

Figure 1: Sequence of Multi-Games Across a Season

values that an  $O_x$  objective variable can take – in the EPL would be:

- **Winning the League** ( $o_1$ ): Awarded to the team who finishes top of the league.
- **Qualifying for the Champions League** ( $o_2$ ): Awarded to the top 4 teams, so in this case the objective relates to teams finishing 2nd-4th.<sup>5</sup>
- **Qualifying for the Europa League** ( $o_3$ ): Another European competition usually awarded to teams who finish between 5th-7th.
- **Top Half Finish** ( $o_4$ ): The financial benefit of finishing higher in the league are huge and therefore teams often aim to finish in the top half of the table (higher than 10th).<sup>6</sup>
- **Avoiding Relegation** ( $o_5$ ): The bottom 3 (18th-20th) teams in the EPL are relegated into the English Football League (EFL) Championship which is the second division of the English football leagues.

To set the objective we can simulate how we expect the season to unfold and create a distribution  $\mathcal{D}$  that allows us to use a Maximum a Posteriori (MAP) estimation [10] for the probability of the team finishing in each position. This then allows us to calculate a set of probabilities for of a team achieving each objective  $\mathcal{P} = \{p(o_1), p(o_2), \dots, p(o_k)\}$ . We then set the  $O_o$  (for a pre-season objective) as the most likely objective that can be achieved by a team that season.

This process can then be re-run after each game-week is completed to give the fluent objective  $O_1$  to  $O_{N-1}$ . Our simulation of the league will include the real-results which will get more accurate as the season progresses and we learn more about each team. This will then mean we have a fluent objective that will change as the

season progresses. At the end of the season, we can compare  $O_0$  to  $O_{N-1}$  to the final outcome  $S_O$  that the team achieves.

### 3.3 Learning From Prior Games

As well as the fluent objective, we can also improve the tactical decision-making in our Bayesian and stochastic games by adding prior knowledge  $P$  that we learn after each game we play and observe. In more general terms we aim to observe and learn from other successful agents and our own actions. This could also be applicable in swarms of UAVs or imitating other agents trading in the financial markets settings.

We can learn a set of weights  $\mathcal{W}$  that relate to how effective given style/formation pairs (actions that are made in the multi-step games) that we select in our games are against given oppositions style/formation pairs. These weights are initially set to 1 and are then increased if found to be effective and decrease if found to be ineffective. These can be updated after each game-week and also updated from the other games that we observe. Our  $P$  value is defined in Equation 1.

$$P = \begin{pmatrix} w_{11} & w_{12} & w_{13} & \dots & w_{1j} \\ w_{21} & w_{22} & w_{23} & \dots & w_{2j} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ w_{i1} & w_{i2} & w_{i3} & \dots & w_{ij} \end{pmatrix} \quad (1)$$

Where  $w \in \mathcal{W}$  and  $i/j$  is the number of possible style/formation pairs. The columns represent the style/formation pair selected by our team and the rows represent the style/formation selected by the opposition (e.g.,  $w_{ij}$  is how effective our style/formation pair  $i$  is against an opposition using style/formation pair  $j$ ).

In the following sections, we give more details into how we calculate our fluent objective  $O$  and how we can learn the weights that make up  $P$ . We explore how these are used in the individual

<sup>5</sup><https://www.premierleague.com/european-qualification-explained>.

<sup>6</sup><https://www.goal.com/en-gb/news/how-much-money-do-premier-league-2019-20-winners-get/19jbaudy17cw1ieojo40yextz>.

football match multi-step game (discussed in Section 2.3) to further optimise the tactical decision-making process.

## 4 CALCULATING THE FLUENT OBJECTIVE

In this section, we discuss how we simulate seasons, calculate the fluent objective, and how this can be used to optimise game tactics.

### 4.1 Simulating Season Outcomes

When we simulate the season outcomes and calculate the distributions of where we expect the team to finish we are interested in predicting all remaining games in the season for both our team and all other teams in the league. To do this we first look at the single-game prediction which is discussed in the next subsection.

