Appendix

A Implementation Details

A.1 The Sports Dataset

In this paper, we use an ice hockey and a soccer dataset. Table A.1 shows a complete list of features of these datasets. The dataset records the movements of each player in professional games. The data sources are game logs and broadcast videos, which are public resources. Personal information of these players, including age, gender, and physical conditions, has not been included or discussed in this paper.

Table A.1: The complete list of game features for the ice hockey dataset and the soccer dataset. The table utilizes adjusted spatial coordinates where negative numbers denote the defensive zone of the acting player and positive numbers denote the offensive zone.

Type	Name	Range	
Ice Hockey	Spatial Features	X Coordinate of Puck	[-100, 100]
		Y Coordinate of Puck	[-42.5, 42.5]
		Velocity of Puck	$(-\infty, +\infty)$
		Angle between the puck and the goal	$[-3.14, 3.14]$
	Temporal Features	Game Time Left	[0, 3,600]
		Event Duration	$(0, +\infty)$
	In-Game Features	Score Differential	$(-\infty, +\infty)$
		Manpower Situation	{Even Strength, Shorted Handed, Power Play}
		Home or Away Team	{Home, Away}
		Action Outcome	{successful, failure}
Soccer	Spatial Features	X Coordinate of ball	[0, 100]
		Y Coordinate of ball	[0, 100]
		Velocity of ball	$(-\infty, +\infty)$
		Angle between the ball and the goal	$[-3.14, 3.14]$
	Temporal Features	Game Time Remaining	[0, 100]
		Event Duration	$(0, +\infty)$
	In-Game Features	Goal Differential	$(-\infty, +\infty)$
		Manpower Situation	[-5, 5]
		Home or Away Team	{Home, Away}
		Action Outcome	{successful, failure}

Ice Hockey Dataset In this paper, we use a play-by-play dataset constructed by Sportlogiq[1]. They capture the information of an on-puck player (player possessing the puck) from broadcast videos with computer vision techniques. In the experiments, we split the games in this dataset into training,

[1] https://sportlogiq.com

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

testing, and validation datasets according to game dates, so the training dataset contains 956 games (from October 3rd, 2018 to February 24th, 2019), and the validation dataset contains 119 games (from February 24th, 2019 to March 12th, 2019), and the testing dataset contains 121 games (from March 12th, 2019 to April 6th, 2019).

Soccer Dataset In this paper, we utilize the F24 play-by-play soccer game dataset provided by Opta[2]. The dataset records the play-by-play information of game events and player actions for the entire 2017-2018 game season from multiple soccer leagues, including English Premier League, Dutch Eredivisie, EFL Championship, Italian Serie A, German Bundesliga, Spanish La Liga, French Ligue 1 and German Bundesliga Zwei.

A.2 Hyper-Parameters

Table A.2: The Architecture of the main components in our model.

Model	Network Component	Hidden Dimensions
Feature extractor	Residual Layer	128
	Leaky Rectified Linear Unit	N/A
	Spectral Normalization	N/A
	Residual Layer	128
	Leaky Rectified Linear Unit	N/A
	Spectral Normalization	N/A
Spline DRQN	LSTM Layer	128
	Fully Connect Layer	128
	Rectified Linear Unit	N/A
	Fully Connect Layer	128
	Rectified Linear Unit	N/A
	Spline function	N/A
A CNF Block (i.e., MADE Layer [1]	Masked Linear Layer	180
	Rectified Linear Unit	N/A
	Masked Linear Layer	180
	Batch Normalization	N/A
	Reverse Layer	N/A

We introduce the hyper-parameters for implementing our distributional RL and FS-CNF models. Table A.2 shows the model architecture.

Distributional RL model. We set the quantile number N to 64 and set the size of hidden layers (in both LSTM and Resnet) to be 128. The max trace length of LMST is set to 10, and the batch size is set to 64. The discount factor is set to 1, and the learning rate is set to 0.0005. The η is set to 1.

FS-CNF. The feature extractor is implemented by Residual layers and Spectral Normalization. To build CNF, we stack 5 layers of CNF blocks. Each block contains a MADE layer [1], a batch normalization layer, and a reverse layer by following the structure in [2]. The size of a hidden layer is set to 180, and the learning rate is set to 0.0001.

