



Interpreting Deep Sports Analytics: Valuing Actions and Players in the NHL

Guiliang Liu^(✉), Wang Zhu, and Oliver Schulte

Simon Fraser University, Burnaby, Canada
{gla68,zhuwangz}@sfu.ca, oschulte@cs.sfu.ca

Abstract. Deep learning has started to have an impact on sports analytics. Several papers have applied action-value Q learning to quantify a team's chance of success, given the current match state. However, the black-box opacity of neural networks prohibits understanding why and when some actions are more valuable than others. This paper applies interpretable Mimic Learning to distill knowledge from the opaque neural net model to a transparent regression tree model. We apply Deep Reinforcement Learning to compute the Q function, and action impact under different game contexts, from 3M play-by-play events in the National Hockey League (NHL). The impact of an action is the change in Q-value due to the action. The play data along with the associated Q functions and impact are fitted by a mimic regression tree. We learn a general mimic regression tree for all players, and player-specific trees. The transparent tree structure facilitates understanding the general action values by feature influence and partial dependence plots, and player's exceptional characteristics by identifying player-specific relevant state regions.

1 Introduction

A fundamental goal of sports statistics is to quantify how much physical player actions contribute to winning in what situation. The advancement of sequential deep learning opens new opportunities of modeling complex sports dynamics, as more and larger play-by-play datasets for sports events become available. Action values based on deep learning are a very recent development, that provides a state-of-the-art player ranking method [12]. Several very recent works [10, 26] have built deep neural networks to model players' actions and value them under different situation. Compared with traditional statistics-based methods for action values [14, 19], deep models support a more comprehensive evaluation because (1) deep neural networks generalize well to different actions and complex game contexts and (2) various network structures (e.g. LSTM) can be applied to model the current game context and its sequential game history.

However, a neural network is an opaque black-box model. It prohibits understanding when or why the player's action is valuable, and which context features are the most influential for this assessment. A promising approach to overcome this limitation is *Mimic Learning* [1], which applies a transparent model to distill

the knowledge from the opaque model to an interpretable data structure. In this work, we train a Deep Reinforcement Learning (DRL) model to learn *action-value functions*, also known as Q functions, which represent a team’s chance of success in a given match context. The *impact* of an action is computed as the difference between two consecutive Q values, before and after the action. To obtain an interpretable model that mimics the Q-value network, we first learn general regression trees for all players, for both Q and impact functions. The results show our trees achieve good mimic performance (small mean square error and variance). To understand the Q functions and impact, we compute the feature importance and use partial dependence plot to analyze the influence of different features with the mimic trees.

To highlight the strengths and weaknesses of an individual player compared to a general player, we construct player-specific mimic trees for Q values and impact. Based on a player-specific tree, we define an interpretable measure for which players are most exceptional overall.

Contribution. The main contributions of our paper are as follows: (1) A Mimic Learning Framework to interpret the action values from a deep neural network model. (2) Both a general mimic model and a player specific mimic model are trained and compared to find the influential features and exceptional players.

The paper is structured as follow: Sect. 2 covers the related work about the player evaluation metrics, Deep Sport Analytics and Interpretable Mimic Learning. Section 3 explains the reinforcement learning model of play dynamics from NHL dataset. Section 4 introduces the procedure of learning the Q values and Impact with DRL model, which completes our review of previous work. We show how to mimic DRL with regression tree in Sects. 5 and 6 discuss the interpretability of Q functions and Impact with Mimic tree. We highlight some exceptional players with the Mimic tree in Sect. 7.

2 Related Work

We discuss the previous work most related to our work.

Player Evaluation Metrics. Numerous metrics have been proposed to measure the players’ performance.

One of the most common is *Plus-Minus* (\pm) [14], which measures how the presence of a player influences the goals of his team. But it considers only goals, and for context only which players are on the ice. *Total Hockey Rating (THoR)* [18] is an alternative metric that evaluates all the actions by whether or not a goal occurred in the following 20s. Using a fixed time window rather makes this approach less useful for low-scoring goals like hockey and soccer. *Expected Possession Value (EPV)* [2] is an alternative metric, developed for basketball, that evaluate all players’ actions by the points that they are expected to score. A POINTWISE Markov model is built to compute the point values with the spatial-temporal tracking data of players’ state and actions. Many recent

works have applied the *Reinforcement Learning (RL)* to compute a Q value to evaluate players actions. [12, 17, 19, 20] built an Markov Decision Model from the sequential video tracking data and applied dynamic programming to learn the Q-functions. Value-above-replacement evaluates how many expected goals or wins the presence of a player adds, compared to a random player, giving rise to the GAR and WAR metrics [7]. Liu and Schulte [7] provide evidence that the Q-value ranking performs better than the GAR and WAR metrics.

