university of groningen
faculty of science and engineering
artificial intelligence

# ENHANCING FOOTBALL SIMULATION PERFORMANCE IN DEEP REINFORCEMENT LEARNING THROUGH ANALYTICS-BASED DENSE REWARD SHAPING

Bachelor's Project Thesis

Andre van Dommele, s3606392, a.van.dommele@student.rug.nl,
Supervisor: Rafael Fernandes Cunha

**Abstract:** Recent research in reinforcement learning (RL) has seen a shift towards experimenting within complex and stochastic environments with large, dynamic state spaces. Football, one of the most popular sports, has garnered significant interest in RL research for these reasons. Football has also seen a rise in data-based analysis to improve real world player and team performance. A common issue in RL for complex environments is accurately modeling the relationship between the ultimate goal and the individual actions that produce desired behavior. Previous research shows that incorporating prior knowledge through reward shaping is essential for efficiently training RL agents to learn complex concepts, as it allows for conditioning individual actions at a local scale. Existing work either incorporates football domain knowledge within the model representation while keeping the sparse reward unchanged, or arbitrarily changes the reward without clear motivation. In this paper, we focus on incorporating football domain knowledge as a dense reward motivated by real world analytics in a proximal policy optimization-based RL scheme. Experimental results through extensive simulations against a fixed opponent show an improved policy in novel scenarios compared to the results published by the environment developers, as well as comparative performance to models with a greater architecture complexity.

## 1 Introduction

Reinforcement Learning (RL) can be classified as the study of understanding and automating goal-directed learning and decision-making in environments of various complexity (Sutton & Barto, 2020). RL and Deep Reinforcement Learning (DRL) algorithms have been becoming increasingly applicable in solving complicated real-world situations successfully such as traffic control signals (TCS) and HVAC management (Sivamayil et al., 2023). While the development of these systems are impressive achievements, from a goal-directed learning perspective these problems are fairly trivial for RL algorithms to solve. For example, investigating a TCS system solution where the ultimate goal is to (safely) minimize the total waiting time for all cars at an intersection over a continuous period, the state space of this environment is relatively limited. Encompassing variables could include the current phase of the traffic light, the number of vehicles in each lane, and the current waiting times of vehicles. This would be sufficient to create a successful model for the task (Bouktif et al., 2023), but still a conceivably small state space. Thus, this can be considered a solved problem with existing RL technology. However, the significance of an agent and its behavior is directly tied to the complexity of the domain in which it operates. As agents learn from interactions within their environments, more complex domains and environments naturally lead to more sophisticated agent behavior. This complex behavior observed through engaging with increasingly advanced environments has led to multiple milestones in understanding and challenging traditional strategy in partially-solved problems such as the game of Go (Silver et al., 2016), which is considerably more difficult to solve due to the magnitude of possible states spaces.

Further research in RL has continued experimenting with increasingly complex environments.

Moving beyond deterministic games where the transition functions are predictable and often static, the focus has shifted to more complex and stochastic environments with larger, dynamic state spaces. These environments require sophisticated learning algorithms capable of managing levels of uncertainty and intricate strategy formations. Football, one of the most popular sports, has seen significant interest in RL for these reasons. Tangentially, football has also seen a rise in utilizing data science based analysis to understand and improve the performance of players and teams. Mathematical models such as the Pitch-Control Model (Fernández & Bornn, 2018) have been developed to quantify spatial value occupation of in-possession and out-of possession players, enhancing strategic approaches and tactical insights in a manner that is highly reproducible. Naturally, this has invoked research into the combination of football analysis and DRL in football environments. Google Research released a physics-based football simulation for RL research purposes (Kurach et al., 2020), which has inspired investigation into many of the current scientific problems facing RL such as sample-efficiency, and sparse reward problems.

Various type of deep neural network techniques have been applied in attempt to overcome the learning difficulties found within DRL in football. Existing studies (B. Liu et al., 2023) have used the Pitch-Control Model to design a knowledge embedded state representation of the environment, applied with a Deformable Convolution Network (DCN) to achieve impressive results. This was assisted by their custom-built shaped reward, which was largely motivated by intuition through common knowledge of real world football. Other studies (J. Liu et al., 2022) have done similar investigations in implementing football knowledge to the model architecture of the training algorithm, but leave the sparse rewards provided by the environment unchanged. A common issue found within RL in complex environments is correctly modeling the relationship between the long horizon goal and the individual actions that produce the desired behavior; the correlation between the two can become less apparent over training time. For this reason, Adam Laud (Laud, 2004) believes that the incorporation of prior knowledge through the use of reward shaping is the key to efficiently train artificial intelligence agents to learn complex concepts, as it allows conditioning for individual actions at a local scale. Given the large growth of data aggregation in the professional football domain, we have sufficient availability to data of the individual actions within a team that are considered ideal play in real world football.

Motivated by the discussions above, this paper will investigate if we can rationalize and implement an effective RL policy, based on a reward function inspired by leveraging analytics from domain knowledge; to provide a more consistent dense reward to assist with the long horizon learning goal in Google Football.

Effectiveness will be measured by:

1. Number of timesteps for policy convergence.
2. The model's ability to generalize and perform in new, unseen scenarios of the environment.
3. Average goal difference compared to existing models

We believe that future applications can benefit from a better understanding on exploring rationales behind shaping an effective reward within the context of a specific environment. As well as how shaped rewards effect training processes in complex, stochastic environments.

## 2 Related Work

Reward shaping as a means to incorporate domain knowledge is a divisive topic among researchers in reinforcement learning. When reasoning about the mechanics of the reward function as a means to promote an ultimate goal, common intuition compels us to believe that this is an appealing strategy. If we shape the reward function to include domain knowledge, this will promote a minimal search radius, and therefore a lower reward horizon; resulting in a faster convergence to an effective policy and greater success in high dimensional environments (Laud, 2004).

