Is Risk-Sensitive Reinforcement Learning Properly Resolved?

Ruwen Zhou [1] **Minghuan Liu** [1] **Kan Ren** [2] **Xufang Luo** [2] **Weinan Zhang** [1] **Dongsheng Li** [2]

Abstract

Due to the nature of risk management in learning applicable policies, risk-sensitive reinforcement learning (RSRL) has been realized as an important direction. RSRL is usually achieved by learning risk-sensitive objectives characterized by various risk measures, under the framework of distributional reinforcement learning. However, it remains unclear if the distributional Bellman operator properly optimizes the RSRL objective in the sense of risk measures. In this paper, we prove that the existing RSRL methods do not achieve unbiased optimization and can not guarantee optimality or even improvements regarding risk measures over accumulated return distributions. To remedy this issue, we further propose a novel algorithm, namely Trajectory Q-Learning (TQL), for RSRL problems with provable convergence to the optimal policy. Based on our new learning architecture, we are free to introduce a general and practical implementation for different risk measures to learn disparate risk-sensitive policies. In the experiments, we verify the learnability of our algorithm and show how our method effectively achieves better performances toward risk-sensitive objectives.

1. Introduction

Reinforcement learning (RL) has shown its success on various tasks (Mnih et al., 2015; Silver et al., 2017; Yang et al., 2022), which usually requires the agent to take enormous trial-and-error steps. However, most real-world applications are sensitive to failure and attach more importance to risk management, and thus need to turn to the help of risk-sensitive reinforcement learning (RSRL). Typically, RSRL can be achieved by building upon existing distributional RL approaches (Bellemare et al., 2017; Dabney et al., 2018a). For instance, risk-sensitive actor-critic methods like Urpí et al. (2021); Dabney et al. (2018a) first learn distributional critics as normal distributional RL methods, but take various distortion risk measures representing different risk preferences as the objective for the actor (e.g., conditional value-at-risk (`CVaR`)), which is computed based on critic output.

However, as we reveal in this paper, such solutions lead to biased optimization, fails to converge to an optimal solution in terms of risk-sensitive returns along the whole trajectory, and can sometimes lead to an arbitrarily bad policy. Therefore, an algorithm for unbiased optimization is desired in RSRL. Although some works have proposed solutions by defining risk in a per-step manner (Bäuerle and Ott, 2011; Chow and Ghavamzadeh, 2014; Tamar et al., 2015), it is still challenging to resolve RSRL problems where a policy is learned to optimize the risk measure over accumulated return distributions.

In this paper, we provide an in-depth analysis on the biased optimization issue of existing RSRL methods, conclude the reason, and present the intuition of the solution. Correspondingly, we propose Trajectory Q-Learning (TQL), a novel RSRL framework that is proven to learn the optimal policy w.r.t. various risk measures. Specifically, TQL learns an historical value function that models the conditional distribution of accumulated returns along the whole trajectory given the trajectory history. We give an extensive theoretical analysis of the learning behavior of TQL, indicating the convergence of policy evaluation, policy iteration, policy improvement, and (no) value iteration. Notably, we prove that the policy iteration of TQL can achieve unbiased optimization in RSRL. To the best of our knowledge, TQL is the first algorithm that can converge to the optimal risk-sensitive policy for all kinds of distortion risk measures. Experimentally, we verify our idea on both discrete mini-grid and continuous control tasks, showing TQL can be practically effective for finding optimal risk-sensitive policies and outperforms existing RSRL learning algorithms.

2. Preliminaries

2.1. Distributional Reinforcement Learning

We consider a Markov decision process (MDP), denoted as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ represents the state and action space, $\mathcal{P}(s'|s,a)$ is the dynamics transi-

[1]Shanghai Jiao Tong University [2]Microsoft Research. Correspondence to: Weinan Zhang <wnzhang@sjtu.edu.cn>, Kan Ren <kan.ren@microsoft.com>.

