Detecting Environment Changes via Quantile Spread in Quantile Regression Deep-Q Networks

Paul-Gabriel Stan
Supervisor(s): Frans A. Oliehoek, Mustafa Celikok
EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 22, 2025

Name of the student: Paul-Gabriel Stan
Final project course: CSE3000 Research Project
Thesis committee: Frans A. Oliehoek, Mustafa Celikok, Annibale Panichella

Abstract

Reinforcement learning agents are trained in well-defined environments and evaluated under the assumption that the test time conditions match those encountered during training. However, even small changes in the environment's dynamics can degrade the policy's performance, even more so in safety critical domains. This work investigates the use of Quantile Regression Deep-Q Networks (QR-DQN) to detect environment shifts by analyzing the uncertainty in return predictions. QR-DQN extends Deep-Q learning by estimating the entire distribution of future returns through quantile regression. We hypothesize that under deterministic settings, the spread of the return distribution, quantified by the inter-quantile range, can determine whether environmental changes took place. The RL agent learns low spread predictions for familiar dynamics, but when deployed in changed environments, the quantile distribution becomes wider. We conduct experiments on the deterministic CartPole-v1 environment by the pole length. We show that the quantile spread is low under small changes, but drastically increases as the dynamics' shifts diverge more from the training setting. Our results indicate the potential of distributional reinforcement learning to enhance reliability and awareness in deployment scenarios.

1 Introduction

Reinforcement learning is the process of learning what to do - how to map situations to actions - to maximize a numerical reward signal [11].Reinforcement learning (RL) is a framework in which agents learn to make decisions by interacting with an environment in order to maximize cumulative reward over time. Agents receive observations and rewards from the environment and learn a policy that maps states to actions based on expected outcomes. In recent years, deep RL approaches have emerged, with Deep Q-Networks (DQN) [7] achieving strong performance by approximating value functions using neural networks. However, standard DQN methods focus solely on estimating the expected return, providing a point estimate that fails to capture uncertainty about outcomes. This can be a limitation in practice, especially when the agent encounters scenarios that differ from those seen during training.

Distributional reinforcement learning (DRL) addresses this weak point by modeling the complete distribution of possible returns. By learning a full distribution, DRL offers richer information that contains insights about variability and uncertainty in returns. One such method is Quantile Regression Deep Q-Networks (QR-DQN) [4] and works by approximating the return distribution using quantile regression. This leads to an expressive representation of uncertainty, which has been shown to improve performance and stability in consistent conditions between training and evaluation.

However, in real-life applications, the training environment of the agent may differ from the one in which it is deployed. Even small changes in the dynamics and physics of the environment such as: object mass, friction coefficient or gravitational force, can impact the agent's performance if the policy has not been exposed to these variations in the training setting. Detecting such environment shifts between training and testing environments is critical, even more so in sensitive environments such as finance and health care. Most RL algorithms do not offer built-in mechanisms to recognize such mismatches.

This work explores whether the quantile spread learned by QR-DQN can serve as a signal for identifying environment shifts and aims to answer the question: Can changes of deterministic environments of reinforcement learning agents be detected by the quantile spread

of QR-DQN? In deterministic environments, the agent learns to expect consistent returns, resulting in tightly clustered quantile predictions. If the environment dynamics change at test time, the agent may encounter new state transitions or outcomes, leading to increased uncertainty and a wider predicted distribution. By tracking the inter-quantile range (IQR) across time steps and episodes, we aim to determine whether consistent deviations from the trainin time spread can reveal environment level changes. Our goal is not to improve decision making with this signal, but to assess its potential as a passive detection mechanism for unseen or altered deployment conditions.

The rest of this paper is organized as follows. Chapter 2 presents background knowledge about the topic, followed by Chapter 3 that describes related work in distributional RL and uncertainty estimation. Chapter 4 describes the experimental methodology. Chapter 5 presents the results, and Chapter 6 concludes with key findings and potential directions for future work. Chapter 7 notes the guidelines to adhere to for responsible research and Chapter 8 addresses the conclusion.