**4.1.1 Single-Game Prediction.** To predict the outcomes of single games in the league we use the model that is defined in [3] which is used for calculating the single-game payoffs. The model uses the team’s tactical style, potential formation and team strength to give probabilities of a team winning the game. The set of features used are: home team style, away team style, home team formation, away team formation and then team strengths are calculated by using the outputs from the model described in [8]. The target class is the final result of the game: home team win, away team win or a draw.

Using these features, we train a multi-class classification deep neural network. The neural network is trained using stochastic gradient descent using a categorical cross-entropy loss function (Equation 2) and a soft-max activation function.

$$-\frac{1}{N} \sum_{i=1}^N \log p_{model}[y_i \in O_{y_i}] \quad (2)$$

where,  $N$  is the number of games that we are using to train the model and  $p_{model}[y_i \in O_{y_i}]$  is the probability that  $y_i$  is in the class  $O$ . This model takes the given teams, possible playing styles and possible formations to estimate the probability of winning, drawing or losing the game. Using these probabilities we can simulate the outcome of the entire season, this is discussed in the next subsection.

**4.1.2 Season Simulation.** To simulate the remaining games of the season, we use the real-world fixture list to ensure that the ordering of the games is correct. We then find the probability of a home win, away win and draw in each game and use a Markov chain Monte Carlo simulation [26] to simulate all remaining games and total up the points that each team will gain (3 points for a win, 1 for a draw and 0 for a loss). This works well as it emulates the randomness that we see in real-world football games. We repeat this process 100,000 times for each simulation which allows us to derive a distribution for the probability that a team will finish in each place in the league in the final standings. An example of this distribution is shown in Figure 2.

### 4.2 Setting the Fluent Objective

Once we have calculated the distributions of possible place outcomes from the MCMC simulation, we use a Maximum a Posteriori (MAP) estimation [10] to set the fluent objective. To do this, we can use the posterior distribution to find interval estimates of the final position for the team in the league. We use the position intervals

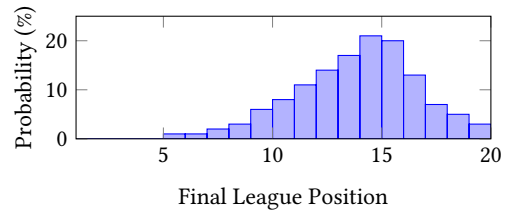


Figure 2: Example League Outcome Probability Distribution.

for the objectives discussed in Section 3.2 and can find the  $o_k \in O$  that maximises the posterior PDF. This then sets the objective  $O_n$  that is used in game-week  $n$  and is updated after each game-week.

### 4.3 Optimising Tactics using the Fluent Objective

Once we have set the fluent objective we can now use this when optimising the team tactics in the multi-step game for optimising individual game tactics in that game-week. In the pre-match Bayesian game outlined in [3], Beal et al. present 3 options that can be used depending on the overall environment. Here we present modified, novel notions of these options, which now employ the fluent objective.

- **Best Response:** Used to maximise the chances of winning a game. This option is selected if a team is currently not on track to achieve their objective for the season and must win games to be able to achieve their goals.
- **Spiteful:** Used to minimise the chances of the opposition winning the game (and therefore improve your chances of drawing/winning). This option is selected if a team is well ahead of their objective and that by preventing losing the game they are more likely to stay on track for their objective across the season.
- **Expectimax:** This is a mixture of the two above and factors both into account (mathematically defined in [3] where referred to as “minmax”).<sup>7</sup> This is selected if a team is on track for their objective and is aiming to stay that way.

In terms of the in-match stochastic game that is also defined in [3] there are two options that can be selected when making in-match decisions.

- **Aggressive Approach:** This is set if a team is losing/drawing a game and wants to win. It will maximise the chance of a team moving to a more positive state. Therefore, if we know that the objective is to win and gain three points we will select this approach.
- **Reserved Approach:** This is set if a team is winning/drawing and is happy with their current state. It is used to maximise the chances of staying in the current state. Therefore this is used if winning or if a point is a good result in the overall environment in relation to the objective.

<sup>7</sup>We rename since the approach does not align with the usual meaning of the term “minimax” or “minmax” in Game Theory.

In the next section, we move on to assess how we can learn from prior games and other games in the environment and how this can be added to our optimising decisions model.