A.3 Gaussian Discriminant Analytic Model

We introduce the implementation of our Gaussian Discriminant Analysis (GDA) in our baselines. GDA is a Gaussian mixture model that builds a single Gaussian for each class $q(e|\tilde{z}_m)$ where 1) the class label $\tilde{z}_m$ is constructed by dividing the expected returns $\mathbb{E}(Z(s,a)) \in [0,1]$ into m classes $\{\tilde{z}_1, \ldots, \tilde{z}_M\}$. 2) we estimate the density for latent features e instead of raw inputs (s,a) since building GDA on spatial-temporal raw features is difficult and the higher-level latent features learned by neural networks can alleviate this difficulty [3]. In order for $q(e|\tilde{z}_m)$ to capture the input density, the distance between data points in latent space must accurately reflect their distance in input space [4]. However, during learning, feature extractors might map the features of OoD inputs to InD regions

[2]https://www.optasports.com/

in the latent space (i.e., Feature Collapse [5]). To fix this issue, we utilize a bi-Lipschitz constraint for the feature extractor $f(x;\omega_{\mathcal{E}})$:

$$\begin{aligned} &\forall x_1, x_2 \in \mathcal{D} \text{ and } x := (s,a), \\ &K_1\|x_1 - x_2\|_I \leq \|f(x_1;\omega_{\mathcal{E}}) - f(x_2;\omega_{\mathcal{E}})\|_F \leq K_2\|x_1 - x_2\|_I \end{aligned} \tag{1}$$

where $\|\cdot\|_I$ and $\|\cdot\|_F$ denote metrics in the input and feature space respectively, and $K1$ and $K2$ denote the lower and upper Lipschitz constants [6]. The lower Lipschitz bound ensures sensitivity to distances in the input space, and the upper Lipschitz bound ensures smoothness in the features, preventing them from becoming too sensitive to input variations and leading to poor generalisation and loss of robustness. We follow [4] to ensure the bi-Lipschitzness in the feature extractor $f(\cdot;\omega_{\mathcal{E}})$ by implementing it with residual connections together with spectral normalisation.

A.4 Computation Resource and Running Time

We ran the experiments on a cluster operated by the Slurm workload manager. The cluster has multiple kinds of GPUs, including Tesla T4 with 16 GB memory, Tesla P100 with 12 GB memory, and RTX 6000 with 24 GB memory. Our algorithm runs with GPUs of at least 12 GB of memory. We use 24 GB of main memory for training the distributional RL and the FS-CNF models. The number of running nodes is 1, and the number of CPUs requested per task is 8. Given the aforementioned resources, the distributional RL program uses around 12 GPU hours to finish the training of 1 random seed for the ice-hockey dataset and 16 GPU hours for the soccer dataset (since the size of the soccer dataset is larger). Based on the well-trained distributional RL program, FS-CNF takes around 8 GPU hours to run the ice-hockey dataset and 10 GPU hours to run the soccer dataset.

Computational Complexity. Table A.2 illustrates the structure of our model. RiGIM is based on the mini-batch gradient descent. Let B be the batch size, M be the total number of data points, N be the number of quantiles, H be the size of hidden layers, I be the input size, T be the maximum trace length of the LSTM, L be the number of layers in the CNF and K be the number of autoregressive layers in the CNF. The computational complexities of a forward pass of Resnet, LSTM, SPL-DQN and FS-CNF are $O(I^2+I)$, $O[T(IH+I^2+I)]$, $O(H^2+NH^2)$ [7] and $O[KIH+K(L-1)H^2]$ [2] respectively. It requires M/B passes to finish one round of training.

Memory Complexity. Following the same notation, the memory complexity of RiGIM is $O[2I^2+4IH+(N+1)H^2+3KIH+K(L-1)H]$. This complexity is based on the number of parameters in our model. In practice, we also need to consider the influence of batch size.