2 Background

RL problems are typically formalized as a Markov Decision Process (MDP), defined as a tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where:

- $\mathcal{S}$ is the set of possible states,
- $\mathcal{A}$ is the set of possible actions,
- $P(s'|s, a)$ is the transition probability from state s to s' under action a,
- $R(s, a)$ is the expected immediate reward after taking action a in state s,
- $\gamma \in [0, 1)$ is the discount factor that balances immediate and future rewards.

The agent's objective is to learn a policy $\pi(a|s)$ that maximizes the expected cumulative discounted return:

$$G_t = \sum_{k=0}^{\infty} \gamma^k R(s_{t+k}, a_{t+k}).$$

A very important part of RL is the action value function (Q-function), defined as:

$$Q^\pi(s, a) = E\left[G_t \mid s_t = s, a_t = a\right],$$

which represents the expected return after taking action a in state s and following policy π.

In DQN [7], the Q-function is approximated using a neural network. This network typically has two components: a feature extractor shared across actions, and an output layer that provides Q-values for all available actions. However, it captures only the expected return and offers no information about variability or uncertainty in the outcomes. Such a limitation becomes problematic when the agent is deployed in environments that differ from the training distribution.

DRL addresses this by learning the full return distribution $Z^\pi(s, a)$ instead of its expectation $Q^\pi(s, a)$. The goal becomes approximating the distributional Bellman equation:

$$Z^\pi(s, a) \overset{D}{=} R(s, a) + \gamma Z^\pi(s', a'),$$

where $a' \sim \pi(\cdot|s')$ and $s' \sim P(\cdot|s,a)$. This formulation enables the agent to capture return uncertainty better, offering robustness.

QR-DQN [4] is a practical and widely used DRL algorithm. It approximates the return distribution by directly learning N quantiles $\{\hat{Z}_i(s,a)\}_{i=1}^{N}$ corresponding to fixed probability levels $\tau_i \in (0,1)$. Instead of learning a single expected value, the QR-DQN policy network outputs $N \times |\mathcal{A}|$ quantile estimates, giving a rich description of possible future returns.

As in DQN, QR-DQN uses a two-network architecture: a main network (online) and a target network, which is periodically synchronized with the main network to stabilize learning. The quantile outputs are trained using the Quantile Huber loss [6, 4], which combines the robustness of quantile regression with the numerical stability of the Huber loss. The loss function is defined as:

$$\rho_\kappa(u) = \begin{cases} \frac{1}{2}u^2 & \text{if } |u| \leq \kappa \\ \kappa(|u| - \frac{1}{2}\kappa) & \text{otherwise} \end{cases}$$

where $u = \hat{Z}_i(s,a) - z_j$ is the quantile regression target error with $\hat{Z}_i(s,a)$ as the predicted i-th quantile of the return distribution for state-action pair (s,a), z_j the target quantile value, computed using the Bellman update and κ is a tunable threshold controlling the transition between the quadratic and linear regions. The complete quantile Huber loss is then weighted based on the target quantile level τ as:

$$\mathcal{L}_\tau(u) = |\tau - I\{u < 0\}| \cdot \rho_\kappa(u).$$

To measure the agent's uncertainty about its q-value prediction, we compute the Interquantile Range (IQR) for the selected action. The IQR is defined as:

$$\text{IQR} = Q_{0.75} - Q_{0.25},$$

the difference between the 75th and 25th percentile of the quantile outputs. This is a robust measure of spread that reflects the central variability of the predicted return distribution while being less sensitive to outliers.

The transitions and rewards of deterministic environments are always predictable. Thus, after sufficient training, QR-DQN tends to produce a low IQR distribution, indicating a a high confidence. However, in test environments that deviate from the training environment, the agent may encounter unfamiliar states. In these states, the model's quantile estimates are less confident, therefore a larger IQR.

This motivates our hypothesis: persistent increases in IQR across episodes and time steps may serve as a reliable indicator of environment level distributional shifts. Unlike methods that require direct access to environment parameters, this approach is model free and grounded in the output of the learned quantile distribution.

We apply this method in the classic CartPole-v1 environment from OpenAI Gym. CartPole is a benchmark that consists of a pole attached to a cart that moves along the track. The agent must learn to balance the pole by applying forces left or right to the cart. The state is defined by four continuous variables: cart position, cart velocity, pole angle, and pole angular velocity. The episode terminates when the pole angle or cart position exceeds a predefined threshold or stays in place for a sufficient amount of time.