## 5 LEARNING FROM PREVIOUS GAMES

In this section, we discuss how we can learn from completed prior games that we play and that other teams in the league play. This allows us to find formation/style combinations that work best against a given formation/style combination that an opposition team may use. To do this we learn a matrix of weights  $P$  that corresponds to estimated successes of the formation/style combinations. To estimate each of the weights  $w \in P$  we factor in both the games that we have played as well as the games that we have observed. Each weight  $w$  corresponds to how effective a given formation/style combination is against a given opposition formation/style. These are computed using Equation 3 where we look at the games won when using the formation/style ( $x$ ) against the given opposition formation/style ( $y$ ), both in games we have played (first fraction) and in games we have observed (second fraction).

$$w_{xy} = \left( \frac{\text{gameswon}}{\text{gamesplayed}} + \frac{\text{observedgameswon}}{\text{observedgames}} \right) \div 2 \quad (3)$$

These weights in  $P$  are updated after each game-week so should become more accurate across the season. In game-week 1 all weights can either be set to 1 or be carried over from the previous season. In the next season, we outline how  $P$  is used to optimise the pre-game tactics in the Bayesian Game and in-match decisions in the stochastic game.

### 5.1 Optimising Tactics using Prior Games

Once we have computed the weights that we use in  $P$ , these can be used when making our pre-match decisions in our Bayesian game. In the optimisation model, a payoff table is computed for each combination of opposition actions to give the probability of the match outcomes based on their selected action of styles  $S$  and formations  $f$ , where  $h$  is home win,  $d$  is a draw and  $a$  is an away win. The payoff for the team is the weighted sum of win and draw probabilities that we store in payoff table made up from the different decision that we can make.

We can then apply the computed weights in  $P$  to the payoff table to weigh each payoff depending on how successful these have been in prior games and in observed games. Therefore, we can optimise the tactical decision based on the weighted payoffs in these tables using either the best approach, spiteful or expectimax approaches which are decided based on our fluent objective. This means that if a formation/style combination has never worked in games we have played or observed the payoff will be weighted by 0 and not be selected. The same approach can be applied when changing the formation and style in the in-match stochastic game and each change made can be weighted by the corresponding element in  $P$ .

In the next section, we perform a number of experiments on our models and assess the performance over a whole given season as well as how the inclusion of  $O$  and  $P$  each game-week can be used to help teams improve their performance and meet their objectives.

## 6 EMPIRICAL EVALUATION

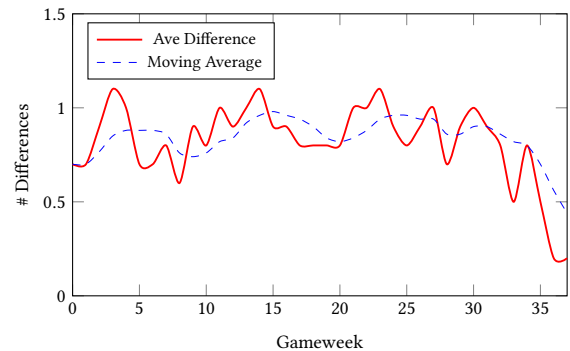
To evaluate our models we use a dataset collected from two seasons (2017/18 and 2018/19) from the English Premier League (EPL).<sup>8</sup> The dataset breaks down each of the games from the tournament into an event-by-event analysis where each event gives different metrics including event type (e.g., pass, shot, tackle), the pitch coordinates of the event and the event outcome. This type of dataset is industry-leading in football and used by top professional teams. Thus, this is a rich real-world dataset that allows us to rigorously assess the value of our model.

### 6.1 Experiment 1: Learning the Fluent Objective

Here, we test our fluent objective model in each game-week. Firstly, we evaluate the individual game prediction model that is used to feed the probabilities of outcomes into our season simulation. Secondly, we evaluate our season simulation prediction model using a Markov-chain Monte-Carlo (MCMC) simulation with respect to its accuracy as the season progresses. In Experiment 2, we test our MAP estimator for setting fluent objectives at each game-week.