However, other researchers disagree with this narrative. A fundamental belief behind reinforcement learning is the idea that agents should learn from a blank slate using a general model (Randlov & Alstrøm, 1998), any additional information you wish to provide to the agent should be encoded in the environment as part of the problem statement.

Barto and Sutton (Sutton & Barto, 2020) agree with this premise, stating “In particular, the reward signal is not the place to impart to the agent prior knowledge about how to achieve what we want it to do”. The skepticism of incorporating domain knowledge in the reward function arises from the greedy nature of policy gradient algorithms, where intentionally introducing a bias may cause adverse affects depending on which state the agent finds itself in (Eschmann, 2021). More general methods of promoting efficient exploration are often considered, such as intrinsic motivation signals, which foster comparative psychological rewards such as curiosity or novelty (Oudeyer et al., 2016). However, we reject the notion that successful reward shaping is incapable to achieve, and believe that in following certain characteristics in context to the underlying environment, the potential to significantly enhance learning outcomes can be unlocked.

As Adam Laud introduces (Laud, 2004), the theory of reward shaping overlaps considerably with the psychological learning processes of conditioning and intermediate reinforcement. This is based on the understanding that contextually relevant and timely rewards allow the correlation between actions and outcomes to become more apparent. Nevertheless, these localized rewards must be consistent with prior knowledge in order for the adoption of the desired behaviors to be faithful to the desired outcome. In Liu et al. (B. Liu et al., 2023), their custom shaped reward features a combination of positive and negative signals for the long horizon reward (scoring), and possession and dribbling sub-tasks. The reward signal polarity for the possession and dribbling sub-tasks are dependent on an arbitrary division of the environment into 5 different regions, with the regions closest to the user goal returning the negative reward signal. While the logic behind this reward captures the essence of an effective football strategy, it is clear that it was strictly motivated through intuition, and is not entirely consistent with prior knowledge. Studies (David Adams & Williams, 2013) exist which contradict the logic behind this reward design, demonstrating a direct performance correlation in top-level professional teams with the number of completed short passes between defenders. This trait, however, would be negatively rewarded under their current design.

# 3 Preliminaries

In this section, we introduce relevant background knowledge to contribute full context to our research. We provide a comprehensive overview of the Google Football Environment and its representations, as well as how those representations function in a mathematical framework for decision-making used in reinforcement learning; the Markov Decision Process. We then introduce the Proximal Policy Optimization (PPO) algorithm, followed by a summary of the default reward functions implemented in the Google Football Environment.

## 3.1 Google Football Environment

In 2019, Google Research released Google Research Football (Kurach et al., 2020). It is a reinforcement learning environment which provides a 3D, physics-based football simulator that follows the real rules of football. It consists of two teams of 11 agents which broadly compete to score more goals than the other team; but also offers additional training scenarios under the ‘Football Academy’ benchmarks. In this paper, we focus on single agent DRL, meaning the algorithm controls a single player on the left side team called the ‘active player’. The active player is generally the player closest to the ball, but more specifically the player of attention. For example, if current active player ‘1’ has the ball and attempts a pass to inactive player ‘2’, the environment will switch activity to player ‘2’ immediately in anticipation of receiving a pass, despite the ball being closer to player ‘1’ for the majority of the pass. All other inactive players on the user team are controlled by a rule-based agent which operates separately from the model our agent is playing against in training. The Google Football Environment provides three different state representations such as pixels, super mini-map (SMM), and a 115-dimensional vector. We will be utilizing the SMM representation, which contains four $72 \times 96$ matrices that encodes various environment observations as the state. This representation includes detailed information about the ball such as its position, direction vector, rotation vector, and possession status. It also encompasses data of both the left and right teams, including player positions, movement vectors, stamina levels, yellow card status, activity status (indicating red cards), and specific player roles. Furthermore, it provides insights into the controlled player's information, detailing any active actions. Additionally, the match state is encapsulated, featuring the current score, remaining steps until the match ends, and the current game mode (such as Normal, KickOff, GoalKick, FreeKick, Corner, ThrowIn, and Penalty). The environment contains a discrete action space of 19 different actions, meaning that the agent can choose from a finite set of distinct actions at each state. The action set can be found in the appendix.

## 3.2 Markov Decision Process

The model of the football environment can be classified as a Markov Decision Process denoted as $M =< S, A, P, r, \gamma >$ (van Otterlo & Wiering, 2012). Each state $s \in S$ is a unique representation of the $72 \times 96 \times 4$ shape described previously, while each action $a \in A$ is from the mentioned action space. P is the state transition function, where $P(s'|s, a)$ gives the probability for reaching $s'$ after taking action $a$ in state $s$. $r$ is an immediate reward, which we will official define later for our case. Lastly, $\gamma \in [0, 1)$ is the discount factor, which determines the emphasis placed on immediate rewards over future return.

## 3.3 Proximal Policy Optimization

Proximal Policy Optimization, or PPO, is a reinforcement learning algorithm designed at OpenAI (Schulman et al., 2017). It has proven itself successful in a wide variety of single-agent tasks from robotic control to complicated video games. PPO is well-versed to handling environments with both discrete and continuous action and state spaces, making it an ideal algorithm for our environment with a discrete action space, but continuous state space. Consequently, PPO is a common algorithm used within the aforementioned studies of Google Research Football, and we will continue to adopt it for our research. This section will outline the mechanisms and techniques used in PPO from a reinforcement learning perspective, highlighting the importance of an increasing data utilization ratio with respect to operating in the Google Football Environment.

In reinforcement learning, the training data generated is dependent on the current policy of the algorithm, as it is collected through the agent's interactions with the environment. Consequently, the data distributions for observations and rewards are continually evolving. This inherent instability compounded with sensitivity to hyperparameters leads to an overall unstable training process. PPO was developed with these problems in mind, as it looked to improve the sample efficiency of no experience buffer policy gradient methods.