Sport Analytics with Deep Models. Modelling sports dynamics with deep sequential neural nets is a rising trend [10, 15]. Dynamical models predict the next event but do not evaluate the expected success from actions, as Q functions do. DRL for learning sports Q functions is a very recent topic [12, 26]. Although these deep models provide an accurate evaluation of player actions, it is hard to understand why the model assigns a large influence to a player in a given situation.

Interpretable Mimic Learning. Complex deep neural networks are hard to interpret. An approach to overcome this limitation is Mimic Learning [1]. Recent works [3, 4] have demonstrated that simple models like shallow feed-forward neural network or decision trees can mimic the function of a deep neural network. Soft outputs are collected by passing inputs to a large, complex and accurate deep neural network. Then we train a mimic model with the same input and soft output as supervisor. The results indicate that training a mimic model with soft output achieves substantial improvement in accuracy and efficiency, over training the same model type directly with hard targets from the dataset.

Table 1. Dataset example

GID	PID	GT	TID	X	Y	MP	GD	Action	OC	P
1365	126	14.3	6	-11.0	25.5	Even	0	Lpr	S	A
1365	126	17.5	6	-23.5	-36.5	Even	0	Carry	S	A
1365	270	17.8	23	14.5	35.5	Even	0	Block	S	A
1365	126	17.8	6	-18.5	-37.0	Even	0	Pass	F	A
1365	609	19.3	23	-28.0	25.5	Even	0	Lpr	S	H
1365	609	19.3	23	-28.0	25.5	Even	0	Pass	S	H

Table 2. Derived features

Velocity	TR	D	Angle	H/A	PN
(-23.4, 1.5)	3585.7	3.4	0.250	A	4
(-4.0, -3.5)	3582.5	3.1	0.314	A	4
(-27.0, -3.0)	3582.2	0.3	0.445	H	4
(0, 0)	3582.2	0.0	0.331	A	4
(-30.3, -7.5)	3580.6	1.5	0.214	H	5
(0, 0)	3580.6	0.0	0.214	H	5

GID = GameId, PID = playerId, GT = GameTime, TID = TeamId, MP = Manpower, GD = Goal Difference, OC = Outcome, S = Succeed, F = Fail, P = Team Possess puck, H = Home, A = Away, TR = Time Remain, PN = Play Number, D = Duration.

3 Play Dynamics in NHL

Dataset. The Q-function approach was originally developed using the publicly available NHL data [17]. Our deep RL model could be applied to this data, but in this paper, we utilize a richer proprietary dataset constructed by SPORT-LOGIQ with computer vision techniques. It provides information about *game events* and *player actions* for the entire 2015–2016 NHL season, which contains

3,382,129 events, covering 30 teams, 1,140 games and 2,233 players. Table 1 shows an excerpt. The data tracks events around the puck, and record the identity and actions of the player, with space and time stamps, and features of the game context. The unit for space stamps are feet and for time stamps seconds. We utilize adjusted spatial coordinates, where negative numbers refer to the defensive zone of the acting player, positive numbers refer to his offensive zone. Adjusted X-coordinates (XAdjcoord) run from -100 to $+100$, Y-coordinates (YAdjcoord) from 42.5 to -42.5 , and the origin is at the ice center. We include data points from all manpower scenarios, not only even-strength, and add the manpower context as a feature. We did not include overtime data. Period information is implicitly represented by game time. We augment the data with derived features in Table 2 and list the complete feature set in Table 3.

Table 3. Complete feature list. Values for the feature Manpower are EV = Even Strength, SH = Short Handed, PP = Power Play.