To predict the outcome probabilities of individual games we use the deep learning neural network model that calculates pay-offs in the Bayesian game.<sup>9</sup> Over the past two EPL seasons the accuracy of the model is 72.99% with a precision of 69.48%, recall of 59.5% and F1 Score of 59.82 %. This model is used to calculate the probability distribution used in our MCMC model for the entire season.

We then run a number of experiments of our MCMC simulation of a season. We predict all remaining games 100,000 times and find the most likely league standings after 38 game-weeks. We can compare this to the final league ranks and compare the differences. In Figure 3, we show an average of all clubs' absolute difference in their actual finishing position and their predicted finishing position. This is run after each game-week so we have more information about the games that have already been completed. Week 0 is the prediction before any games have been played and week 37 is the final prediction after 37 out of 38 games have been played.



**Figure 3: 2018/19 EPL Actual League Standings vs MCMC Predictions**

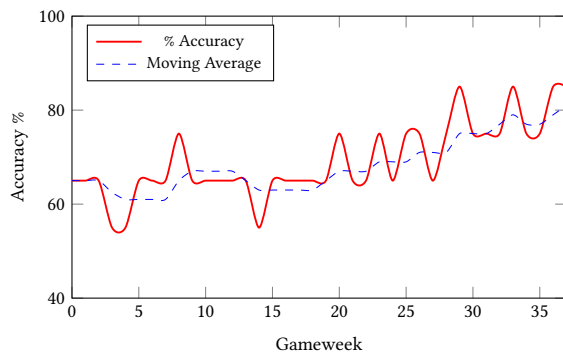
<sup>8</sup>All data provided by StatsBomb - [www.statsbomb.com](http://www.statsbomb.com).

<sup>9</sup>We use a fully-connected feed-forward NN with 3 layers & a ReLU activation function.

As shown in Figure 3, we can see how in the first half of the season the league standings remain fairly unpredictable due to the number of different possible combinations that we are attempting to predict – there are a total of  $2.43 \times 10^{18}$  different combinations of team order that the league could finish in.<sup>10</sup> We do see however that as the season unfolds and we have a better idea of team performance the simulation accuracy improves. This is also to be expected as we are simulating fewer games later into the season and we have more evidence from those having taken place in the real world. This shows that we have a suitable method to extract a distribution of where we expect a team to finish and can therefore derive the fluent objective using a MAP estimation to get our objective. This is shown in the next experiment.

## 6.2 Experiment 2: Setting the Fluent Objective

To test our MAP estimation, after each game-week simulation we set the fluent objective for all 20 EPL teams. We then assess if their objective was met at that game-week and show the percentage of teams that were successful in meeting their objectives. This is shown in Figure 4 where week 0 is the prediction before any games and week 37 is the final prediction.



**Figure 4: Accuracy of Setting the Fluent Objective (2018/19 EPL Season).**

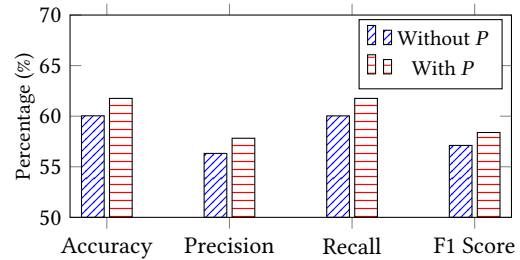
As we can see in Figure 4 the fluent objective accuracy rises as the season progresses and from week 15 onwards we see the accuracy of the fluent objective setting rise more clearly. This shows that we can set realistic to aim for as the season progresses in relation to the actual league outcomes and what was achieved by the teams. One thing to note in this experiment is that not every team in the league can meet their objective as there may be more teams aiming for something than can achieve it (e.g., 3 teams aiming to win the league). Also, 3 teams must always be relegated which the minimum objective is to avoid, meaning that even in the best case only 85% of teams will achieve their objective. We find that in weeks 36 and 37, we reach the maximum 85% of teams meeting their objectives.