B Proof

Let's assume we have vector valued random variables $\boldsymbol{Z}$, $\boldsymbol{R}$ with distribution space $\mathcal{P}(\mathbb{R})^{|S||A|}$, so

$$\begin{aligned} H(\boldsymbol{Z}) &\overset{\text{(a)}}{=} H[(I-\gamma\boldsymbol{P}^{\pi})^{-1}\boldsymbol{R}] \\ &\overset{\text{(b)}}{=} \log|\det[(I-\gamma\boldsymbol{P}^{\pi})^{-1}]| + H[\boldsymbol{R}] \\ &\overset{\text{(c)}}{=} \log|\det[\frac{\mathbf{d}^{\pi}}{1-\gamma}]| + H[\boldsymbol{R}] \\ &\overset{\text{(d)}}{=} -|A||S|\log(1-\gamma) + \log|\det[\mathbf{d}^{\pi}]| + H[\boldsymbol{R}] \end{aligned}$$

- (a) holds by following the Bellman consistency $\boldsymbol{Z}^{\pi} = \boldsymbol{R} + \gamma\boldsymbol{P}^{\pi}\boldsymbol{Z}^{\pi}$. To see that the $(I-\gamma\boldsymbol{P}^{\pi})$ is invertible, it suffices to show that for any non-zero vector $\boldsymbol{x} \in \mathbb{R}^{|S|\times|A|}$:

$$\begin{aligned} \|(I-\gamma\boldsymbol{P}^{\pi})\boldsymbol{x}\|_{\infty} &= \|\boldsymbol{x} - \gamma\boldsymbol{P}^{\pi}\boldsymbol{x}\|_{\infty} \\ &\geq \|\boldsymbol{x}\|_{\infty} - \gamma\|\boldsymbol{P}^{\pi}\boldsymbol{x}\|_{\infty} \\ &\geq \|\boldsymbol{x}\|_{\infty} - \gamma\|\boldsymbol{x}\|_{\infty} \\ &= (1-\gamma)\|\boldsymbol{x}\|_{\infty} \\ &\geq 0 \end{aligned}$$

which implies $I - \gamma \boldsymbol{P}^{\pi}$ is full rank.

- (b) holds by following the differential entropy proprieties and the fact that $(I - \gamma \boldsymbol{P}^{\pi})$ is invertible.
- (c) holds by defining $\mathbf{d}^{\pi} = (1-\gamma)(I - \gamma \boldsymbol{P}^{\pi})^{-1}$. Here, we would like to show that $\mathbf{d}^{\pi} \in [0,1]^{|S||A| \times |S||A|}$ is the induced matrix for distributions over state-action tuples by following policy π. In other words, the $(s,a)^{th}$ row of $\mathbf{d}^{\pi}$ is an induced distribution over states and actions when following π after starting with $s_0 = s$ and $a_0 = aa$. This follows from its definition:

$$\mathbf{d}^{\pi} = (1-\gamma) \sum_{t=1}^{\infty} (\gamma \boldsymbol{P}^{\pi})^t \tag{2}$$

$$= \frac{(1-\gamma)[1 - (\gamma \boldsymbol{P}^{\pi})^{\infty}]}{1 - (\gamma \boldsymbol{P}^{\pi})} \tag{3}$$

$$= \frac{(1-\gamma)}{1 - (\gamma \boldsymbol{P}^{\pi})} \tag{4}$$

C Complementary Results

Methods	Assist	Goal	GWG	OTG	SHG	PPG
$+/-$	0.181 ± 0	0.189 ± 0	0.187 ± 0	0.028 ± 0	0.071 ± 0	-0.047 ± 0
EG	0.239 ± 0	0.303 ± 0	0.264 ± 0	0.130 ± 0	-0.053 ± 0	184 ± 0
SI	0.237 ± 0	0.596 ± 0	0.409 ± 0	0.123 ± 0	0.095 ± 0	0.361 ± 0
VAEP	0.238 ± 0.017	0.454 ± 0.013	0.225 ± 0.009	0.06 ± 0.005	0.053 ± 0.006	0.315 ± 0.004
T0-GIM	0.397 ± 0.014	0.394 ± 0.016	0.139 ± 0.009	0.16 ± 0.006	0.151 ± 0.008	0.223 ± 0.021
GIM	0.456 ± 0.029	0.408 ± 0.029	0.167 ± 0.017	0.158 ± 0.007	0.134 ± 0.018	0.248 ± 0.014
Na-RiGIM(0.5)	0.593 ± 0.026	0.476 ± 0.01	0.223 ± 0.013	0.173 ± 0.008	0.152 ± 0.014	0.314 ± 0.012
GDA-RiGIM(0.5)	0.591 ± 0.026	0.475 ± 0.011	0.221 ± 0.014	0.174 ± 0.01	0.152 ± 0.013	0.314 ± 0.012
RiGIM(0.5)	0.675 ± 0.002	0.477 ± 0.008	0.266 ± 0.006	0.184 ± 0.003	0.11 ± 0.007	0.355 ± 0.003
RiGIM(c^*)	0.68 ± 0.002	0.477 ± 0.008	0.269 ± 0.004	0.187 ± 0.003	0.107 ± 0.006	0.357 ± 0.003

