximizes equation (3) for every choice of $\pi_o \in P_o$. We can therefore identify Jansen's optimal pitch by choosing the π_o that minimizes (3) given that Phillips' will choose the optimal τ . This optimal pitch takes into consideration Jansen's ability to hit his intended target, but we also found the best possible pitch that Jansen could throw under the assumption that he would hit his intended target with certainty. Figure 3b shows the optimal pitch, the best possible pitch, and the actual pitch that Jansen threw to Phillips.

All three pitches were cutters. For both the optimal pitch and the best possible pitch, the best response from Phillips is to swing. The expected utility of the optimal pitch is 0.007. If Jansen managed to throw the best possible pitch, the expected run value (4) would be -0.067. Both of these are better than the expected run value of the pitch he actually threw, which was 0.037.

Note, the optimal pitch here assumes that Phillips always makes the maximizing swing decision. We will show in section 7 that this is not always the case. That means that the expected utility of the optimal pitch is really a worst case scenario for Jansen. In reality, the utility of that pitch would be even smaller because Phillips will not always make the optimal swing decision. That also means that there may be a better pitch for Jansen if we adjust for Phillips' true ability to make optimal decisions. With that being said, the optimal pitch had a better expected utility than the pitch that Jansen actually threw, and that expected utility represents a worst case scenario for Jansen, so we believe Jansen would've benefitted from using the Stackelberg equilibrium strategy.

3.2.3. Decision Point Game Analysis

In our final game, we assume that Phillips will observe the pitch's decision point but not the pitch type, meaning he will have some uncertainty about the full trajectory of the pitch after the decision point. Using the methods described in section 8, we solve for a mixed strategy over possible pitches for Jansen and a mixed swinging strategy for each possible decision point for Phillips.

Figures 3c and 3d show Jansen's mixed strategy solution. One notable thing about Jansen's strategy is that the pitch he should choose to throw with highest probability is a slider well below the strike zone. One might argue that this is a wasted pitch because a professional hitter is not likely to swing at a pitch half a foot below the strike zone. Recall that the solution to this game gives Phillips an optimal strategy for any observed decision point. It turns out that the optimal swing strategy for Phillips in the decision point for this slider is a pure swinging strategy, meaning Phillips should swing every time he observes a pitch that passes through that decision point. This strange result appears to be caused by the threat of sinkers and cutters being thrown through the same decision point. If Phillips swings at the Jansen slider in this decision point, the expected run value is -0.10. If

he takes that slider, the expected run value is 0.05. If he swings at a sinker or a cutter in this decision point, the expected run value is 0.03 either way, and if he takes a sinker or a cutter in this decision point, the expected run value is -0.2 and -0.18 respectively. Thus, the cost of taking the sinker is much higher than the cost of swinging at the slider. Additionally, we can see that this slider and the two sinkers in the bottom half of the strike zone in Jansen's mixed strategy all pass through similar decision points in figure 3c. When we consider Jansen's accuracy skill, it is fair to say that all three of these pitches are going to pass through a similar set of decision points. Because Jansen is mixing the slider and the sinker, the threat of Phillips observing a sinker in this decision point is very real, and since it is much more costly for Phillips to take a sinker in this decision point than it is

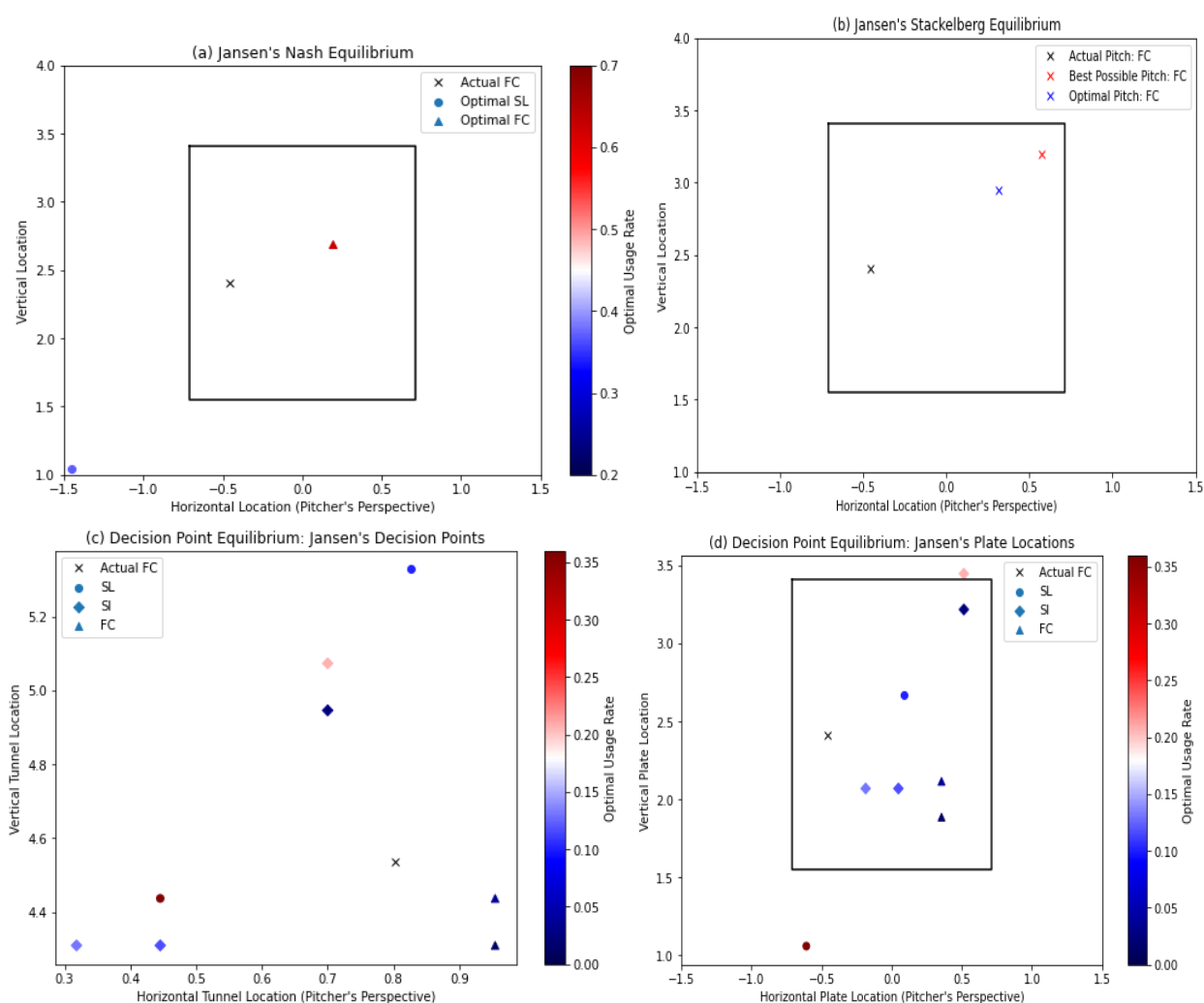


Figure 3: Jansen's equilibria against Phillips

to swing at a slider, we can see why Phillips might be incentivized to swing every time he observes a pitch in this decision point.

For any remaining skeptics, we tested to see if Phillips really should swing at every pitch in that decision point by varying Phillips' swing rate strategy in that decision point and seeing how that affects the resulting expected utility in (1). Figure 4 shows the result of that test. Clearly, Phillips' utility increases as his swing rate increases in the slider's decision point location.