## 6.3 Experiment 3: Learning from Observing Games

To test the impact of the addition of the weights  $w$  that we estimate in  $P$ , we evaluate how the weights are able to boost our ability to

<sup>10</sup>The vast number of possible combination is why we use position differences rather than the overall accuracy of the entire standings after each game-week.

predict the outcomes of games based on the tactical decisions and therefore improve our payoff model. To evaluate our  $P$  weights, we compare the accuracy of the predictions of the model presented in [8] both with and without  $P$  (this model makes up part of the feature set that is used for calculating the payoffs). We then assess the differences in terms of the models' ability to be able to accurately predict the outcome of the game running the tests over 1046 games. In both cases, the prediction is the highest probability that is given to one of the results (home win, away win and draw). The results from this experiment are shown in Figure 5.<sup>11</sup>



**Figure 5: Payoff Model Performance Comparison.**

As we can see in Figure 5 by using the weights in  $P$  we are able to boost in the accuracy of the model, and therefore the accuracy of our payoffs, achieving a boost of 1.76%. We also see that there is an increase in the precision, recall and F1 of our model by 1.50%, 1.72% and 1.27% respectively. Even though this represents a fairly small increase to the results of the model in [8], it shows that by learning from what tactics have worked (both for your team and others), we can boost our ability to calculate the tactical decision pay-off and therefore our ability to optimise decisions made. Over a large scale of time such as a 38 game-week season, a 1.76% boost in performance could be the difference between finishing a place higher in the league which can have huge financial gain and help to achieve our set fluent objective.

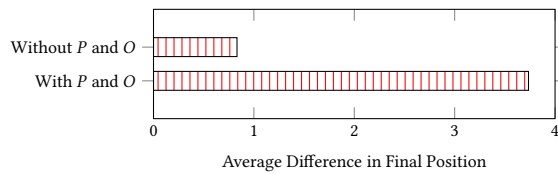
## 6.4 Experiment 4: Optimising Team Long-Term Performance

Our final experiment assesses how we incorporate the fluent objective  $O$  and weights in  $P$  into the tactical decision-making optimisation model presented in [3] and evaluate how this improves team performance to help them meet their objective. To test this we simulate an entire season week by week and apply our model to a single team in the simulation. After each game-week we simulate the remaining games and recalculate  $O$  and  $P$  as outlined in Figure 1. We then compare our results using the new model across a simulated season against a simulation where we do not use the  $O$  and  $P$ . We show the results from this when running separate simulations for a set of different teams<sup>12</sup> (the team we use is the only team using the new model in each simulation) in Figure 6. We show the average difference in the mean-expected finishing position from

<sup>11</sup>The precision, recall and F1 score are computed as a weighted average of the ability to predict each outcome using SciKit Learn's multi-class support.

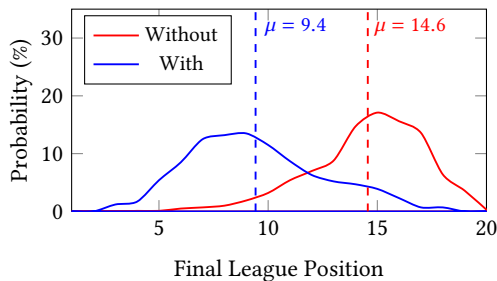
<sup>12</sup>We use the bottom 8 teams in the 2018/19 EPL season to show we can improve their performance.

the distribution of each team that we run our season simulation for, both using the new model and without.



**Figure 6: Payoffs of Real-World vs. Optimised Decisions**

This shows how our model can improve the probability of teams' finishing positions and see that on average there is a 2.90 position improvement when using  $O$  and  $P$  compared to without for our test set of teams. This is achieved as by using  $O$  and  $P$  teams can add more context to their decisions, also by selecting the optimal tactics each week in the simulation using the model in [3] we would also expect to see a boost to the performance. Below, we highlight an example of the distribution improvement of the simulation when aiming to optimise the performance of Southampton FC (only team using the optimisation model in the simulation). Figure 7 shows the distribution with  $O$  and  $P$  applied and not applied.



**Figure 7: Example League Outcome Probability Distribution for Southampton FC in 2018/19.**

As we can see from the example shown in Figure 7, we can use the fluent objectives to help teams boost their probabilities of winning games that matter, and thus boost their expected finishing position, increasing the mean of the expected finishing distribution by up to 35.6%. We see similar improvements to this across our test set of teams. In the next section, we will further discuss these results, the real-world implications and some further findings.