Methods	Point	SHP	PPP	PIM	TOI	S
$+/-$	0.206 ± 0	0.119 ± 0	-0.071 ± 0	-0.014 ± 0	0.021 ± 0	0.038 ± 0
EG	0.322 ± 0	0.023 ± 0	0.226 ± 0	-0.112 ± 0	0.153 ± 0	0.534 ± 0
SI	0.452 ± 0	0.066 ± 0	0.274 ± 0	0.138 ± 0	0.224 ± 0	0.405 ± 0
VAEP	0.382 ± 0.017	-0.0 ± 0.001	0.321 ± 0.01	0.027 ± 0.007	0.086 ± 0.002	0.362 ± 0.012
T0-GIM	0.455 ± 0.017	0.153 ± 0.013	0.295 ± 0.024	0.058 ± 0.008	0.356 ± 0.023	0.387 ± 0.022
GIM	0.501 ± 0.024	0.137 ± 0.01	0.345 ± 0.028	0.061 ± 0.018	0.395 ± 0.037	0.431 ± 0.032
Na-RiGIM(0.5)	0.625 ± 0.019	0.175 ± 0.018	0.453 ± 0.02	0.115 ± 0.018	0.597 ± 0.047	0.611 ± 0.036
GDA-RiGIM(0.5)	0.623 ± 0.02	0.174 ± 0.019	0.452 ± 0.02	0.113 ± 0.018	0.593 ± 0.048	0.609 ± 0.037
RiGIM(0.5)	0.678 ± 0.005	0.141 ± 0.007	0.529 ± 0.002	0.146 ± 0.005	0.68 ± 0.008	0.7 ± 0.006
RiGIM(c^*)	0.681 ± 0.004	0.141 ± 0.007	0.531 ± 0.002	0.147 ± 0.005	0.685 ± 0.007	0.707 ± 0.005

Table C.1: The mean±standard deviation (std) of correlations between the player evaluation metrics and standard measures for the **ice hockey** dataset. The metrics with zero standard deviation are computed with dynamic programming and game statistics.

C.1 Correlations with Standard Success Measures

Table C.2 shows the complete results for the correlations between player evaluation metrics and standard success measures. We report only the standard deviation for the learning-based metrics across 5 independent runs.

C.2 Player Ranking for All games

We show the ranking for all the games in the 2018-19 NHL season. Table C.3 shows the ranking of top-20 players. Nikita Kucherov, who was elected as the Most Valuable Player (MVP) and won the hart memorial trophy, is included in this ranking.

Methods	Goals	Assists	SpG	PS%	KeyP	Drb
$+/-$	0.284 ± 0	0.318 ± 0	0.199 ± 0	0.288 ± 0	0.218 ± 0	0.119 ± 0
EG	0.422 ± 0	0.173 ± 0	0.328 ± 0	0.164 ± 0	0.278 ± 0	0.013 ± 0
SI	0.585 ± 0	0.153 ± 0	0.438 ± 0	-0.140 ± 0	0.052 ± 0	0.050 ± 0
VAEP	0.093 ± 0.037	0.290 ± 0.058	0.121 ± 0.063	-0.111 ± 0.017	0.116 ± 0.005	0.059 ± 0.002
T0-GIM	0.614 ± 0.008	0.455 ± 0.008	0.715 ± 0.007	0.148 ± 0.008	0.472 ± 0.005	0.431 ± 0.004
GIM	0.627 ± 0.022	0.462 ± 0.024	0.72 ± 0.014	0.149 ± 0.013	0.473 ± 0.017	0.437 ± 0.011
Na-RiGIM(0.5)	0.646 ± 0.035	0.507 ± 0.055	0.741 ± 0.024	0.144 ± 0.036	0.503 ± 0.059	0.445 ± 0.06
GDA-RiGIM(0.5)	0.649 ± 0.051	0.506 ± 0.062	0.725 ± 0.031	0.132 ± 0.048	0.478 ± 0.058	0.421 ± 0.064
RiGIM(0.5)	0.671 ± 0.021	0.577 ± 0.015	0.756 ± 0.01	0.181 ± 0.01	0.574 ± 0.005	0.53 ± 0.005
RiGIM(c^*)	0.682 ± 0.009	0.583 ± 0.014	0.757 ± 0.011	0.186 ± 0.008	0.575 ± 0.005	0.531 ± 0.006