CartPole is a deterministic environment and thus given a state and action, the transition to the next state will always yield the same result. This makes it ideal for studying distributional methods like QR-DQN, as any uncertainty in the return predictions can be attributed to policy uncertainty or environmental shifts rather than stochasticity.

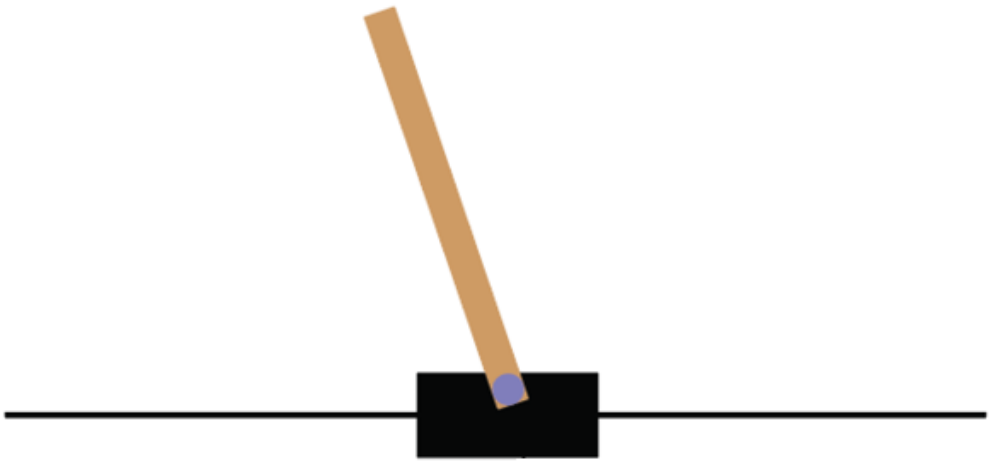

Figure 1: CartPole Environment

By perturbing a single parameter pole length, we alter the test environment's dynamics. These changes serve as systematic deviations from the training distribution, enabling us to evaluate whether increases in the predicted quantile spread can reliably signal an environment shift. Since all randomness is removed, any observed increase in spread strongly suggests that the agent is operating in states or dynamics it has not experienced during training.

3 Related Work

Deep Q-Networks (DQN). Value based methods have been at the core of reinforcement learning (RL), with Deep Q-Networks (DQN) emerging as one of the most influential advancements [7]. DQN combines Q-learning with deep neural networks to handle high-dimensional input spaces such as raw pixels, and has demonstrated human level performance in Atari games. Despite its success, standard DQN only models the expected return, discarding potentially useful information about the return distribution. This makes it less robust under distributional shifts or uncertainty.

Distributional Reinforcement Learning. To address this limitation, distributional reinforcement learning (DRL) has been proposed, where the full distribution over returns is modeled instead of just the expectation. Bellemare et al. introduced the Categorical DQN (C51) algorithm [1], which approximates the value distribution using a fixed support and a categorical distribution. Other DRL approaches include Implicit Quantile Networks (IQN) [3] and Full Quantile Functions [12], which offer more flexible distribution approximations. These methods have shown improved sample efficiency and stability, especially in deterministic settings.

Quantile Regression Deep Q-Networks (QR-DQN). Building on the benefits of DRL, QR-DQN [4] learns to approximate the quantile function of the return distribution using quantile regression and Huber loss. Unlike C51, which uses fixed bins, QR-DQN directly

estimates quantiles, offering greater expressiveness and robustness while maintaining the architectural efficiency of DQN. The spread between quantiles, such as the interquartile range (IQR), provides a built-in measure of uncertainty, which can be valuable in downstream applications like risk aware decision making [10] and robustness assessment [2]. QR-DQN thus merges the scalability of DQN with the flexibility of distributional RL, making it a better fit for evaluating distributional shifts.