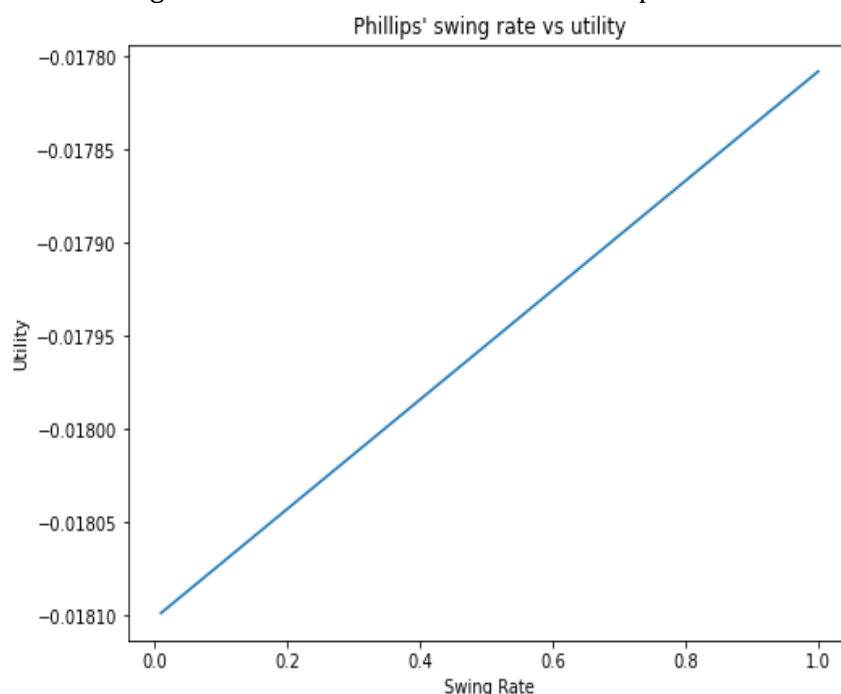


Figure 4: Result of varying swing rate in the Jansen slider tunnel

Imagine if Jansen had thrown his slider in that location, and instead of hitting a game winning single, Phillips swung and missed, ending the game with a strikeout. It is likely that many disappointed Rays fans would have angrily cursed at Phillips, wondering how a professional hitter could swing at a pitch so far below the strike zone. But if we assume that Phillips has to make his swing decision based only on the decision point that he observes, which seems like a fairly reasonable assumption, then the reality is that Rays fans have nothing to complain about. Phillips made the best decision he could have made with the information available to him.

We'll discuss some of the assumptions of the third game in more detail in section 8. It has some of the same issues that the second game has, particularly that batters are not actually able to behave optimally under these assumptions. However, we think it is reasonable to conclude that Jansen would have been much better off throwing that slider down below the zone to Phillips. If Phillips decided to swing at that pitch, as his equilibrium strategy would recommend, then the result would have been a pitch with a negative expected run value. If Phillips had not swung at the pitch, the result would have likely been a ball, which may not be a good outcome for Jansen, but at least it isn't a walk-off single.

4. Pitch Outcome Model

A truly optimal solution to the pitch sequencing question has to consider the effect of the batter on pitch outcomes. Different batters have different strengths and weaknesses. An optimal pitch sequence for right-handed hitting Aaron Judge may not be optimal against left-handed hitting Bryce Harper. Additionally, an optimal solution has to consider the effect that the prior pitches in the plate appearance may have on the outcome of the next pitch in the sequence. We cannot evaluate or improve upon the "hard in, soft away" sequencing strategy, or any other existing sequencing strategy, if our utility function does not consider the effect of prior pitches. Thus, we developed a novel recurrent neural network (RNN) that incorporates information about the batter and all prior pitches in the plate appearance when predicting the outcome of the next pitch. We use this model to estimate $P(x \mid \tilde{\pi}_o, \tilde{\pi}_u, \tau, c)$ in our utility function (2). We can think of this RNN as an artificial intelligence that aids in optimal pitch selection, so we henceforth will refer to it as OptimusPitch (OP).

OP's architecture is shown in Figure 5. It is made up of two main components: the outcome estimator and the hidden state calculator. Let t be the pitch number in the sequence. x_t is a vector containing the relevant information about pitch t , specifically the release speed, spin rate, movement, plate location, the ball and strike count, and the batter and pitcher handedness. Note, throughout this paper when we say a pitcher chooses a pitch type in T , what we really mean is that they are choosing the release speed, spin rate, and movement. B is an index representing the batter in the plate appearance. $y_t \in X$ is the outcome of pitch t . Finally, h_t is the hidden state for pitch t . Note that h_1 is initialized to be a vector of all zeros.

In the outcome estimator, we first pass the batter index through an embedding layer as in [6]. Then, x_t , the batter embedding, and h_t are concatenated together and passed through a feed forward network. This feed forward network is made up of five linear layers with ReLU activations between them. The final output is then passed through a softmax activation resulting in a probability distribution over X .

In the hidden state calculator, we take the actual result of pitch t , given by y_t , and pass that through an embedding layer. We then concatenate that embedding onto x_t and pass the resulting vector through another feed forward network. This network is also made up of five linear layers with ReLU activations between them. We finally add the output of this feed forward network onto h_t to get h_{t+1} . This final addition ensures that our hidden state at time step $t+1$ is a function of all of the previous pitches in the sequence and the outcomes of those pitches. The resulting h_{t+1} would then be passed into the output estimator for the next pitch in the sequence.

For each pitch in a plate appearance, OP estimates the output using information about the batter through a batter embedding and information about all previous pitches in the sequence through the hidden state, h_t . Thus, this model satisfies our requirement that our pitch outcome predictions are batter specific and consider the effect of previous pitches on the outcome of the next pitch in the sequence.

4.1. Training and Evaluating OptimusPitch

MLB pitch level data from 2021 to 2022 were acquired from Statcast using the pybaseball package [5][1][19]. We randomly chose 75% of the plate appearances in this data to make up our training set, and the remaining 25% were used as a validation set.