Methods	Crosses	Fouled	Yel	Red	Off	OwnG
$+/-$	0.017 ± 0	0.035 ± 0	0.001 ± 0	-0.069 ± 0	0.053 ± 0	-0.001 ± 0
EG	0.040 ± 0	-0.026 ± 0	0.534 ± 0	0.034 ± 0	-0.124 ± 0	-0.008 ± 0
SI	0.216 ± 0	-0.065 ± 0	0.114 ± 0	-0.089 ± 0	-0.249 ± 0	-0.102 ± 0
VAEP	0.082 ± 0.021	-0.00 ± 0.001	0.024 ± 0.003	0.133 ± 0.023	-0.055 ± 0.006	-0.051 ± 0.011
T0-GIM	0.161 ± 0.01	0.355 ± 0.004	-0.007 ± 0.01	-0.027 ± 0.003	-0.346 ± 0.01	-0.168 ± 0.007
GIM	0.169 ± 0.016	0.358 ± 0.019	-0.0 ± 0.036	-0.025 ± 0.01	-0.336 ± 0.017	-0.154 ± 0.013
Na-RiGIM(0.5)	0.177 ± 0.05	0.391 ± 0.048	0.101 ± 0.078	0.007 ± 0.018	-0.309 ± 0.078	-0.144 ± 0.028
GDA-RiGIM(0.5)	0.161 ± 0.048	0.389 ± 0.047	0.147 ± 0.088	0.018 ± 0.015	-0.259 ± 0.075	-0.125 ± 0.037
RiGIM(0.5)	0.239 ± 0.007	0.448 ± 0.006	-0.092 ± 0.039	-0.039 ± 0.009	-0.451 ± 0.028	-0.185 ± 0.019
RiGIM(c^*)	0.238 ± 0.007	0.446 ± 0.006	-0.101 ± 0.04	-0.042 ± 0.007	-0.455 ± 0.03	-0.184 ± 0.022

Table C.2: The mean±standard deviation (std) of correlations between the player evaluation metrics and standard measures for the **soccer** dataset. The metrics with zero standard deviation are computed with dynamic programming and game statistics.

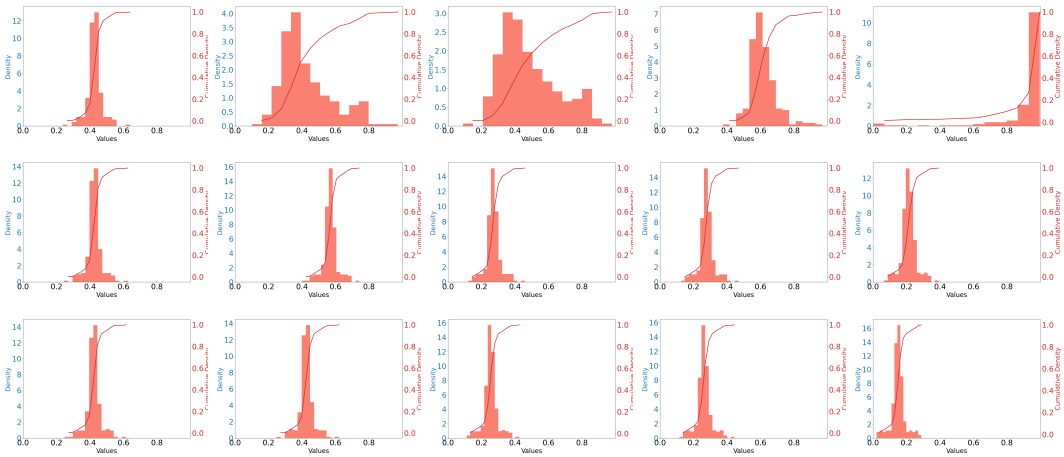

Figure 1: Visualization of action-value distributions. From top to bottom, the actions are shot (top row), carry (middle row) and pass (bottom row). We show the 5 samples for each action.

C.3 Visualization of Action Distributions

C.4 The Scale of Uncertainty during Training

We measure the epistemic uncertainty by the feature-space density estimator. In this section, we compute the scale of the density of the testing game as more games are observed during training. Figure 2 shows the scale of epistemic uncertainty after training the model with different numbers of games. We find the scale of epistemic uncertainty decreases as more games are observed, which shows the model becomes more confident about its predictions after observing richer training data. This is evidence that data plays an important role in supporting decision-making. This result is consistent with the findings in [8].

