Uncertainty-Aware Reinforcement Learning for Risk-Sensitive Player Evaluation in Sports Game

Anonymous Author(s)
Affiliation
Address
email

Abstract

A major task of sports analytics is player evaluation. Previous methods commonly measured the impact of players' actions on desirable outcomes (e.g., goals or winning) without considering the risk induced by stochastic game dynamics. In this paper, we design an uncertainty-aware Reinforcement Learning (RL) framework to learn a risk-sensitive player evaluation metric from stochastic game dynamics. To embed the risk of a player's movements into the distribution of action-values, we model their 1) *aleatoric uncertainty*, which represents the intrinsic stochasticity in a sports game, and 2) *epistemic uncertainty*, which is due to a model's insufficient knowledge regarding Out-of-Distribution (OoD) samples. We demonstrate how a distributional Bellman operator and a feature-space density model can capture these uncertainties. Based on such uncertainty estimation, we propose a Risk-sensitive Game Impact Metric (RiGIM) that measures players' performance over a season by conditioning on a specific confidence level. Empirical evaluation, based on over 9M play-by-play ice hockey and soccer events, shows that RiGIM correlates highly with standard success measures and has a consistent risk sensitivity.

1 Introduction

The advancement of player tracking and object detection systems enables data-driven analytics for professional sports players. A common approach to evaluating the contribution of players is to quantify their action impacts. Previous performance metrics [1, 2, 3, 4] computed the expected impact of an action on scoring or winning a game. However, actions with significantly different distributions of impact can have the same expectations. As a result, the expectation-based metrics cannot differentiate the risk-seeking actions from the risk-averse ones. How to distinguish these actions and assign proper credits to the players remains a fundamental challenge in sports analytics.

An important step toward a risk-sensitive evaluation metric is to model the distributions of action values. To achieve this goal, distributional Reinforcement Learning (RL) [5] predicts the supporting quantiles of action-value distributions. Previous distributional RL methods [6, 7, 8, 9, 10] mainly studied the virtual environments with deterministic transitions (e.g., Atari [11] or Mujoco [12]), whereas sports games are real environments with stochastic game dynamics and complex context features. Moreover, player evaluation, as a data-driven task, requires learning from a fixed dataset without exploration. The model must be able to handle the distribution shift experienced at test time.

To mitigate the impact of distribution shift, Offline RL algorithms [13] typically strive for conservative policies that discourage visits to Out-of-Distribution (OoD) states by penalizing corresponding values. However, this approach cannot be scaled to player evaluation since 1) our goal is not controlling (i.e., improving players' policy), but using RL as an analytical tool to evaluate observed actions in professional games, and 2) penalizing actions in OoD states distorts the evaluation.

In this paper, we design an uncertainty-aware RL framework for risk-sensitive player evaluation. Figure 1 shows a real-world example introducing our key motivations. Instead of directly influencing

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

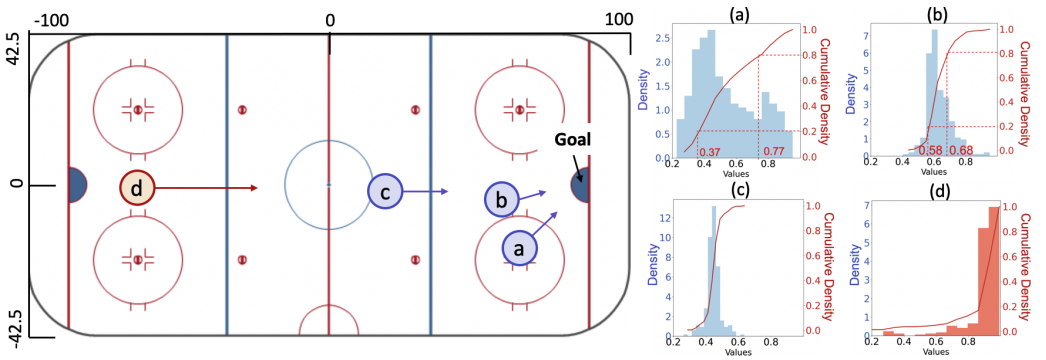

Figure 1: The predicted distribution of future goals in an ice hockey game between Blues and Coyotes, 2018-19 NHL season. The shots are made in the positions (a) - (d), providing important *motivations* for 1) **risk-sensitive evaluation**: Distributions (a) and (b) have *the same expectation* (around 0.6), but the first shot has a larger risk-averse estimate (at the confidence 0.8, we find $0.58 > 0.37$) and a smaller risk-seeking estimate (at the confidence 0.2, we find $0.68 < 0.77$), and thus they have *different impact on risk-sensitive evaluation*. 2) **Post-hoc calibration**: the event of shooting from the position (d) (the back-court) is rare in an ice hockey game, and thus this event is likely to be OoD, leading to a biased prediction at (d) (the predicted scoring chances are too large).

players' actions like other RL algorithms, we perform a post-hoc calibration of the learned action values. The main idea of our framework is to model important types of uncertainty in sports games:

1) *Aleatoric uncertainty* captures the intrinsic stochasticity of game dynamics caused by stochastic rewards, transition dynamics, and policies. We show that this stochasticity can be captured by a distributional Bellman operator and propagated between action-value distributions by implementing Temporal-Difference (TD) learning with distributional RL.

2) *Epistemic uncertainty* is due to the finite training samples and OoD state-action pairs during testing. Online RL algorithms can overcome this uncertainty given sufficient exploration in the environment [8]. However, when we have only a demonstration dataset with limited samples, the influence of epistemic uncertainty cannot be ignored. Striving for simplicity and scalability, we model the epistemic uncertainty with a Feature-Space Conditional Normalizing Flow (FS-CNF).

Based on the uncertainty estimations, we develop a Risk-sensitive Game Impact Metric (RiGIM) for player evaluation. RiGIM filters the predictions for OoD samples and computes the impact of players' actions by conditioning on a confidence level. Empirical evaluation shows that RiGIM is highly correlated with standard measures when compared to other baselines. We measure the accuracy of action-value predictions by matching them empirically with game results and evaluate the risk-sensitivity of RiGIM by its correlations with standard measures at different confidence levels.

Contributions. 1) We design an uncertainty-aware RL framework that enables post-hoc calibrations on action values according to their uncertainties. 2) We demonstrate how the distributional Bellman operator captures the aleatoric uncertainty with action-value distributions from both a theoretical and an empirical perspective. 3) To the best of our knowledge, RiGIM is the first risk-sensitive metric for player evaluation. Although this work mainly focus professional sports games, our method is general and can be scaled to other stochastic environments or domains.

2 Related Works

In this section, we introduce previous works that are most related to our approach.