## 7 DISCUSSION

One interesting finding from further experiments is when we simulate the season with all teams using the model discussed in this paper to select their tactics. When we run this simulation, we find that the results cancels itself out and the final standings are very similar to what we see when we run the simulation without the new fluent objective and prior game weights. We see that there is a boost of under 1 position on average per team when every team uses the model in the same season. This shows that teams can gain a boost in their performance over the season but only if they utilise the game theoretic approaches while all others are not.

Another observation we see in our results is when we compare the increase in the positional distribution using the model between the stronger top-half teams and the teams who are in the lower half of the league and aiming to stay in the division. When using the model for the latter, we observe a substantial boost of up to 35.6% in long-term performance. This may be due to the algorithm helping teams using the new model gain positive results in the closer games at the bottom of the table when playing teams of similar ability and thus preventing them gaining points by taking all 3 for yourself. In turn, higher up the league teams often win the games they are expected to against weaker teams so the performance boost is lower.

It is also worth noting that across the season there are also a number of other variables that can affect team decision-making both tactically and off the pitch. As teams re-assess their objectives in the season, there are decisions off the pitch that can help boost their performing as well as the tactical decision optimisation that helps on it. One example is a change in managers/coaches, this is often a measure taken for an underperforming team and can help boost performance. If a team is doing well and wants to push higher up the table or is struggling and needs new players, then during January teams are able to invest money into new players to improve their team and again improve. These types of decisions could be added into the model to help decision makers at clubs subjectively decide when to invest more money or make changes.

## 8 CONCLUSIONS AND FUTURE WORK

This paper presents a novel model for the long-term tactical decisions that are made in football and helps teams to optimise their decisions by adding more long-term context. We introduce the concept of a *fluent objective* that allows us to re-evaluate team performance and base decisions based on a wider environment. We find that we can build models that are able to predict the final outcome of the table on a regular basis, and then using a MAP estimation to effectively set the fluent objective each week. We also learn from other games that happen in the overall environment and find that this can boost the performance of pay-off models in our multi-step games. Overall, we find that our model can be used for football teams who are looking to improve their overall expected league position (on average improves teams by 2.90 positions) and, show that the concept of a fluent objective can help to optimise long-term performance in a competitive league setting.

Due to the success we show when using fluent objectives for an application to football in this paper, in future work we intend to test our approach in other domains. For example, they could be used in security games and UAV swarms as the objective also often change over a given time frame. This testing will help to further verify how the modelling of objectives can aid long-term performance. We also aim to further improve our  $P$  weights with applications of the observational learning and reinforcement learning as presented in [6]. Finally, the reinforcement learning techniques presented in [17, 24] could be used to further optimise team performance.

## ACKNOWLEDGMENTS

We would like to thank the reviewers for their comments. This research is supported by the AXA Research Fund and the EPSRC NPIF doctoral training grant number EP/S515590/1.