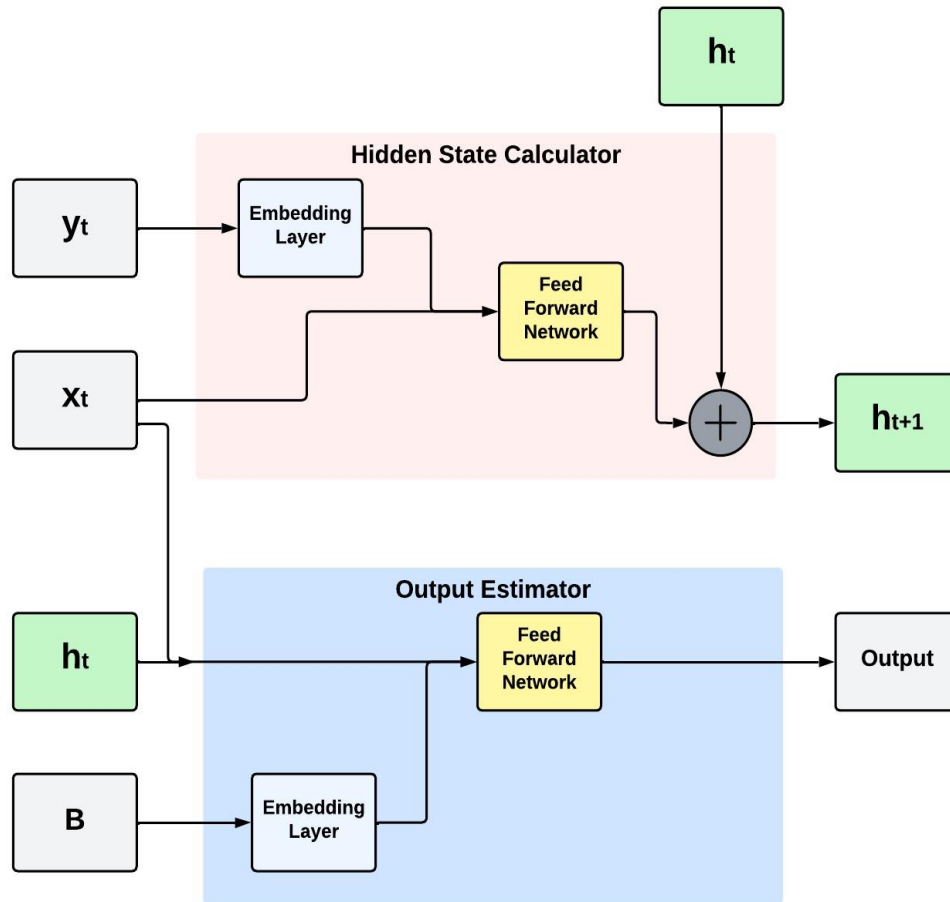


Figure 5: OptimusPitch

Since OP is used in our utility function to estimate the probability of each possible pitch outcome, we needed to verify that OP's probability predictions were well-calibrated. We ran OP on the validation set and then created probability calibration plots for each of the nine possible pitch outcomes [4]. These plots are shown in figure 6. Clearly, OP's outputs are well-calibrated and can reasonably be used in our utility function.

The main innovation in OP is not that it can reliably estimate pitch outcome probabilities or even that it makes batter specific predictions; it is that it considers the effect of prior pitches on the outcome of the next pitch in the sequence. Thus, we needed to verify that including prior pitch information actually helped OP make better predictions on later pitches in the sequence. We compared OP to a memoryless version of OP. The memoryless version was trained on the same training set as OP, but for each pitch in the sequence, h_t was reset to the initial vector of zeros, h_1 . This memoryless version has the same structure as OP, but it does not consider any information about the prior pitches in the sequence. We then ran OP and the memoryless version on the validation set and compared the cross entropy loss of the predictions [3].

Overall, the two models actually performed similarly well on the validation set. OP's average cross entropy loss was 1.11, whereas the memoryless version's average cross entropy loss was 1.12.

However, when we compared average cross entropy loss by pitch number, it became apparent that OP does a better job than the memoryless version at making predictions on later pitches in the sequence. Figure 7 shows the difference between OP's cross entropy loss and the memoryless model's cross entropy loss grouped by pitch number. As we get deeper into the pitch sequence, OP's

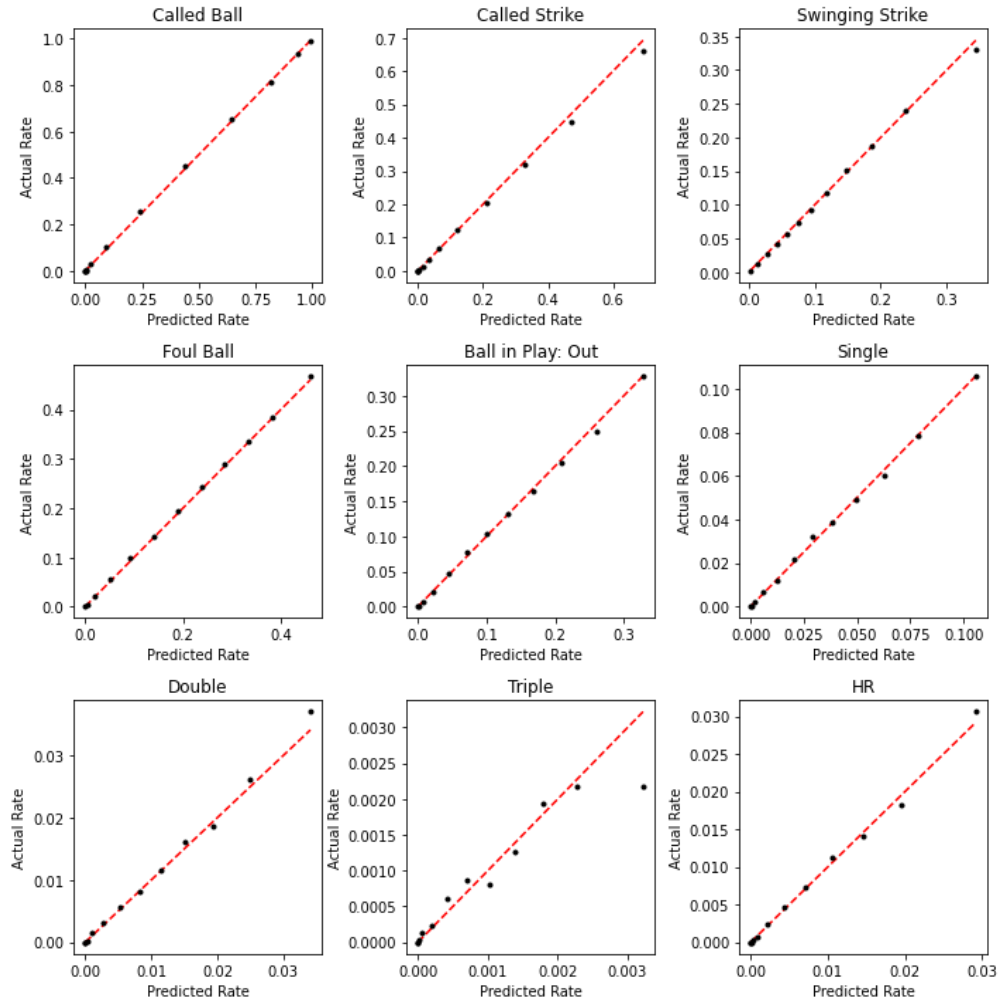


Figure 6: Probability calibration of OptimusPitch on the validation set

loss decreases relative to the memoryless model's loss. Thus, even though the models perform similarly well on the validation set, it is clear that OP is better at making predictions on pitches that are deeper into a pitch sequence.

We conclude that not only do prior pitches in the sequence have an effect on the outcome of the next pitch in the sequence, but also that OP is the first model we know of that is able to learn something about this effect and apply it to its outcome predictions.

4.2. OptimusPitch Example

On July 6, 2022, Edwin Diaz of the New York Mets faced off against Mike Moustakas of the Cincinnati Reds. Diaz struck Moustakas out in five pitches. Notably, Diaz effectively utilized the “hard in, soft away” strategy on the final two pitches of the at bat, throwing a 100 mph fastball inside for a ball and then following that with a 92 mph slider outside for a swinging strike three. We ran both OP and the memoryless model on this at bat and compared the predictions for the final pitch. Using OP’s estimated outcome distribution in (4) returned an expected run value on the final