Uncertainty Estimation for RL. Uncertainty estimates have been widely used in RL for guiding exploration and stabilizing policies. To achieve this goal, an effective approach is to measure the uncertainty of *future returns*: [14] designed an uncertainty Bellman equation that estimates the variance of the Q-value posterior distributions. Distributional RL methods [6, 7, 8, 9, 10, 15, 16, 17] directly model the distribution of future returns by computing corresponding quantities. Bootstrapped DQN methods [18, 19, 20, 21] learn ensembles of action-value Q functions to capture uncertainty. Some following works [22, 23] extend the Q-ensemble methods to offline RL settings by learning from a fixed dataset. Instead of focusing on the returns' uncertainty, an alternative approach is to

measure the uncertainty of *model dynamics*: [24, 25] proposed model-based RL approaches that predict the uncertainty of dynamics models and penalize the actions leading to uncertain returns. Instead of separately capturing the epistemic and aleatoric uncertainties, these methods often quantify the overall uncertainty with a unified measure (e.g., variance or entropy).

Another line of approaches that heavily relies on uncertainty estimates is Risk-Sensitive RL (RSRL) [26]. The RSRL agents avoid states with high costs by estimating a risk measure (e.g., variance, Value at Risk (VaR), or Conditional VaR [27]). These algorithms [28, 29] often learn a controlling policy based on a known MDP instead of an offline dataset (with unknown dynamics).

Player Evaluation. The most common approach to player evaluation is to quantify the impact of their actions on game results [30]. Previous works measured action impacts by predicting 1) whether a goal will be scored within a fixed look-ahead horizon [3], 2) the change of winning chances [31], and 3) the expected number of goals within a possession [32]. Some recent works also trained action-value Q-functions by dynamic programming [1], deep Sarsa [2, 33] and Inverse RL [4]. These methods compute an expected action value without modeling their potential risk, and they commonly assume the training and testing datasets are identically distributed.

3 Uncertainty-Aware RL framework for Player Evaluation

We represent the dynamics in sports games with a Markov Game model and introduce the motivation of estimating the aleatoric uncertainty and the epistemic uncertainty.

3.1 Finite-Horizon Markov Game Model

Player evaluation metrics commonly evaluate players by how much their actions influence the opportunity of scoring the next goal [2, 3, 34]. Following this setting, we divide a sports game into *goal-scoring episodes*, so that each episode 1) begins immediately after a goal (or at the beginning of the game), and 2) terminates when the next goal is scored (or the end of the game is reached). From an algorithmic perspective, this setting allows us to bound the support of future-goals distribution (Section 4.1) into $[0, 1]$, which leads to faster model convergence and more accurate evaluations.

For a scoring episode of length T_H, we model its dynamics with a finite-horizon Markov game model [35]: $G = (\mathcal{S}, \boldsymbol{\mathcal{A}}, P_{\mathcal{T}}, \boldsymbol{R}, \mathcal{O}, T_H, \gamma)$. At a time step $t \in [0, T_H]$, an agent k performs an action $a_{k,t} \in \mathcal{A}_k$ at a game state $s_t \in \mathcal{S}$ after receiving an observation $o_t \in \mathcal{O}$. This process generates the next state $s_{t+1} \sim P_{\mathcal{T}}(\cdot|s_t, a_t)$ and a reward $r_{k,t} = R_k(s_t, a_{k,t})$. γ is a discount factor. In this paper, we consider two agents $k \in \{Home, Away\}$ representing the home and away teams. The observed data $\mathcal{D} = [(o_1, a_{k,1}, \boldsymbol{r}_1), (o_2, a_{k,2}, \boldsymbol{r}_2), \ldots, (o_t, a_{k,t}, \boldsymbol{r}_t), \ldots]$ records the action $a_{k,t}$ performed by the team k who possesses the puck. To alleviate the partial observability, a game state includes the game history: $s_t := (o_t, a_{t-1}, o_{t-1}, \ldots, o_0)$. The reward $\boldsymbol{r}_t$ is a 1-of-2 indicator vector that specifies which team ($Home$, $Away$) scores. We assign zeros to $\boldsymbol{r}_t$ until a team scores at the end of an episode.

3.2 Uncertainty-Aware RL for Player Evaluation

Learning from Offline Data. As a data-driven behavior analytic tool, the player evaluation model assigns values to players' actions by learning from an offline dataset. Under this setting, previous works [1, 2, 33, 3] commonly assumed the training and testing datasets are sampled from the same underlying distribution. However, in practice, since the behaviors of players may change when some team members (especially core players or the coach) are traded or hired during a season, there is no guarantee that the visitation frequency of state-action pairs is consistent in different games. It is natural to assume a distributional shift between the games in the training and testing dataset: while the value function is trained under one distribution, it will be evaluated on a different distribution.

Calibration with Uncertainty. To alleviate the influence of distribution shift, offline RL algorithms [13] commonly discourage the visit to OoD state-action pairs by lowering their values [36] or penalising their rewards [24] or constraining the updated policy [37, 22]. However, to evaluate players' performance, RL is employed as a policy evaluation tool instead of controlling players. The trajectories in testing games correspond to observed players' actions that cannot be changed or influenced by penalties. Instead, we perform a post-hoc calibration of the predicted action values by modeling their *epistemic uncertainty*, which is due to a lack of knowledge about OoD samples, and thus the resulting model is uncertain about the returns (Section 4.2). Since our goal is to develop a risk-sensitive player evaluation metric, we estimate distributions of action values to model their

aleatoric uncertainty, which is due to the intrinsic stochasticity in the game dynamics (Section 4.1). In practice, the quantification of aleatoric uncertainty can be influenced by the epistemic uncertainty of input samples, so we filter the OoD samples by utilizing our density estimator (Section 5.1) .

4 Modelling the Uncertainty of Action Values

In this section, we introduce our method of modeling the aleatoric and epistemic uncertainty with a distributional-RL model and a feature-space density estimator (Figure 2).

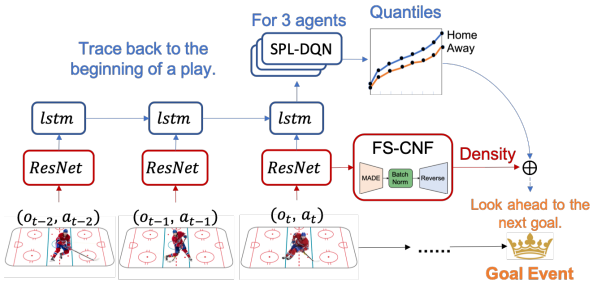

Figure 2: Model architecture. A play is a turn where one team attacks and the other defends. We add Spectral Normalization to ResNet outputs.