Name	Type	Range
X Coordinate of puck	Continuous	$[-100, 100]$
Y Coordinate of puck	Continuous	$[-42.5, 42.5]$
Velocity of puck	Continuous	$(-\text{inf}, +\text{inf})$
Time remaining	Continuous	$[0, 3600]$
Score differential	Discrete	$(-\text{inf}, +\text{inf})$
Manpower	Discrete	{EV, SH, PP}
Event duration	Continuous	$[0, +\text{inf})$
Action outcome	Discrete	{successful, failure}
Angle between puck and goal	Continuous	$[-3.14, 3.14]$
Home/Away team	Discrete	{Home, Away}

Reinforcement Learning Model. Our notation for RL concepts follows [17]. There are two agents *Home* resp. *Away* representing the home and away team, respectively. The **reward**, represented by goal vector \mathbf{g}_t , is a 1-of-3 indicator vector that specifies which team scores. For readability, we use *Home*, *Away*, *Neither* to denote the team in a goal vector (e.g. $g_{t,Home} = 1$ means that the home team scores at time t). An **action** a_t is one of 13 types, including shot, assist, etc., with a mark that specifies the team executing the action, e.g. *Shot(Home)*. An observation is a feature vector \mathbf{x}_t for discrete time step t specifies a value for the 10 features listed in Table 3. A **sequence** s_t is a list $(x_0, a_0, \dots, x_t, a_t)$ of observation-action pairs.

We divide NHL games into **goal-scoring episodes**, so that each episode (1) begins at the beginning of the game, or immediately after a goal, and (2) terminates with a goal or the end of the game. We define a Q function to represent the conditional probability of the event that the home resp. away team *scores*

the goal at the end of the current goal-scoring episode (denoted $goal_{Home} = 1$ resp. $goal_{Away} = 1$), or neither team does (denoted $goal_{Neither} = 1$):

$$Q_{team}(s, a) = P(goal_{team} = 1 | s_t = s, a_t = a).$$

4 Q-Values and Action Impact

We review learning Q values and impact, using neural network Q-function approximation. A Tensorflow script is available on-line [13].

4.1 Compute Q Functions with Deep Reinforcement Learning

We apply the on policy Temporal Difference (TD) prediction method Sarsa [22] to estimate $Q_{team}(s, a)$ for current policies π_{home} and π_{away} . The neural network has three fully connected layers connected by a ReLu activation function. The number of input nodes equals the sum of the dimensions of feature vector \mathbf{s} and action vector \mathbf{a} . The number of output nodes is three, including \hat{Q}_{Home} , \hat{Q}_{Away} and $\hat{Q}_{Neither}$, which are normalized to probability. The parameters θ of neural network are updated by minibatch gradient descent with optimization method Adam. Using mean squared error function, the Sarsa Gradient Descent at training step i is based on the square of TD error:

$$\begin{aligned} \mathcal{L}(\theta_i) &= 1/B \sum_t^B (g_t + \hat{Q}(s_{t+1}, a_{t+1}, \theta_i) - \hat{Q}(s_t, a_t, \theta_i))^2 \\ \theta_t &= \theta_t + \alpha \nabla_{\theta} \mathcal{L}(\theta_t) \end{aligned}$$

where B is the batch size and α is the learning rate optimized by the Adam algorithm [8]. For post-hoc interpretability [11] for the learned Q function, we illustrate its temporal and spatial projections in Figs. 1 and 2.

Temporal Projection. Figure 1 plots a value ticker [6] that represents the evolution of the action-value Q function (including Q values for home, away team and neither) from the 3rd period of a randomly selected match between Penguins (Home) and Canadians (Away), Oct.13, 2015. Sports analysts and commentators use ticker plots to highlight critical match events [6]. We mark significant changes in the scoring probabilities and their corresponding events.

Spatial Projection. The neural network generalizes from observed sequences and actions to sequences and actions that have not occurred in our dataset. So we plot the learned smooth value surface $\hat{Q}^{Home}(s_{\ell}, shot(team))$ over the entire rink for home team shots in Fig. 2. Here s_{ℓ} represents the average play history for a shot at location ℓ , which runs in unit steps from $x_{axis} \in [-100, 100]$ and $y_{axis} \in [-42.5, 42.5]$. It can be observed that (1) The chance that the home team scores after a shot is shown to depend on the angle and distance to the goal. (2) Action-value function generalizes to the regions where shots are rarely observed (At the lower or upper corner of the rink).