Environment Shift Detection. Detecting changes between training and deployment environments is critical for building reliable RL systems. Prior work has studied domain adaptation and meta-learning for adapting to new tasks [9], and bootstrapped ensembles like Bootstrapped DQN [8] provide uncertainty estimates through multiple Q-heads. Bayesian methods [5] decompose uncertainty into epistemic and aleatoric components, but they often require architectural changes. Unlike these approaches, we focus on an unsupervised signal: the quantile spread. This makes our approach lightweight, interpretable, and applicable to deterministic environments where return uncertainty is tightly coupled with familiarity of the encountered states.

4 Methodology

We aim to evaluate whether quantile spread produced by QR-DQN can serve as an indicator for environmental shifts in deterministic reinforcement learning tasks. We perform controlled experiments in the CartPole-v1 environment by changing the most significant parameter (pole length in this case) and analyzing the resulting spread of predicted return over time steps.

4.1 Training Setup

We start by training the QR-DQN agent on the default version of the CartPole-v1 environment, where the pole length parameter has a length of 0.5, and use the Stable Baselines3 Contrib library and the following hyper-parameters: During training, we can also visualize

Parameter	**Value**
policy	`MlpPolicy`
env	`env`
learning_rate	0.0023
buffer_size	100,000
learning_starts	1,000
batch_size	64
gamma	0.99
train_freq	256
gradient_steps	128
exploration_fraction	0.16
exploration_final_eps	0.04
target_update_interval	10
policy_kwargs	`net_arch=[256, 256], n_quantiles=10`
verbose	1

Table 1: Hyper-parameters used for training the agent.

the environment and we leave the agent to train until convergence to ensure performance and high confidence in the predicted return distributions.

Moreover, all experiments are seeded for reproducibility: environment initialization, action sampling and model loading. Thus, consistency is ensured across repeated runs and eliminates variance due to randomness.

4.2 Evaluation Protocol

Once trained, the QR-DQN agent is evaluated across multiple cart pole lengths. More specifically, we test it on $0.1, 0.45, 0.55, 1.0, 2.0, 3.0, 10, 20$. The length of 0.5 is the default pole, and longer and shorter aim to simulate increasingly different versions of the task. Each configuration is evaluated on 100 episodes with at most steps per episodes.

4.3 Quantile Spread Analysis

During evaluation, we extract the predicted quantiles for the chosen action at each time step. We then compute the interquantile range (IQR), defined as the difference between the 75th quantile and 25th quantile. This spread plays the role of a proxy for uncertainty in the agent's return prediction.

To also account for the variation found in episode length, we focus our analysis on the second half of the episode, where the agent usually deviates more from known behavior. Then we compute the average quantile spread over the second half and aggregate the results across episodes to create the mean spread for each pole length.

4.4 Environment Shift Detection

We consider the default pole length (0.5) as a baseline. Then, for each configuration we compare the average second half spread to this baseline. If the difference exceeds a threshold ($\Delta > 0.01$), we classify the environment as different, otherwise we consider it similar.

This rule is used to assess the capability of QRDQN's quantile spread to detect subtle or significant environment changes without retraining or parameter inspection.

4.5 Visualization

We generate plots of the mean spread trajectory over time for each test environment. We also include standard deviation bands and allow for a visual comparison of how the uncertainty evolves as the agent interacts with the environments.

5 Results

The table below summarizes the results:

Pole Length	Mean Spread
0.10	0.5991
0.45	0.1855
0.50	0.1853
0.55	0.1853
1.00	2.1801
2.00	2.6374
3.00	2.1947
10.00	1.4493
20.00	1.0339

Table 2: Mean quantile spread for different pole lengths

5.1 Environment Shift Detection

To test our detection hypothesis, we calculated the difference between the average spread of each environment and the baseline (pole length 0.5). Using a simple decision rule: flag as "DIFFERENT" if the increase in spread exceeded 0.01; we obtained the following classification:

Pole Length	Δ Spread (vs. 0.50)	Shift Detected
0.10	0.4137	**DIFFERENT**
0.45	0.0002	SAME
0.55	-0.0000	SAME
1.00	1.9948	**DIFFERENT**
2.00	2.4521	**DIFFERENT**
3.00	2.0093	**DIFFERENT**
10.00	1.2640	**DIFFERENT**
20.00	0.8486	**DIFFERENT**

Table 3: Environment shift detection based on change in quantile spread relative to pole length 0.50

5.2 Analysis

The quantile spread behaved as hypothesized: as the environment's pole length deviated to a greater extent from the training length of 0.5, the spread increased, reflecting rising uncertainty in the learned value estimates. Figure 2 shows environments where the agent performs reliably, reaching the episode length limit of 500 steps. In these environments, the spread remains small, with negligible differences across neighboring lengths like 0.45 and 0.55. This supports the idea that minor changes in dynamics do not meaningfully affect the learned policy's confidence.