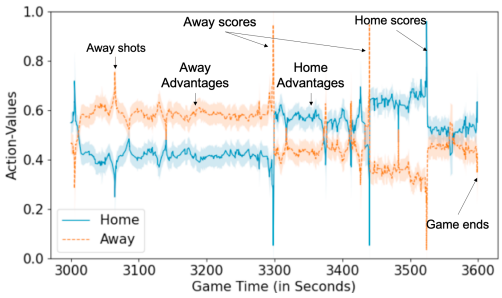

Figure 3: Illustrating the predicted distributions by showing the corresponding mean $\pm$ standard deviation at each time step in a sports game.

4.1 Distributional RL for Capturing Aleatoric Uncertainty

Distributional RL learns the distribution of the random variable $Z_k(s_t, a_t)$ that corresponds to the number of future goals when a player of team k performs action a_t in state s_t. In other words, we can think of $Z_k(s_t, a_t)$ as a random variable with outcomes corresponding to the sum of discounted rewards $\sum_{\iota=t}^{T_H} \gamma^\iota R_{k,\iota}(S_\iota, A_\iota)$, where $S_\iota = s_\iota$, $A_\iota = a_\iota$, $S_{\iota+1}$ is distributed according to $P_{\mathcal{T}}(\cdot|S_\iota, A_\iota)$ and A_ι is distributed according to $\pi(\cdot|S_\iota)$. Following the Quantile-Regression (QR)-DQN method [6], we represent the distribution of Z by a uniform mixture of N supporting quantiles by $\hat{Z}_k(s_t, a_t) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\theta_{k,i}(s_t,a_t)}$, where $\theta_{k,i}$ estimates the quantile at the quantile level (or quantile index) $\hat{\tau}_i = \frac{\tau_{i-1}+\tau_i}{2}$ ($1 \leq i \leq N$, and $\tau_i = i/N$) and $\delta_{\theta_{k,i}}$ denotes a Dirac distribution at $\theta_{k,i}$. The model outputs $[\theta_{k,1}, \ldots, \theta_{k,N}]$ are monotonically increasing quantile values computed with the spline DQN (SPL-DQN) by following [17].

Distributional Bellman Operator. When the player of a team k performs an action a_t at a state s_t, the agent receives a reward $R_k(s_t, a_t)$ and moves to a future state $s_{t+1} \sim P_{\mathcal{T}}(S_{t+1}|s_t, a_t)$ where the agent's next action $a_{t+1} \sim \pi(A_{t+1}|S_{t+1})$. This stochastic process can be captured by a distributional Bellman operator $\mathcal{T}^\pi$ [6]:

$$\mathcal{T}^\pi Z_k(s_t, a_t) :\stackrel{\Delta}{=} R_k(s_t, a_t) + \gamma Z_k(S_{t+1}, A_{t+1}) \tag{1}$$

where $X :\stackrel{\Delta}{=} Y$ indicates that random variables X and Y follow the same distribution. Based on the distributional Bellman operator, we estimate the supporting quantiles of Z by minimizing the quantile Huber loss (with threshold η):

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{i'=1}^{N} \rho^\eta_{\hat{\tau}_i}(r + \gamma\theta_{i',k}(s_{t+1}, a_{t+1}) - \theta_{i,k}(s_t, a_t)) \text{ where}$$
$$\rho^\eta_\tau(\sigma) = |\tau - \mathbb{I}_{\sigma<0}|\mathcal{L}_\eta(\sigma) \;\text{ with } \mathcal{L}_\eta(\sigma) = \begin{cases} \frac{1}{2}\sigma^2, & |\sigma| \leq \eta \\ \eta(|\sigma| - \frac{1}{2}\eta), & \text{otherwise.} \end{cases} \tag{2}$$

Illustration of Temporal Projection. Figure 3 illustrates the mean $\pm$ standard deviation of the action-values sampled from the predicted distributions $\hat{Z}(s, a)$, where s and a follow the players' movements in a match between the Flyers (Home team) and the Maple Leafs (Away team) on March 15, 2019. The figure plots values of the two output nodes. We highlight critical events and match contexts to show the context-sensitivity of our predictions.

We show that the predicted distribution of action values can measure the aleatoric uncertainty.

Proposition 1. *Assume the Bellman consistency holds by $\hat{\boldsymbol{Z}} \overset{\Delta}{:=} \boldsymbol{R} + \gamma \boldsymbol{P}^{\pi} \hat{\boldsymbol{Z}}$ where $\hat{\boldsymbol{Z}}$, $\boldsymbol{R}$ are vector-valued random variables and $\boldsymbol{P}^{\pi}$ is the transition matrix of the stationary policy π, so $P^{\pi}_{(s,a),(s',a')} = P(s'|a,s)\pi(a'|s')$, the uncertainty of action-value distributions $\hat{\boldsymbol{Z}}$ under an entropy measure $H(\cdot)$ can be given by:*

$$H(\hat{\boldsymbol{Z}}) = H[\boldsymbol{R}] - |\mathcal{A}||\mathcal{S}| \log(1-\gamma) + \log |det(\mathbf{d}^{\pi})| \tag{3}$$

where $\mathbf{d}^{\pi} = (1-\gamma)(I - \gamma \boldsymbol{P}^{\pi})^{-1} \in [0,1]^{|S||A| \times |S||A|}$ is the induced matrix for distributions over state-action tuples by following policy π and transition $P_{\mathcal{T}}$.

The proof is in Appendix B.1. Proposition 1 disentangles the entropy of $\boldsymbol{Z}$ into 1) the entropy of reward variables that quantifies the uncertainty of current rewards, 2) the uncertainty induced by the discount factor, which determines how much the current uncertainty estimation should be influenced by the stochasticity of future rewards or transitions (i.e., a small γ reduces this influence), and 3) a log-absolute determinant of the induced distribution matrix, which measures the amount of stretch or change that the transition function $P_{\mathcal{T}}$ and the policy π apply to the initial state-action distribution.

Proposition 1 suggests that the key components for representing the aleatoric uncertainty can be captured by Z when the Bellman consistency is reached by learning. However, in practice, the estimation of Z cannot be well generalized to all samples because of insufficient exploration or limited training data, so we need to estimate their epistemic uncertainty.

4.2 Density Estimator for Capturing Epistemic Uncertainty

By definition, epistemic uncertainty stems from limited training data and is inherent to the model fitting these data. A common measure of epistemic uncertainty is $I(\theta; y|x, \mathcal{D})$ [38, 39, 40]: the amount of information gained when the model θ observes the true label y of an input x. To estimate this uncertainty measure, previous works [41, 42, 43] utilized deep ensemble models and treated each ensemble as a sample from the posterior $p(\theta|\mathcal{D})$. However, adding additional deep ensemble layers to a distributional RL model (i.e., as a *unified* estimator for the joint distribution $p(Z_{1,\ldots,K}, \Theta|\mathcal{D})$) significantly increases the model complexity. Striving for model simplicity and scalability for large datasets, we build a feature space density estimator [40] to detect OoD samples.