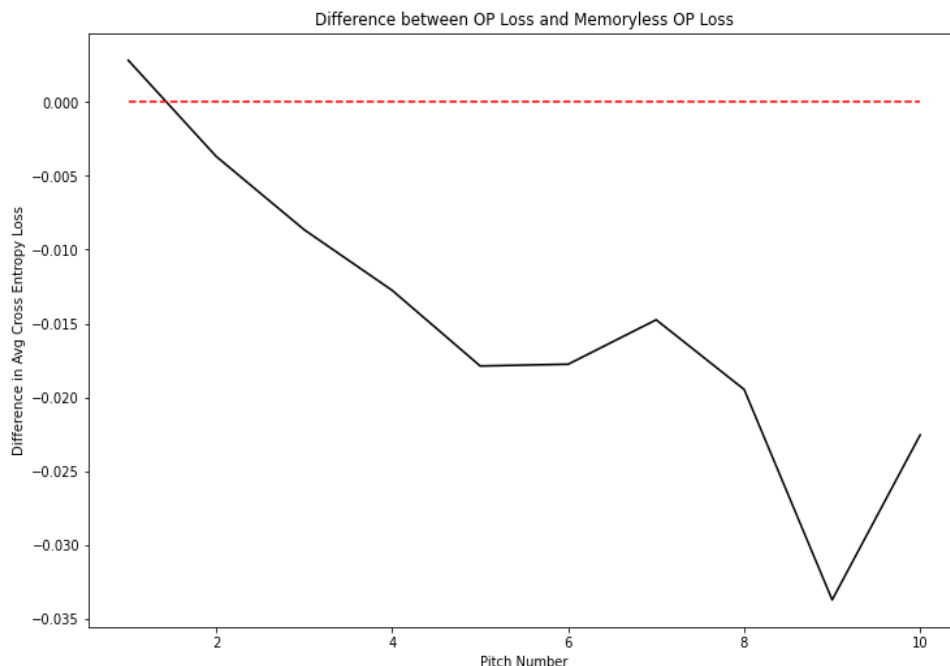


Figure 7: Difference in average cross entropy loss by pitch number

pitch of -0.14 . On the other hand, the estimated outcome distribution from the memoryless model gave an expected run value of -0.07 . Something about the prior sequence of pitches led OP to believe that the final strikeout slider was a better pitch than the memoryless model thought it was. Perhaps OP has learned that an outside slider is better when it is preceded by an inside fastball, as the “hard in, soft away” strategy would suggest.

5. Pitcher Accuracy Estimation

In each of the three games we are considering, the pitcher can choose a target location for a given pitch, but when the pitcher throws a pitch, there will be some amount of error, generally resulting in a different actual pitch location than was intended. This error, or noise, is random, and its characteristics will be different for each pitcher. In this section we describe how parameters of this error were estimated for each pitcher and pitch type for use in the game models.

First, we note that if we had data that indicated the aiming or target location for each pitch, it would be fairly simple to estimate the characteristics of the noise for each pitcher. The most significant noise characteristic is the standard deviation of the noise, which indicates how close the actual pitch locations are to their targets. However, the data that we have access to only records the actual pitch locations, not the target locations.

Prior research has focused on solving precisely this problem, estimating what was termed the *execution skill* of a player in a game [7, 8]. These previous methods all take as input a set of observations of the player acting in a domain. Each observation contains the specific action that was executed by a player along with numerical utility values for all possible actions that could have been executed by the player. For the current baseball setting, each observation consists of all of the

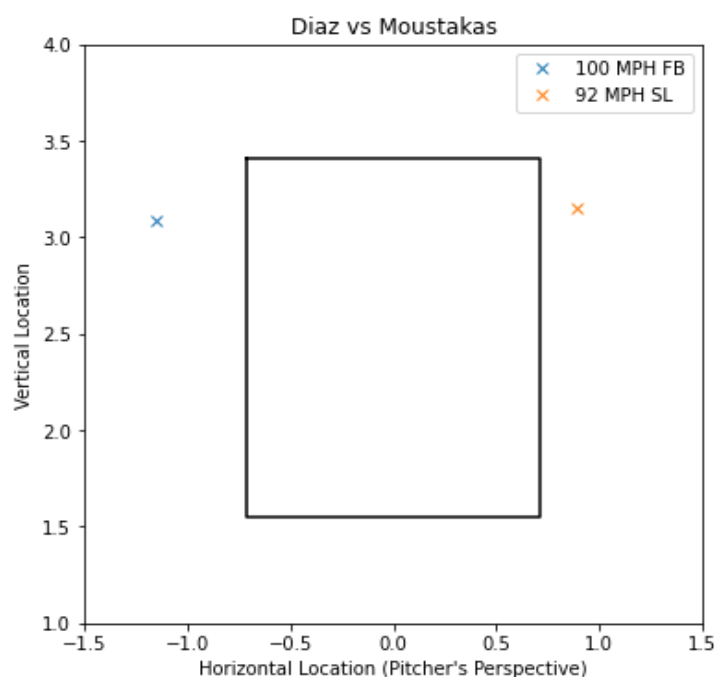


Figure 8: The final two pitches in Diaz's strikeout to Moustakas

details about a single pitch of a given type (cutter, sinker, slider, etc.), by a pitcher. The set of possible actions the pitcher could have executed consists of all the locations where the pitch could cross the plate, while the executed action is the actual realized location of the pitch as it crossed the plate. The numerical utility values were obtained for a grid of possible pitch locations by keeping all other data about each pitch constant while varying the location of the pitch. The r_v function in equation (4) was then used to give the expected utility for each of these alternatively-located pitches against the batter in the observation. With the two required components defined, any of the methods described in previous work can be utilized. We use the most robust and successful method previously investigated, which is called The Bayesian Approach (TBA) [8].

TBA models a player's error as noise added to the intended action, where this noise is drawn from a symmetric two-dimensional Gaussian distribution $N(\mu, \Sigma)$, where $\mu = [0, 0]$ and $\Sigma = \sigma I_2$. TBA maintains a probability distribution over a set of *hypothesis* σ values that the pitcher could have. The available observations for each pitcher and pitch type are processed sequentially and TBA repeatedly updates the probability of each hypothesis σ value as follows:

$$P(\sigma|z) \propto \left[\beta \left(\frac{1}{m} \sum_{f \in F} P(z|f, \sigma) \right) + \frac{(1 - \beta)}{W} \right] P(\sigma)$$

In this equation z indicates the executed action for the observation, m is the number of focal point targets, $f \in F$, W is the total number of possible targets considered, β is a parameter that can be tuned, and $P(\sigma)$ is the previous probability of the hypothesis σ value. The estimate of the pitcher's actual σ value is computed as the mean hypothesis σ value with respect to this probability distribution. This was called the TBA-EES estimate in [8].

For the baseball experiments reported in this work, we utilized a set of 66 hypothesis σ values distributed between 0.17 ft (2 inches) and 2.81 ft. 2.81 ft was chosen as an upper limit as with this σ value the strike zone can only be hit 20% of the time when aiming in the middle. This seemed to be far lower than the success rate that would be required for anyone to be a major league pitcher. 60 of the hypothesis σ values were distributed between 0.17 ft and 1.0 ft, and the remaining 6 were distributed between 1.0 ft and 2.81 ft. This was due to the fact that for σ values lower than 1 ft, the best pitch location can vary, while for values greater than 1 ft, the best location doesn't change, remaining near the middle of the strike zone.

The total set of possible pitch locations we utilized for the TBA method stretches horizontally from -2.13 ft to 2.13 ft and vertically from -2.5 ft to 6.6 ft, discretized in both directions with a resolution of 0.5 inches. These ranges were chosen to enclose at least 95% of all the observed pitch locations in the data from the 2021 MLB season. Pitches landing outside this region were assigned the minimum utility from among all the locations within the region.

TBA utilizes a set of *focal* actions, which are presumed to be likely aiming points for the pitcher. For the baseball domain, we have chosen the set of points shown in Figure 9. In addition, for each pitch, we included the location yielding the maximum expected value for each σ hypothesis, as long as it was at least 2 inches from the center target and previous maximum aiming points added.