Policy Gradient Loss is defined as the expectation of the product of the log of the parameterized policy output from our network ($\pi_\theta$), and the estimate of the advantage function ($\hat{A}_t$), which measures the relative value of the selected action. In this context, $\pi$ represents the strategy used by the agent to decide actions, while $\theta$ denotes the parameters of said strategy.

$$L^{PG}(\theta) = \mathbb{E}_t \left[ \log \pi_\theta(a_t|s_t) \hat{A}_t \right] \tag{3.1}$$

Expanding $\hat{A}_t$, its value is obtained using generalized advantage estimation (GAE) (Schulman et al., 2018), which combines the current discounted return $\gamma$ and the estimated future returns $\lambda \in (0, 1)$.

$$\hat{A}_t^{\mathrm{GAE}^{(\gamma,\lambda)}} = \delta_t + (\gamma\lambda)\delta_{t+1} + \cdots + (\gamma\lambda)^{T-t-1}\delta_{T-1} \tag{3.2}$$

Here, $\delta_t$ represents the temporal difference error, or the difference between the predicted value of the current state and the updated estimate of the next state. The equation sums the difference errors from the current timestep $t$ to the final timestep $T$. $\delta_t$ is calculated as

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t) \tag{3.3}$$

where $r_t$ is the reward at time $t$, and $V(s_t)$ is the estimated value at $t$ given by the critic network. This allows for balancing between bias and variance of advantage estimations.

Expanding $\log \pi_\theta(a_t|s_t)$, as $\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\mathrm{old}}}(a_t|s_t)}$, we end up with the objective function or surrogate objective, who's goal is to be maximized within a constraint of the policy update.

$$\max_\theta \mathbb{E}_t \left[ \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\mathrm{old}}}(a_t \mid s_t)} \hat{A}_t \right] \tag{3.4}$$

$$\text{constraint } \mathbb{E}_t \left[ \mathrm{KL} \left[ \pi_{\theta_{\mathrm{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t) \right] \right] \leq \delta \tag{3.5}$$

However, setting a constraint function that generalizes well across a problem within the course of training is difficult, and proposed modifications are made. The proposed method of PPO is to alter the surrogate objective by clipping the probability ratio, removing the incentive for our policy output $\log \pi_\theta(a_t|s_t)$ to move outside the intervals $[1-\epsilon, 1+\epsilon]$, where $\epsilon$ is an importance sampling clipping parameter. The updated objective function can be defined below, where the result is a function that benefits from the gradient clip operation with its update process being relatively stable.

$$\begin{gathered} L^{PG}(\theta) = \mathbb{E}_t[\min(\log \pi_\theta(a_t|s_t)\hat{A}_t^{\mathrm{GAE}^{(\gamma,\lambda)}}, \\ \operatorname{clip}\left(\log \pi_\theta(a_t|s_t)\right), 1-\epsilon, 1+\epsilon)\, \hat{A}_t^{\mathrm{GAE}^{(\gamma,\lambda)}})] \end{gathered} \tag{3.6}$$

### 3.4 Default Environment Rewards

The Google Football Environment comes supported with two different extrinsic reward functions, each with their own characteristics in respect to the ultimate goal. The 'Scoring' reward corresponds to the natural reward in football where each team obtains +1 for scoring a goal, and −1 for conceding a goal. In the context of the environment, this is an extremely sparse reward. Numerous steps within the simulation can occur before a positive, or negative signal from the reward function is given, due to the general complexity of the task. In contrast, the 'Checkpoint' reward corresponds to the football domain knowledge that scoring is a result of advancing the ball closer to the opponents goal. Specifically, the opponents field is divided into 10 regions based on the Euclidean distance to the opposition goal. Each initial time the agent possesses the ball in one of the checkpoint regions, a reward of +0.1 is given, for a maximum of +1, the same as scoring a goal. All non-collected checkpoints are also given if a goal is scored, to avoid penalizing the agent for achieving the ultimate goal outside the confines of the checkpoints. This can be considered a dense reward, as it allows for more frequent signals from the reward function over the course of many episodes, which can be highly beneficial for algorithms based on policy gradient methods (Sehnke et al., 2010).

## 4 Methods

In this section, we detail and motivate the implementation and design characteristics behind our shaped reward. We introduce our methodology behind the sourcing and processing of football data to obtain meaningful analytics, which we then appropriate and illustrate in a Markov Decision Process framework.

### 4.1 Reward Function Design Motivation

To apply the principles in the previous section successfully, we need to design our reward function with certain characteristics in mind. First, we must clearly define the long-term goal of our agent and understand the implications of this goal within the state space of the environment. In our context, the primary objective is to score more goals than the opposing team. While achieving this objective could involve various strategies, including robust defensive modeling to prevent the opponent from scoring, it is important to ensure that the reward function aligns with the desired outcome and accounts for dynamic behavior. This involves focusing on rewarding processes which offer endorsement for the long-horizon goal, rather than offering small incentives to avoid large negative consequences, as it may be counterproductive in a high-dimensional environment (Yuan et al., 2021). Furthermore, we should avoid generalized state punishment for trivial reasons to maintain a healthy balance of exploration and exploitation and promote a faster convergence (Dayal et al., 2022). Additionally, we are operating in a realistic simulation of a physical game, which is highly popular. Assuming the simulation faithfully replicates a real-world game of football (A. Scott et al., 2021), we have access to millions of samples of ideal play from the highest level in the domain. While our algorithm will simulate thousands of games during training, leveraging this extensive pool of data can significantly enhance our approach.