Preprint version.

tion function and when it is deterministic we use $s' = M(s,a)$ to represent the transition, $r(s,a)$ is the reward function, and γ denotes the discount factor. The return $Z^\pi(s,a) = \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)$ is a random variable representing the sum of the discounted rewards. The history $h_t = \{s_0, a_0, \cdots, s_t\}$ is state-action sequences sampled by agents in the environment, and its space $\mathcal{H} = \bigcup_t \left[\left(\prod_{i=0}^{t-1}(\mathcal{S}\times\mathcal{A})\right)\times\mathcal{S}\right]$. The objective of reinforcement learning (RL) is to learn a policy to perform the action $a \sim \pi$ on a given state or history that maximizes the expected cumulative discounted reward $\mathbb{E}_\pi\left[Z^\pi(s,a)\right]$. The optimization typically requires to compute the state-action value function $Q(s,a) = \mathbb{E}_\pi\left[Z^\pi(s,a)\right]$, which can be characterized by the Bellman operator $\mathcal{T}_B^\pi$: $\mathcal{T}_B^\pi Q^\pi(s,a) := \mathbb{E}\left[r(s,a)\right] + \gamma\mathbb{E}_{s'\sim\mathcal{P},a'\sim\pi}\left[Q^\pi(s',a')\right]$. The optimal policies can be obtained by learning the optimal value $Q^* = Q^{\pi^*}$ through the Bellman optimality operator $\mathcal{T}_B^*$: $\mathcal{T}_B^* Q(s,a) := \mathbb{E}\left[r(s,a)\right] + \gamma\mathbb{E}_{\mathcal{P}}\max_{a'} Q(s',a')$.

Instead of utilizing a scalar value function Q^π, which can be seen as optimizing the expectation of the distribution over returns, distributional RL considers modeling the whole distribution (Bellemare et al., 2017; Dabney et al., 2018a). From a distributional perspective, we regard $Z^\pi \sim \mathcal{Z}$ as a mapping from state-action pairs to distributions over returns, named the value distribution. Analogous to traditional RL, the goal is seeking a policy that maximizes the expected return over trajectories:

$$\pi^* \in \arg\max_\pi \mathbb{E}_{S_0, A_0\sim\pi(\cdot|S_0)}\left[Z^\pi(S_0, A_0)\right], \quad (1)$$

where $Z^\pi(S_0,\cdot)$ represents the return distribution of the trajectory starting from a random initial state S_0 and following π. We can define a distributional Bellman operator $\mathcal{T}^\pi$ that estimates the return distribution Z^π

$$\mathcal{T}^\pi Z(s,a) \stackrel{D}{:=} R(s,a) + \gamma Z(S', A'), \quad (2)$$

where $A \stackrel{D}{:=} B$ means that random variables A and B follows the same distribution, $R(s,a)$ is reward distribution, $S' \sim \mathcal{P}(\cdot|s,a)$ and $A' \sim \pi(\cdot|s')$. Correspondingly, the distributional Bellman optimality operator is

$$\mathcal{T}^* Z(s,a) \stackrel{D}{:=} R(s,a) + \gamma Z\left(S', \arg\max_{a'\in\mathcal{A}} \mathbb{E}\left[Z(S',a')\right]\right), \quad (3)$$

In this paper, we use capital letters to denote random variables and emphasize their random nature.

2.2. Distortion Risk Measure and Risk-Sensitive RL

As a type of risk measure, a *distortion risk measure* (Wang, 1996) β for a random variable X with the cumulative distribution function (CDF) $F_X(x)$ is defined as $\beta[X] = \int_{-\infty}^{\infty} x \frac{\partial}{\partial x}(h_\beta \circ F_X)(x)\,\mathrm{d}x$, where $h_\beta : [0,1] \to [0,1]$, called a distortion function, is a continuous non-decreasing function that transforms the CDF of X into $(h_\beta \circ F_X)(x)$. Intuitively, a distortion function distorts the probability density of a random variable to give more weight to either higher or lower-risk events. For example, `mean` and `CVaR` are the most commonly used distortion risk measures. For readers unfamiliar with distortion risk measures, we list some typical examples and their definitions in Appendix A.1. Thereafter, risk-sensitive reinforcement learning (RSRL) is natural to combine various distortion risk measures with distributional RL for achieving a risk-sensitive behavior. In the sequel, a risk-sensitive optimal policy with distortion risk measure β can be defined as a deterministic policy π_β^* by the risk-sensitive return over random variable $S_0 \sim \rho_0$ representing the initial state:

