Quantile Geometry Regularization for Distributional Reinforcement Learning

Zhaofan Zhang[1], **Minghao Yang**[1], **Rufeng Chen**[1], **Sihong Xie**[1]*, **Hui Xiong**[1,2]*
[1]Thrust of Artificial Intelligence,
The Hong Kong University of Science and Technology (Guangzhou), China
[2]Department of Computer Science and Engineering,
The Hong Kong University of Science and Technology, Hong Kong SAR, China
{zzhang398,myang272,rchen514}@connect.hkust-gz.edu.cn;
sihongxie@hkust-gz.edu.cn; xionghui@ust.hk

Abstract

Quantile-based distributional reinforcement learning methods learn return distributions through sampled quantile regression, but their bootstrapped target quantiles may induce distorted or degenerate distribution estimates. We propose Robust Quantile-based Implicit Quantile Networks (RQIQN), a lightweight Wasserstein distributionally robust enhancement boosted from a quantile estimation perspective. We first reinterpret a snapshot of IQN loss as a collection of local empirical quantile estimation problems over sampled current fractions. We then formulate each local fitting problem through Wasserstein distributionally robust quantile estimation, deriving a closed-form, fraction-dependent correction. For stable learning, we adopt a bounded adaptation that regularizes quantile geometry and empirically mitigates distribution degeneration without changing the underlying value objective or requiring additional sample-set reconstruction. Finally, experiments on a chain MDP demonstrate RQIQN's improved distributional estimation accuracy, while evaluations on risk-sensitive navigation and Atari games show that RQIQN outperforms IQN on most tasks and achieves competitive performance against other quantile-based distributional reinforcement learning methods.

1 Introduction

Distributional reinforcement learning (DistRL) models the full distribution of discounted returns rather than only its expectation, providing a richer basis for learning under stochasticity and risk. A useful way to understand practical DistRL algorithms is through the distinction between return distributions and statistics of those distributions. The distributional Bellman operator acts on full return distributions, whereas scalable algorithms usually propagate finite-dimensional summaries, such as atoms, quantiles, or expectiles. Existing DistRL algorithms differ mainly in how they represent and update return distributions, including categorical atoms, quantile functions, expectiles, and sample-set representations. Among these choices, quantile-based methods are particularly appealing because quantile functions naturally align with Wasserstein geometry, do not require a fixed value support, and provide direct access to different regions of the return distribution. Within this scope, Implicit Quantile Networks (IQN) [1] further extends QR-DQN [2] with fixed-grid quantiles by learning a continuous quantile function through sampled quantile fractions $\tau \sim U([0, 1])$. Generally, a set of statistics is Bellman-closed if the statistics of the Bellman-updated distribution can be computed solely from the same statistics of the next-state distribution and the reward, without reconstructing

*Corresponding authors.
The source code of RQIQN is available at https://github.com/FANnn100/RQIQN.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

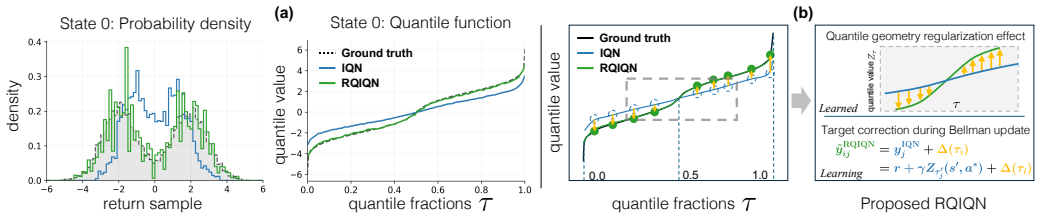

Figure 1: Illustration of **(a)** distribution degeneration at state 0 and **(b)** quantile geometry modulation under the exact Wasserstein-robust correction. The fitted return distributions in **(a)** are obtained from a *four-state chain MDP* with deterministic, sequential transitions and a single action. Rewards are zero except at the terminal state, where $r \sim \frac{1}{2}\mathcal{N}(-2,1) + \frac{1}{2}\mathcal{N}(2,1)$.

the full distribution. This nature is crucial since it allows recursive dynamic programming (defined in Eq. (1)) in the chosen statistic space. However, finite quantile statistics are not Bellman-closed [3], which can lead to distribution degeneration and biased estimates of distributional spread (Figure 1(a)).

In quantile-based DistRL, deep Q-learning [4] is extended from scalar value regression to distributional quantile fitting. For each transition, the agent forms pairwise TD errors between current quantile estimates and bootstrapped target quantiles, and minimizes a quantile regression loss. Thus, the Bellman update has an explicit quantile-estimation interpretation, with each sampled fraction corresponding to a local quantile fitting problem. In this work, we revisit IQN through the lens of local quantile estimation. Specifically, the optimization objective of IQN can be interpreted exactly at the empirical-loss level as a collection of empirical quantile-estimation slots, one for each sampled current quantile fraction τ_i. Each slot fits the τ_i located quantile of the bootstrapped Bellman target distribution induced by target quantile fractions $\{\tau'_j\}_{j=1}^{N'}$. This view reveals that IQN performs local quantile estimation over a model-generated empirical target distribution formed by finitely many target quantile values. From this perspective, distribution degeneration can be understood as accumulated bias in the learned quantile values, which distorts the geometry of the represented return.

We propose RQIQN (Figure 1(b)) to address the distribution degeneration problem, namely biased quantile representation, by robustifying each local quantile fitting problem against worst-case perturbations of the empirical Bellman target distribution within a Wasserstein ambiguity set. The resulting closed-form quantile correction regularizes the learned return distribution by stabilizing finite-sample quantile estimates. Rather than modifying the distributional Bellman operator or the return-distribution representation as in existing work [3, 5, 6], RQIQN preserves the implicit quantile representation and robustifies each local quantile-regression slot. For the check-loss formulation[2], this robust slot admits a closed-form fraction-dependent correction $\Delta_p(\tau;\epsilon)$, where p is the Wasserstein order, and ϵ is the robustness radius that decays over time. Geometrically, the correction is median-antisymmetric and monotone in τ. Hence it preserves the uniform quantile average while expanding upper–lower quantile gaps, providing a mean-neutral regularization of distributional spread.

1.1 Related Work

DistRL has developed rapidly as a powerful alternative to expectation-based value learning, with representative approaches modeling return distributions using categorical atoms [7, 8], quantiles [2, 1, 9], expectiles [3, 5], sample or particle-based representations [6, 10]. Among them, IQN [1] has become particularly influential due to its flexible implicit quantile-function representation, which supports arbitrary quantile sampling and naturally enables risk-sensitive policies through distortion risk measures. Therefore, it has been widely applied in tasks that require risk-sensitive decision-making, such as autonomous navigation [11, 12, 13, 14] and locomotion [15]. However, IQN still inherits a limitation of quantile-based DistRL: finite quantile statistics are not Bellman-closed.

Several lines of work have addressed the difficulty caused by non-Bellman-closed distributional statistics by either changing the learned statistics or introducing an explicit statistics-to-samples interface. Expectiles, introduced as asymmetric least-squares location statistics [16], are attractive because they provide smoother L_2-based fitting than check-loss quantile regression, and their symmetric case $\tau = 0.5$ coincides with the mean, which is useful for stable value estimation. However, a finite set of

[2]Namely, quantile loss. It's different from quantile Huber loss (Huberized quantile regression loss).

expectiles is still not Bellman-closed in general, and expectile values cannot be directly interpreted as samples from the return distribution. To make expectiles usable for Bellman backups, ER-DQN [3] recover Bellman target samples through solving costly nonlinear systems. IEQN [5] instead jointly learns expectiles and quantiles to retain efficient L_2-based learning while avoiding explicit imputation, at the cost of additional prediction heads and auxiliary coupling losses. Sample-set methods, such as MMDQN [6] and MWG [10], take a different route by operating on explicit return samples, thereby bypassing the need to propagate non-closed statistics as samples. However, deterministic sample sets may limit stochastic target diversity, while Gaussian-mixture variants rely on EM-style projection [17] and sample augmentation. Besides, another line of work enhances the statistical validity of distributional estimates such as monotonicity [18, 19, 20, 21], by introducing constraints or additional network structure. These works either do not address the degeneration problem from Bellman non-closure or require additional computational and algorithmic complexity.

2 Problem Setup and Main Analysis

2.1 Distributional Reinforcement Learning

We consider a Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathbb{R}, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, $\mathcal{P}(\cdot \mid s, a)$ is the transition kernel, $\mathbb{R}(\cdot \mid s, a)$ denotes the reward distribution, and $\gamma \in [0, 1)$ is the discount factor. For a policy π, the state-action return is the random variable $Z^\pi(s, a) \overset{D}{=} \sum_{t=0}^{\infty} \gamma^t R_t$, $S_0 = s$, $A_0 = a$, where $\overset{D}{=}$ denotes equality in distribution and the transition process can be explained by $S_{t+1} \sim \mathcal{P}(\cdot \mid S_t, A_t)$, $R_t \sim \mathbb{R}(\cdot \mid S_t, A_t)$, $A_t \sim \pi(\cdot \mid S_t)$ for $t \geq 1$. Standard reinforcement learning typically optimizes only the first moment of this return distribution. In particular, the action-value function is defined as $Q^\pi(s, a) = \mathbb{E}_Z[Z^\pi(s, a)]$. The optimal action-value function satisfies the Bellman optimality equation $Q(s, a) = \mathbb{E}_{R \sim \mathbb{R}(\cdot|s,a),\, S' \sim \mathcal{P}(\cdot|s,a)}[R + \gamma \max_{a' \in \mathcal{A}} Q(S', a') \,|\, S = s, A = a]$. This expectation-based formulation compresses the full return distribution into a single scalar, discarding distributional information induced by stochastic transitions and rewards.