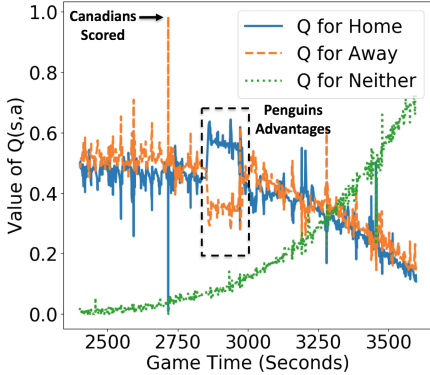


Fig. 1. Temporal projection: evolution of scoring probabilities for the next goal, including the chance that neither team scores another goal.

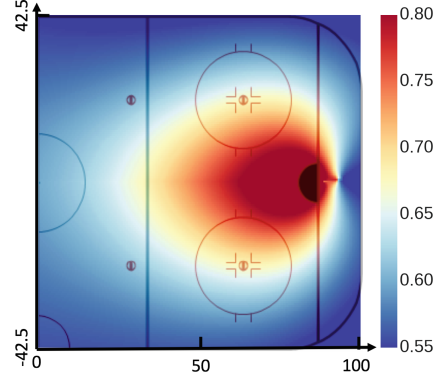


Fig. 2. Spatial projection for the shot action: the probability that the home team scores the next goal after taking a shot at a rink location, averaged over possible game states.

4.2 Evaluate Players with Impact Metric

We follow previous work [12] and evaluate players by how much their actions change the expected return of their team’s in a given game state [12]. This quantity is defined as the **Impact** of an action under current environment (observation) s_t . Players’ overall performance can be estimated by summing the impact of players throughout a game season. The resulting metric is named **Goal Impact Metric (GIM)**.

$$\begin{aligned} \text{impact}^{\text{team}}(s_t, a_t) &= \hat{Q}^{\text{team}}(s_t, a_t) - \hat{Q}^{\text{team}}(s_{t-1}, a_{t-1}) \\ \text{GIM}^i(D) &= \sum_{s,a} n_D^i(s, a) \times \text{impact}^{\text{team}_i}(s, a) \end{aligned}$$

Table 4 shows the top 10 players ranked by GIM. Our purpose in this paper is to interpret the Q values and the impact ranking, not to evaluate them. Previous work provides extensive evaluation [12, 17, 19, 20]. We summarize some of the main points. (1) The impact metric passes the “eye test”. For example the players in Table 4 are well-known top performers. (2) The metric correlates strongly with various quantities of interest in the NHL, including goals, points, Time-on-Ice, and salary. (3) The metric is consistent between and within seasons. (4) The impact is assessed for *all* actions, including defensive and offensive actions. It therefore not biased towards forwards. For instance, defenceman Erik Karlsson appears at the top of the ranking.

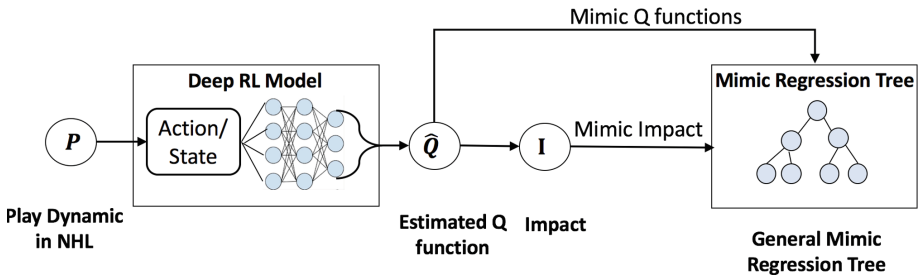
5 Mimicking DRL with Regression Tree

We apply Mimic Learning [1] and train a transparent regression tree to mimic the black-box neural network. As it is shown in Fig. 3, our framework aims at

Table 4. 2015–2016 Top-10 player impact scores

Name	GIM	Assists	Goals	Points	\pm	Age	Team	Salary
Taylor Hall	96.40	39	26	65	-4	24	EDM	\$6,000,000
Joe Pavelski	94.56	40	38	78	25	31	SJS	\$6,000,000
Johnny Gaudreau	94.51	48	30	78	4	22	CGK	\$925,000
Anze Kopitar	94.10	49	25	74	34	28	LAK	\$7,700,000
Erik Karlsson	92.41	66	16	82	-2	25	OTT	\$7,000,000
Patrice Bergeron	92.06	36	32	68	12	30	BOS	\$8,750,000
Mark Scheifele	90.67	32	29	61	16	23	WPG	\$832,500
Sidney Crosby	90.21	49	36	85	19	28	PIT	\$12,000,000
Claude Giroux	89.64	45	22	67	-8	28	PHI	\$9,000,000
Dustin Byfuglien	89.46	34	19	53	4	31	WPG	\$6,000,000

mimicking Q functions and impact. We first train the *general tree model* with the deep model’s input/output for all players and then use it to initialize the *player-specific model* for an individual player (Sect. 7). The transparent tree structure provides much information for understanding the Q functions and impact.

