Progress Report

210099V
De Silva DAV

Page Content Overview

1 Introduction

Reinforcement Learning (RL) has become a key part of modern artificial intelligence. It allows agents to learn the best actions by interacting with changing environments. Traditional RL methods like Q-Learning and Deep Q-Networks (DQN) estimate the expected return. This works for maximizing long-term average rewards when risk is not a factor. However by reducing the full range of possible returns to a single average, these methods overlook the natural variability and uncertainty found in real-world tasks.

To overcome this limitation, the concept of Distributional Reinforcement Learning (DRL) was introduced. This approach models the entire distribution of returns instead of just their average. Algorithms like C51 showed the benefits of this view by achieving top performance on the Atari 2600 benchmark suite. Expanding on this idea, the Quantile Regression Deep Q-Network (QR-DQN) proposed by Dabney et al. (2017) offered a solid method for reducing the Wasserstein distance between predicted and target return distributions. QR-DQN surpassed earlier models, showing the advantages of learning value distributions directly.

This project aims to improve QR-DQN with risk-sensitive goals, allowing agents to change their behavior based on different risk profiles. The expected result is an agent that achieves strong average performance and shows stability under uncertainty.

2. Literature Review

2.1 Expected-Value RL to Distributional RL

Traditional reinforcement learning (RL) methods aim to maximize expected returns by viewing reward optimization as a point estimation problem. However, this approach does not account for the uncertainty and risk involved in sequential decision-making in random environments. The rise of distributional reinforcement learning and risk-sensitive learning methods has created new opportunities for building stronger and more applicable RL algorithms.

2.2 Distributional Reinforcement Learning

Distributional reinforcement learning marks a significant change from traditional RL by modeling the entire distribution of returns instead of just their expected values. The key research by Bellemare et al. [1] laid the groundwork for distributional RL. They showed that learning return distributions gives more detailed information than point estimates, which can lead to better performance in different areas.

The distributional Bellman operator which is the foundation of this approach works with probability distributions instead of scalar values. This change allows algorithms to account for uncertainty, multiple outcomes and the complete statistical structure of returns. Later research by Rowland et al. [2] offered a more detailed analysis of the statistical properties and sample complexity of distributional RL algorithms.

Dabney et al. [3] introduced Quantile Regression DQN (QR-DQN). This method uses quantile regression to learn approximations of the return distribution. It has computational benefits over categorical methods while still preserving distributional information, making it especially useful for real-world applications. The quantile regression supplies the distributional information needed for decisions that take risk into account since quantiles are tied directly to risk measures.

2.3 Risk-Sensitive Reinforcement Learning

Risk-sensitive reinforcement learning tackles the issue that traditional RL approaches overlook the risks linked to different actions. Classic RL methods aim to maximize expected returns, but many real-world situations need attention to worst case scenarios and risks. This gap has led to developing various risk measures and optimization methods designed for sequential decision making problems.

Conditional Value-at-Risk (CVaR) has become a key risk measure in the RL literature. Recent theoretical research has determined minimax regret rates for CVaR-based RL, indicating that regret increases with the risk tolerance parameter. Wang et al. [4] offered near-minimax-optimal algorithms for risk-sensitive RL with CVaR goals, showing both theoretical support and practical effectiveness.

Iterated CVaR RL represents a risk-sensitive method that seeks to maximize the tail of reward-to-go at each step. It emphasizes managing risks throughout the entire decision process. This approach is crucial in situations where risk management matters at every stage, not just when looking at the overall expectation of entire episodes [5].

2.4 Integration of Distributional and Risk-Sensitive Approaches

The natural connection between distributional RL and risk-sensitive methods has drawn considerable research interest. Distributional RL provides the essential distributional information needed to calculate risk measures. Meanwhile, risk-sensitive frameworks offer clear approaches for making decisions in uncertain situations. Liang et al. [6] introduced one of the first thorough methods for risk-averse distributional RL using CVaR optimization. Their work showed better performance compared to traditional risk-neutral methods.

The main benefit of this integration is the availability of complete distributional information. This allows for precise computation of various risk measures without additional approximation errors. Unlike methods that try to estimate risk measures indirectly, distributional methods can calculate CVaR, VaR, and other risk metrics directly from the learned return distributions.

2.5 Safety and Robustness Considerations

Recent work has highlighted the use of CVaR as a way to measure risk in safe RL algorithms. This approach has led to better performance and greater robustness in both theory and practice. Yang et al. [7] suggested limiting CVaR to guarantee safety in RL applications, which is especially important in safety-critical fields like autonomous systems and robotics.

Robust risk-sensitive RL has been adapted to address model uncertainty through robust MDPs. This method focuses on optimizing CVaR objectives under the worst-case model scenarios. This dual robustness, which considers both random outcomes and model uncertainty, marks a significant improvement in creating RL systems that can be practically implemented [8]

3. Methodology Outline

3.1 Baseline Algorithm

The baseline is the Quantile Regression Deep Q-Network (QR-DQN) as proposed in the provided arXiv paper (Dabney et al., 2017).