DistRL extends standard RL by modeling the *return distribution* Z instead of only its expectation. Formally, Z can be updated by dynamic programming, with distributional Bellman operator[3] defined by

$$\mathcal{T}^\pi Z(s, a) \overset{D}{=} R(s, a) + \gamma Z(S', A'). \tag{1}$$

We can compute the optimal return distribution by using the distributional Bellman optimality operator $\mathcal{T}$ defined as

$$\mathcal{T} Z(s, a) \overset{D}{=} R(s, a) + \gamma Z\left(S', a^*\right), \qquad a^* = \operatorname*{argmax}_{a'} \mathbb{E}_Z[Z(S', a')]. \tag{2}$$

As a representative quantile-based DistRL method, Implicit Quantile Networks (IQN) [1] provides a flexible distributional representation through the quantile function of the return distribution. Let $F^{-1}_{Z(s,a)}(\tau)$ denote the quantile function of the random return $Z(s, a)$ at quantile fraction τ. For notational simplicity, we write $Z_\tau(s, a) := F^{-1}_{Z(s,a)}(\tau)$, so that when $\tau \sim U([0, 1])$, the sampled quantile value $Z_\tau(s, a)$ follows the return distribution $Z(s, a)$. Based on this implicit quantile representation, IQN parameterizes this quantile function with a neural network $Z_\tau(x, a; \theta) \approx Z_\tau(s, a)$. For brevity, we use $Z_\tau(s, a)$ directly. Since IQN samples quantile fractions explicitly, risk-sensitive behavior can be induced by modifying the quantile sampling range. A common choice is the conditional value-at-risk (CVaR) distortion, which maps a uniformly sampled fraction $\tau \sim U([0, 1])$ to $\tilde{\tau} = \eta\tau$, thereby restricting sampling to the lower-tail quantile region $U([0, \eta])$ for risk-averse control. In this work, unless explicitly stated, fractions are sampled without distortion.

Throughout this work, we consider two quantile regression losses related to DistRL. The first is the check loss

$$\rho_\tau(u) = u\left(\tau - \mathbb{1}_{\{u<0\}}\right) = \left|\tau - \mathbb{1}_{\{u<0\}}\right| |u|,$$

which gives the exact quantile-regression objective. The second is the standard quantile Huber [22] loss used in practical QR-DQN/IQN implementations:

$$\rho^\kappa_\tau(u) = \left|\tau - \mathbb{1}_{\{u<0\}}\right| \frac{\mathcal{H}_\kappa(u)}{\kappa}, \qquad \mathcal{H}_\kappa(u) = \begin{cases} \frac{1}{2}u^2, & |u| \leq \kappa, \\ \kappa\left(|u| - \frac{1}{2}\kappa\right), & |u| > \kappa. \end{cases}$$

[3] Uppercase letters S and A denote random variables, and lowercase letters denote their realized values.

Here, $\kappa > 0$ is the Huber threshold. The check loss preserves the exact quantile-estimation interpretation, while the Huberized loss provides a smoother surrogate for numerical stability and achieving stronger overall return performance.

2.2 Distribution Degeneration with Quantile-based Representation

As discussed above, finite quantile sets are not Bellman-closed, and quantiles are statistics rather than samples. Thus, unless the Bellman-updated distribution is explicitly projected back to the quantile representation or reconstructed through an imputed target distribution, approximation bias may accumulate and distort the learned return distribution [3]. In practice, this issue is exacerbated by the Huberized loss, which no longer preserves the exact guarantees of quantile regression. The learned distribution may consequently collapse toward its mean, discarding tail information.

The degeneration risk of return distribution motivates a robustness-oriented view of quantile-based distributional learning, in which mitigating degeneration amounts to stabilizing the quantile estimates produced by the process of Bellman updates. In Figure 1, IQN (depicted in blue) suffers from worse variance degradation, whereas the RQIQN demonstrates robustness in terms of distribution geometry. The detailed formulation of RQIQN is provided in Section 3, and the complete four-state chain experimental results with are reported in Appendix C.

2.3 The Quantile Estimation Inspired Loss

Given a transition (s, a, r, s'), IQN samples *current fractions* $\{\tau_i\}_{i=1}^{N}$ for the quantile levels to be fitted at (s, a) and *target fractions* $\{\tau'_j\}_{j=1}^{N'}$ for bootstrapped next-state quantile targets, both from $U([0, 1])$, and defines the IQN loss with TD error δ_{ij} as

$$\mathcal{L}_{\mathrm{IQN}}(s, a, r, s') = \frac{1}{N'} \sum_{i=1}^{N} \sum_{j=1}^{N'} \rho_{\tau_i}^{\kappa}(\delta_{ij}) = \sum_{i=1}^{N} \mathbb{E}_{\tau'} \left[\rho_{\tau_i}^{\kappa}(\delta_{ij}) \right], \quad \delta_{ij} = r + \gamma Z_{\tau'_j}(s', a') - Z_{\tau_i}(s, a). \tag{3}$$

Similar to the TD loss in deep Q-learning [4], it is mentioned in QR-DQN [2] that a nature in the loss function of quantile-based DistRL is to employ quantile regression temporal difference learning (QRTD). The update is computed for all pairs of $(Z_{\tau'_j}(s', a'), Z_{\tau_i}(s, a))$. At its essence, quantile estimation seeks the value corresponding to a given quantile level τ from samples drawn from a distribution. With respect to the TD error in Eq. (3), if the quantile fraction τ_i is fixed, the update target is regarded as a sample, then the update can be interpreted as estimating the quantile at level τ_i. Therefore, the overall IQN loss performs quantile estimation over all sampled quantile levels during the distributional Bellman update, which in turn yields a representation of the return distribution.

Lemma 1 (Snapshot of Implicit Quantile Network loss as quantile estimation)**.** *For any transition (s, a, r, s'), consider the check-loss counterpart of IQN snapshot $\mathbb{E}_{\tau'} \left[\rho_{\tau_i}(\delta_{ij}) \right]$ in Eq.* (3) *with sampled target fractions $\tau'_j \sim U([0, 1])$ and a fixed current fraction $\tau_i \in (0, 1)$:*

$$\delta_{ij} = y_j - q_i, \qquad y_j := r + \gamma Z_{\tau'_j}(s', a'), \qquad q_i := Z_{\tau_i}(s, a).$$

Let the empirical Bellman target distribution be $\widehat{\mu}_{s,a} = \frac{1}{N'} \sum_{j=1}^{N'} \delta_{y_j}$, where δ_{y_j} denotes a Dirac measure at y_j. For fixed τ_i and targets $\{y_j\}_{j=1}^{N'}$, the corresponding check-loss component is $\frac{1}{N'} \sum_{j=1}^{N'} \rho_{\tau_i}(y_j - q) = \mathbb{E}_{Y \sim \widehat{\mu}_{s,a}}[\rho_{\tau_i}(Y - q)]$, with $q = Z_{\tau_i}(s, a)$. Viewing q as a free scalar therefore yields the empirical τ_i-quantile estimation problem:

$$q^0_{\tau_i} \in \arg\min_{q \in \mathbb{R}} \mathbb{E}_{Y \sim \widehat{\mu}_{s,a}} \left[\rho_{\tau_i}(Y - q) \right], \tag{4}$$

where $\rho_\tau(u) = u(\tau - \mathbb{1}_{\{u<0\}})$ and any minimizer $q^0_{\tau_i}$ satisfies the empirical coverage condition

$$\mathbb{P}_{Y \sim \hat{\mu}_{s,a}}(Y < q^0_{\tau_i}) \le \tau_i \le \mathbb{P}_{Y \sim \hat{\mu}_{s,a}}(Y \le q^0_{\tau_i}).$$

Lemma 1 shows that the check-loss IQN TD objective can be interpreted, at the empirical-loss level, as a collection of local quantile estimation problems over sampled fractions $\{\tau_i\}_{i=1}^{N}$.

2.4 Robustness of Quantile Geometry Regularization

Building on the degeneration analysis in Section 2.2 and the local quantile-estimation view formalized in Lemma 1, we interpret distributional degeneration as the accumulation of errors in local quantile estimates during Bellman target fitting. To address this problem, we formulate each local fitting as a Wasserstein-robust quantile estimation centered at the empirical Bellman target distribution.

Robust quantile estimation [23, 24, 25] has been extensively studied in statistics literature, and is closely related to the robustness of IQN loss at a specific snapshot. More recently, Wasserstein distributionally robust quantile regression (WDRQR) [26] shows that, under a type-p Wasserstein ambiguity set, distributional robustness induces an exact regularized reformulation for the check-loss quantile regression objective. More importantly, this robustness effect is not merely a global penalty. The robust solution differs from the corresponding regularized nominal solution through a radius-dependent location adjustment, while the check loss is essentially the unique convex loss class that admits such an additive Wasserstein regularization phenomenon. Next, we present Theorem 1 in a concise form based on WDRQR, making its connection to the statistical intuition more explicit.

Theorem 1 (Wasserstein-robust local quantile correction)**.** *For $\epsilon \geq 0$, $p \in (1, \infty]$, consider the Wasserstein distributionally robust quantile estimation slot at τ_i:*

$$q_{\tau_i}^{\epsilon} \in \arg\min_{q \in \mathbb{R}} \sup_{\nu \in B_p(\hat{\mu}_{s,a}, \epsilon)} \mathbb{E}_{Y \sim \nu}[\rho_{\tau_i}(Y - q)], \tag{5}$$

where $B_p(\hat{\mu}_{s,a}, \epsilon)$ denotes the type-p Wasserstein ambiguity set $B_p(\hat{\mu}_{s,a}, \epsilon) := \{\nu \in \mathcal{P}(\mathbb{R}) : W_p(\nu, \hat{\mu}_{s,a}) \leq \epsilon\}$. For each sampled fraction τ_i, any robust minimizer admits the explicit location correction

$$q_{\tau_i}^{\epsilon} = q_{\tau_i}^{0} + \Delta_p(\tau_i, \epsilon), \tag{6}$$

where the robust term Δ_p can be expressed as

$$\Delta_p(\tau, \epsilon) = \frac{\epsilon}{q}\left(\tau^q - (1-\tau)^q\right) c_{\tau,p}^{1-q}, \qquad \frac{1}{p} + \frac{1}{q} = 1, \tag{7}$$

and

$$c_{\tau,p} = \begin{cases} \left(\tau^q(1-\tau) + \tau(1-\tau)^q\right)^{1/q}, & p \in (1, \infty), \\ 2\tau(1-\tau), & p = \infty. \end{cases} \tag{8}$$

For each sampled fraction τ_i, the Wasserstein distributionally robust quantile estimation problem in Theorem 1 can be written as $q_{\tau_i}^{\epsilon} \in \arg\min_{q \in \mathbb{R}} \frac{1}{N'} \sum_{j=1}^{N'} \rho_{\tau_i}(y_j + \Delta_p(\tau_i, \epsilon) - q)$.