This compounds into the motivation behind our reward function design, which is ultimately to encourage the active agent to occupy positions that historically led to goal-assisting passes. We will reward the agent for productive existence in these spaces, while providing an additional reward to the

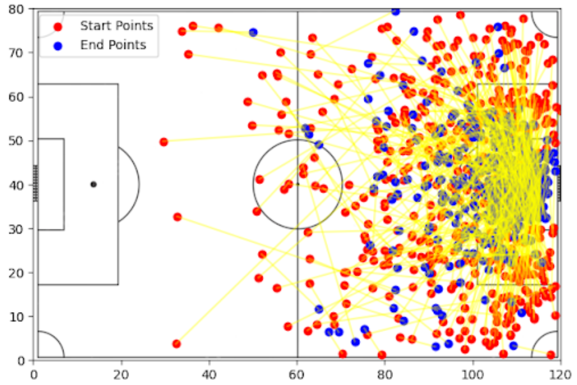

**Figure 4.1: Positional graph of start and end passing locations that resulted in assists. Red dots denote the starting positions and blue dots denote the ending positions, with yellow lines connecting the coordinate pairs.**

agent for completing a transition between active players from the goal-assisting starting spaces, to the goal-assisting ending spaces; ultimately modeling an assist in the environment. The distinction between the regions will be determined statistically using Gaussian Kernel Density Estimation (D. W. Scott, 1992), which has shown to be an effective representation of spacial structure in sports (Mortensen, 2020). The weights of the distributions within the reward will be scaled to ensure that the magnitude reflects the relative importance between the starting states, ending states, and the ultimate goal - scoring.

## 4.2 Analytics

We sourced positional data of events that precede scoring in real-world professional play using an open-source database from *StatsBomb*. We collected all instances of goal-assisting passes from open play during the 2015/2016 seasons of the English Premier Division and the French Ligue 1 division. This resulted in 474 data points of start and end positions plotted in a space ranging from 0 to 120 for the x-axis and 0 to 80 for the y-axis, illustrated in 'Figure 4.1'.

The scale of the football field in the Google Football Environment is naturally different than the scale of the collected data points. So we must normalize the data points to a space of -1 to 1 for the x-axis, and -0.42 to 0.42 for the y-axis.

Next, we perform Gaussian Kernel Density Estimation (KDE) on the coordinate pairs to determine the perceived spatial influence of the areas. Since our shaped reward depends on modeling the process of an assist, we apply the KDE function to the set of starting positions and ending positions separately. This process results in two density distributions, each contributing to the reward function.

The Gaussian Kernel Density Estimation (KDE) is defined by the formula:

$$\hat{f}(x,y) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2\pi h^2} e^{-\frac{(x-x_i)^2+(y-y_i)^2}{2h^2}} \qquad (4.1)$$

The exponential function is the Gaussian function, which measures the squared euclidean distance between a given point $(x, y)$ and each data point $(x_i, y_i)$. This function is multiplied by a factor of $\frac{1}{2\pi h^2}$ for normalization in two dimensions. Lastly, the contributions of each data point are summed and normalized by the number of data points in the set $n$, yielding the estimated density at a given point, $\hat{f}(x, y)$. $h$ is the bandwidth parameter of the kernel, which controls smoothness of the density estimation. We found Normal Distribution Approximation (Silverman, 2018) to be appropriate for our use case, which is defined as

$$h = \left(\frac{4\sigma^5}{3n}\right)^{1/5} \qquad (4.2)$$

where $\sigma$ is the standard deviation of the data and $n$ is the number of data points. The results are two individual density estimation matrices for the starting position and ending positions, which can be viewed in 'Figure 4.2'.

## 4.3 Reward Function Implementation

In this section, we will define our reward function implementation within a MDP framework to illustrate its calculation during a step in the environment. Some of the variables' notations are derived directly from the current observation of the environment; however, we will also introduce new variables in our shaped reward. These additional variables will augment the state, allowing us to access them as part of the observation.

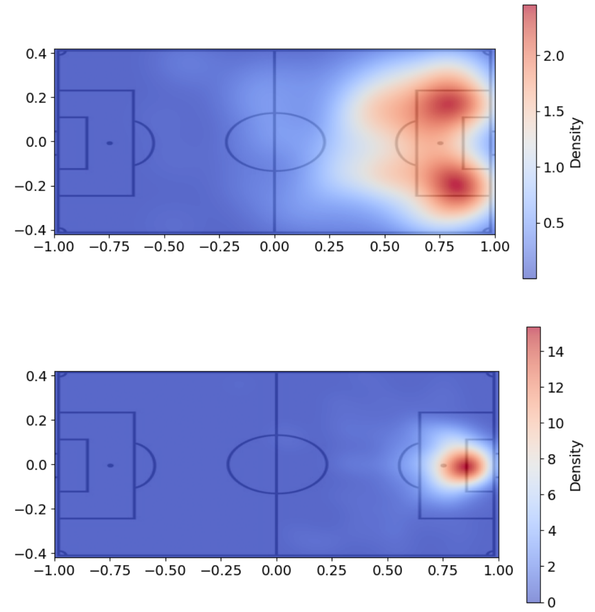

**Figure 4.2: Gaussian Kernel Density Estimation of Assist Starting (Top) and Ending (Bottom) Passing Positions**

First we define our long horizon goal of scoring and conceding. We leave the scaled value of scoring unchanged from the default reward implementations to preserve consistency, as well as it being a generally intuitive value to place on a goal being scored. In the environment, a goal is considered scored if the ball's coordinates surpass the coordinates of the left or right goal, located at -1 and 1 on the X-axis, respectively. The goalposts span between -0.044 and 0.044 on the Y-axis.

$$scoring(s,a) = \begin{cases} +1 & \text{if goal scored} \\ -1 & \text{if goal conceded} \end{cases} \tag{4.3}$$

Next, we define a boolean variable to ensure the active player has possession of the ball at the current state, this variable is directly obtained from the environment observation.

$$active(s,a) = \begin{cases} 1 & \text{if at observation } s \text{ the active player has the ball} \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

From the observation of the environment at state $s$, we are able to obtain the $(x, y)$ coordinates of the ball. We will denote the position of the ball as follows

$$O_b(s) = s_b(x), s_b(y) = x, y \tag{4.5}$$

where $s_b(x)$ represents the x-coordinate of the ball at state $s$, and $s_b(y)$ the y-coordinate of the ball at state $s$.