3.2 Proposed Enhancements

Few enhancement techniques will be tried as following:

- **Risk-Sensitive Policy on Learned Quantiles**
  Retain QR-DQN as the distribution estimator and replace mean-based action selection with a risk-aware decision rule computed from the same quantiles.

- **Loss Function Enhancement**
  Keep quantile Huber loss but reweight it to emphasize tails.

- **Architecture Modifications**
  Multi-head outputs: a second lightweight head that can predict a risk-summary.

- **Training Strategy Enhancements**
  Two-stage schedule: pretrain with mean-greedy policy, then fine-tune with the risk-aware decision rule.

3.3 Evaluation

Evaluate on the full **Atari-57** benchmark for headline results (use a smaller subset only for early runs).

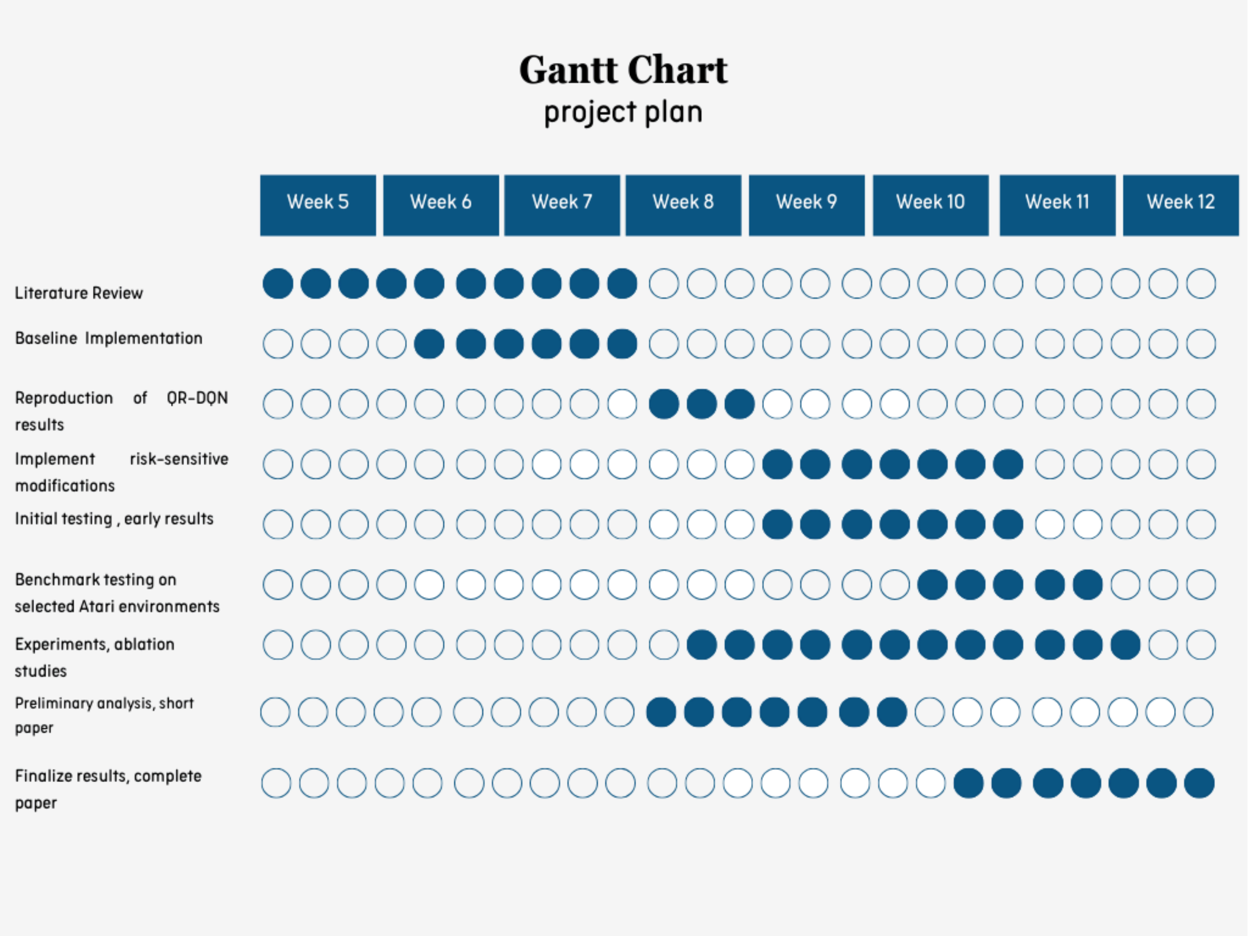
Evaluation metrics

Primary Metric : Suite median human-normalized score under the Best-Agent protocol (also include suite mean). This mirrors QR-DQN's Atari reporting and is directly comparable across agents.