We ran TBA with $\beta \in \{0.5, 0.75, 0.85, 0.9, 0.95, 0.99\}$, and found that in this baseball setting the different β values all resulted in very similar estimates. The results with $\beta = 0.95$ were used in the other experiments.

We ran TBA on up to the 1000 most recent pitches in the data for a given pitcher and pitch type. Fewer pitches were used when that was all that appeared in the data. The final estimate produced by TBA after processing these pitches was utilized as the actual σ value for that pitcher and pitch type for use in the game models investigated in the remainder of the paper.

5.1. The Effect of Execution Skill

To help visualize the effect of this execution noise on the situation facing the pitcher, Figures 10 and 11 show examples of the utility for different pitch locations, as estimated by the r_v function as described. Figure 10 shows the utility of different locations assuming perfect execution ability, along with the actual location of the pitch.

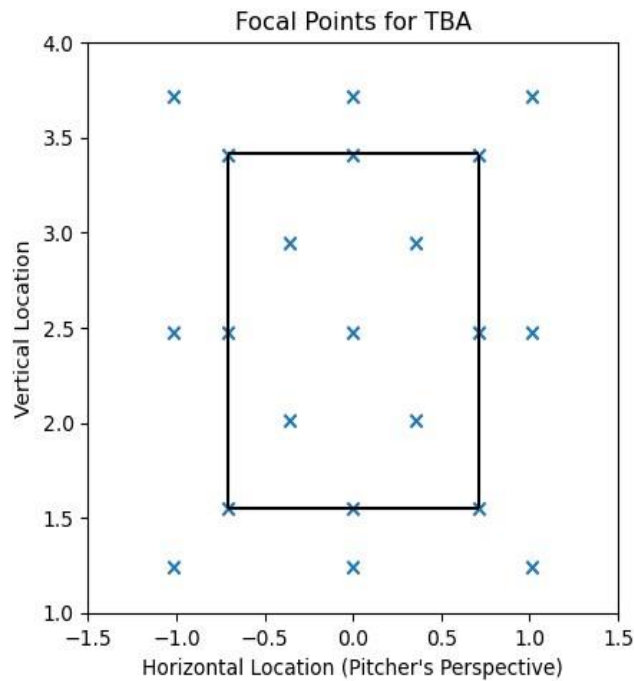


Figure 9: Focal Points for TBA method, where the black box indicates the border of the typical strike zone

As the accuracy of the pitcher decreases, the expected utility of aiming in each location can be calculated by convolving the Gaussian pitcher noise distribution with the perfect execution utility function. Figure 11 shows these different expected utility plots for different σ levels for Aroldis Chapman on a slider (SL) from the data. In each figure the pitch location which yields the highest expected utility for that σ level is marked with an X. It can be seen that, as the execution noise increases, the expected reward decreases, and the optimal pitch converges to aiming somewhere in the middle.

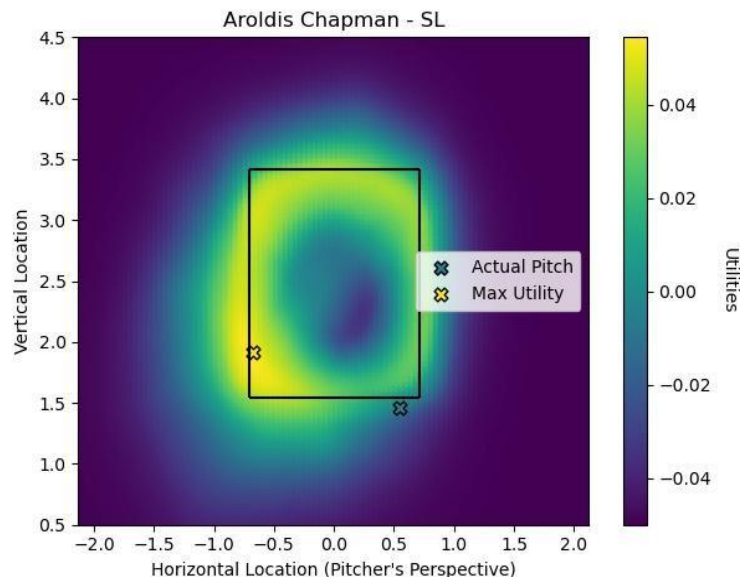


Figure 10: Utility Plot for Aroldis Chapman. X indicates the actual pitch location.

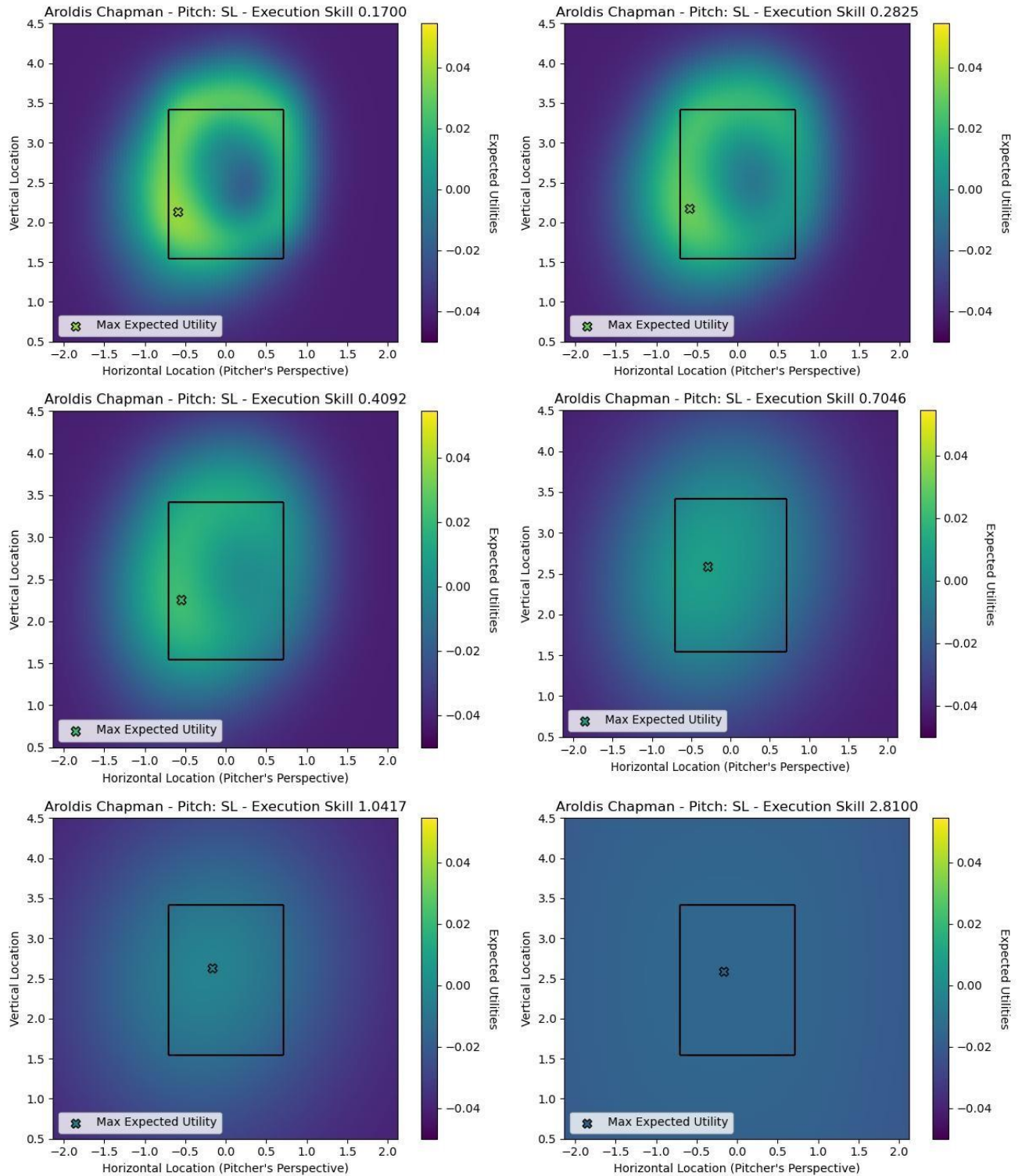


Figure 11: Expected utilities for different execution skills. Colors indicate the expected utility of a target location. The X indicates the optimal pitch location.