These coordinates of the ball will directly correspond to a value calculated from the previously defined density distributions. We scale the values of each distribution to reflect the relative importance of the perceived actions associated with them, in the context of the value of scoring a goal.

For our starting position reward, this value is scaled to a maximum of 0.001. This value is significantly smaller than the reward for scoring due to its large spatial size in proportion to the environment. Consequently, relative to scoring, the maximum reward this distribution can provide at any given step will be worth 1/1000 of a goal. This can be represented as:

$$\hat{f}_s(x,y) = \frac{0.001}{v_{\max}} KDE_{\text{starting}}$$

$$\hat{f}_s(O_b(s,a)) = k \quad \text{where} \quad k \in [0, 0.001] \tag{4.6}$$

For our ending position reward, this value is scaled to a maximum of 0.1. We scale this value to be significantly larger than the starting position reward due to the positional importance in the context of the ultimate goal and the greater difficulty in achieving this reward compared to the starting position, which will be realized in the complete function.

$$\hat{f}_e(x,y) = \frac{0.1}{v_{\max}} KDE_{\text{ending}}$$

$$\hat{f}_e(O_b(s,a)) = k \quad \text{where} \quad k \in [0, 0.1] \tag{4.7}$$

Lastly, we define a check to confirm if the active player in the current state is different from the active player in the previous state. This augments the state with a variable that tracks the index of the active player with possession of the ball in the previous state, to access in the next state. Essentially this models the transition of the ball between active players between $s$ and $s'$.

$$transition(s, s') = \begin{cases} 1 & \text{if } O_{active}(s') \neq O_{active}(s) \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

The aggregation of these functions results in our complete reward function, which is defined in its entirety below.

$$R(s, a, s') = scoring(s, a) + active(s, a) \times \\ (\hat{f}_s(O_b(s, a)) + (\hat{f}_e(O_b(s, a)) \times transition(s, s'))) \tag{4.9}$$

This function can be decomposed into three distinct components that collectively contribute to the total reward. The first being the natural reward for scoring or conceding a goal. The second being the value from the starting position density estimation, dependent on the active player having possession of the ball. With the third being the value from the ending position density estimation, given a successful transition between active players with possession of the ball. This approach effectively captures the essence of rewarding a pass from common assist starting locations to common assist ending locations, motivating the agent to occupy these spaces without overshadowing the long-term goal of scoring a goal.

# 5 Experimental Setup

To evaluate our analytics-based shaped reward in terms of its ability to converge to an effective policy, and generalize to scenarios, we train multiple models on various Football Academy scenarios of interest, as well as on different difficulty levels of a full 11v11 game. We compare the performance of the models trained with the analytics-based shaped reward to similar models with various specifications and enhancements in fresh scenarios. This section describes the finer details of the environment scenarios used, as well as the experimental setup, encompassing training and hyperparameter details.

## 5.1 Scenarios Used

We train our model using the analytics-based shaped reward on five different scenarios provided by the Google Football Environment, all of these scenarios involve training against a fixed opponent. Three of these are Football Academy scenarios and can be viewed in 'Figure 5.1' and 'Figure 5.2', namely:

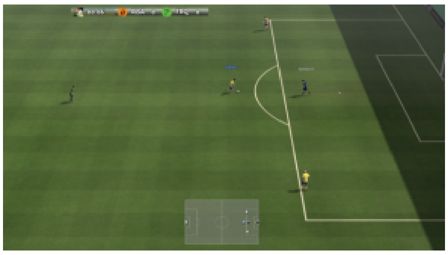

**Figure 5.1: 3v1 with Keeper Scenario**

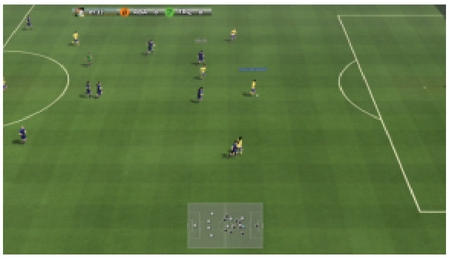

**Figure 5.2: Counterattack Scenario**

1. Academy 3v1 with Keeper - Three user players try to score from the edge of the box, one on each side, and the other at the center. Initially, the player at the center has the ball and is facing the defender. There is an opponent keeper.

2. Academy Counterattack Easy - 4 versus 1 counter-attack with keeper; all the remaining players of both teams run back towards the ball.

3. Academy Counterattack Hard - 4 versus 2 counter-attack with keeper; all the remaining players of both teams run back towards the ball.

We chose to investigate these scenarios due to comparative studies reporting low performance in models with a shorter training time compared to models with a longer training time, suggesting difficulties in effective policy convergence. The Football Academy scenarios run for a maximum of 400 steps per episode, with early stopping if the ball goes out of play, or a goal is scored/conceded.

The full 11v11 game scenario attempts to replicate a simulation of a real football game. The ball starts in the center of the field, and the active player(s) needs to possess the ball, overcome the opposition, and eventually score. The simulation runs continuously for 3000 steps. We train a model for both the “11v11 stochastic easy” and “11v11 stochastic” scenarios in order to make an effective comparison to results published by the Google Football Environment developers, and to other models competing in the Google Research Football Competition with Manchester City F.C. (Addison Howard, 2020).

## 5.2 Model

For our PPO model, we use the convolutional neural network architecture as described in the original paper by the environment developers, inspired by (Espeholt et al., 2018). We opted to keep the model unchanged for simplicity and comparison purposes. The input state, with dimensions 72 × 96 × 16, is normalized by dividing by 255. This input is then processed through a convolutional layer with a $3 \times 3$ kernel, a stride of 1, and 16 channels, followed by a ReLU activation. This is followed by a max-pooling layer with a 3 × 3 kernel, stride of 1, and ReLU activation, maintaining the number of channels as [16, 32, 32, 32]. The output of the residual blocks is fed into a fully connected layer with 256 units and a ReLU activation. Finally, two separate fully connected layers predict the value function and the policy from the features produced by the previous layer. A diagram of the model can be found in ‘Figure 5.3’.