In this work, we design a Feature Space Conditional Normalizing Flow (FS-CNF) to estimate sample density in the training distribution with a minimum overhead. The main components are:

Feature Extractor. FS-CNF shares the same feature extracting layers with the distributional RL models. Note that a common reason why traditional feature extractors might fail to capture epistemic uncertainty is feature collapse [44], which maps OoD samples to iD regions in feature space. To prevent this phenomenon, an ideal feature extractor f_{θ} must be subjected to a bi-Lipschitz constraint:

$$\beta_1 \|x_1 - x_2\|_I \geq \|f_{\theta}(x_1) - f_{\theta}(x_2)\|_F \leq \beta_2 \|x_1 - x_2\|_I \; \forall x_1, x_2 \in \mathcal{D} \tag{4}$$

where 1) the lower Lipschitz bound ensures sensitivity to distances in the input space (i.e., sensitive to OoD samples) and 2) the upper Lipschitz bound ensures smoothness in the feature space (i.e., prevents overfitting to the input variations). To ensure this bi-Lipschitz condition in practice, we follow [44] and use residual networks with spectral normalisation as our feature extractor.

Density Estimator. Based on the extracted features, FS-CNF utilizes the Masked Auto-regressive Flow (MAF) [45] design that estimates the density of input variables in the training data distribution with an auto-regressive constraint. Additionally, to make FS-CNF more sensitive to the abnormal predictions in OoD samples, the density model is conditioned on the expected returns of action-value distributions, so $p(\boldsymbol{x}|\boldsymbol{z}_E) = \sum_i p(x_i|x_{1:i-1}, \boldsymbol{z}_E)$ where $\boldsymbol{z}_E := \{\mathbb{E}[Z_k(s,a)]\}_{k=1}^{K}$ and $\boldsymbol{x} := (o, a)$ (we use o instead of s since a large input dimension influences the accuracy of estimation). We implement $p(x_i|x_{1:i-1}, \boldsymbol{z}_E) = \mathcal{N}(x_i|\mu_i, (\exp(\alpha_i))^2)$ where $\mu_i = \psi_{\mu_i}(x_{1:i-1}, \boldsymbol{z}_E)$ and $\alpha_i = \psi_{\alpha_i}(x_{1:i-1}, \boldsymbol{z}_E)$. The neural function ψ is implemented by stacking multiple MADE layers [46].

5 Player Evaluation

In this section, we introduce our player evaluation metric and risk-sensitive rankings.

5.1 Risk-sensitive Impact Metric

Measuring Risk. We measure the risk of a player's movement with the aleatoric uncertainty since modeling the intrinsic stochasticity of the game dynamics is consistent with the goal of sports analytics. However, as Figure 4 shows, when we quantify the aleatoric uncertainty with predictions from distributional RL, the performance is influenced by the density of input samples (i.e., OoD samples have a lower accuracy).

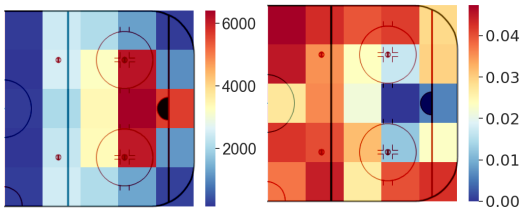

Figure 4: We discretize the ice hockey rink into 5×5 regions. For each region, the *left* heatmap shows the number of shots in the *training dataset*, and the *right* heatmap shows the Mean Absolute Error (MAE) between the estimated and the real aleatoric uncertainty for shot values in the *testing dataset*. We observe a negative correlation (-0.761) between the density and the MAE across these regions.

An effective approach to verify whether $\hat{Z}_k(s,a)$ accurately captures the true aleatoric uncertainty is to check the epistemic uncertainty for each input data (s,a) [40]: 1) a high input density ($p(\cdot|\boldsymbol{z}_E) \geq \epsilon$) indicates low epistemic uncertainty, (i.e., (s,a) is inD) and we can trust the aleatoric uncertainty estimated by distributional RL. 2) a low input density ($p(\cdot|\boldsymbol{z}_E) < \epsilon$) indicates high epistemic uncertainty (i.e., the input is OoD, and we do not use the model prediction). In practice, ϵ is determined based on the validation dataset.

Risk-Sensitive Action Impact. The *impact* $\phi(s,a)$ measures how much an action a changes the future return of a player's team. In terms of the value function, this is the change in action value due to a player's movement. Previous works [1, 2, 3] computed the impact with the expected next-goal return $Q(s_t,a_t) = \mathbb{E}[Z(s_t,a_t)]$. $Q(\cdot)$ does not take into account the inherent variability of the returns and thus cannot estimate the risk of an action. To understand how players respond to risk, we propose a Risk-sensitive Game Impact Metric (RiGIM) based on $\hat{Z}_k(s,a)$ and $p(\cdot|\boldsymbol{z}_E)$:

$$\phi_k(s_{t+1},a_{t+1},c) = \left[\hat{Z}_k^c(s_{t+1},a_{t+1}) - \hat{Z}_k^c(s_t,a_t)\right]\mathbb{I}_{p(\cdot|\boldsymbol{z}_E)\geq\epsilon}$$
$$RiGIM_l(c) = \sum_{(s,a)\in\mathcal{D}'} n(s,a,l) \times \phi_k(s,a,c) \tag{5}$$

where $c \in [0,1]$ is the confidence level, Z^c denotes the $(1-c)^{th}$ quantile in $Z(\cdot)$ and $n(s,a,l)$ denotes the number of times that a player l performs action a at a state s in the testing dataset $\mathcal{D}'$. We omit the term r_t since $r_t = 0$ except at scoring step T, and s_T is the terminal state of a scoring episode. $\phi(\cdot)$ can either be 1) risk-averse (with a large c), with better sensitivity to bad outcomes or 2) risk-seeking (with a small c), with better sensitivity to positive outcomes.

5.2 Case Study: Player Ranking in Testing Games

We rank players according to their RiGIM scores in the NHL testing games (see the experiment setting in Section 6) at different confidence levels. Table 1 shows a *risk-seeking* ranking (confidence $c = 0.2$), which favors offensive players (e.g., Centres (C)) with strong scoring ability. Aleksander Barkov, who scores the most points in these games, is captured by this ranking. When we set confidence c to 0.8, Table 2 shows a *risk-averse* ranking which highlights players in defensive positions (e.g., Defensive (D)). John Klingberg, the defenseman with the most assists, is listed in the top-10 players.