Standard IQN estimates $q_{\tau_i}^{0}$ by treating the bootstrapped target values $\{y_j\}_{j=1}^{N'}$ as empirical samples from the Bellman target distribution. In contrast, our goal is to estimate a quantile that remains stable under local distributional perturbations of the Bellman target. Thus, instead of directly robustifying the infinite-dimensional return distribution, we robustify the scalar quantile estimator that represents each local slice of the learned quantile function.

Remark 1 (Mean-neutral geometry modulation)**.** *The robust correction term in Eq.* (7) *is antisymmetric around the median: $\Delta_p(1-\tau; \epsilon) = -\Delta_p(\tau; \epsilon), \Delta_p(1/2; \epsilon) = 0$. Hence, under uniform quantile sampling, $\mathbb{E}_{\tau \sim \mathcal{U}(0,1)}[\Delta_p(\tau; \epsilon)] = \int_0^1 \Delta_p(\tau; \epsilon) d\tau = 0$. Therefore, the robust correction preserves the risk-neutral quantile average $\mathbb{E}_\tau\left[Z_\tau(s,a) + \Delta_p(\tau; \epsilon)\right] = \mathbb{E}_\tau\left[Z_\tau(s,a)\right]$. Moreover, since $\Delta_p(\tau; \epsilon)$ is nondecreasing in τ, for any $0 < \tau_l < \tau_h < 1$, $[Z_{\tau_h}(s,a) + \Delta_p(\tau_h; \epsilon)] - [Z_{\tau_l}(s,a) + \Delta_p(\tau_l; \epsilon)] = Z_{\tau_h}(s,a) - Z_{\tau_l}(s,a) + \Delta_p(\tau_h; \epsilon) - \Delta_p(\tau_l; \epsilon)$, with non-negative item $\Delta_p(\tau_h; \epsilon) - \Delta_p(\tau_l; \epsilon) \geq 0$. Thus, the robust correction expands upper–lower quantile gaps while leaving the quantile average unchanged in expectation.*

The corresponding proofs for this section are provided in Appendix A.

3 Algorithm

In this section, we propose the practical algorithm for **RQIQN**. It can be readily adapted to other variants of QRTD based DistRL.

Algorithm 1: Robust Quantile-based Implicit Quantile Network Loss

Input: (s, a, r, s'), $\gamma \in [0, 1)$, N, N', K, timestep $t > 0$, $\epsilon_t \geq 0$, $p \in (1, \infty]$, β, Z

$\tilde{\tau}_k \sim \beta(\cdot), \quad k = 1, \ldots, K$ // Sample fractions for action selection

$a^* \leftarrow \arg\max_{a' \in \mathcal{A}} \frac{1}{K} \sum_{k=1}^{K} Z_{\tilde{\tau}_k}(s', a')$ // Select greedy next action

$\tau_i, \tau'_j \sim \mathcal{U}([0, 1]), \quad i = 1, \ldots, N, \ j = 1, \ldots, N'$ // Sample quantile fractions

$\Delta_i \leftarrow \Delta_p(\tau_i; \epsilon_t)$ // Compute robust local correction

$\widetilde{\delta}_{ij} \leftarrow r + \gamma Z_{\tau'_j}(s', a^*) + \Delta_i - Z_{\tau_i}(s, a), \quad \forall i, j$ // Compute robust TD error

Output: $\frac{1}{N'} \sum_{i=1}^{N} \sum_{j=1}^{N'} \rho_{\tau_i} \left(\widetilde{\delta}_{ij} \right)$

Corollary 1 (The RQIQN Loss)**.** *Following the form of Eq.* (6)*, robust quantile-regression-based loss function for RQIQN is given by*

$$\mathcal{L}_{\mathrm{RQIQN}}(s, a, r, s') = \frac{1}{N'} \sum_{i=1}^{N} \sum_{j=1}^{N'} \rho_{\tau_i} \Big(r + \gamma Z_{\tau'_j}(s', a^*) + \Delta_p(\tau_i, \epsilon) - Z_{\tau_i}(s, a) \Big). \tag{9}$$

A key issue arises when correction-related term $c_{\tau,p}$ in Eq. (8) is applied to continuously sampled quantile fractions in RQIQN. For $p \in (1, \infty)$, the closed-form correction may become unbounded as $\tau \to 0$ or 1. This is undesirable for RQIQN distributional learning, where quantile fractions are sampled from a continuous space and bootstrapped targets are affected by non-stationary network updates. Unbounded endpoint corrections may introduce numerical instability and amplify noisy tail estimates. To address this problem, we then design an applicable correction form.

As discussed in Remark 1, the robust correction provides stronger correction intensity near the distributional tails, where quantiles correspond to extreme return outcomes and are more sensitive to finite-sample uncertainty. In our implementation, we focus on two representative cases, $p = 2$ and $p = \infty$. The case $p = \infty$ yields the bounded correction $\Delta_\infty(\tau; \epsilon) = \epsilon(2\tau - 1)$. For $p = 2$, the raw correction contains a denominator that vanishes near the endpoints. We therefore use a bounded adaptation by reversing the endpoint behavior of the denominator:

$$\Delta_2(\tau; \epsilon) = \frac{\epsilon}{2} \frac{1 - 2\tau}{\sqrt{\tau^2 + (1 - \tau)^2}}.$$

This adapted term remains finite for all $\tau \in [0, 1]$, while preserving the median-antisymmetry and monotonicity required by the geometric interpretation in Remark 1. For numerical stability, we adopt this bounded adaptation of the $p = 2$ correction. This practical form preserves median antisymmetry and induces a mean-neutral contraction of quantile gaps, with stronger local contraction near the median than in the tails.

Statistically, WDRQR [26] suggests a power-law shrinkage of the Wasserstein radius with the empirical sample size, i.e., $\epsilon_N = O(N^{-1/2})$, while DistRL methods induce non-stationary empirical targets through replay sampling, bootstrapping, and network updates. We therefore decouple ϵ from the nominal batch size and use a decay schedule, preserving strong early-stage robustness and gradually annealing the correction as training stabilizes. Specifically, to implement adaptive robustness during training rather than using a fixed radius ϵ, we adopt a time-dependent reverse-logistic decay schedule

$$\epsilon_t = \frac{\epsilon_0}{1 + \exp(k(t - t_0))},$$

where t is the training step, ϵ_0 denotes the initial robustness scale, $k > 0$ controls the decay sharpness, and t_0 specifies the midpoint of the reverse-logistic transition. For navigation and Atari, the time-dependent reverse-logistic factor is additionally multiplied by the absolute mean of the current quantile predictions for the sampled state-action pair.

Check loss versus Huberized loss. Recent WDRQR theory shows that, for $p > 1$, the check loss is essentially the unique convex loss that admits an exact additive Wasserstein regularization under a location-adjusted objective. The quantile Huber loss, although standard in quantile-based methods like QR-DQN and IQN, is not an affine transformation of the check loss because it is locally quadratic

and globally linear. Therefore, the exact distributionally robust quantile regression equivalence is not feasible to extend to the Huber variant.

Corollary 1 characterizes RQIQN under the original check loss, we also adopt a variant with Huberized quantile regression objective for improved optimization stability. Specifically, given N current quantile fractions and N' target quantile fractions, the RQIQN-Huber loss is defined as

$$\mathcal{L}_{\text{RQIQN-Huber}} = \frac{1}{N'} \sum_{i=1}^{N} \sum_{j=1}^{N'} \rho_{\tau_i}^{\kappa} \left(\widetilde{\delta}_{ij} \right), \quad \widetilde{\delta}_{ij} = r + \gamma Z_{\tau'_j}(s', a^*) + \Delta_i - Z_{\tau_i}(s, a).$$

4 Empirical Results

In this section, we report experimental results. *Additional implementation details and experimental results are provided in Appendix B, with robustness decay schedule studies presented in Appendix C.2.*

4.1 Risk-Sensitive Control: Autonomous Surface Vehicle Navigation

IQN has emerged as a powerful paradigm for robotics tasks [13] such as uncertainty-aware autonomous navigation. We evaluate RQIQN in a representative safety-critical marine navigation environment [12], where static obstacles, and vortex-induced flow disturbances are simulated according to physically motivated environmental settings [27]. Following the original learning-based evaluation protocol, we deploy the learned policy for unmanned surface vehicle control and compare RQIQN against IQN and DQN [28].

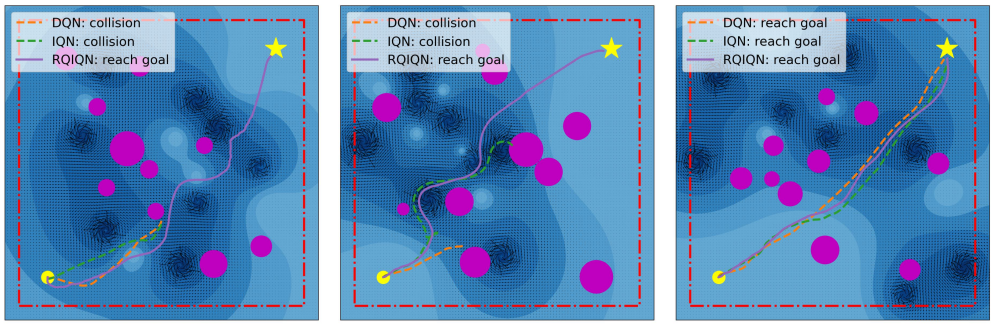

Figure 2: Qualitative trajectory results of RL agents. The yellow circle denotes the start position, and the yellow star denotes the goal. Magenta circles indicate static obstacles, while the background vector field represents vortex-induced flow disturbances. Trajectories for IQN and RQIQN are shown under the adaptive setting, where both achieve stronger performance.