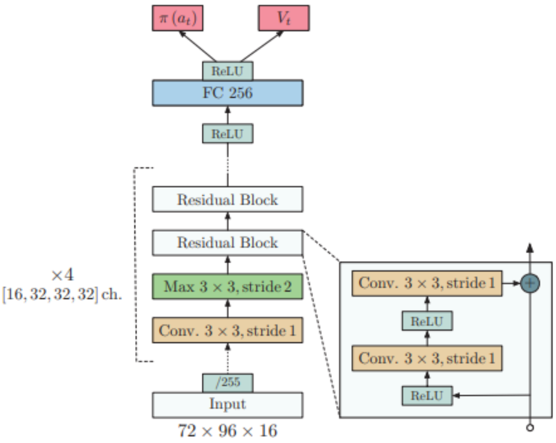

**Figure 5.3: Architecture used for PPO Experiments.**

## 5.3 Training Details and Hyperparameters

In PPO, the primary mechanism for improving control and learning efficiency involves increasing the number of collected experiences per update (Yu et al., 2022). This principle contributes to our rationale for the 11v11 hyperparameter selection, as detailed in ‘Table 5.1’. It is important to note the disparity between the ending conditions for the environmental scenarios. Unlike the Academy scenarios, which have dynamic end cases depending on the state, the continuous end case in the 11v11 scenarios affects variance throughout the PPO training. In environments with long episodes, a larger batch size and more frequent updates are essential to accommodate the extensive variability and to ensure that the model can learn from a more comprehensive and representative set of state experiences across the entire episode length. Our motivation for the Academy hyperparameters selection was from a parameter sweep done in the original Google Research Football paper (Kurach et al., 2020).

We elected to use an annealing learning rate on a linear decay schedule over the course of training. The reasoning behind this choice while somewhat nuanced, generally aligns with the consensus that it improves training stability. Studies suggest that gradient estimates may poorly correlate with the true gradient, and despite the potential for inducing degenerate agent behavior, optimal gradient estimates often require a lower learning rate (Engstrom et al., 2020).

We used the PPO implementation by Stable-Baselines3. Each Academy scenario model was trained for one million timesteps, while each 11v11 scenario model was trained for five million timesteps. Training was done on a Nvidia V100 with 8 cores and 32GB of memory, completing roughly one million steps in four hours.

# 6 Experimental Results

Results for our experiments and comparisons for Academy and 11v11 scenarios are presented in 'Figure 6.1', 'Figure 6.2', and 'Figure 6.3'. Average goal difference was determined by recording the scoring performance of each trained model across 100 new scenarios for each scenario type.

## 6.1 Football Academy Results

'Figure 6.1' presents the average goal difference at 1 million steps for our model trained with our analytics-based shaped reward against the model results published by Google Football with the default scoring reward. In the Academy 3v1 scenario, our shaped reward model greatly outperforms the default scoring reward, with an average goal difference of 0.53 compared to 0.09. A similar trend is observed in the counter-attack scenarios across both difficulty levels. While the default scoring reward failed to exceed 0.0 on either difficulty, our shaped reward model achieved an average goal difference of 0.09 and 0.03 on the easy and hard difficulties, respectively.

## 6.2 11v11 Results

'Figure 6.2' presents the average goal difference in the 11v11 medium difficulty scenario, comparing our shaped reward model against the results of default scoring model and a default checkpoint model both published by Google Football, each with varying timesteps. Our shaped reward model with 5 million timesteps fails to outperform either the scoring reward model or the checkpoint reward model at 20 million timesteps. The exact values of the three models are -1.29, -0.71, and -0.29, respectively. Due to computational constraints, there is a substantial difference in the number of training timesteps between our reported results and those in the original Google Football paper. Future investigations could consider a simulation with 20 million timesteps. 'Figure 6.3' presents the average goal difference in the 11v11 easy difficulty scenario at 5 million steps, comparing our shaped reward model against other models created through research in the same domain. KESR-DCFP, 16SMM-DCFP, and KESR-RESNET are all models created by the work done in Liu et al. (B. Liu et al., 2023). PARIS(Seungeunrho, 2020) is a model that was created and made open source as a result of the Google Research Football Competition Addison Howard, 2020, which it achieved 6th place in. Our shaped reward model achieved a average goal difference of -1.02, which is a similar score to most of the models reported results at this timestep.

**Table 5.1: Scenario Hyperparameters For PPO**

| | Academy | 11v11 |
|---|---|---|
| lr | $decay(3.43e-4)$ | $decay(3.43e-4)$ |
| nsteps | **512** | **1024** |
| mbatch | **8** | **64** |
| n_epochs | **2** | **4** |
| discount | 0.993 | 0.993 |
| GAE | 0.95 | 0.95 |
| clip_range | 0.08 | 0.08 |
| ent_coef | 0.00155 | 0.00155 |
| vf_coef | 0.5 | 0.5 |
| max_grad | 0.64 | 0.64 |

# 7 Discussion

The experimental results highlight several key findings regarding the performance of our analytics-based shaped reward model in comparison to existing models in both Academy and 11v11 scenarios. In the chosen Football Academy scenarios, our shaped reward model consistently outperforms the default scoring reward model, achieving a significant increase in goal difference in the 3v1 scenario compared to the default model. This trend continues in the counter-attack scenarios, where our model outperforms the default model on both easy and hard difficulties, achieving a goal difference higher than 0 at 1 million timesteps. Notably, there is a difference in complexity between these two scenario types. The 3v1 scenario always operates with 5 agents (including goalkeeper) total, while the counter-attack scenarios always initializes agent positions in a 4vX situation, but all 22 agents are present as if it were a 11v11 game. This suggests that the analytics-based shaped reward approach is more effective in scenarios with fewer agents and therefore, a simpler state space.