Table 1: Top 10 players with confidence 0.2.

Player Name	Position	Team	P	A	G	RiGIM
Jonathan Toews	C	CHI	10	5	5	14.72
Anze Kopitar	C	LAK	12	9	3	14.55
Vincent Trocheck	C	FLA	8	5	3	14.02
Tomas Hertl	C	SJS	12	8	4	13.97
John Tavares	C	TOR	12	3	9	13.92
Tyler Seguin	C	DAL	18	12	6	13.71
Leon Draisaitl	C	EDM	16	8	8	13.16
Aleksander Barkov	C	FLA	19	14	5	12.63
Sean Couturier	C	PHI	11	6	5	12.62
Nathan MacKinnon	C	COL	12	6	6	12.48

Table 2: Top 10 players with confidence 0.8.

Player Name	Position	Team	P	A	G	RiGIM
Radek Faksa	C	DAL	6	3	3	2.74
Leon Draisaitl	C	EDM	16	8	8	2.51
John Klingberg	D	DAL	10	9	1	2.46
Esa Lindell	D	DAL	3	1	2	2.29
Connor McDavid	C	EDM	18	11	7	2.23
Tomas Hertl	C	SJS	12	8	4	1.93
Miro Heiskanen	D	DAL	5	3	2	1.86
Elias Pettersson	C	VAN	8	6	2	1.79
Tyler Seguin	C	DAL	18	12	6	1.78
Roope Hintz	LW	DAL	11	7	4	1.77

6 Empirical Evaluation

Dataset. Our experiments utilize both a *ice-hockey* and a *soccer* dataset from the National Hockey League (NHL) and major European soccer leagues, which contains 9,213,371 events, covering 195

teams, 4,172 games, and 6,513 players. These datasets consist of events around the ball. Each event records the identity and action of the player possessing the ball, with time stamps and features of the game context (see all the game features in Appendix A.1). To the best of our knowledge, this is the *most extensive study for player evaluation*. Note that we *do not utilize virtual environments* like Atari [11] or Mujoco [12] because 1) the dynamics in these environments are deterministic without uncertainty to be modeled. 2) this paper mainly studies sports games that have stochastic dynamics [30], which are valid test-beds for our method.

Experiment Settings. We divide the dataset into a training set (80%), a validation set (10%), and a testing set (10%) according to game dates, so that games in the testing set happened after the games in the training and validation set. To predict the action values in the testing games, the metric must remain robust to OoD data points. We report the results averaged over 5 independent runs.

Comparison Methods. We employ an ablation design that iteratively removes parts from RiGIM . **GIM** removes the uncertainty estimator by directly using a Deep Recurrent Q-Network for estimating action values [2]. **T0-GIM** removes the recurrent model and uses a Deep Q-Network (DQN) for the value function. We then replace the RL framework with a supervised learning framework for estimating action values by following **VAEP** [3]. Instead of using function approximators, **Scoring Impact (SI)** [1] implements the tabular-based value iteration algorithm for computing action values from discretized spatial and temporal features. **Expected-Goal (EG)** metric directly uses the expected action values instead of impact values for measuring player performance. The last metric **Plus-Minus ($+/-$)** is based on game statistics and measures the goal-gain with and without the player on court. We summarize these metrics in Table 3.

To study how well SP-CNF boosts model performance, we compare 1) a Gaussian Discriminant Analysis **(GDA)-RiGIM** metric that replaces SP-CNF with GDA [40], and 2) a Naive **(Na)-RiGIM** metric that removes the epistemic uncertainty estimator and uses all the predicted distributions to compute players' impact (see Appendix A.2 and A.3 for more details.).

Table 3: A summary of the baseline Methods for player evaluation.

Method	Risk-Aware	History-Aware	RL-Based	Continuous Feature	Impact-Based	Context-Aware
$+/-$	✗	✗	✗	✗	✗	✗
EG	✗	✗	✗	✗	✗	✓
SI	✗	✗	✗	✗	✓	✓
VAEP	✗	✓	✗	✓	✓	✓
T0-GIM	✗	✗	✓	✓	✓	✓
GIM	✗	✓	✓	✓	✓	✓

6.1 Player Evaluation Performance: Correlations with Standard Measures

We compute the correlations between player ranking metrics and standard measures on the testing games in a game season. We study 11 *success* measures for ice hockey and 8 *success* measures for soccer. To make the results more comprehensive, we also add 5 *penalty measures*. The studied measures are popular measures from the NHL and soccer statistics websites [1]. To derive a fair comparison, we set the confidence level of RiGIM to 0.5.

Tables 4 and 5 show the average correlations in 5 independent runs on the testing dataset (see Tables C.1 and C.2 in Appendix for the complete mean $\pm$ standard deviation results). RiGIM achieves the highest correlations with 14 out of 19 success measures and the smallest correlations with 3 out of 5 penalty measures in ice hockey and soccer games. This observation shows that RiGIM is a comprehensive metric that can detect both the positive and the negative influence of a player. If we remove SP-CNF or replace it with other uncertainty estimators, most correlations become weaker except for the correlations with the SHP and SHG measures. This is because SHP and SHG rarely happen in a season (scoring with fewer players on ice is difficult), SP-CNF detects this phenomenon and assigns small densities to corresponding events. Some of these events can be filtered by SP-CNF, and thus our correlations with SHP and SHG are less significant. For the risk-neutral methods, their performance is generally less satisfying when compared with risk-aware methods, especially for the $+/-$ metric, which shows the importance of capturing risk and modeling the context features.

6.2 Sensitivity to Risk: Correlations Conditioning on Different Confidence Levels

We measure whether RiGIM is sensitive to the risk by its correlations with the standard measures conditioning on a specific confidence level c (from 0 to 1), for example, RiGIM(c), which indicates with probability c that the players' impact should be at least RiGIM(c).

[1] `http://www.nhl.com/stats/skaters` and `https://www.whoscored.com/statistics`

Table 4: Correlations with standard measures in the **ice hockey** games. The *success* measures are assist, goal, Game Winning Goal (GWG), Overtime Goal (OTG), Short-handed Goal (SHG), Power-play Goal (PPG), Point (P), Short-handed Point (SHP), Power-play Point (PPP), Time On Ice (TOI), and Shots (S). The *penalty* measure is Penalty Minute (PIM).