6. Game 1: Simultaneous Game

Recall in our first game, we assume the batter can observe nothing about the pitch before making their decision to swing. We have $P_o = \emptyset$ and $P_u = T \times L$ where T is the pitcher's pitch types and L is the set of possible plate locations (see Figure 1). Then the equilibrium in (1) is equivalent to:

$$p_u^* \in \operatorname{argmin}_{p_u \in \Delta P_u} \max_{s \in \Delta S} p_u^T A s \quad (5a)$$

$$s^* \in \operatorname{argmax}_{s \in \Delta S} \min_{p_u \in \Delta P_u} p_u^T A s \quad (5b)$$

where A is the $|P_u| \times |S|$ matrix whose $(i, j)^{th}$ entry is given by $R(p_i, \tau_j, c)$ for $p_i \in P_u$ and $\tau_j \in S$. This formulation is known as the normal form representation of the game. We can set up and solve a linear programming problem to determine the equilibrium, (p_u^*, s^*) [30].

Although we now have the tools to solve for the Nash equilibrium in a batter/pitcher matchup, we do not consider it a worthwhile endeavor. As we saw in the Jansen and Phillips example of Section 3.2 assuming the batter is randomly choosing to swing at a fixed rate can result in a pitching strategy that requires throwing pitches way outside of the strike zone. In the real world, hitters rarely swing at pitches like that, and they just get called balls. Although some Nash pitching strategies may work fairly well, and in fact the other pitch in the optimal strategy for Jansen likely would have worked out better for him than the pitch he actually threw, we think that the assumptions in the next two games are more realistic, and therefore the equilibria would make more effective pitch sequencing strategies.

7. Game 2: Sequential Game

Recall in this game, we assume that the batter can observe the full trajectory of the pitch before making their decision to swing. We have $P_o = T \times L$ and $P_u = \emptyset$. In this case, the equilibrium in (1) is equivalent to:

$$\pi_o^* \in \operatorname{argmin}_{\pi_o \in P_o} \max_{\tau \in S} R(\pi_o, \tau, c)$$

$$\tau^* \in \operatorname{argmax}_{\tau \in S} \min_{\pi_o \in P_o} R(\pi_o, \tau, c).$$

Here the batter is assumed to know which action in S is the best response to the observed pitch, π_o , so they deterministically choose that action. The pitcher has nothing to gain by randomizing their choice of action if the batter is going to respond optimally to any observed pitch, so the pitcher also chooses their action deterministically to minimize the utility of the batter's best response. We solve for the equilibrium (π_o^*, τ^*) , known as the Stackelberg equilibrium, with an exhaustive search over P_o and S .

7.1. Sequential Game Analysis Example

On October 8, 2022, Jacob deGrom of the New York Mets faced off against Juan Soto of the San Diego Padres in the top of the first inning of game two of the NL Wild Card Series. The five-pitch at bat concluded with a 102 mph fastball just outside of the strike zone that Soto swung at and missed for

strike three. The best response for Soto to that pitch would have been to take it, which would have given an expected run value of about 0. The expected run value for a swing was -0.08. Juan Soto is among the best hitters in all of MLB, and he is particularly well-known for not chasing pitches outside of the strike zone. If any major league hitter is capable of always making the optimal swing decision, it is likely Juan Soto. Despite that, deGrom got him to swing at a pitch outside of the strike zone.

As it turns out, no hitter is able to make the optimal swing decision on every pitch under the assumptions of the sequential game. We determined the best response strategy for every pitch in the validation data from Section 4. The batter made the optimal swing decision on 70% of these pitches. Additionally, we identified the optimal pitch for every pitcher that we had execution scores for in the validation data. We found that these pitchers chose the optimal pitch type just 48% of the time.

If batters are not behaving optimally, pitchers may be incentivized to deviate from the equilibrium strategy. The deGrom and Soto at bat provides a good example of this. Figure 12 shows the equilibrium strategy for the final strikeout pitch. The optimal pitch is a changeup right down the middle, which gives an expected run value of -0.02 if Soto chooses the best response action. Recall that the expected run value on the pitch that deGrom actually threw was a much better -0.08 because Soto did not make the optimal decision. Thus we see that if deGrom had some reason to believe that Soto would choose the suboptimal action in response to his outside fastball, then it was actually rational for deGrom to deviate from the equilibrium strategy.

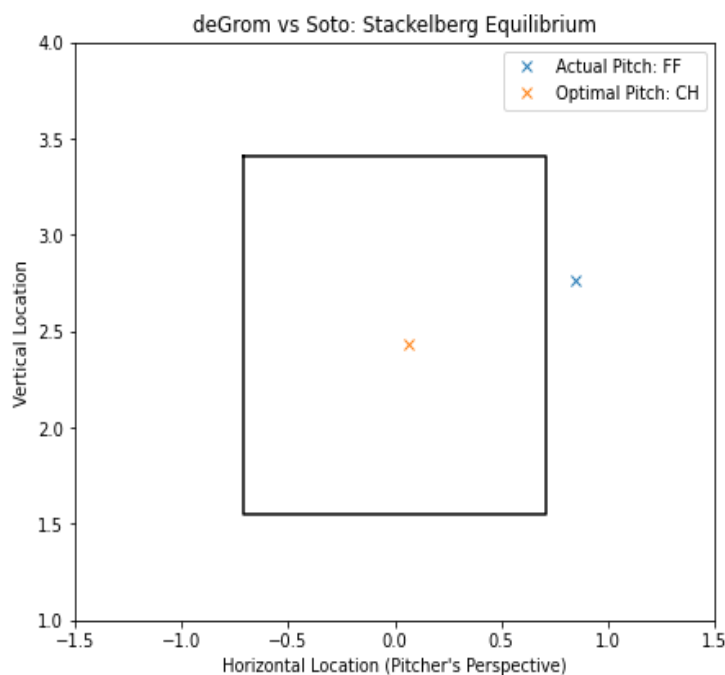


Figure 12: Stackelberg equilibrium for deGrom vs Soto

As mentioned in Section 3, the Stackelberg equilibrium strategy gives an upper bound on how well the batter can respond to a pitch. If deGrom had chosen to throw the optimal pitch of Figure 12, the largest possible expected utility that Soto could have achieved is -0.02, which is a win for deGrom.

Thus, although we have shown that batters will not always behave optimally in the sequential game, we still consider the Stackelberg equilibrium to be a viable pitch sequencing strategy because it minimizes the upper bound of the batter's response. In other words, it minimizes the utility if the batter is a perfect version of themselves who always makes the right decision. Given that batters made the right decision a majority of the time in the validation data, it probably is not a bad idea for pitchers to choose the action that minimizes the best response.