The 11v11 scenarios presents a more complex scenario, where our shaped reward trained with 5 million timesteps did not outperform the de-

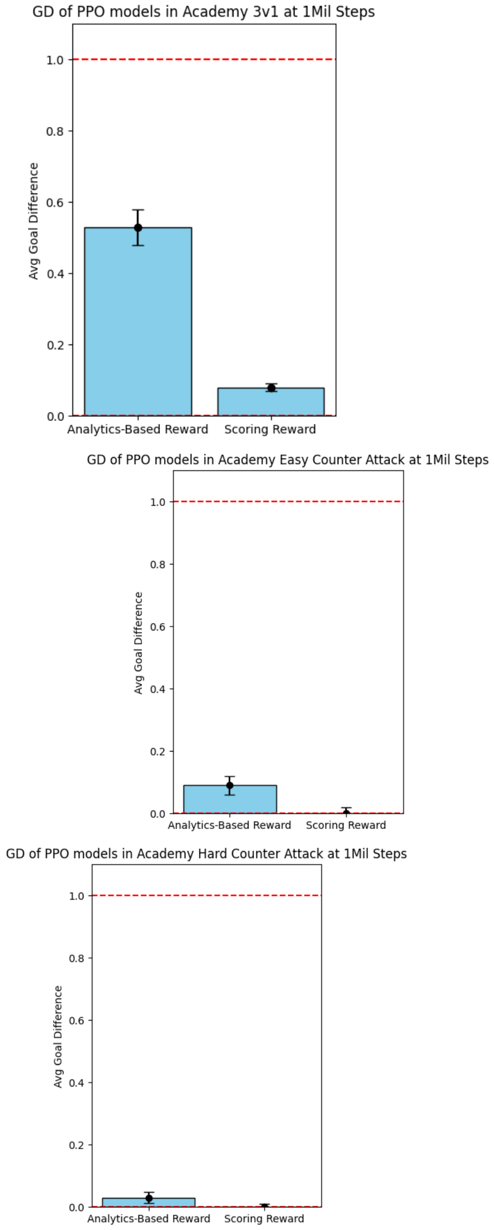

**Figure 6.1: Academy Scenario goal difference comparisons by reward type. Note that the Analytics-Based Reward outperforms the default Scoring reward in all scenarios.**

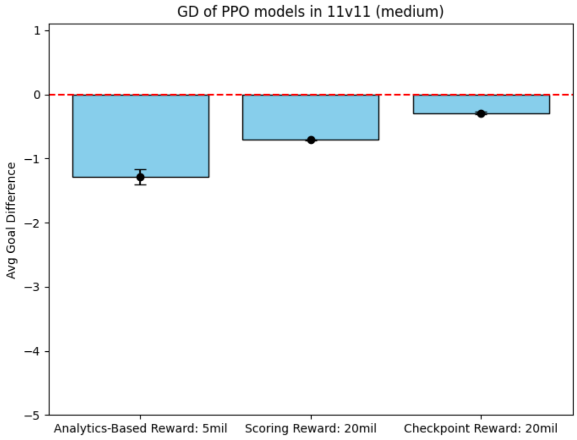

**Figure 6.2: Average goal difference on medium difficulty by reward type. Note the difference in trained timesteps.**

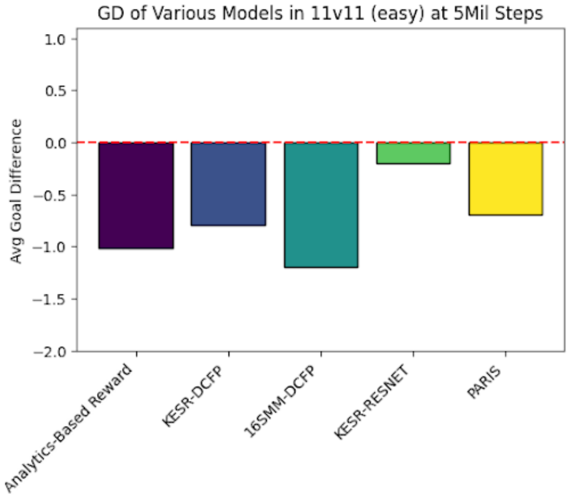

**Figure 6.3: Average goal difference on easy difficulty by model technique.**

fault models trained with 20 million timesteps in medium difficulty. Despite this, the results at 5 million timesteps in the easy difficulty scenario indicate that our model's performance is comparable to other models created in the same domain. Many of the models used for comparison, most of which were trained to a standard of approximately 20 million timesteps, successfully converged to a positive goal difference in the full 11v11 scenario. Our model achieving similar performance suggests it has the potential to do the same. However, this assumption is only based on observing the increasing average return over training, as it is difficult to interpret model stability from other training metrics. The complexity of the environment means that even the highest performing parameters from the original Google Football parameter sweep do not yield the most optimal or expected training metrics typically seen in simpler RL environments. This highlights a broader issue in the RL domain: the challenge of generalizing expected training metrics across diverse environments. Studies have shown that RL algorithms often do not operate or model as intended based on traditional training metrics, advocating for a shift beyond evaluation methods solely based on these metrics (Engstrom et al., 2020).

Initially, we avoided the use of negative reward signals for the purpose of promoting exploration, and for a general dissatisfaction of the negative reward modeling done by Liu et al. (B. Liu et al., 2023). However, studies exist (Dayal et al., 2022) which show that models trained with a negative reward class tend to converge with lower variance in training, despite having a slower overall convergence. This could be beneficial for future investigation, by using domain analytics to shape an effective negative reward.

## 8 Conclusions

In this paper, we present the conception and implementation of a dense reward function motivated by specific analytical domain knowledge. Experiment results confirm that our method was able to improve performance of trained models in certain scenarios, achieving a higher goal difference performance in environmental scenarios compared to the results reported using the default rewards. Our reward function was also able to achieve similar performance to comparative models with higher architecture complexity, showing that reward shaping can be an effective strategy in RL for navigating complex, stochastic environments.