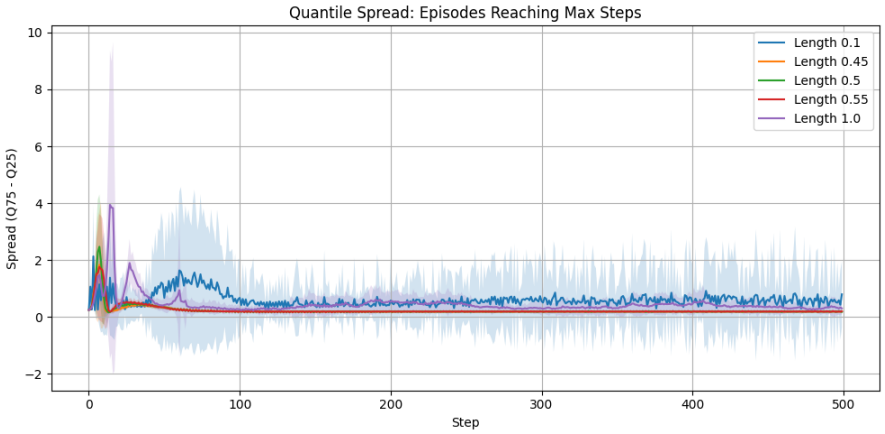

Figure 2: Quantile spread over time for environments where the agent reaches the maximum episode length

On the other hand, Figure 3 visualizes environments in which the agent fails to maintain balance consistently. These unknown conditions (pole lengths 2.0 and above) favor early termination and are associated with a higher spread. The distribution of Q-values becomes wider across steps, compared to the training environment's distribution, indicating increased disagreement among quantiles, and hence indicating a bigger possibility for suboptimal actions being taken.

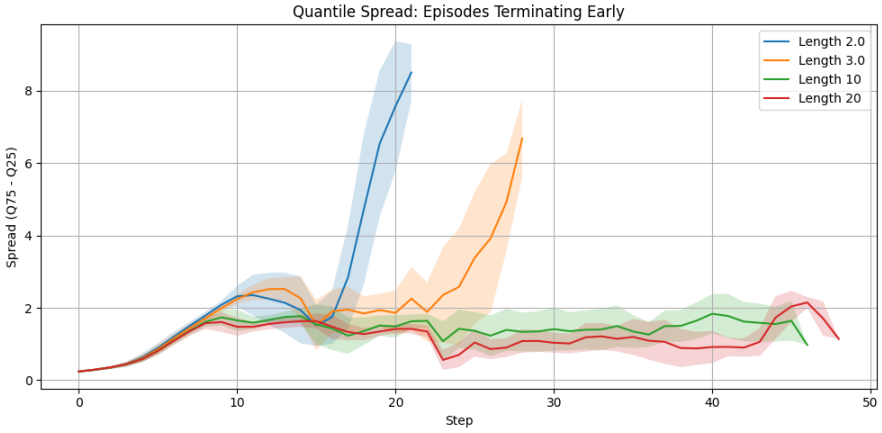

Figure 3: Quantile spread over time for environments where episodes terminate early due to failure

Interestingly, the pole length 0.1 stands out. Even though the agent consistently completes the full 500 steps, the average quantile spread is higher than in other environments that reach the maximum steps. This could be related to the altered dynamics at very short pole lengths, where the system becomes highly responsive to small control inputs. Therefore, even if the agent can balance the pole, the predictions have greater uncertainty.

As a whole, the results align with our initial hypothesis that the spread in QR-DQN conveys meaningful information about changes in environment dynamics. When the environment changes drastically, as with longer pole lengths, the spread increases accordingly. Moreover, the method remains stable under small changes, such as those near the training pole length (0.45 and 0.55). This ability to distinguish between impactful and negligible changes makes it a promising tool for identifying distributional shifts in deterministic environments without overreacting to noise or irrelevant variation.