Table C.3: Top 20 players in all games in the 2018-19 NHL season with confidence 0.2.

Player Name	Position	Team	P	A	G	RiGIM
Aleksander Barkov	C	FLA	96	61	35	51.39
Jack Eichel	C	BUF	82	54	28	47.43
Nathan MacKinnon	C	COL	99	58	41	46.87
Mark Scheifele	C	WPG	84	46	38	46.73
Dylan Larkin	C	DET	73	41	32	43.15
Roman Josi	D	NSH	56	41	15	43.0
Connor McDavid	C	EDM	116	75	41	42.95
Mika Zibanejad	C	NYR	74	44	30	42.15
Johnny Gaudreau	LW	CGY	99	63	36	42.11
Leon Draisaitl	C	EDM	105	55	50	41.55
Nikita Kucherov	RW	TBL	128	87	41	41.39
Sebastian Aho	C	CAR	83	53	30	40.73
Mathew Barzal	C	NYI	62	44	18	40.23
Anze Kopitar	C	LAK	60	38	22	39.81
Bo Horvat	C	VAN	61	34	27	38.89
Keith Yandle	D	FLA	62	53	9	38.84
William Karlsson	C	VGK	56	32	24	38.8
John Carlson	D	WSH	70	57	13	38.71
Jonathan Toews	C	CHI	81	46	35	38.42
Kevin Hayes	C	NYR	55	36	19	38.06

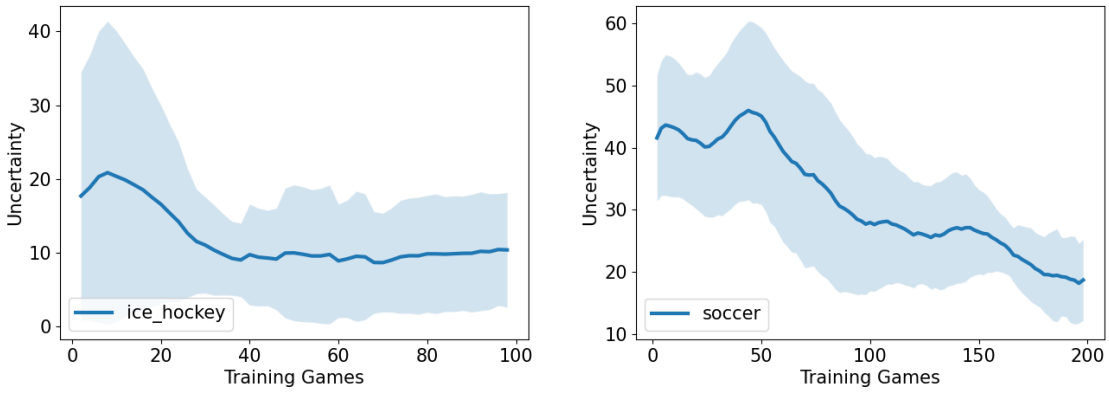

Figure 2: Illustrating the scale of epistemic uncertainty after observing more training games. The uncertainty is measured by the negative log-likelihood $-\log p(s, a|z)$ based on the outputs from the feature-space density estimator. We show the mean$\pm$std uncertainty computed with the games in the testing dataset.

D Negative Social Impact

We expect the main impact outside of the machine learning community to be in professional sports. As part of the entertainment industry, professional sports increase the quality of life for many people. Fans, media, and clubs can be better engaged in sports games with a player ranking system, but putting players' performance under intense scrutiny may yield more pressure on the professional players. The overwhelming pressure might cause anxiety and thus contradicts the original purpose of evaluating players, which is about improving their playing skills and market value.

References

[1] Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. MADE: masked autoencoder for distribution estimation. In *International Conference on Machine Learning (ICML)*, volume 37, pages 881–889, 2015.

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

[3] 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.

[4] Jeremiah Z. Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax-Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. In *Neural Information Processing Systems (Neurips)*, 2020.

[5] 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.

[6] Mihaela Rosca, Theophane Weber, Arthur Gretton, and Shakir Mohamed. A case for new neural network smoothness constraints. *CoRR*, abs/2012.07969, 2020.

[7] 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.

[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.