## REFERENCES

- [1] Per Bak, Maya Paczuski, and Martin Shubik. 1997. Price variations in a stock market with many agents. *Physica A: Statistical Mechanics and its Applications* 246, 3-4 (1997), 430–453.
- [2] Albert Bandura. 2008. Observational learning. *The international encyclopedia of communication* (2008).
- [3] Ryan Beal, Georgios Chalkiadakis, Timothy J Norman, and Sarvapali D Ramchurn. 2020. Optimising Game Tactics for Football. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*. 141–149.
- [4] Ryan Beal, Narayan Changder, Timothy J Norman, and Sarvapali D Ramchurn. 2020. Learning the Value of Teamwork to Form Efficient Teams. In *Proceedings of AAAI 2020*. 7063–7070.
- [5] Ryan Beal, Timothy J. Norman, and Sarvapali D. Ramchurn. 2019. Artificial intelligence for team sports: a survey. *The Knowledge Engineering Review* 34 (2019).
- [6] Diana Borsa, Nicolas Heess, Bilal Piot, Siqi Liu, Leonard Hasenclever, Remi Munos, and Olivier Pietquin. 2019. Observational Learning by Reinforcement Learning. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. 1117–1124.
- [7] Tom Decroos, Lotte Bransen, Jan Van Haaren, and Jesse Davis. 2020. VAEP: An Objective Approach to Valuing On-the-Ball Actions in Soccer. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*.
- [8] Mark Dixon and Stuart Coles. 1997. Modelling Association Football Scores and Inefficiencies in the Football Betting Market. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 46, 2 (1997), 265–280.
- [9] Javier Fernández, Luke Bornn, and Dan Cervone. 2019. Decomposing the Immeasurable Sport: A deep learning expected possession value framework for soccer. *MIT Sloan Sports Analytics Conference* (2019). <http://www.sloansportsconference.com/wp-content/uploads/2019/02/Decomposing-the-Immeasurable-Sport.pdf>
- [10] J-L Gauvain and Chin-Hui Lee. 1994. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE transactions on speech and audio processing* 2, 2 (1994), 291–298.
- [11] Min Jang and Sungzoon Cho. 1999. Ensemble learning using observational learning theory. In *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339)*, Vol. 2. IEEE, 1287–1292.
- [12] Jeremy D Jordan, Sharif H Melouk, and Marcus B Perry. 2009. Optimizing football game play calling. *Journal of Quantitative Analysis in Sports* 5, 2 (2009).
- [13] Gary Kagan, Herbert Mayo, and Robert Stout. 1995. Risk-adjusted returns and stock market games. *The Journal of Economic Education* 26, 1 (1995), 39–50.
- [14] Christopher Kiekintveld, Manish Jain, Jason Tsai, James Pita, Fernando Ordóñez, and Milind Tambe. 2009. Computing optimal randomized resource allocations for massive security games. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. 689–696.
- [15] Fangzhen Lin. 2008. Situation calculus. *Foundations of Artificial Intelligence* 3 (2008), 649–669.
- [16] Thomas Lux and Michele Marchesi. 1999. Scaling and criticality in a stochastic multi-agent model of a financial market. *Nature* 397, 6719 (1999), 498–500.
- [17] Tim Matthews, Sarvapali D. Ramchurn, and Georgios Chalkiadakis. 2012. Competing with humans at fantasy football: Team formation in large partially-observable domains. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.
- [18] Praveen Paruchuri, Jonathan P Pearce, Janusz Marecki, Milind Tambe, Fernando Ordóñez, and Sarit Kraus. 2008. Playing games for security: An efficient exact algorithm for solving Bayesian Stackelberg games. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*. International Foundation for Autonomous Agents and Multiagent Systems, 895–902.
- [19] Bilal Piot, Matthieu Geist, and Olivier Pietquin. 2013. Learning from demonstrations: Is it worth estimating a reward function?. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 17–32.
- [20] Sarvapali D Ramchurn, Trung Dong Huynh, Feng Wu, Yukki Ikuno, Jack Flann, Luc Moreau, Joel E Fischer, Wenchao Jiang, Tom Rodden, Edwin Simpson, et al. 2016. A disaster response system based on human-agent collectives. *Journal of Artificial Intelligence Research* 57 (2016), 661–708.
- [21] Anand Ranganathan and Roy H Campbell. 2003. An infrastructure for context-awareness based on first order logic. *Personal and Ubiquitous Computing* 7, 6 (2003), 353–364.
- [22] Stuart Russell. 1998. Learning agents for uncertain environments. In *Proceedings of the eleventh annual conference on Computational learning theory*. 101–103.
- [23] Eric Anyung Shieh, Bo An, Rong Yang, Milind Tambe, Craig Baldwin, Joseph DiRenzo, Ben Maule, and Garrett Meyer. 2012. PROTECT: An application of computational game theory for the security of the ports of the United States. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.
- [24] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* 529, 7587 (2016), 484.
- [25] Kwang Mong Sim and Chung Yu Choi. 2003. Agents that react to changing market situations. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 33, 2 (2003), 188–201.
- [26] Jasper A Vrugt, James M Hyman, Bruce A Robinson, Dave Higdon, Cajo JF Ter Braak, and Cees GH Diks. 2008. *Accelerating Markov chain Monte Carlo simulation by differential evolution with self-adaptive randomized subspace sampling*. Technical Report. Los Alamos National Lab.(LANL), Los Alamos, NM (United States).