Specifically, the RL agent receives observations encoding the vessel state and sonar-based frontal obstacle distances without environmental prior knowledge, and outputs discrete motion-control actions for goal-directed navigation. The reward encourages goal-reaching progress and safe navigation. The objective is to learn a policy that reaches the target efficiently while avoiding collisions and maintaining robust behavior under invisible flow-induced disturbances. Due to space limitations, the full MDP formulation and other experimental details are provided in Appendix B.2.

In this study, the default IQN and RQIQN agents use the natural full quantile support, i.e., $\tau \sim U([0, 1])$. Following the existing navigation task, we additionally evaluate the *adaptive variants* in Table 1, where the CVaR threshold η is adjusted according to the perceived distance to the nearest obstacle. Smaller obstacle distances induce more conservative lower-tail sampling and RQIQN under both sampling reaches improved navigation results and safety. As shown in Figure 2, RQIQN remains robust in dense-obstacle scenarios where vortex-induced flows perturb actions and state transitions. To further assess the feasibility of extending our method to continuous action space and the benefits of its distributionally robust formulation for actor–critic algorithms, we present additional comparative experiments in Appendix C.3.

Table 1: Evaluation under the *hard-mode* navigation setting with 10 static obstacles and 8 vortex cores. Results are reported as mean ± standard deviation over 5 independently trained seeds, with each seed evaluated on 500 episodes using different environment layouts.

Method	Success(%) ↑	Collision(%) ↓	Timeout(%) ↓	Return ↑	$\text{Time}_{\text{succ}}$(s) ↓	$\text{Energy}_{\text{succ}}$ ↓
DQN	67.24 ± 4.79	32.72 ± 4.77	**0.04 ± 0.09**	12.97 ± 3.65	**33.48 ± 0.33**	**98.95 ± 5.76**
IQN	64.80 ± 22.66	21.92 ± 9.54	13.28 ± 29.47	-13.17 ± 51.78	39.08 ± 1.22	122.56 ± 17.29
+Adaptive	71.04 ± 23.35	15.44 ± 6.13	13.52 ± 29.11	-12.19 ± 51.20	42.24 ± 1.55	133.31 ± 17.04
RQIQN	83.24 ± 1.19	16.48 ± 1.14	0.28 ± 0.27	**19.20 ± 1.73**	37.55 ± 1.25	111.24 ± 6.34
+Adaptive	**85.64 ± 2.27**	**14.20 ± 2.02**	0.16 ± 0.26	18.57 ± 3.90	39.19 ± 1.87	115.83 ± 6.41

4.2 Atari Games

We compare our algorithm against several representative quantile-based DistRL baselines that have demonstrated strong performance on standard RL benchmarks. DQN [28] serves as the expectation-based baseline that learns only scalar action values. C51 [7] represents the return distribution with a categorical distribution over fixed supports. QR-DQN[2] learns a fixed set of quantile locations via quantile regression, while IQN extends this idea by sampling quantile fractions to approximate an implicit quantile function. Rainbow [8] is included as a strong aggregate baseline that combines C51 with several orthogonal improvements to DQN. RQIQN is evaluated as our robust extension of IQN, using the implicit quantile representation with proposed Wasserstein-robust local correction.

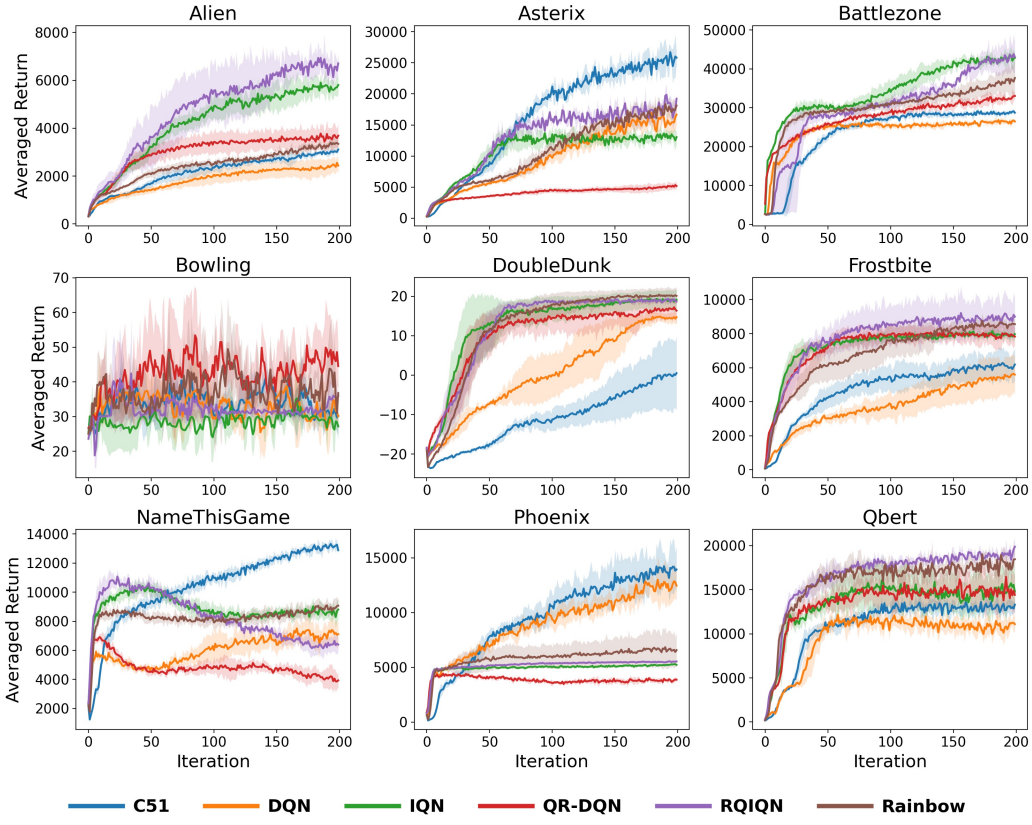

Figure 3: Performance comparison on 9 Atari games. RQIQN results are averaged over 3 random seeds and compared against reference values from Castro et al. [29].

We evaluate on several Atari games, including the Atari-5 subbenchmark [30], a compact subset of the Arcade Learning Environment (ALE) designed to approximate trends on the full Atari-57 suite with representativeness and substantially lower computational cost. For reproducible evaluation, all baselines and our method are implemented in the Dopamine framework [29] and trained for 200M

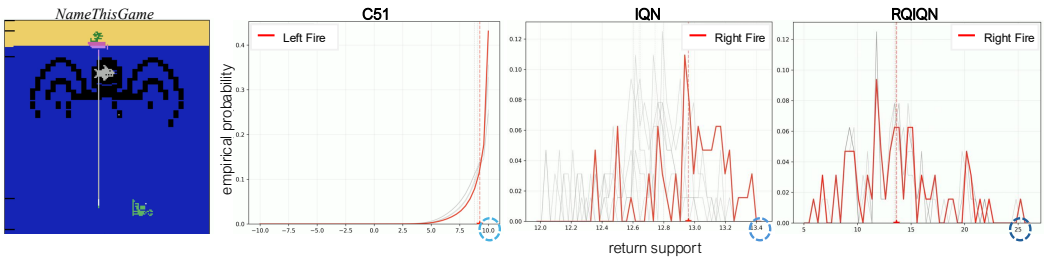

Figure 4: Visualization of the action-return distributions and the corresponding decisions of different agents under a representative state in NAMETHISGAME. The vertical dashed lines indicate the means of the corresponding return distributions, and red highlight denotes the action selected by the agent.

frames under the *sticky-actions protocol* [31]. Under this protocol, the ALE repeats the previously executed action with probability 0.25 instead of always applying the agent's newly selected action, injecting controlled stochasticity and reducing reliance on deterministic Atari dynamics.

As shown in Figure 3, RQIQN improves over IQN on most Atari games, with the exception of NAMETHISGAME. Notably, it achieves stronger performance on ALIEN, BATTLEZONE, FROSTBITE, and QBERT compared with the evaluated baselines. These results suggest that the proposed robust quantile correction can further enhance IQN, indicating that quantile-level robustness remains beneficial even when the quantile Huber loss already provides a stronger practical surrogate than the pure check loss. Additional comparisons are demonstrated in Appendix C.4.

The Suitability of Distribution Representation: Case Study. In Figure 3, we observe that on NAMETHISGAME, methods with more advanced schemes exhibit unexpected performance fluctuations and converge to suboptimal final returns. This indicates that RQIQN, despite its overall gains, may still be vulnerable to performance degradation in specific environments.

As shown in Figure 4, in the representative state, C51 selects the more reasonable left-fire action, which guides the diver toward the oxygen line while attacking the hostile shark. In contrast, IQN and RQIQN are biased toward the high-return region of the right-fire action, resulting in a suboptimal decision that moves the diver away from the oxygen line. One possible explanation is due to the differences in the value ranges. C51 represents the return distribution with 51 fixed atoms on a bounded support, typically $[V_{\min}, V_{\max}] = [-10, 10]$. The projection of Bellman targets back onto this support naturally clips or compresses extreme high-return estimates, which may reduce the influence of unstable high-return tails. By contrast, IQN and RQIQN model returns through sampled quantile fractions without an explicit fixed value support, making their action ranking more susceptible to inflated upper-tail estimates. This can render the induced policy overly optimistic in NAMETHISGAME. A natural remedy is to introduce task-dependent fraction distortion, which adaptively reshapes the sampling distribution over quantile fractions, allowing the agent to emphasize different risk regions of the return distribution according to the task structure.