Methods	Assist	Goal	GWG	OTG	SHG	PPG	Point	SHP	PPP	TOI	S	PIM
$+/-$	0.181	0.189	0.187	0.028	0.071	0.063	0.206	0.119	-0.071	0.021	0.038	-0.014
EG	0.239	0.303	0.264	0.130	-0.053	0.163	0.322	0.023	0.226	0.153	0.534	-0.112
SI	0.237	**0.596**	**0.409**	0.123	0.095	0.351	0.452	0.066	0.274	0.224	0.405	0.138
VAEP	0.238	0.454	0.225	0.06	0.053	0.326	0.382	-0.0	0.321	0.086	0.362	0.027
T0-GIM	0.397	0.394	0.139	0.16	0.151	0.216	0.455	0.153	0.295	0.356	0.387	0.058
GIM	0.456	0.408	0.167	0.158	0.134	0.246	0.501	0.137	0.345	0.395	0.431	0.061
Na-RiGIM(0.5)	0.593	0.476	0.223	0.173	**0.152**	0.313	0.625	**0.175**	0.453	0.597	0.611	0.115
GDA-RiGIM(0.5)	0.591	0.475	0.221	0.174	**0.152**	0.315	0.623	0.174	0.452	0.593	0.609	0.113
RiGIM(0.5)	**0.675**	0.477	0.266	**0.184**	0.11	**0.355**	**0.678**	0.141	**0.529**	**0.68**	**0.7**	0.146

Table 5: Correlations with standard measures in the **soccer** dataset. The *success* measures are goal, assist, Shots per Game (SpG), Pass Success percentage (PS%), Key Passes (KeyP), Dribbles (Drb), Crosses and (being) Fouled. The *penalty* measures are Yellow (Yel) and Red Card Received, Offsides (Off) and Own Goals (OwnG).

Methods	Goal	Assist	SpG	PS%	KeyP	Drb	Crosses	Fouled	Yel	Red	Off	OwnG
$+/-$	0.284	0.318	0.199	**0.288**	0.218	0.119	0.017	0.035	0.001	-0.069	0.053	-0.001
EG	0.422	0.173	0.328	0.164	0.278	0.013	0.040	-0.026	0.534	0.034	-0.124	-0.008
SI	0.585	0.153	0.438	-0.140	0.052	0.050	0.216	-0.065	0.114	-0.089	-0.249	-0.102
VAEP	0.093	0.290	0.121	-0.111	0.116	0.059	0.082	-0.00	0.024	0.133	-0.055	-0.051
T0-GIM	0.614	0.455	0.715	0.148	0.472	0.431	0.161	0.355	-0.007	-0.027	-0.346	-0.168
GIM	0.627	0.462	0.72	0.149	0.473	0.437	0.169	0.358	-0.0	-0.025	-0.336	-0.154
Na-RiGIM(0.5)	0.646	0.507	0.741	0.144	0.503	0.445	0.177	0.391	0.101	0.007	-0.309	-0.144
GDA-RiGIM(0.5)	0.649	0.506	0.725	0.132	0.478	0.421	0.161	0.389	0.147	0.018	-0.259	-0.125
RiGIM(0.5)	**0.671**	**0.577**	**0.756**	0.181	**0.574**	**0.530**	**0.239**	**0.448**	-0.092	-0.039	-0.451	-0.185

Figures 5 and 6 show the correlations at different confidence levels for ice-hockey and soccer games. RiGIM is sensitive to risk, in the sense that it has different correlations with standard measures at these confidence levels, whereas GIM, as a risk-neutral metric, is unaware of the risk, and thus its correlations remain unchanged. Compared to other baselines, our RiGIM generally maintains higher correlations with success measures and lower correlations with penalty measures. The exceptions are the correlations with the SHP, SHG, and OTG, which are rare events with small densities, and thus SP-CNF may filter them during testing. We find when c becomes smaller, RiGIM(c) becomes risk-seeking, and thus achieves a higher correlation with success measures. However, the correlations drop when c approaches 0. This observation is consistent with the fact that an overly risk-seeking estimate cannot reflect the real contributions of players.

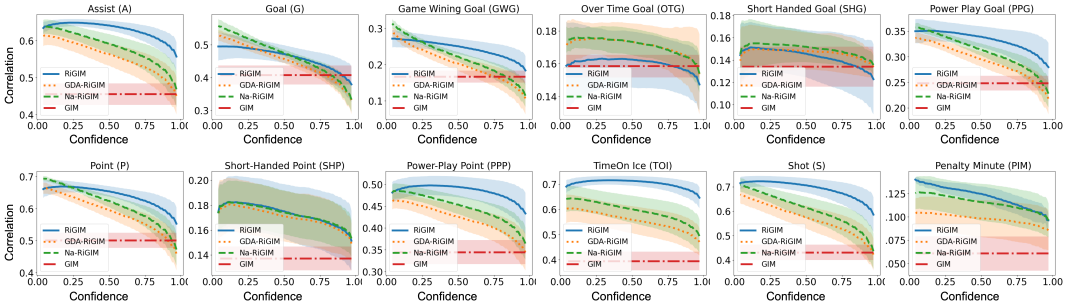

Figure 5: Correlations (Mean $\pm$ standard deviation) with success measures (the first 11 plots) and penalty measures (the last plot) at different confidence levels in **ice-hockey** games.

6.3 The Prediction Accuracy: Match Future Scoring Frequencies With Action-Values

We study how well the predicted action-value distributions match the real next-goal scoring frequencies under discrete game contexts. These game contexts are constructed by dividing the continuous state space into discrete bins. To calculate the empirical scoring frequency associated with each bin, we assign an observed state s to a bin B according to the values of three discrete context features in the current observation: Manpower Differential, Goal Differential, and Period. The empirical and estimated scoring probabilities for a bin (with size $|B|$) are defined as follows:

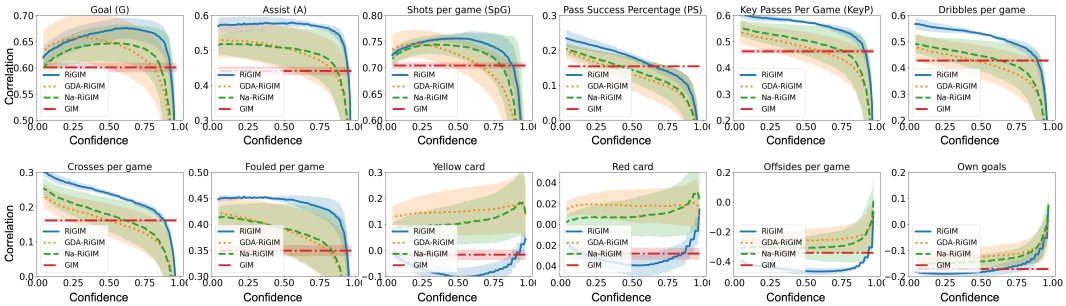

Figure 6: Correlations (Mean $\pm$ standard deviation) with success measures (the first 8 plots) and penalty measures (the last 4 plots) at different confidence levels for in **soccer** games.

a) *Empirical Scoring Chances*: for each $s \in B$, we set $g_k(s) = 1$ if the observed scoring-episode containing state s ends with a goal by team k. Then $G_k^*(B) = \frac{1}{|B|}\sum_{s\in B} g_k(s)$.

b) *Estimated Scoring Chances*: for each $s \in B$, given N samples from the calibrated distribution $z_k(s) \sim \hat{Z}_k(s,a)\mathbb{I}_{p(\cdot|\boldsymbol{z}_E)\geq\epsilon}$, the estimated chances are: $\hat{G}_k(B) = \frac{1}{N|B|}\sum_{s\in B}\sum_{n=1}^{N} z_{k,n}(s)$.