**Fig. 3.** Interpretable Mimic Learning Framework

We focus on two mimicking targets: *Q functions* and *Impact*. For *Q functions*, we fit the mimic tree with the NHL play data and their associated soft outputs (Q values) from our DRL model (neural network). The last 10 observations (determined experimentally) from the sequence are extracted, and CART regression tree learning is applied to fit the soft outputs. This is a multi-output regression task, as our DRL model outputs a Q vector containing three Q values ($\hat{Q}_t = \langle \hat{Q}_t^{home}, \hat{Q}_t^{away}, \hat{Q}_t^{end} \rangle$) for an observation features vectors (s_t) and an action (a_t). A straightforward approach for the multi-target regression problem is training a separate regression model for each Q value. But separate trees for each Q function are somewhat difficult to interpret. An alternative approach to reduce the total tree size is training a Multi-variate Regression Tree (MRTs) [5], which fits all three Q values simultaneously in a regression tree. An MRT can also model the dependencies between the different Q variables [21]. For Impacts, we have only one output ($impact_t$) for each sequence (s_t) and current action (a_t) at time step t .

We examine the mimic performance of regression tree for the Q functions and impact. A common problem of regression trees is over-fitting. We use the Mean Sample Leaf (MSL) to control the minimum number of samples at each leaf node. We apply ten-fold cross validation to measure the performance of our mimic regression tree by Mean Square Error (MSE) and variance. As is shown in Table 5, the tree achieves satisfactory performance when MSL equals 20 (the minimum MSE for Q functions, small MSE and variance for impact).

Table 5. Performance of General Mimic Regression Tree (RT) with different Minimum Samples in each Leaf node (MSL). We apply ten-fold cross validation and report the regression result with format: Mean Square Error (Variance)

Model	Q_home	Q_away	Q_end	Impact
RT-MSL1	3.35E-04(1.43E-09)	3.21E-04(1.26E-09)	1.74E-04(2.18E-09)	1.33E-03(5.43E-09)
RT-MSL5	2.59E-04(1.07E-09)	2.51E-04(0.89E-09)	1.35E-04(1.87E-10)	9.84E-04(2.72E-09)
RT-MSL10	2.38E-04(1.02E-09)	2.30E-04(0.89E-09)	1.25E-04(2.30E-10)	8.66E-04(2.17E-09)
RT-MSL20	2.31E-04(0.92E-09)	2.22E-04(0.82E-09)	1.23E-04(2.05E-10)	7.92E-04(1.45E-09)
RT-MSL30	2.35E-04(0.98E-09)	2.27E-04(0.85E-09)	1.27E-04(2.32E-10)	7.67E-04(1.16E-09)
RT-MSL40	2.39E-04(0.96E-09)	2.30E-04(0.85E-09)	1.29E-04(2.19E-10)	7.58E-04(1.10E-09)

6 Interpreting Q Functions and Impact with Mimic Tree

We now show how to interpret Q functions and Impact using the general Mimic tree, by deriving feature importance and a partial dependence plot.

6.1 Compute Feature Importance

In CART regression tree learning, variance reduction is the criterion for evaluating the quality of a split. Therefore we compute the importance of a target feature by summing the variance reductions at each split using the target feature [3]. We list the top 10 important features in the mimic tree for Q values and impact in Table 6. The frequency of a feature is the number of times the tree splits on the feature. The notation $T - n : f$ indicates that a feature occurs n time steps before the current time. We find that the Q and impact functions agree on nearly half of the features, but their importance values differ. For Q values, time remaining is the most influence features with significantly larger importance value than other. This is because less time means fewer chance of any goals (see Fig. 1). But for impact, time remaining is much less important, because impact is the difference of consecutive Q values, which cancels the time effect and focuses only on the influence of a player’s action a : Near the end of the match, players still have a chance to make actions with high impact. The top three important features for impact are (1) Goal: if the player scores a goal. (2) Shot-on-Goal Outcome: if the player’s shot is on target (3) X Coordinate: the x -location of the puck (goal-to-goal axis). Thus the impact function recognizes

players for shooting, successful actions, and for advancing the puck towards the goal of their opponent. A less intuitive finding is that the duration of an action affects its impact. Notice that for both Q values and impact, the top ten important features contain historical features (with $T - n$ for $n > 0$), which supports the importance of including historical data in observation sequence s .