Variants of RQIQN. To examine the sensitivity of RQIQN to key implementation choices, we further conduct ablation studies on QBERT. Specifically, we compare the check-loss and Huberized variants, and evaluate two representative type-p Wasserstein ambiguity sets, corresponding to $p = 2$ and $p = \infty$. As shown in Figure 5, RQIQN achieves the best performance under the default configuration, using the check loss with a type-2 Wasserstein ambiguity set. The Huberized variant with $p = 2$ performs comparably, with only a small performance gap, and both variants consistently outperform IQN. In contrast, when the ambiguity set is defined with $p = \infty$, both the check-loss and Huberized variants exhibit perfor-

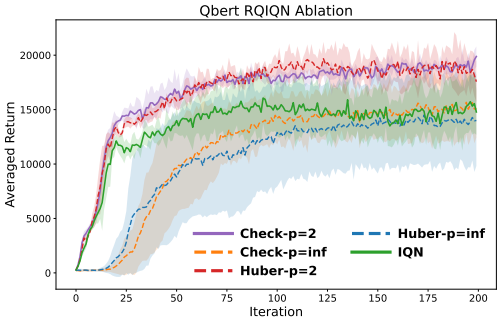

Figure 5: Performance of RQIQN variants.

mance degradation. These results suggest that RQIQN is more sensitive to the choice of Wasserstein geometry than to the choice between check and Huberized losses.

Table 2: Distributional estimation errors of IQN and RQIQN at states 0, 1, and 2. Lower values indicate better accuracy.

State	Method	W_1	W_2	D_{25}	D_{50}	D_{75}
0	IQN	1.5655	1.7527	1.9677	0.5061	1.1275
	RQIQN	**0.1542**	**0.2143**	**0.0915**	**0.0050**	**0.0810**
1	IQN	1.4091	1.5727	1.8165	0.4381	0.9767
	RQIQN	**0.1730**	**0.2429**	**0.0463**	**0.0391**	**0.1553**
2	IQN	1.3232	1.4581	1.6875	0.3035	0.9991
	RQIQN	**0.1538**	**0.1863**	**0.1175**	**0.1098**	**0.0847**
Mean	IQN	1.4326	1.5945	1.8239	0.4159	1.0344
	RQIQN	**0.1603**	**0.2145**	**0.0851**	**0.0513**	**0.1070**

4.3 Chain MDP

Figure 1(a) illustrates the effectiveness of RQIQN in mitigating quantile degeneration compared with IQN. We further investigate a chain MDP with a complex multimodal return distribution to evaluate distributional estimation accuracy.

We consider a single-action chain with eight decision states followed by an absorbing terminal state. Transitions are deterministic and sequential. Intermediate transitions yield independent rewards drawn from a normal distribution $\mathcal{N}(0, 0.3^2)$, while the final transition yields a reward drawn from the asymmetric trimodal mixture:

$$0.45\mathcal{N}(-3, 0.4^2) + 0.35\mathcal{N}(0.5, 1.0^2) + 0.20\mathcal{N}(4, 1.8^2).$$

Since the ground-truth return distribution is known, we directly evaluate distributional estimation errors using Wasserstein distances (W_1, W_2) and quantile errors (D_{25}, D_{50}, D_{75}).

As shown in Table 2, RQIQN consistently achieves lower Wasserstein distances and quantile errors across evaluated states, indicating more accurate approximation of complex multimodal return distributions.

5 Conclusion

In this work, we introduce RQIQN to improve the robustness of quantile fitting against finite-sample perturbations in bootstrapped Bellman targets. Motivated by Wasserstein-robust quantile estimation, RQIQN employs a lightweight correction to regularize quantile geometry and mitigate distribution degeneration. Experiments on a chain MDP demonstrate improved distributional estimation accuracy, while evaluations on Atari games and risk-sensitive ASV navigation further support its practical effectiveness. Overall, our findings highlight the importance of robust quantile estimation for distributional reinforcement learning.

Acknowledgments and Disclosure of Funding

Hui Xiong was supported in part by the National Natural Science Foundation of China (Grant No.92370204), in part by the National Key R&D Program of China (Grant No.2023YFF0725001), in part by Guangdong Provincial Key Laboratory of Frontier Basic Science for All-domain Intelligence, in part by the guangdong Basic and Applied Basic Research Foundation (Grant No.2023B1515120057), in part by the Key-Area Special Project of Guangdong Provincial Ordinary Universities (2024ZDZX1007). Sihong Xie was supported by the Department of Science and Technology of Guangdong Province (2023CX10X079), National Key R&D Program of China (Grant No.2023YFF0725001), the Guangzhou-HKUST(GZ) Joint Funding Program (Grant No.2023A03J0008), and Education Bureau Guangzhou Municipality. The authors declare no competing interests.

References

[1] Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit quantile networks for distributional reinforcement learning. In *International conference on machine learning*, pages 1096–1105. PMLR, 2018.

[2] Will Dabney, Mark Rowland, Marc Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[3] Mark Rowland, Robert Dadashi, Saurabh Kumar, Rémi Munos, Marc G Bellemare, and Will Dabney. Statistics and samples in distributional reinforcement learning. In *International Conference on Machine Learning*, pages 5528–5536. PMLR, 2019.

[4] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[5] Sami Jullien, Romain Deffayet, Jean-Michel Renders, Paul Groth, and Maarten de Rijke. Distributional reinforcement learning with dual expectile-quantile regression. *arXiv preprint arXiv:2305.16877*, 2023.

[6] Thanh Tang Nguyen, Sunil Gupta, and Svetha Venkatesh. Distributional reinforcement learning with maximum mean discrepancy. *Association for the Advancement of Artificial Intelligence (AAAI)*, 2020.

[7] Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International conference on machine learning*, pages 449–458. Pmlr, 2017.

[8] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[9] Derek Yang, Li Zhao, Zichuan Lin, Tao Qin, Jiang Bian, and Tie-Yan Liu. Fully parameterized quantile function for distributional reinforcement learning. *Advances in neural information processing systems*, 32, 2019.

[10] Weijian Zhang, Jianshu Wang, and Yang Yu. Distributional reinforcement learning with sample-set bellman update. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2852–2858. IEEE, 2024.

[11] Cheng Liu, Erik-Jan van Kampen, and Guido CHE De Croon. Adaptive risk-tendency: Nano drone navigation in cluttered environments with distributional reinforcement learning. *arXiv preprint arXiv:2203.14749*, 2022.

[12] Xi Lin, John McConnell, and Brendan Englot. Robust unmanned surface vehicle navigation with distributional reinforcement learning. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6185–6191. IEEE, 2023.

[13] Zhaofan Zhang, Minghao Yang, Sihong Xie, and Hui Xiong. Perturbation-mitigated usv navigation with distributionally robust reinforcement learning. *arXiv preprint arXiv:2512.00030*, 2025.

[14] Xi Lin, Yewei Huang, Fanfei Chen, and Brendan Englot. Decentralized multi-robot navigation for autonomous surface vehicles with distributional reinforcement learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8327–8333. IEEE, 2024.

[15] Jiyuan Shi, Chenjia Bai, Haoran He, Lei Han, Dong Wang, Bin Zhao, Mingguo Zhao, Xiu Li, and Xuelong Li. Robust quadrupedal locomotion via risk-averse policy learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11459–11466. IEEE, 2024.

[16] Whitney K Newey and James L Powell. Asymmetric least squares estimation and testing. *Econometrica: Journal of the Econometric Society*, pages 819–847, 1987.

[17] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22, 1977.

[18] Fan Zhou, Jianing Wang, and Xingdong Feng. Non-crossing quantile regression for distributional reinforcement learning. *Advances in neural information processing systems*, 33:15909–15919, 2020.

[19] Fan Zhou, Zhoufan Zhu, Qi Kuang, and Liwen Zhang. Non-decreasing quantile function network with efficient exploration for distributional reinforcement learning. *arXiv preprint arXiv:2105.06696*, 2021.

[20] Yudong Luo, Guiliang Liu, Haonan Duan, Oliver Schulte, and Pascal Poupart. Distributional reinforcement learning with monotonic splines. In *International Conference on Learning Representations*, 2021.

[21] Thibaut Théate, Antoine Wehenkel, Adrien Bolland, Gilles Louppe, and Damien Ernst. Distributional reinforcement learning with unconstrained monotonic neural networks. *Neurocomputing*, 534:199–219, 2023.

[22] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pages 492–518. Springer, 1992.

[23] Nikos Tzavidis, Stefano Marchetti, and Ray Chambers. Robust estimation of small-area means and quantiles. *Australian & New Zealand Journal of Statistics*, 52(2):167–186, 2010.

[24] Onyedikachi O John. Robustness of quantile regression to outliers. *American Journal of Applied Mathematics and Statistics*, 3(2):86–88, 2015.

[25] Christian Galarza Morales, Victor Lachos Davila, Celso Barbosa Cabral, and Luis Castro Cepero. Robust quantile regression using a generalized class of skewed distributions. *Stat*, 6(1):113–130, 2017.

[26] Chunxu Zhang, Tiantian Mao, and Ruodu Wang. Wasserstein distributionally robust quantile regression. *arXiv preprint arXiv:2603.14991*, 2026.

[27] David J Acheson. *Elementary fluid dynamics*. Oxford University Press, 1990.

[28] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[29] Pablo Samuel Castro, Subhodeep Moitra, Carles Gelada, Saurabh Kumar, and Marc G Bellemare. Dopamine: A research framework for deep reinforcement learning. *arXiv preprint arXiv:1812.06110*, 2018.

[30] Matthew Aitchison, Penny Sweetser, and Marcus Hutter. Atari-5: Distilling the arcade learning environment down to five games. In *International Conference on Machine Learning*, pages 421–438. PMLR, 2023.

[31] Marlos C Machado, Marc G Bellemare, Erik Talvitie, Joel Veness, Matthew Hausknecht, and Michael Bowling. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *Journal of Artificial Intelligence Research*, 61:523–562, 2018.