Table 6: The difference between the empirical and the estimated scoring chances at different contexts.

	Ice-Hockey			Soccer		
Manpower Differential	Short-Handed	Even-Strength	Power-Play	Short-Handed	Even-Strength	Power-Play
GIM	0.115 ± 0.078	0.094 ± 0.082	0.099 ± 0.085	0.211 ± 0.034	$\mathbf{0.114} \pm 0.05$	0.199 ± 0.037
Na-RiGIM	0.133 ± 0.016	0.064 ± 0.016	$\mathbf{0.013} \pm 0.009$	0.226 ± 0.019	0.136 ± 0.019	0.175 ± 0.028
GDA-RiGIM	0.148 ± 0.035	0.072 ± 0.029	0.017 ± 0.011	0.216 ± 0.022	0.151 ± 0.011	0.18 ± 0.013
RiGIM	$\mathbf{0.080} \pm 0.020$	$\mathbf{0.058} \pm 0.008$	0.047 ± 0.046	$\mathbf{0.204} \pm 0.005$	0.133 ± 0.007	$\mathbf{0.147} \pm 0.033$
Goal Differential	-1	0	1	-1	0	1
GIM	0.238 ± 0.122	0.105 ± 0.084	0.271 ± 0.059	0.155 ± 0.047	0.155 ± 0.054	0.221 ± 0.049
Na-RiGIM	0.238 ± 0.006	0.045 ± 0.015	0.108 ± 0.031	0.157 ± 0.02	$\mathbf{0.104} \pm 0.024$	0.16 ± 0.017
GDA-RiGIM	0.236 ± 0.007	0.045 ± 0.016	0.11 ± 0.027	0.165 ± 0.018	0.117 ± 0.017	0.175 ± 0.007
RiGIM	$\mathbf{0.193} \pm 0.021$	$\mathbf{0.029} \pm 0.015$	$\mathbf{0.092} \pm 0.019$	$\mathbf{0.152} \pm 0.008$	0.109 ± 0.004	$\mathbf{0.149} \pm 0.013$
Period	3	2	1	2st half	1nd half	N/A
GIM	$\mathbf{0.095} \pm 0.055$	0.111 ± 0.086	0.114 ± 0.084	$\mathbf{0.191} \pm 0.037$	0.104 ± 0.059	
Na-RiGIM	0.139 ± 0.018	0.044 ± 0.015	0.024 ± 0.015	0.237 ± 0.013	0.061 ± 0.03	
GDA-RiGIM	0.143 ± 0.028	0.050 ± 0.025	0.033 ± 0.025	0.238 ± 0.012	0.059 ± 0.026	
RiGIM	0.143 ± 0.011	$\mathbf{0.032} \pm 0.005$	$\mathbf{0.014} \pm 0.009$	0.226 ± 0.011	$\mathbf{0.058} \pm 0.009$	

Table 6 shows the average absolute difference between $\hat{G}$ and G^* based on 5 independent runs. Our baselines are the learning-based action-values metrics for the context-aware evaluation. RiGIM achieves a minimum distance in 12 out of 17 bins. This is because 1) FS-CNF outperforms GDA by computing a more accurate uncertainty estimator for filtering OoD states and 2) the distribution estimates contain richer information than expectation estimates, allowing $\tilde{Z}(\cdot)$ to better match the scoring frequencies than $Q(\cdot)$. This observation is consistent with the findings in [6].

6.4 Limitations

A Study for Sports Games. This work uses stochastic sports games as the testbeds for uncertainty estimation, but we argue the same methods can be easily migrated to other applications in stochastic environments such as autonomous driving or healthcare.

Evaluation Instead of Controlling. Our method focuses on player evaluation instead of control. We believe both tasks are challenging with respect to different aspects. Evaluation requires the action values to accurately reflects an agent's real contribution to game-winning. An ideal evaluation metric can provide in-game predictions of game outcomes, which is important for the sports industry.

7 Conclusion and Future Work

In this paper, we designed an RL framework for quantifying the aleatoric uncertainty and the epistemic uncertainty from stochastic sports datasets. This framework enabled distributional RL and an FS-CNF model to estimate both uncertainties, with which we proposed a risk-sensitive evaluation metric RiGIM . Empirical results show that RiGIM correlates well with success measures and the correlation is sensitive different risks. A direction of future work is to extend our model to other domains.

References

[1] Kurt Routley and Oliver Schulte. A markov game model for valuing player actions in ice hockey. In *The Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 782–791, 2015.

[2] Guiliang Liu and Oliver Schulte. Deep reinforcement learning in ice hockey for context-aware player evaluation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3442–3448, 2018.

[3] Tom Decroos, Lotte Bransen, Jan Van Haaren, and Jesse Davis. Actions speak louder than goals: Valuing player actions in soccer. In *The 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1851–1861, 2019.

[4] Yudong Luo, Oliver Schulte, and Pascal Poupart. Inverse reinforcement learning for team sports: Valuing actions and players. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3356–3363, 2020.

[5] Marc G. Bellemare, Will Dabney, and Mark Rowland. *Distributional Reinforcement Learning*. MIT Press, 2022.

[6] Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning (ICML)*, volume 70, pages 449–458, 2017.

[7] Will Dabney, Mark Rowland, Marc G. Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *AAAI Conference on Artificial Intelligence*, pages 2892–2901, 2018.

[8] Borislav Mavrin, Hengshuai Yao, Linglong Kong, Kaiwen Wu, and Yaoliang Yu. Distributional reinforcement learning for efficient exploration. In *International Conference on Machine Learning (ICML)*, volume 97, pages 4424–4434, 2019.

[9] Fan Zhou, Jianing Wang, and Xingdong Feng. Non-crossing quantile regression for distributional reinforcement learning. In *Neural Information Processing Systems (Neurips)*, 2020.

[10] Fan Zhou, Zhoufan Zhu, Qi Kuang, and Liwen Zhang. Non-decreasing quantile function network with efficient exploration for distributional reinforcement learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3455–3461, 2021.