Table 6. Top 10 features for Q values (left) and Impact (right). The notation $T - n : f$ indicates that a feature occurs n time steps before the current time.

Feature Name	Frequency	Importance	Feature Name	Frequency	Importance
T: Time Remaining	12,524	0.817431	T: Goal	1	0.160595
T-1: Manpower	93	0.070196	T: Shot-on-Goal	1	0.099482
T-1: Team Identifier	57	0.020504	T: X Coordinate	7,142	0.077410
T: Manpower	346	0.017306	T: X Velocity	8,087	0.041903
T: Shot	31	0.011159	T-1: X Coordinate	3,591	0.041847
T: Score Differential	3,229	0.009568	T: Angle to Goal	7,525	0.041607
T: X Coordinate	11,797	0.006968	T: Time Remaining	8,669	0.036289
T-1: X Coordinate	3,406	0.006963	T: Duration	7,411	0.028831
T-2: Manpower	82	0.005045	T: Home/Away Team	378	0.027177
T: Home/Away Team	135	0.003755	T: Y Coordinate	6,890	0.027597

6.2 Draw Partial Dependence Plot

A partial dependence plot is a common visualization to determine qualitatively what a model has learned and thus provides interpretability [3, 11]. The plot approximates the prediction function for a single target feature, by marginalizing over the values of all other features. We select X Coordinate (of puck), Time Remaining and X Velocity (of puck), three continuous features with high importance for both the Q and the impact mimic tree. As it is shown in Fig. 4, Time Remaining has significant influence on Q values but very limited effect on impact. This is consistent with our findings for feature importance. For X Coordinate, as a team is likely to score the next goal in the offensive zone, both Q values and impact increase significantly when the puck is approaching its opponent's goal (larger X Coordinate). And compared to the position of the puck, velocity in X-axis has limited influence on Q values but it does affect the impact. This shows that the impact function uses speed on the ice as an important criterion for valuing a player. We also observe the phenomenon of home advantage [23] as the Q value (scoring probability) of the home team is slightly higher than that of the away team.

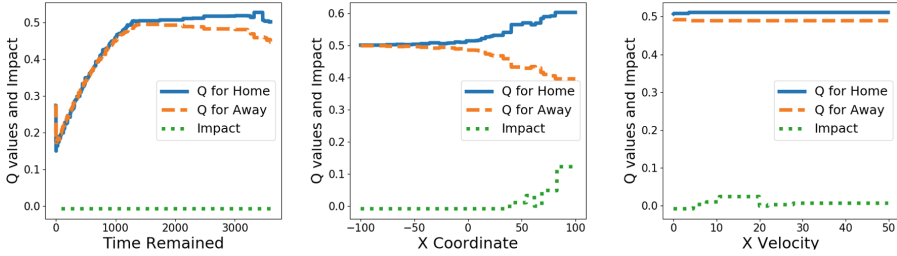


Fig. 4. Partial dependence plot for Time Remaining (left), X Coordinate (middle) and X Velocity (right)

7 Highlighting Exceptional Players

Our approach to quantifying which players are exceptional is based on a partition the continuous state space into a discrete set of m disjoint regions. Given a Q or impact function, exceptional players can be found by region-wise comparison of a player’s expected impact to that of a random player’s. For a specific player, this comparison highlights match settings in which the player is especially strong or weak. The formal details are as follows.