$$\pi_\beta^* \in \arg\max_\pi \mathbb{E}_{S_0\sim\rho_0, A_0\sim\pi}\left[\beta\left[Z^\pi(S_0, A_0)\right]\right]. \quad (4)$$

We call Eq. (4) the RSRL objective, as it seeks a policy that maximizes the risk measure of accumulated return over whole trajectory given the initial state distribution. Such a formulation was initially implemented in (Dabney et al., 2018a), by directly changing the objective to risk measures computed from the value distribution. Some other works define and optimize risk in a per-step manner (Bäuerle and Ott, 2011; Chow and Ghavamzadeh, 2014; Tamar et al., 2015), but this paper only focuses on the RSRL objective, as it is the natural risk-sensitive extension of RL. For readers interested in per-step risk definition, we give a brief introduction in Appendix A.4.

2.3. Metrics for Convergence

In distributional RL, since the value function is modeled as a distribution, researchers utilize a maximal form of the Wasserstein metric to establish the convergence of the distributional Bellman operators (Bellemare et al., 2017; Dabney et al., 2018b) $\bar{d}_p(Z_1, Z_2) := \sup_{x,a} d_p(Z_1(x,a), Z_2(x,a))$, where $Z_1, Z_2 \in \mathcal{Z}$ are two value distributions and $\mathcal{Z}$ denotes the space of value distributions with bounded moments. The p-Wasserstein distance d_p is the L_p metric on inverse CDF, i.e., quantile functions (Müller, 1997), which is defined as an optimal transport metric for random variables U and V with quantile functions F_U^{-1} and F_V^{-1} respectively: $d_p(U,V) = \left(\int_0^1 |F_U^{-1}(\omega) - F_V^{-1}(\omega)|^p d\omega\right)^{1/p}$. This can be realized as the minimal cost of transporting mass to make the two distributions identical.

Requiring the distributional Bellman operators to converge in the metric of $\bar{d}_p$ indicates that we must match the value distribution. While in policy evaluation the distributional Bellman operator $\mathcal{T}^\pi$ (Eq. (2)) is shown to be a contraction in p-Wasserstein, in the control setting proving the distributional Bellman optimality operator $\mathcal{T}^*$ (Eq. (3)) is hard (see Bellemare et al. (2017) for more details) and is not always necessary in practical cases. Instead, we may only need to achieve convergence in the sense of distributional statistics or measures. For example, we only require the learned value distribution to have the same `mean` of the optimal value distribution so that the policy learns to achieve the optimal return expectation, or we match a risk measure (like `CVaR`) of the optimal value distribution to learn a policy that achieves the optimal risk preference of the return distribution. As these measures upon value distributions are real functions w.r.t. states and actions, the convergence of distributional Bellman operators only need to lie in the infinity norm, a L_∞ metric: $\|f_1 - f_2\|_\infty = \sup_{x,a} \|f_1(x,a) - f_2(x,a)\|$.

3. Mismatch in RSRL Optimization

Although the RSRL objective Eq. (4) seems reasonable, *existing dynamic programming (DP) style algorithms does not optimize Eq. (4) properly*, as we will reveal in this section.