## References

Addison Howard, B. R. C. F. G. G. S. M. M. P. S., Anton Raichuk. (2020). *Google research football with manchester city f.c.* Kaggle. Retrieved from `https://kaggle.com/competitions/google-football`

Bouktif, S., Cheniki, A., Ouni, A., & El-Sayed, H. (2023). Deep reinforcement learning for traffic signal control with consistent state and reward design approach. *Knowledge-Based Systems*, *267*, 110440.

David Adams, J. S. S. M., Ryland Morgans, & Williams, M. D. (2013). Successful short passing frequency of defenders differentiates between top and bottom four english premier league teams. *International Journal of Performance Analysis in Sport*, *13*(3), 653–668. doi: 10.1080/24748668.2013.11868678

Dayal, A., Cenkeramaddi, L. R., & Jha, A. (2022). Reward criteria impact on the performance of reinforcement learning agent for autonomous navigation. *Applied Soft Computing*, *126*, 109241.

Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., & Madry, A. (2020). *Implementation matters in deep policy gradients: A case study on ppo and trpo.*

Eschmann, J. (2021). Reward function design in reinforcement learning. In B. Belousov, H. Abdulsamad, P. Klink, S. Parisi, & J. Peters (Eds.), *Reinforcement learning algorithms: Analysis and applications* (pp. 25–33). Cham: Springer International Publishing. doi: 10.1007/978-3-030-41188-6_3

Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., . . . Kavukcuoglu, K. (2018). *Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures.*

Fernández, J., & Bornn, L. (2018, 03). Wide open spaces: A statistical technique for measuring space creation in professional soccer..

Kurach, K., Raichuk, A., Stańczyk, P., Zajac, M., Bachem, O., Espeholt, L., … et al. (2020, Apr). Google research football: A novel reinforcement learning environment. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(04), 4501–4510.

Laud, A. (2004). Theory and application of reward shaping in reinforcement learning..

Liu, B., Pu, Z., Zhang, T., Wang, H., Yi, J., & Mi, J. (2023). Learning to play football from sports domain perspective: A knowledge-embedded deep reinforcement learning framework. *IEEE Transactions on Games*, *15*(4), 648-657.

Liu, J., Niu, Y., Shi, Y., & Zhu, J. (2022). Graph neural network based agent in Google Research Football. In L. Zhu (Ed.), *2nd international conference on artificial intelligence, automation, and high-performance computing (aiahpc 2022)* (Vol. 12348, p. 123483B). SPIE. Retrieved from `https://doi.org/10.1117/12.2641817` doi: 10.1117/12.2641817

Mortensen, J. (2020). *Statistical methods for tracking data in sports.* Simon Fraser University.

Oudeyer, P.-Y., Gottlieb, J., & Lopes, M. (2016). Chapter 11 - intrinsic motivation, curiosity, and learning: Theory and applications in educational technologies. In B. Studer & S. Knecht (Eds.), *Motivation* (Vol. 229, p. 257-284). Elsevier.

Randlov, J., & Alstrøm, P. (1998, Jan). Learning to drive a bicycle using reinforcement learning and shaping. In *Proceedings of the 15th international conference on machine learning* (p. 463-471).

Schulman, J., Moritz, P., Levine, S., Jordan, M., & Abbeel, P. (2018). *High-dimensional continuous control using generalized advantage estimation.*

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *CoRR*, *abs/1707.06347*.

Scott, A., Fujii, K., & Onishi, M. (2021). How does AI play football? an analysis of RL and real-world football strategies. *CoRR*, *abs/2111.12340*.

Scott, D. W. (1992, Aug). Multivariate density estimation. *Wiley Series in Probability and Statistics*. doi: 10.1002/9780470316849

Sehnke, F., Osendorfer, C., Rückstieß, T., Graves, A., Peters, J., & Schmidhuber, J. (2010, May). Parameter-exploring policy gradients. *Neural Networks*, *23*(4), 551–559. doi: 10.1016/j.neunet.2009.12.004

Seungeunrho. (2020). *Google research football competition:liveinparis team.* Retrieved from `https://github.com/seungeunrho/football-paris`

Silver, D., Huang, A., Maddison, C., Guez, A., Sifre, L., Driessche, G., … Hassabis, D. (2016, 01). Mastering the game of go with deep neural networks and tree search. *Nature*, *529*, 484-489. doi: 10.1038/nature16961

Silverman, B. (2018). *Density estimation for statistics and data analysis.* doi: 10.1201/9781315140919

Sivamayil, K., Rajasekar, E., Aljafari, B., Nikolovski, S., Vairavasundaram, S., & Vairavasundaram, I. (2023). A systematic study on reinforcement learning based applications. *Energies*, *16*(3).

Sutton, R. S., & Barto, A. G. (2020). *Reinforcement learning: An introduction.* The MIT Press.

van Otterlo, M., & Wiering, M. (2012). Reinforcement learning and markov decision processes. In M. Wiering & M. van Otterlo (Eds.), *Reinforcement learning: State-of-the-art* (pp. 3–42). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-642-27645-3_1

Yu, C., Velu, A., Vinitsky, E., Gao, J., Wang, Y., Bayen, A., & Wu, Y. (2022). *The surprising effectiveness of ppo in cooperative, multi-agent games.*

Yuan, M., Pun, M.-O., Chen, Y., Wang, D., & Li, H. (2021). Multimodal reward shaping for efficient exploration in reinforcement learning. *CoRR*, *abs/2107.08888*.

# A Appendix

We thank the Center for Information Technology of the University of Groningen for their support and for providing access to the Hábrók high performance computing cluster.

| | | | |
|---|---|---|---|
| Top | Bottom | Left | Right |
| Top-Left | Top-Right | Bottom-Left | Bottom-Right |
| Short Pass | High Pass | Long Pass | Shot |
| Keeper Rush | Sliding | Dribble | Stop-Dribble |
| Sprint | Stop-Moving | Stop-Sprint | Do-Nothing |

**Table A.1: Action Set**