Let n_D be the number of actions by player P , of which n_ℓ fall into discrete state region $\ell = 1, \dots, m$. For a function f , let \hat{f}_ℓ be the value of f estimated from *all* data points that fall into region ℓ , and let \hat{f}_ℓ^P be the value of f estimated from the n_ℓ data points for region ℓ and player P . Then the weighted squared f -difference is given by:

$$\sum_{\ell} n_{\ell}/n_D (\hat{f}_{\ell} - \hat{f}_{\ell}^P)^2. \quad (1)$$

Regression trees provide an effective way to discretize a Q-function for a continuous state space [25]: Each leaf forms a partition cell in state space (constructed by the splits with various features along the path from root to the leaf). The regression trees described in Sect. 5 could be used, but they represent *general* discretizations learned for all the players over a game season, which means that they may miss distinctions that are important for a specific player. For example, if an individual player is especially effective in the neutral zone, but the average player’s performance is not special in the neutral zone, the generic tree will not split on “neutral zone” and therefore will not be able to capture the individual’s special performance. Therefore we learn for each player, a *player-specific* regression tree.

The General Tree is learned with all the inputs and their corresponding Q or Impact values (soft labels). The Player Tree is initialized with the General Tree and then fitted with the n_D datapoints of a specific player P and their corresponding Q values ($f_Q^P(f_{\hat{Q}}, s_t^P, a_t^P) \rightarrow \text{range}(\hat{Q}_t^P)$) or Impact values ($f_I^P(f_I, s_t^P, a_t^P) \rightarrow \text{range}(\text{Impact}_t^P)$). It inherits the tree structure of the general

model RT-MSL20 in Sect. 5, uses the target player data to prune the general tree, then expands the tree with further splits. Initializing with the general tree assumes players share relevant features and prevents over-fitting to a player’s specific data. A Player Tree defines a discrete set of state regions, so we can apply Eq. 1 with the Q or impact functions. Table 7 shows the weighted squared differences for the top 5 players in the GIM metric.

Table 7. Exceptional players based on tree discretization

Player	Q_home	Q_away	Q_end	Impact
Taylor Hall	1.80E-04	2.49E-04	2.28E-04	6.66E-05
Joe Pavelski	4.64E-04	2.90E-04	3.04E-04	1.09E-04
Johnny Gaudreau	2.12E-04	1.96E-04	1.43E-04	6.77E-05
Anze Kopitar	2.58E-04	2.00E-04	2.43E-04	8.28E-05
Erik Karlsson	2.97E-04	1.89E-04	1.86E-04	2.00E-04

We find that (1) Joe Pavelski, who scored the most in the 2015–2016 game season, has the largest Q values difference and (2) Erik Karlsson, who had the most points (goal+assists), has the largest Impact difference. They are the two players who differ the most from the average players by Q-value and Impact.

8 Conclusion and Future Work

This paper applies Mimic Learning to understand the Q function and impact from Deep Reinforcement Learning Model in valuing actions and players. To study the influence of a feature, we analyze a general mimic model for all players by feature importance and partially dependence plot. For individual players, performance in state regions defined by the player specific tree is implemented to find exceptional players. With our interpretable Mimic Learning, coaches and fans can understand what the deep models have learned and thus trust the results. While our evaluation focuses on ice hockey, our techniques apply to other continuous-flow sports such as soccer and basketball.

In *future work*, the player trees can be used to highlight match scenarios where a player shows exceptionally strong or weak performance, in both defense and offense. A limitation of our current model is that it pools all data from the different teams, rather than modelling the differences among teams. A hierarchical model for ice hockey can be used to analyze how teams are similar and how they are different, like those that have been built for other sports (e.g., cricket [16].) Another limitation is that players get credit only for recorded individual actions. An influential approach to extend credit to all players on the rink has been based on regression [9, 14, 24]. A promising direction for future work is to combine Q-values with regression.