A Proofs

A.1 Proof of Lemma 1

Lemma 1. For any transition (s, a, r, s'), consider the check-loss counterpart of IQN snapshot $\mathbb{E}_{\tau'}\left[\rho_{\tau_i}\left(\delta_{ij}\right)\right]$ in Eq. (3) with sampled target fractions $\tau'_j \sim U([0,1])$ and a fixed current fraction $\tau_i \in (0,1)$:

$$\delta_{ij} = y_j - q_i, \qquad y_j := r + \gamma Z_{\tau'_j}(s', a'), \qquad q_i := Z_{\tau_i}(s, a).$$

Let the empirical Bellman target distribution be $\widehat{\mu}_{s,a} = \frac{1}{N'}\sum_{j=1}^{N'} \delta_{y_j}$, where δ_{y_j} denotes a Dirac at y_j. Then the snapshot has the same form with empirical quantile estimation slot of Eq. (3) at τ_i:

$$q^0_{\tau_i} \in \arg\min_{q\in\mathbb{R}} \mathbb{E}_{Y\sim\widehat{\mu}_{s,a}}\left[\rho_{\tau_i}(Y - q)\right], \tag{10}$$

where $\rho_\tau(u) = u(\tau - \mathbb{1}_{\{u<0\}})$ and any minimizer $q^0_{\tau_i}$ satisfies the empirical coverage condition

$$\mathbb{P}_{Y\sim\hat{\mu}_{s,a}}(Y < q^0_{\tau_i}) \le \tau_i \le \mathbb{P}_{Y\sim\hat{\mu}_{s,a}}(Y \le q^0_{\tau_i}).$$

Proof. Conditioned on the transition (s, a, r, s'), the sampled target fractions $\{\tau'_j\}_{j=1}^{N'}$, and the target-side values $y_j = r + \gamma Z_{\tau'_j}(s', a')$ are fixed scalar samples. Hence, for any scalar prediction q, the empirical expectation under

$$\widehat{\mu}_{s,a} = \frac{1}{N'}\sum_{j=1}^{N'} \delta_{y_j}$$

satisfies

$$\mathbb{E}_{Y\sim\widehat{\mu}_{s,a}}\left[\rho_{\tau_i}(Y - q)\right] = \frac{1}{N'}\sum_{j=1}^{N'} \rho_{\tau_i}(y_j - q).$$

Using the TD snapshot $\delta^{\tau_i,\tau'_j} = y_j - q_i$, the check-loss contribution of Eq. (3) associated with the fixed fraction τ_i can therefore be written as

$$\frac{1}{N'}\sum_{j=1}^{N'} \rho_{\tau_i}\left(\delta^{\tau_i,\tau'_j}\right) = \mathbb{E}_{Y\sim\widehat{\mu}_{s,a}}\left[\rho_{\tau_i}(Y - q_i)\right].$$

Thus, if q_i is viewed as a free scalar output variable, fitting the i-th output slot amounts to minimizing the empirical risk

$$q^0_{\tau_i} \in \arg\min_{q\in\mathbb{R}} \mathbb{E}_{Y\sim\widehat{\mu}_{s,a}}\left[\rho_{\tau_i}(Y - q)\right].$$

It remains to verify the coverage condition. For the check loss $\rho_\tau(u) = u(\tau - \mathbb{1}_{\{u<0\}})$, the objective $R_{\tau_i}(q) := \mathbb{E}_{Y\sim\widehat{\mu}_{s,a}}\left[\rho_{\tau_i}(Y - q)\right]$ is convex in q. Its subdifferential is

$$\partial R_{\tau_i}(q) = \left[\mathbb{P}_{Y\sim\hat{\mu}_{s,a}}(Y < q^0_{\tau_i}) - \tau_i,\ \mathbb{P}_{Y\sim\hat{\mu}_{s,a}}(Y \le q^0_{\tau_i}) - \tau_i\right].$$

The first-order optimality condition for a convex function is $0 \in \partial R_{\tau_i}(q^0_{\tau_i})$, this condition is equivalent to

$$\mathbb{P}_{Y\sim\hat{\mu}_{s,a}}(Y < q^0_{\tau_i}) \le \tau_i \le \mathbb{P}_{Y\sim\hat{\mu}_{s,a}}(Y \le q^0_{\tau_i}).$$

which proves the empirical quantile coverage condition.

Finally, summing the output-level risks over all sampled fractions $\{\tau_i\}_{i=1}^{N}$ gives the check-loss counterpart of the IQN snapshot objective. Therefore, Eq. (3) can be interpreted, at the output level and under a TD snapshot, as simultaneously fitting a collection of local quantile regression risks. □

A.2 Proof of Theorem 1

Theorem 1. For $\epsilon \geq 0$, $p \in (1, \infty]$, consider the Wasserstein distributionally robust quantile regression slot at τ_i:

$$q^{\epsilon}_{\tau_i} \in \arg\min_{q \in \mathbb{R}} \sup_{\nu \in B_p(\hat{\mu}_{s,a}, \epsilon)} \mathbb{E}_{Y \sim \nu}[\rho_{\tau_i}(Y - q)],$$

where $B_p(\hat{\mu}_{s,a}, \epsilon)$ denotes the type-p Wasserstein ambiguity set

$$B_p(\hat{\mu}_{s,a}, \epsilon) := \{\nu \in \mathcal{P}(\mathbb{R}) : W_p(\nu, \hat{\mu}_{s,a}) \leq \epsilon\}.$$

For each sampled fraction τ_i, any robust minimizer admits the explicit location correction

$$q^{\epsilon}_{\tau_i} = q^{0}_{\tau_i} + \Delta_p(\tau_i, \epsilon),$$

where the robust term Δ_p can be expressed as

$$\Delta_p(\tau, \epsilon) = \frac{\epsilon}{q}\big(\tau^q - (1-\tau)^q\big)\, c_{\tau,p}^{1-q}, \qquad \frac{1}{p} + \frac{1}{q} = 1,$$

and

$$c_{\tau,p} = \begin{cases} \big(\tau^q(1-\tau) + \tau(1-\tau)^q\big)^{1/q}, & p \in (1, \infty), \\ 2\tau(1-\tau), & p = \infty. \end{cases}$$

Proof. The proof follows Proposition 5 in the work [26], since IQN TD fitting slot is consistent with the form of Proposition 5. Fix a transition $e = (s, a, r, s')$ and a sampled fraction $\tau_i \in (0, 1)$. Let

$$Q^{0}_{\tau_i} := \arg\min_{q \in \mathbb{R}} \mathbb{E}_{Y \sim \widehat{\mu}_e}\left[\rho_{\tau_i}(Y - q)\right]$$

denote the nominal empirical quantile solution set. The robust counterpart is

$$Q^{\epsilon}_{\tau_i} := \arg\min_{q \in \mathbb{R}} \sup_{\nu \in B_p(\widehat{\mu}_e, \epsilon)} \mathbb{E}_{Y \sim \nu}\left[\rho_{\tau_i}(Y - q)\right].$$

This is a one-dimensional Wasserstein distributionally robust quantile estimation problem. Applying the Wasserstein distributionally robust quantile regression result to the location-only model gives that the robust solution set is a translated version of the nominal empirical quantile solution set:

$$Q^{\epsilon}_{\tau_i} = Q^{0}_{\tau_i} + \Delta_p(\tau_i; \epsilon).$$

Equivalently, for every $q^{\epsilon}_{\tau_i} \in Q^{\epsilon}_{\tau_i}$, there exists $q^{0}_{\tau_i} \in Q^{0}_{\tau_i}$ such that

$$q^{\epsilon}_{\tau_i} = q^{0}_{\tau_i} + \Delta_p(\tau_i; \epsilon).$$

Let q denote the Hölder conjugate of p, i.e.,

$$\frac{1}{p} + \frac{1}{q} = 1.$$

The location correction from Proposition 5 in WDRQR is

$$\Delta_p(\tau; \epsilon) = \frac{\epsilon}{q}\left(\tau^q - (1-\tau)^q\right) c_{\tau,p}^{1-q},$$

where

$$c_{\tau,p} = \begin{cases} \left(\tau^q(1-\tau) + \tau(1-\tau)^q\right)^{1/q}, & p \in (1, \infty), \\ 2\tau(1-\tau), & p = \infty. \end{cases}$$

This proves the claimed closed-form Wasserstein-robust local quantile correction. □

A.3 Proof of Remark 1

Remark 1. The robust correction term in Eq. (7) is antisymmetric around the median:

$$\Delta_p(1-\tau;\epsilon) = -\Delta_p(\tau;\epsilon), \qquad \Delta_p(1/2;\epsilon) = 0.$$

Hence, under uniform quantile sampling,

$$\mathbb{E}_{\tau\sim\mathcal{U}(0,1)}[\Delta_p(\tau;\epsilon)] = \int_0^1 \Delta_p(\tau;\epsilon)d\tau = 0.$$

Therefore, the robust correction preserves the risk-neutral quantile average

$$\mathbb{E}_\tau\left[Z_\theta(s,a,\tau) + \Delta_p(\tau;\epsilon)\right] = \mathbb{E}_\tau\left[Z_\theta(s,a,\tau)\right].$$

Moreover, since $\Delta_p(\tau;\epsilon)$ is nondecreasing in τ, for any $0 < \tau_l < \tau_h < 1$,

$$\begin{aligned}&[Z_\theta(s,a,\tau_h) + \Delta_p(\tau_h;\epsilon)] - [Z_\theta(s,a,\tau_l) + \Delta_p(\tau_l;\epsilon)]\\ &= Z_\theta(s,a,\tau_h) - Z_\theta(s,a,\tau_l) + \Delta_p(\tau_h;\epsilon) - \Delta_p(\tau_l;\epsilon)\end{aligned}$$

where $\Delta_p(\tau_h;\epsilon) - \Delta_p(\tau_l;\epsilon) \geq 0$. Thus, the robust correction expands upper–lower quantile gaps while leaving the quantile average unchanged in expectation.