5.3 Future Work

Even though our paper focuses on deterministic environments, we expect that the use of quantile spread to capture environmental shifts to extend to non-deterministic settings as well. In these types of stochastic environments transitions include an intrinsic randomness, thus making it harder to know if the increased IQR is caused by environment shifts or the variance of the setting. Future work will need to explore how to calibrate and adapt the thresholds in the presence of noise in the environment.

Despite this challenge, being able to interpret quantile spread has great potential for the improvement of agent's robustness in dynamic and safety critical domains. One direction is to use the IQR increases to fall back on a more conservative policy that prioritizes safety while the agent is retrained or adapts in the background. This type of policy can be pretrained on a variety of conditions, but also using heuristics until the agent regains confidence could work. This approach allows the system to work even under environmental shifts and without causing failures.

On top of this, the quantile spread increase could be used as a way to trigger human intervention for the systems. In the event the agent encounters a region where the IQR increases, the system may ask for confirmation or overseeing from a human operator before taking action. This could improve human intervention and make it more scalable by reducing constant monitoring and focusing only on times where spread is anomalous.

Considering the above mentioned, the quantile spread over time of the quantile distribution is not only an indicator of uncertainty in a model, but also a key gateway in providing safety in reinforcement learning systems.

6 Responsible Research

This study is concerned with the analysis of deterministic environmental shift detection in RL using simulated environments such as CartPole. Therefore, it does not involve human subjects, real data or interactions with the physical world, thus minimizing the direct ethical risks.

Nevertheless, we must acknowledge the implications of environmental change detection in RL systems. As more reinforcement learning agents are deployed in different domains, including safety critical ones, the system's ability to detect when agents are operating outside their training distribution is critical for avoiding unsafe decisions.

Another critical aspect is reproducibility. All of the experiments in this work were seeded to ensure that the results were consistent across multiple runs. This includes seeding the environment, model initialization, and seeding during training and testing (seed 420). Seeding was explicitly used to minimize randomness and the setup and parameters used are documented, which helps others replicate the work. All code used for training and evalu-

ation is available at: `https://github.com/Tdr13/research_project_25/tree/paul/rp_paulstan`

Additionally, while writing this report, LLMs were used to selectively rephrase and clarify pieces of information via prompts such as "Rephrase the following part:". However, AI was not involved in designing the experiment or interpreting the results. Prompts were limited to writing support only and all implementations, conclusions and technical ideas were developed independently by the author of the paper.

7 Conclusion

In this work, we investigated the use of quantile spread in QR-DQN as a signal for detecting environment changes. The experiments conducted indicated that the spread reliably increases with shifts in the dynamics and physics of the environment, but also that it remains stable under minor and inconsequential changes. Therefore, this may allow the agent to assess the reliability of its predictions without external validation or access to the environment's parameters. The results support our hypothesis that quantile spread captures uncertainty in an interpretable and useful way.

References

[1] Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. *International Conference on Machine Learning (ICML)*, 2017.

[2] William R Clements, Marc G Bellemare, and Rémi Munos. Estimating uncertainty in deep reinforcement learning. In *arXiv preprint arXiv:1906.09664*, 2019.

[3] Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit quantile networks for distributional reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2018.

[4] Will Dabney, Mark Rowland, Marc G Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. *AAAI Conference on Artificial Intelligence*, 2018.

[5] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International conference on machine learning*, pages 1050–1059, 2016.

[6] Peter J. Huber. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35(1):73–101, 1964.

[7] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[8] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In *Advances in neural information processing systems*, volume 29, pages 4026–4034, 2016.

[9] Kate Rakelly, Aurick Zhou, Deirdre Quillen, Sergey Levine, and Chelsea Finn. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International conference on machine learning*, pages 5331–5340, 2019.

[10] Mark Rowland, Will Dabney, Marc G Bellemare, and Rémi Munos. An analysis of categorical distributional reinforcement learning. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

[11] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2nd edition, 2018.

[12] Ziyu Yang, Zichuan Zhang, and Dong Xu. Fully parameterized quantile function for distributional reinforcement learning. In *Advances in Neural Information Processing Systems*, 2019.