References

1. Ba, J., Caruana, R.: Do deep nets really need to be deep? In: *Advances in Neural Information Processing Systems*, pp. 2654–2662 (2014)
2. Cervone, D., D’Amour, A., Bornn, L., Goldsberry, K.: Pointwise: predicting points and valuing decisions in real time with NBA optical tracking data. In: *Proceedings of the 8th MIT Sloan Sports Analytics Conference*, Boston, MA, USA, vol. 28, p. 3 (2014)
3. Che, Z., et al.: Interpretable deep models for ICU outcome prediction. In: *AMIA Annual Symposium Proceedings*, vol. 2016, p. 371. AMIA (2016)
4. Dancey, D., Bandar, Z.A., McLean, D.: Logistic model tree extraction from artificial neural networks. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **37**(4), 794–802 (2007)
5. De’Ath, G.: Multivariate regression trees: a new technique for modeling species-environment relationships. *Ecology* **83**(4), 1105–1117 (2002)
6. Decroos, T., Dzyuba, V., Van Haaren, J., Davis, J.: Predicting soccer highlights from spatio-temporal match event streams. In: *AAAI*, pp. 1302–1308 (2017)
7. Gerstenberg, T., Ullman, T., Kleiman-Weiner, M., Lagnado, D., Tenenbaum, J.: Wins above replacement: Responsibility attributions as counterfactual replacements. In: *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 36 (2014)
8. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 315–323 (2011)
9. Kharrat, T., Pena, J.L., McHale, I.: Plus-minus player ratings for soccer. arXiv preprint [arXiv:1706.04943](https://arxiv.org/abs/1706.04943) (2017)
10. Le, H.M., Carr, P., Yue, Y., Lucey, P.: Data-driven ghosting using deep imitation learning. In: *MIT Sloan Sports Analytics Conference* (2017)
11. Lipton, Z.C.: The myths of model interpretability. arXiv preprint [arXiv:1606.03490](https://arxiv.org/abs/1606.03490) (2016)
12. Liu, G., Schulte, O.: Deep reinforcement learning in ice hockey for context-aware player evaluation. In: *Proceedings IJCAI-18*, pp. 3442–3448, July 2018. <https://doi.org/10.24963/ijcai.2018/478>
13. Liu, G., Schulte, O.: Drl-ice-hockey (2018). <https://github.com/Guiliang/DRL-ice-hockey>
14. Macdonald, B.: A regression-based adjusted plus-minus statistic for NHL players. *J. Quant. Anal. Sports* **7**(3), 29 (2011)
15. Mehra, N., Zhong, Y., Tung, F., Bornn, L., Mori, G.: Deep learning of player trajectory representations for team activity analysis. In: *MIT Sloan Sports Analytics Conference* (2018)
16. Perera, H., Davis, J., Swartz, T.: Assessing the impact of fielding in twenty20 cricket. *J. Oper. Res. Soc.* **69**, 1335–1343 (2018)
17. Routley, K., Schulte, O.: A Markov game model for valuing player actions in ice hockey. In: *Proceedings Uncertainty in Artificial Intelligence (UAI)*, pp. 782–791 (2015)
18. Schuckers, M., Curro, J.: Total hockey rating (THoR): a comprehensive statistical rating of national hockey league forwards and defensemen based upon all on-ice events. In: *7th Annual MIT Sloan Sports Analytics Conference* (2013)
19. Schulte, O., Khademi, M., Gholami, S., Zhao, Z., Javan, M., Desaulniers, P.: A Markov game model for valuing actions, locations, and team performance in ice hockey. *Data Min. Knowl. Discovery* **31**(6), 1735–1757 (2017)

20. Schulte, O., Zhao, Z., Javan, M., Desaulniers, P.: Apples-to-apples: clustering and ranking NHL players using location information and scoring impact. In: MIT Sloan Sports Analytics Conference (2017)
21. Struyf, J., Džeroski, S.: Constraint based induction of multi-objective regression trees. In: Bonchi, F., Boulicaut, J.-F. (eds.) KDID 2005. LNCS, vol. 3933, pp. 222–233. Springer, Heidelberg (2006). https://doi.org/10.1007/11733492_13
22. Sutton, R.S., Barto, A.G.: Introduction to Reinforcement Learning, vol. 135. MIT Press, Cambridge (1998)
23. Swartz, T.B., Arce, A.: New insights involving the home team advantage. *Int. J. Sports Sci. Coaching* **9**(4), 681–692 (2014)
24. Thomas, A., Ventura, S., Jensen, S., Ma, S.: Competing process hazard function models for player ratings in ice hockey. *Ann. Appl. Stat.* **7**(3), 1497–1524 (2013). <https://doi.org/10.1214/13-AOAS646>
25. Uther, W.T., Veloso, M.M.: Tree based discretization for continuous state space reinforcement learning. In: AAAI/IAAI, pp. 769–774 (1998)
26. Wang, J., Fox, I., Skaza, J., Linck, N., Singh, S., Wiens, J.: The advantage of doubling: a deep reinforcement learning approach to studying the double team in the NBA. In: MIT Sloan Sports Analytics Conference (2018)