8. Game 3: Decision Point Game

In the previous section, we found that batters do not always make the best swing decision under the assumption that they can observe the full pitch. One possible reason for this is that batters cannot actually observe the full pitch before deciding to swing. It takes some small amount of time for a batter to complete their swing. Naturally, there is a decision point along a pitch's trajectory where if the batter has not decided to swing by that point, they no longer have enough time to complete their swing before the ball reaches the catcher's glove. Rather than assuming that batters can observe the full pitch before deciding to swing, it is therefore more realistic to assume that they can only observe the full pitch up to the decision point. A pitcher can throw multiple pitch types through a decision point, so the batter does not know exactly where the pitch will end up after the decision point because they are not sure which pitch type was thrown through the decision point. For instance, a fastball that ends up at the top of the strike zone may share a decision point with a curveball that ends up at the bottom of the strike zone (Figure 1).

In this third and final game, we assume that the batter can observe the decision point of the pitch, but not the pitch type. Thus, we have $P_o = D$ and $P_u = T$, where D is the set of possible decision points, and T is the set of pitch types. We defined the decision point as the point 24 feet from home plate based on [27]. Note that pitchers often use a sequencing strategy known as tunneling where they try to throw consecutive pitches of different types along a similar trajectory up to the decision point [25]. We will therefore use the terms "decision point" and "tunnel point" interchangeably throughout this section.

We defined a mapping, $f: T \times L \rightarrow D$, using Alan Nathan's physics of baseball work, particularly the trajectory and movement calculators [18][22][23]. We used that mapping to create a training dataset containing pitches and their corresponding tunnels. We then used linear regression to define a mapping, $g: T \times D \rightarrow L$. We used g to determine the plate location of a pitch given its pitch type and its decision point location, which allowed us to calculate the expected run values in (4) for an inputted pitch type and decision point.

Recall in (1b), the batter's equilibrium strategy s^* is a function of the actual observed portion of the pitch, π_o . In this game, we are assuming $P_o = D$, so the equilibrium strategy in this game requires an optimal strategy for all $\pi_o \in D$.

Let A be a matrix who's $(i, j)^{th}$ entry is given by $R(\pi_o^i, \pi_u^j, \tau_j, c)$, where (π_o^i, π_u^j) is the i^{th} entry in $P_o \times P_u$ and where $\tau_j \in S$ is the action of taking a pitch in tunnel $j/2$ if j is even and swinging at a pitch in tunnel $(j + 1)/2$ if j is odd. For instance, τ_1 corresponds to swinging at a pitch in tunnel 1, and τ_2 corresponds to taking a pitch in tunnel 1. Thus, A is $|D| \times |T| \times 2|D|$ because there is a row for every choice of $(\pi_o^i, \pi_u^j) \in P_o \times P_u$ and two columns for every tunnel in D , one corresponding to swinging at a pitch in that tunnel, and one corresponding to taking a pitch in that tunnel.

A allows us to rewrite the expected utility function in (1b) as

$$\mathbb{E}_\pi[U(\pi, s, c)] = p^T A s$$

where the i^{th} entry of p gives the probability that the pitcher chooses action (π_o^i, π_u^i) and where the j^{th} entry of s is equal to the probability that the batter uses action τ_j given his strategy s and given that the observed pitch is in the corresponding tunnel ($j/2$ if j is even and $(j + 1)/2$ if j is odd). Thus, in this case s is of length $2|D|$, where the i and $i + 1$ entry are given by (1b) with π_o^i plugged in for π_o .

Writing the utility function in this way allows us to set up and solve a linear programming problem for the decision point equilibrium, just as we did in Section 6. We have

$$\begin{aligned} \max_{s, v} \quad & v \\ \text{s.t.} \quad & A s \geq v \\ & s_i + s_{i+1} = 1, \forall i \in D \\ & s \geq 0 \end{aligned}$$

The notable difference in this formulation compared to the linear programming problem we solve for the Nash equilibrium is the equality constraint, $s_i + s_{i+1} = 1$ for every $i \in D$, which simply requires that the batter's strategy in each tunnel sums to 1. The solution to this problem gives us the decision point equilibrium strategy for the batter, s^* , as well as the decision point utility, v^* . We get the decision point equilibrium pitcher strategy, p^* , by solving the dual of this problem.

8.1. Sequential Game Analysis Example

On October 28, 2022, The Philadelphia Phillies and the Houston Astros played in game one of the World Series. In the bottom of the first inning, Jose Altuve faced off against Aaron Nola in a five-pitch at bat that culminated with a swinging strike three at a cutter a full foot outside of the strike zone. The decision point equilibrium strategy in that pitch's tunnel was a pure take strategy, so Altuve failed to make the optimal swing decision. Figure 13 shows Nola's mixed strategy for this pitch, as well as the pitch he actually threw. Nola's mixed strategy does not include his cutter, so he also chose a suboptimal strategy.

Just as we saw in Section 7, Altuve is not the only batter who does not always behave optimally. Finding the decision point equilibrium takes a bit longer than the Stackelberg equilibrium, so we focused our analysis on just the first pitch of ten batter/pitcher matchups between batters and pitchers who had a fair number of plate appearances against each other in both the training and validation sets from Section 4. We compared the batters' equilibrium strategies with the actions they actually took. We only looked at pitches that went through tunnels where the batter had a pure strategy for that tunnel. We also only looked at pitches that were inside of our pre-defined grid of tunnels, D . We ended up with 255 pitches to analyze. Of those pitches, the batter made the optimal equilibrium decision 71% of the time, and none of the ten batters made the optimal decision on every pitch.

Batters never really had a chance to behave optimally in this third game. It is one thing to be able to use a pure strategy in an observed pitch tunnel, which we have already seen batters cannot do, but imagine how a hitter would go about using a mixed strategy in a tunnel. Altuve, for instance, needs to randomly swing 12% of the time at the fastball (FF) in the bottom middle part of the strike zone in Figure 13. In order to achieve that, Altuve would first need to be able to identify the tunnel he is observing and its strategy. Then he would need to somehow sample from Bernoulli(0.12) and use that sample to decide if he should swing. And he would need to do all of this in the fraction of a second between when the pitch passes through the decision point and when it reaches the catcher's glove. In our efforts to provide batters the most realistic set of information to use in making their swing decisions, we have inadvertently created an impossible batter strategy.

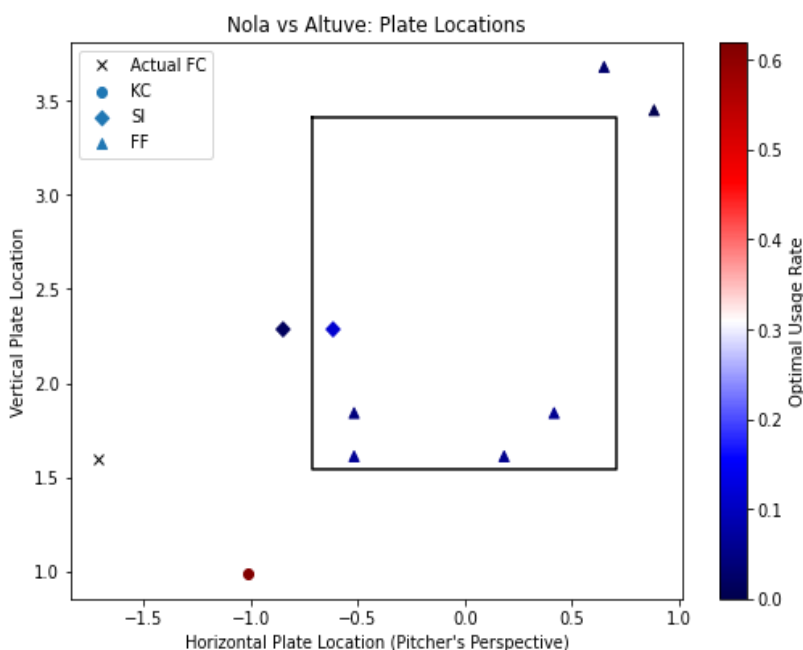


Figure 13: Decision point equilibrium for Nola vs Altuve

Just like in the Stackelberg equilibrium of the deGrom and Soto at bat, if batters are not behaving optimally, pitchers may be incentivized to deviate from the equilibrium. Nola threw a cutter way outside of the strike zone, which certainly was not part of his equilibrium strategy, but it ended up with an expected run value of -0.16 because Altuve made the wrong swing decision. The expected utility using the decision point equilibrium strategy was just -0.01 in comparison. Obviously, if Altuve is going to swing at a pitch a foot outside of the strike zone, then a pitcher should take advantage of that and outperform the decision point equilibrium.