Risk-Sensitive Metrics

- CVaR (worst-case average)
- Worst-k% mean
- Catastrophic-episode rate

4. Project Planning & Timeline



Gantt Chart

project plan

Week 5 | Week 6 | Week 7 | Week 8 | Week 9 | Week 10 | Week 11 | Week 12

Literature Review	●●●●● ●●●●● ○○○○○ ○○○○○ ○○○○○
Baseline Implementation	○○○○● ●●●●● ○○○○○ ○○○○○ ○○○○○
Reproduction of QR-DQN results	○○○○○ ○○○○○ ●●●○○ ○○○○○ ○○○○○
Implement risk-sensitive modifications	○○○○○ ○○○○○ ○○○●● ●●●●● ○○○○○
Initial testing , early results	○○○○○ ○○○○○ ○○○●● ●●●●● ○○○○○
Benchmark testing on selected Atari environments	○○○○○ ○○○○○ ○○○○○ ○○●●● ●●○○○
Experiments, ablation studies	○○○○○ ○○○○○ ○●●●● ●●●●● ●●●○○
Preliminary analysis, short paper	○○○○○ ○○○○○ ●●●●● ●●○○○ ○○○○○
Finalize results, complete paper	○○○○○ ○○○○○ ○○○○○ ○○○●● ●●●●●

5. Project Planning

5.1 Literature Review

Establish background and gap for risk-sensitive distributional RL. Noting the following:

- Key papers (DRL, QR-DQN, CVaR/risk-aware control)
- Research gaps
- Risk metrics
- Evaluation planning

5.2 Baseline Implementation

Implement QR-DQN baseline with reproducible training loops, quantile loss, target network updates, and reliable logging.

5.3 Reproduce QR-DQN Results

Verify the baseline against a few Atari games with multiple seeds to ensure behavior is consistent with published trends.

5.4 Implement Risk-Sensitive Modifications

Integrate the proposed enhancement techniques (mentioned in section 3.2) with the baseline.

5.5 Initial Testing and Early Results

Run small experiments on a few games to confirm the correctness and stability of the risk-aware enhancements. This stage checks that the modification works as intended, improving reliability without unexpectedly lowering average performance.

5.6 Benchmark Testing on a Selected Atari Subset

Scale to a representative group of 10 to 15 Atari games using the finalized protocols. For each seed and game, evaluate frozen checkpoints to produce Best-Agent scores.

5.7 Experiments and Ablations

Systematically explore what drives improvements by adjusting the risk level, comparing different decision rules. Tweak optional components. Perform light tuning of hyperparameters including learning-rate schedules, optimizers. Evaluate efficiency using metrics like AUC and frames-to-threshold, along with stability measured by variance and catastrophes. This analysis helps to choose a fixed configuration based on headline and risk metrics.

5.8 Preliminary Analysis and Short Paper

Consolidate subset results and ablations into a clear narrative. Put together a concise mid-evaluation paper that details methods, findings, limitations and the setup chosen evaluation.

5.9 Finalize Results and Complete Paper

Run the locked configuration on the full Atari-57 suite. Finalize the evaluation analyses. Produce the final paper.

5. References

[1] M. G. Bellemare, W. Dabney, and R. Munos, "A distributional perspective on reinforcement learning," in *Proc. 34th Int. Conf. Machine Learning*, vol. 70, 2017, pp. 449-458.

[2] M. Rowland, M. Bellemare, W. Dabney, R. Munos, and Y. W. Teh, "An analysis of categorical distributional reinforcement learning," in *Proc. 21st Int. Conf. Artificial Intelligence and Statistics*, vol. 84, 2018, pp. 29-37.

[3] W. Dabney, M. Rowland, M. G. Bellemare, and R. Munos, "Distributional reinforcement learning with quantile regression," in *Proc. 32nd AAAI Conf. Artificial Intelligence*, 2018, pp. 2892-2901.

[4] R. Wang, R. Yang, C. Jin, and Z. Wang, "Near-minimax-optimal risk-sensitive reinforcement learning with CVaR," in *Proc. 40th Int. Conf. Machine Learning*, vol. 202, 2023, pp. 36207-36256.

[5] Y. Liang, M. Xu, and T. Zhang, "Provably efficient risk-sensitive reinforcement learning: Iterated CVaR and worst path," *arXiv preprint arXiv:2206.02678*, 2022.

[6] J. Liang, O. Nachum, M. Ghavamzadeh, and S. Levine, "Risk-averse distributional reinforcement learning: A CVaR optimization approach," *IEEE Trans. Neural Networks and Learning Systems*, vol. 31, no. 12, pp. 5796-5809, 2020.

[7] T. Yang, L. Feng, B. Zhang, and J. Luo, "Towards safe reinforcement learning via constraining conditional value-at-risk," in *Proc. 35th AAAI Conf. Artificial Intelligence*, 2021, pp. 12282-12289.

[8] H. Xu, Y. Li, and L. Xie, "Robust risk-sensitive reinforcement learning with conditional value-at-risk," *arXiv preprint arXiv:2405.01718*, 2024.