Proof. By the definition of $c_{\tau,p}$, we have $c_{1-\tau,p} = c_{\tau,p}$. Together with $(1-\tau)^{q_p} - \tau^{q_p} = -\big(\tau^{q_p} - (1-\tau)^{q_p}\big)$, this gives the median antisymmetry $\Delta_p(1-\tau;\epsilon) = -\Delta_p(\tau;\epsilon)$, and hence $\Delta_p(1/2;\epsilon) = 0$. Therefore, under uniform quantile sampling,

$$\mathbb{E}_{\tau\sim U([0,1])}[\Delta_p(\tau;\epsilon)] = \int_0^{1/2} \big[\Delta_p(\tau;\epsilon) + \Delta_p(1-\tau;\epsilon)\big]d\tau = 0.$$

By linearity of expectation, for any quantile estimate $Z_\tau(s,a)$,

$$\mathbb{E}_\tau\left[Z_\tau(s,a) + \Delta_p(\tau;\epsilon)\right] = \mathbb{E}_\tau\left[Z_\tau(s,a)\right],$$

so the correction preserves the risk-neutral quantile average.

For any $0 < \tau_l < \tau_h < 1$, direct subtraction yields

$$\begin{aligned}&\big[Z_{\tau_h}(s,a) + \Delta_p(\tau_h;\epsilon)\big] - \big[Z_{\tau_l}(s,a) + \Delta_p(\tau_l;\epsilon)\big]\\ &\quad = Z_{\tau_h}(s,a) - Z_{\tau_l}(s,a) + \Delta_p(\tau_h;\epsilon) - \Delta_p(\tau_l;\epsilon).\end{aligned}$$

Since $\Delta_p(\tau;\epsilon)$ is nondecreasing in τ, the last term is nonnegative. Thus, the correction weakly expands upper–lower quantile gaps while leaving the uniform quantile average unchanged. □

B Implementation Details

B.1 Hyperparameter Settings

RQIQN is implemented on top of the standard IQN architecture. Unless otherwise specified, we use the check loss for theoretical consistency and set more general $p = 2$ for the closed-form Wasserstein correction. The initial robustness radius $\epsilon_0 = 1$ is natural and stable across the considered tasks. For Atari experiments, we use a reverse-logistic decay schedule with midpoint $t_0 = 3.75 \times 10^6$ training steps and sharpness $k = 1.2 \times 10^{-6}$. For the navigation experiments, which run for 3×10^6 total training steps, we use the same sharpness $k = 1.2 \times 10^{-6}$ and set the midpoint to $t_0 = 5.9 \times 10^5$.

The hyperparameters used for agents' training in two tasks are summarized in Table 3. For Atari Games, we follow the deployment of Dopamine [29], including hyperparameter and structural settings.

Table 3: Training and reward settings for the four-state chain, eight-step chain, and navigation experiments. Navigation hyperparameters refer to IQN and RQIQN.

Setting	4-state chain	8-step chain	Navigation
Training budget	3×10^4 updates	6×10^4 updates	3×10^6 steps
Learning rate	10^{-3}	10^{-3}	10^{-4}
Replay buffer / sampling	Synthetic sampler	Synthetic sampler	10^6 transitions
Minibatch size	128	128	32
Discount factor	0.99	0.99	0.99
Optimizer	Adam	Adam	Adam
TD quantile samples (N/N')	$32/32$	$32/32$	$8/8$
Quantile samples for action selection	–	–	32
Quantile-curve evaluation grid	201	201	–
IQN / RQIQN loss	Huber / Check	Huber / Check	Huber / Check
Huber threshold κ	1	1	1
Intermediate reward (chains)	0	$\mathcal{N}(0, 0.3^2)$	–
Terminal reward (chains)	$\mathcal{R}_4$	$\mathcal{R}_8$	–
RQIQN Wasserstein order p	2	2	2
RQIQN base radius ϵ_0	0.1	0.5	1.0
Radius schedule	Constant	Reverse-logistic	Reverse-logistic
Decay slope k	–	5×10^{-4}	1.2×10^{-6}
Decay midpoint t_0	–	1.5×10^4	5.9×10^5

B.2 Autonomous Surface Vehicle (ASV) Navigation

Simulated Marine Environment for Navigation. We evaluate our method in an autonomous navigation setting: a risky marine environment for ASV navigation simulation based on prior work on robotics [12]. We use a purely sensor-based navigation scheme without any prior knowledge. The ASV perceives obstacles using a forward-looking sonar configured with $11\,\mathrm{beams}$ spanning a $2\pi/3\,\mathrm{rad}$ sector-shaped field of view, with an effective sensing range of up to about $10\,\mathrm{m}$, returning beam-wise range measurements within the forward arc. Additionally, it uses a DVL for speed estimation, and an IMU, compass, and GPS to obtain motion information and the relative direction to the goal. To align with the research focus on return distribution modeling and its overall performance, we configure the simulation scenarios to their maximum difficulty levels with 10 static obstacles and 8 dynamic vortices. To prevent overfitting, the layout of obstacles and uncertainty fields is stochastically re-initialized at the start of each episode.

MDP formulation. To learn a policy that moves toward the goal while avoiding obstacles, the reward function is designed as: $r_t = r_{\text{step}} + \alpha(d_{t-1} - d_t) + r_{\text{collision}}\mathbb{I}_{\text{collision}} + r_{\text{goal}}\mathbb{I}_{\text{goal}}$, where d_t represents the distance between the robot and the goal at time t. The indicator functions $\mathbb{I}_{\text{collision}}$ and $\mathbb{I}_{\text{goal}}$ take the value of 1 if a collision or goal is reached at t, respectively, and 0 otherwise. The components of the reward function are set as $r_{\text{step}} = -1.0$, $r_{\text{collision}} = -50.0$, $r_{\text{goal}} = 100.0$, and $\alpha = 1.0$. Based on real-world device conditions, the observation at timestep t is given by: $O_t = (O_{\text{velocity}}, O_{\text{goal}}, O_{\text{sonar}})$, where O_{velocity} denotes the robot's seafloor-relative velocity. The term O_{goal} represents the goal position in the robot's frame. The component O_{sonar} consists of sonar reflections, which provide information about obstacles. At timestep t, the vehicle selects a discrete action $a_t = (\alpha, \omega)$, where α changes the speed magnitude and ω changes the velocity direction: $\alpha \in \{-0.4, 0.0, 0.4\}\ \mathrm{m/s^2}$, $\omega \in \{-0.52, 0.0, 0.52\}\ \mathrm{rad/s}$. The forward speed is clipped to $[0, v_{\max}]$ with $v_{\max} = 2.0\,\mathrm{m/s}$, and the vortex-induced relative flow velocity is bounded by $v_{\mathrm{rel,max}} = 1.0\,\mathrm{m/s}$.

Metrics. We evaluate our approach using six metrics spanning performance, safety, and efficiency: success rate, collision rate, timeout rate, return, average completion time, and average energy consumption. Completion time and energy are averaged only over successful episodes.

Strategy. We compare the standard greedy strategy used in IQN and RQIQN by default, which always selects the action with the highest expected return, with an adaptive strategy proposed by [12]. The adaptive strategy adaptively sets the CVaR threshold for the distortion function. This threshold reflects different levels of risk sensitivity, measured by the robot's distance to the nearest obstacles. When no obstacles are detected, the adaptive strategy behaves like a normal one.

C Further Experimental Results

C.1 Four-State Chain MDP

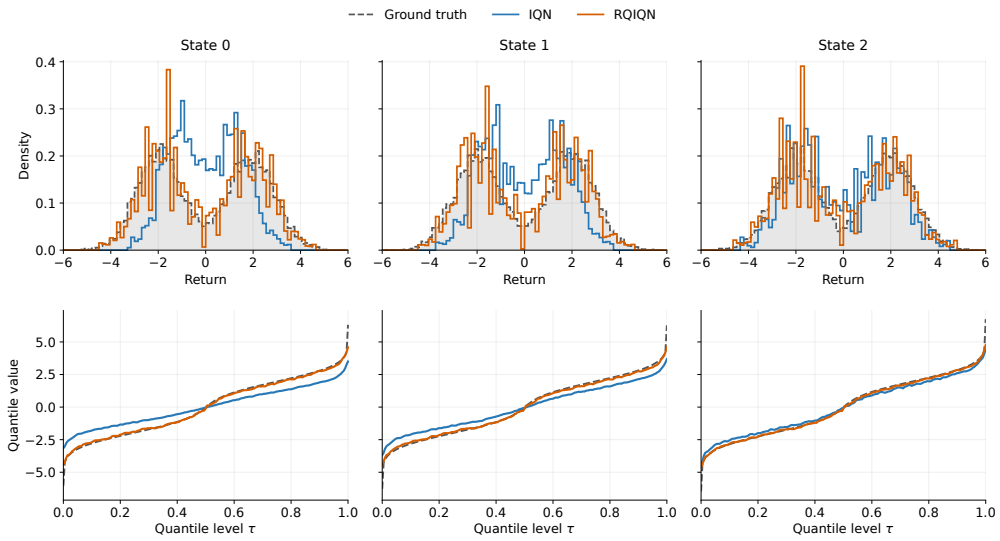

Figure 6: Visualization of the fitted return distributions of IQN and RQIQN at states 0, 1, and 2.

C.2 Experiments for the Robustness Decay Schedule

We added experiments for the decay schedule (different decay midpoints t_0, varying decay slope k while keeping t_0 fixed at its default value) on the ASV navigation task and a representative Atari game Q*bert. T denotes the full training horizon.

C.2.1 ASV navigation

Settings and results. We sweep six decay midpoints spanning from no removal to very early removal: $t_0/T \in \{1.25, 0.5, 0.25, 0.2, 0.15, 0.1\}$, where $t_0 = 1.25T = 3.75$M lies beyond the horizon so the perturbation is effectively never removed. Each variant is trained with 5 seeds under the identical protocol and evaluated under both the risk-neutral greedy policy and the adaptive-CVaR policy. We evaluate the final checkpoint on a fixed set of 30 scenarios, each containing five obstacles and eight vortices. The same scenarios are used for periodic evaluation during training. Five independently trained policies are evaluated on the same scenarios and initial states, with one rollout per scenario, yielding 150 trajectories in total. For greedy evaluation, exploration is disabled ($\epsilon = 0$), and the CVaR level is set to 1. Actions are selected by maximizing the mean of 32 sampled quantile values. We average the discounted returns across the 30 scenarios for each seed and report the mean and population standard deviation across the five seed-level averages.