Although batters do not behave optimally, we still consider the decision point equilibrium to be a viable sequencing strategy for pitchers. Batters obviously couldn't hope to deploy a mixed strategy in an observed tunnel, but depending on what sort of random number generating tools are allowed in a dugout during a game, a pitcher may have enough time to sample from their equilibrium mixed strategy before throwing the next pitch. We can interpret the equilibrium in this game as the pitcher strategy that minimizes the upper bound of the utility of the batter's best response. In other

words, the pitcher strategy is the most effective strategy that a pitcher could use against a perfect batter who always makes the right swing decision. If a pitcher uses their strategy against a batter that does not always take the best response action, their expected utility will not get any worse.

9. Conclusion

In this paper we proposed a novel method for utilizing the tools of game theory – specifically the computation of game-theoretically optimal equilibrium strategies in zero-sum games – to model the interaction between a pitcher and batter in baseball. We explore three different zero-sum game models, which differ in the amount of information available to the batter before they must make their swing decision. The recommendations that are proposed by these models are individualized to the specific batter and pitcher and realistically represent the ability of a pitcher to throw a pitch where intended. We feel that grounding pitcher decision-making in the principles of game-theory will lead to better outcomes for pitchers than methods currently used, and that this work constitutes important first steps towards getting the best performance possible from each pitcher in baseball.

References

- [1] Baseball savant. <https://baseballsavant.mlb.com/>.
- [2] Catcher framing: Glossary. <https://www.mlb.com/glossary/statcast/catcher-framing>.
- [3] Crossentropyloss¶. <https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html>.
- [4] Probability calibration. <https://scikit-learn.org/stable/modules/calibration.html>.
- [5] Statcast. <https://www.mlb.com/glossary/statcast>.
- [6] Michael Alcorn. (batter|pitcher)2vec: statistic-free talent modeling with neural player embeddings. In *MIT Sloan Sports Analytics Conference*. MIT Sloan, 2017.
- [7] Christopher Archibald and Delma Nieves-Rivera. Execution skill estimation. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1859–1861, 2018.
- [8] Christopher Archibald and Delma Nieves-Rivera. Bayesian execution skill estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6014–6021, 2019.
- [9] Robert Arthur. The art and science of sequencing: sabermetrics’ undiscovered country. <https://>

[//www.baseballprospectus.com/news/article/23023/the-art-and-science-of-sequencing-sabermetrics-undiscovered-country/](http://www.baseballprospectus.com/news/article/23023/the-art-and-science-of-sequencing-sabermetrics-undiscovered-country/), March 2014.

- [10] Joel R Bock. Pitch sequence complexity and long-term pitcher performance. *Sports*, 3(1):40–55, 2015.
- [11] Kyle Boddy. Choosing the correct pitch sequences: Datadriven decisions. <https://www.drivelinebaseball.com/2012/05/choosing-the-correct-pitch-sequences-data-driven-decisions/>, May 2012.
- [12] Peter Bonney. Defining the pitch sequencing question. <https://tft.fangraphs.com/defining-the-pitch-sequencing-question/>, March 2015.
- [13] Russell Carleton. Baseball therapy: the power of changing speeds. <https://www.baseballprospectus.com/news/article/25494/baseball-therapy-the-power-of-changing-speeds/>, February 2015.
- [14] Ben Clemens. Can “hard in and soft away” make your troubles go away? <https://blogs.fangraphs.com/can-hard-in-and-soft-away-make-your-troubles-go-away/>, June 2022.
- [15] Connor Douglas, Everett Witt, Mia Bendy, and Yevgeniy Vorobeychik. Computing an optimal pitching strategy in a baseball at-bat. *arXiv preprint arXiv:2110.04321*, 2021.
- [16] Glenn Healey. A bayesian method for computing intrinsic pitch values using kernel density and nonparametric regression estimates. *Journal of Quantitative Analysis in Sports*, 15(1):59–74, 2019.
- [17] Glenn Healey and Shiyuan Zhao. Using pitchf/x to model the dependence of strikeout rate on the predictability of pitch sequences. *Journal of Sports Analytics*, 3(2):93–101, 2017.
- [18] David Kagan and Alan M. Nathan. Statcast and the baseball trajectory calculator, Mar 2017.
- [19] James LeDoux. Introducing pybaseball: An open source package for baseball data analysis. <https://jamesrledoux.com/projects/open-source/introducing-pybaseball/>, Jul 2017.
- [20] Ethan Moore. Revamping my pitch quality metric. <https://towardsdatascience.com/revamping-my-pitch-quality-metric-66cb2dbe8d8a>, Sep 2020.
- [21] J Morehouse. No pitch is an island: pitch prediction with sequenceto-sequence deep learning. <https://community.fangraphs.com/no-pitch-is-an-island-pitch-prediction-with-sequence-to-sequence-deep-learning/>, January 2022.

- [22] Alan Nathan. Pitch movement, spin efficiency, and all that, Aug 2018.
- [23] Alan M Nathan. The physics of baseball.
- [24] Arnav Prasad. Decoding mlb pitch sequencing strategies via directed graph embeddings. In *MIT Sloan Sports Analytics Conference*. MIT Sloan, 2021.
- [25] Jon Roegel. The effects of pitch sequencing. <https://tbt.fangraphs.com/the-effects-of-pitch-sequencing/>, Nov 2014.
- [26] Eno Sarris. The pitcher report: What exactly is stuff+? featuring rich hill, sam long, and more. <https://theathletic.com/2641834/2021/06/11/the-pitcher-report-what-exactly-is-stuff-featuring-rich-hill-sam-long-and-more/>, Jun 2021.
- [27] Bret Sayre, Harry Pavlidis, Jonathan Judge, and Jeff Long. Prospectus feature: Introducing pitch tunnels. <https://www.baseballprospectus.com/news/article/31030/prospectus-feature-introducing-pitch-tunnels/>, Jan 2022.
- [28] Joe P Sheehan. More run values. http://baseballanalysts.com/archives/2008/02/writing_about_t.php, Feb 2008.
- [29] Tom Tango. Run values by pitch count. <http://tangotiger.com/index.php/site/article/run-values-by-pitch-count>.
- [30] Robert J. Vanderbei. *Linear Programming: Foundations and extensions*. Springer, 2020.
- [31] PI Yifan. Predict next baseball pitch type with rnn.
- [32] Keisuke Yoshihara and Kei Takahashi. Pitch sequences in baseball: Analysis using a probabilistic topic model. *Available at SSRN 3728430*, 2020.
- [33] Chih-Chang Yu, Chih-Ching Chang, and Hsu-Yung Cheng. Decide the next pitch: A pitch prediction model using attention-based lstm. In *2022 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pages 1–4. IEEE, 2022.