[11] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

[12] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pages 5026–5033. IEEE, 2012.

[13] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *CoRR*, abs/2005.01643, 2020.

[14] Brendan O'Donoghue, Ian Osband, Rémi Munos, and Volodymyr Mnih. The uncertainty bellman equation and exploration. In *International Conference on Machine Learning (ICML)*, volume 80, pages 3836–3845, 2018.

[15] Yunhao Tang and Shipra Agrawal. Exploration by distributional reinforcement learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2710–2716, 2018.

[16] Shangtong Zhang and Hengshuai Yao. QUOTA: the quantile option architecture for reinforcement learning. In *AAAI Conference on Artificial Intelligence*, pages 5797–5804, 2019.

[17] Yudong Luo, Guiliang Liu, Haonan Duan, Oliver Schulte, and Pascal Poupart. Distributional reinforcement learning with monotonic splines. In *International Conference on Learning Representations (ICLR)*, 2022.

[18] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped DQN. In *Neural Information Processing Systems (Neurips)*, pages 4026–4034, 2016.

[19] Richard Y. Chen, Szymon Sidor, Pieter Abbeel, and John Schulman. UCB and infogain exploration via q-ensembles. *CoRR*, abs/1706.01502, 2017.

[20] Ian Osband, John Aslanides, and Albin Cassirer. Randomized prior functions for deep reinforcement learning. In *Neural Information Processing Systems (Neurips)*, pages 8626–8638, 2018.

[21] Felipe Leno da Silva, Pablo Hernandez-Leal, Bilal Kartal, and Matthew E. Taylor. Uncertainty-aware action advising for deep reinforcement learning agents. In *AAAI Conference on Artificial Intelligence*, pages 5792–5799, 2020.

[22] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. In *Neural Information Processing Systems (Neurips)*, pages 11761–11771, 2019.

[23] Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble. In *Neural Information Processing Systems (Neurips)*, pages 7436–7447, 2021.

[24] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y. Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: model-based offline policy optimization. In *Neural Information Processing Systems (Neurips)*, 2020.

[25] Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. In *Neural Information Processing Systems (Neurips)*, 2020.

[26] Oliver Mihatsch and Ralph Neuneier. Risk-sensitive reinforcement learning. *Machine learning*, 49(2):267–290, 2002.

[27] R Tyrrell Rockafellar, Stanislav Uryasev, et al. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.

[28] Yun Shen, Michael J. Tobia, Tobias Sommer, and Klaus Obermayer. Risk-sensitive reinforcement learning. *Neural Computing*, 26(7):1298–1328, 2014.

[29] Yinlam Chow, Aviv Tamar, Shie Mannor, and Marco Pavone. Risk-sensitive and robust decision-making: a cvar optimization approach. In *Neural Information Processing Systems (Neurips)*, pages 1522–1530, 2015.

[30] Jim Albert, Mark E Glickman, Tim B Swartz, and Ruud H Koning. *Handbook of Statistical Methods and Analyses in Sports*. CRC Press, 2017.

[31] Peter Xenopoulos, Harish Doraiswamy, and Cláudio T. Silva. Valuing player actions in counter-strike: Global offensive. In *IEEE International Conference on Big Data (IEEE BigData)*, pages 1283–1292. IEEE, 2020.

[32] Daniel Cervone, Alex D'Amour, Luke Bornn, and Kirk Goldsberry. A multiresolution stochastic process model for predicting basketball possession outcomes. *Journal of the American Statistical Association*, 111(514):585–599, 2016.

[33] Guiliang Liu, Yudong Luo, Oliver Schulte, and Tarak Kharrat. Deep soccer analytics: learning an action-value function for evaluating soccer players. *Data Mining Knowledge Discovery*, 34(5):1531–1559, 2020.

[34] Xiangyu Sun, Jack Davis, Oliver Schulte, and Guiliang Liu. Cracking the black box: Distilling deep sports analytics. In *The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3154–3162, 2020.

[35] Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine Learning, Proceedings of the 11th International Conference*, pages 157–163, 1994.

[36] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Neural Information Processing Systems (Neurips)*, 2020.

[37] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning ICML*, volume 97, pages 2052–2062, 2019.

[38] Lewis Smith and Yarin Gal. Understanding measures of uncertainty for adversarial example detection. In *The Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 560–569, 2018.

[39] Joost van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International Conference on Machine Learning (ICML)*, volume 119, pages 9690–9700, 2020.

[40] Jishnu Mukhoti, Andreas Kirsch, Joost van Amersfoort, Philip H. S. Torr, and Yarin Gal. Deterministic neural networks with appropriate inductive biases capture epistemic and aleatoric uncertainty. *CoRR*, abs/2102.11582, 2021.

[41] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Neural Information Processing Systems (Neurips)*, pages 6402–6413, 2017.

[42] Yeming Wen, Dustin Tran, and Jimmy Ba. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. In *International Conference on Learning Representations (ICLR)*, 2020.

[43] Michael Dusenberry, Ghassen Jerfel, Yeming Wen, Yi-An Ma, Jasper Snoek, Katherine A. Heller, Balaji Lakshminarayanan, and Dustin Tran. Efficient and scalable bayesian neural nets with rank-1 factors. In *International Conference on Machine Learning ICML*, volume 119, pages 2782–2792, 2020.

[44] Joost van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International Conference on Machine Learning ICML*, volume 119, pages 9690–9700. PMLR, 2020.

[45] George Papamakarios, Iain Murray, and Theo Pavlakou. Masked autoregressive flow for density estimation. In *Neural Information Processing Systems (Neurips)*, pages 2338–2347, 2017.

[46] Madson L. D. Dias, César Lincoln C. Mattos, Ticiana L. Coelho da Silva, José Antônio Fernandes de Macêdo, and Wellington C. P. Silva. Anomaly detection in trajectory data with normalizing flows. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020.

8 Submission of papers to NeurIPS 2022

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Check Section 6.4.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Check our Appendix D.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Check Proposition 1 and the proof in Appendix B.
   (b) Did you include complete proofs of all theoretical results? [Yes] Check Proposition 1 and the proof in Appendix B.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Check a detailed dataset introduction in Appendix A.1 and the instructions in Appendix A.2.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Check the experiment settings in Section 6 and Appendix A.2.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We use 5 random seeds for all experiments. Check the experiment settings in Section 6, our experiment results and Table C.1 and C.2 in Appendix.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Check Appendix A.4.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] There is not formal citation, so We add the link to the data source. Check our dataset introduction in Appendix A.1.
   (b) Did you mention the license of the assets? [N/A] There is no such licence that we can cite for this dataset.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] We have corporated on many projects.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] Check our dataset introduction in Appendix A.1.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]