Table 4 reports the final evaluation (last evaluation per seed, mean $\pm$ std over seeds), alongside the IQN baseline. Table 5 examines the sensitivity of our method to the decay slope k. A slope that is too small results in insufficient decay and degrades policy performance. Beyond this threshold, performance remains stable across different slopes, indicating that the method is relatively insensitive to this hyperparameter and does not require careful tuning.

C.2.2 Atari Q*bert

Analysis. The hyperparameter sensitivity results on Q*bert are presented in Table 6. Overall, RQIQN remains stable across a range of decay midpoints and slopes on both ASV Navigation and Q*bert, with most configurations achieving comparable performance. The default hyperparameters yield intermediate but competitive results rather than the best result in every case, indicating that they were not selected through task-specific tuning. These results suggest that the gains from

Table 4: Selection of the logistic decay midpoint t_0 on the USV task. Final evaluation over 5 seeds (mean $\pm$ std). All variants use $k = 1.2 \times 10^{-6}$ except $t_0/T = 0.5$, which uses $k = 5 \times 10^{-6}$ so that the perturbation is still fully removed before the end of training.

	Greedy		Adaptive CVaR	
Configuration	Reward	Success	Reward	Success
RQIQN, no decay	82.919 ± 2.351	0.953 ± 0.027	79.689 ± 2.406	0.927 ± 0.013
RQIQN, $t_0/T = 0.5$	84.714 ± 3.287	0.967 ± 0.021	80.166 ± 6.328	0.940 ± 0.044
RQIQN, $t_0/T = 0.25$	83.979 ± 2.645	0.953 ± 0.016	86.569 ± 2.497	0.980 ± 0.016
RQIQN, $t_0/T = 0.2$ (default)	83.674 ± 1.602	0.947 ± 0.016	86.747 ± 3.286	0.980 ± 0.027
RQIQN, $t_0/T = 0.15$	80.102 ± 2.168	0.933 ± 0.021	82.342 ± 3.247	0.953 ± 0.027
RQIQN, $t_0/T = 0.1$	82.428 ± 2.601	0.947 ± 0.016	87.867 ± 1.269	0.993 ± 0.013
IQN baseline	81.379 ± 5.898	0.953 ± 0.027	82.898 ± 4.877	0.973 ± 0.025

Table 5: Sensitivity analysis on the robustness decay slope k with the decay midpoint fixed at the default $t_0 = 5.9 \times 10^5$. Values are mean $\pm$ std over five seeds; $k = 1.2 \times 10^{-6}$ is the default setting.

Slope k	Policy	Reward	Success	Collision	Time	Energy
0.6×10^{-6}	greedy	78.258 ± 5.331	0.907 ± 0.039	0.093 ± 0.039	22.021 ± 0.963	34.115 ± 5.927
0.6×10^{-6}	adaptive	83.855 ± 1.842	0.960 ± 0.013	0.040 ± 0.013	23.159 ± 0.576	36.260 ± 5.745
1.2×10^{-6}	greedy	83.674 ± 1.602	0.947 ± 0.016	0.053 ± 0.016	22.204 ± 0.408	33.737 ± 2.768
1.2×10^{-6}	adaptive	86.747 ± 3.286	0.980 ± 0.027	0.020 ± 0.027	23.032 ± 0.736	35.554 ± 3.075
2.4×10^{-6}	greedy	84.927 ± 3.672	0.960 ± 0.025	0.040 ± 0.025	22.469 ± 0.567	32.927 ± 1.461
2.4×10^{-6}	adaptive	84.633 ± 2.903	0.967 ± 0.021	0.033 ± 0.021	23.245 ± 0.868	34.663 ± 2.084
5.0×10^{-6}	greedy	84.463 ± 4.963	0.960 ± 0.039	0.040 ± 0.039	22.760 ± 0.664	37.075 ± 3.244
5.0×10^{-6}	adaptive	84.249 ± 2.984	0.967 ± 0.021	0.033 ± 0.021	23.462 ± 0.702	38.496 ± 2.728

quantile robustness of RQIQN are not highly sensitive to the exact decay schedule and require little hyperparameter adjustment.

Table 6: Sensitivity analysis of the decay midpoint t_0/T and slope k on Q*bert. For the midpoint study, $\pm$ denotes the standard deviation over two seeds; for the slope study, it denotes the fluctuation over the final 10 training iterations.

Midpoint position t_0/T		*Decay slope k*	
Configuration	**Return**	**Configuration**	**Return**
No decay (fixed radius)	18030.4 ± 375.8	$k = 0.3 \times 10^{-6}$	18228.0 ± 553.2
$t_0/T = 0.3$	18968.9 ± 809.9	$k = 0.6 \times 10^{-6}$	19994.3 ± 470.4
$t_0/T = 0.15$	18054.2 ± 985.1	$k = 1.2 \times 10^{-6}$ (default)	17945.5 ± 237.8
$t_0/T = 0.075$ (default)	18442.5 ± 497.1	$k = 2.4 \times 10^{-6}$	17996.6 ± 389.3
$t_0/T = 0.025$	16309.8 ± 1111.5		

C.3 Extension to Continuous-Action Actor–Critic Settings

Experimental Setup. To evaluate the applicability of our method to continuous-action settings, we develop ContiIQN and RQContiIQN by extending IQN and RQIQN with a deterministic actor. The critic retains quantile-fraction sampling, while the actor maximizes the mean of sampled quantiles. The actor uses two hidden layers of 128 units and outputs two actions in $[-1, 1]$, scaled to acceleration $[-0.4, 0.4]\,\mathrm{m/s^2}$ and angular velocity $[-\pi/6, \pi/6]\,\mathrm{rad/s}$. Other settings remain unchanged.

We also include QR-DQN and C51 to assess the effectiveness of quantile robustness against alternative distributional representations. For C51, we use 51 uniformly spaced atoms over $[-100, 100]$ to cover extreme navigation rewards.

Results. As shown in Table 7, QR-DQN and C51 outperform IQN in success rate and exhibit lower variability, while RQIQN achieves further improvements. In the continuous-action setting,

Table 7: Performance comparison in ASV navigation. Results for existing methods are taken from the main text (five seeds), while additional methods use two seeds. Each seed is evaluated on 500 random environments. Values report mean ± standard deviation across seeds. Adaptive denotes dynamic CVaR adjustment.

Algorithm	Success Rate (%)	Return	$\text{Time}_{\text{succ}}$ (s)
IQN	64.80 ± 22.66	-13.17 ± 51.78	39.08 ± 1.22
IQN + Adaptive	71.04 ± 23.35	-12.19 ± 51.20	42.24 ± 1.55
QR-DQN	75.70 ± 2.97	13.80 ± 5.10	37.49 ± 1.27
C51	80.00 ± 1.13	7.88 ± 1.77	42.98 ± 2.08
RQIQN	83.24 ± 1.19	19.20 ± 1.73	37.55 ± 1.25
RQIQN + Adaptive	$\mathbf{85.64 \pm 2.27}$	18.57 ± 3.90	39.19 ± 1.87
ContiIQN	90.20 ± 1.70	23.81 ± 0.55	37.73 ± 0.46
RQContiIQN	$\mathbf{91.60 \pm 0.85}$	22.39 ± 1.41	39.60 ± 0.36

RQContiIQN improves the success rate over ContiIQN, although with a slightly lower average return. These results provide preliminary evidence that the proposed quantile-robustness mechanism can also benefit actor–critic learning in continuous-action settings.

C.4 Additional Comparisons with MMDQN on Atari Games

Table 8: Relative improvements of RQIQN and MMDQN over IQN on nine Atari games. Each method–seed run is evaluated using the mean of its 10 highest iteration-average returns across 200 iterations. Positive improvements of RQIQN are highlighted in bold.

Game	RQIQN vs. IQN	MMDQN vs. IQN
Alien	**+16%**	-13%
Asterix	**+32%**	$+3200\%$
BattleZone	-1%	$+12\%$
Bowling	**+26%**	$+70\%$
DoubleDunk	**+2%**	-59%
Frostbite	**+12%**	-26%
NameThisGame	**+4%**	$+29\%$
Phoenix	-23%	$+1506\%$
Qbert	**+17%**	$+25\%$

We further compare RQIQN with MMDQN [6], a related distributional RL method discussed in Section 1.1 and we use IQN as the common reference for comparison. As shown in Table 8, RQIQN outperforms IQN in seven of the nine games. While MMDQN achieves larger improvements in some games, RQIQN exhibits more consistent gains across the evaluated tasks.

D Experiments Compute Resources

The experiments were completed using a server with NVIDIA A6000 GPUs and AMD EPYC 7542 CPU. For a single environment, one complete RQIQN training run takes about four days on Atari games and about two days in the navigation environment.

E Discussion on the Limitations

Although RQIQN improves quantile-geometry stability and empirical performance, the NAMETHIS-GAME case study reveals that upper-tail estimates can still be inflated under certain return structures. Task-dependent fraction distortion or adaptive risk-sensitive fraction sampling may help address this issue. Moreover, RQIQN inherits IQN's quantile-based representation, mitigating rather than eliminating quantile degeneration. Addressing the fundamental limitations arising from the lack of Bellman closedness remains an important direction for future work.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: As stated in the abstract and introduction, the paper proposes RQIQN, theoretically characterizes it as a lightweight Wasserstein-robust quantile-geometry regularization method, and empirically evaluates its ability to mitigate distribution degeneration in quantile-based DistRL across risk-sensitive navigation and Atari games.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of the work in Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All theorem statements include or explicitly reference their assumptions, and complete proofs are provided in Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All the experimental results in this paper can be reproduced. We have provided the implementation details in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We are going to release our code once the paper is accepted to facilitate reproducibility.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies all the training and test details necessary to understand the results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provided the mean and standard deviation over several random seeds in the experiment to demonstrate statistical significance.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: For each experiment, we provide sufficient information on the computer resources needed to reproduce the experiments in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We fully adhere to NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: As this paper focuses on understanding and enhancing existing RL algorithms, we believe there is no negative societal impacts involved.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: For all the datasets and algorithm baselines used in the paper, we have cited the original papers and provided the license, copyright information, and terms of use in the package in our code repository.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.