3.1. What are Current RSRL Algorithms Optimizing?

Recalling the RL objective Eq. (1) or considering setting β as `mean` in the RSRL objective Eq. (4), we can optimize π by Bellman equation in a dynamic programming style following the distributional Bellman optimality operator, i.e., there is a deterministic policy that maximizes the return at every single step for a given return distribution Z:

$$\pi_{\texttt{mean}}(s) \in \arg\max_{a\in\mathcal{A}} \mathbb{E}\left[Z\left(s,a\right)\right] \ . \tag{5}$$

And the distributional Bellman optimality operator is equivalent to:

$$\mathcal{T}^* Z\left(s,a\right) \stackrel{D}{:=} R\left(s,a\right) + \gamma Z\left(S', \pi_{\texttt{mean}}(S')\right) \ , \ S' \sim \mathcal{P} \ . \tag{6}$$

Although Bellemare et al. (2017) have shown that $\mathcal{T}^*$ itself is not a contraction in $\bar{d}_p$ such that it cannot be used for finding the optimal value distribution, we can realize $\mathcal{T}^*$ as a "contraction" in L_∞ from the perspective of `mean`, which induces a pointwise convergence. In other words, the `mean` of the value distribution $\mathbb{E}Z$ will converge to the `mean` of the value distribution $\mathbb{E}Z^*$.

Lemma 3.1 (Value iteration theorem)**.** *Recursively applying the distributional Bellman optimality operator $Z_{k+1} = \mathcal{T}^* Z_k$ on arbitrary value distribution Z_0 solves the objective Eq. (4) when β is exactly `mean` where the optimal policy is obtained via Eq. (5), and for $Z_1, Z_2 \in \mathcal{Z}$, we have:*

$$\|\mathbb{E}\mathcal{T}^* Z_1 - \mathbb{E}\mathcal{T}^* Z_2\|_\infty \leq \gamma \|\mathbb{E}Z_1 - \mathbb{E}Z_2\|_\infty \ , \tag{7}$$

and in particular $\mathbb{E}Z_k \to \mathbb{E}Z^*$ exponentially quickly.

The proof is just the proof of value iteration and Lemma 4 in Bellemare et al. (2017). For completeness, we include it in Appendix C.1. In the context of distributional RL, we can explain it as the `mean` of value will converge to the `mean` of optimal value. Motivated by and simply resemble Eq. (5), previous implementation like Dabney et al. (2018a) and Urpí et al. (2021) optimized a risk-sensitive policy:

$$\pi_\beta(s) \in \arg\max_{a\in\mathcal{A}} \beta\left[Z\left(s,a\right)\right] \ . \tag{8}$$

From a practical perspective, this can be easily achieved by only a few modifications to distributional RL algorithms towards any given distortion risk β, which implies a dynamic programming style updating following a risk-sensitive Bellman optimality operator $\mathcal{T}^*_\beta$ w.r.t. risk measure β:

$$\mathcal{T}^*_\beta Z(s,a) \stackrel{D}{:=} R(s,a) + \gamma Z(S', A') \ , \tag{9}$$

where $S' \sim \mathcal{P}(\cdot|s,a)$ and $A' \sim \pi_\beta(\cdot|s')$ are random variables. Note that the optimal risk-sensitive policy defined in Eq. (8) is generally different from Eq. (4). The key difference is that Eq. (8) tends to maximize the risk measure everywhere inside the MDP, yet Eq. (4) only requires finding a policy that can maximize the risk measure of trajectories started from the initial state S_0. Although this is equivalent when β is `mean`, when it is not, the equivalence does not ever hold when updating follows $\mathcal{T}^*_\beta$ in a dynamic programming style, which leads to the divergence in RSRL optimization, as we will reveal below.

3.2. $\mathcal{T}^*_\beta$ Leads to Biased Optimization

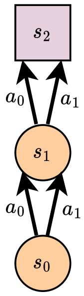
$\mathcal{T}^*_\beta$ is not contraction except β is `mean`. To show why $\mathcal{T}^*_\beta$ leads to biased optimization, we first provide an analysis that optimizing towards the Bellman optimality operator $\mathcal{T}^*_\beta$ w.r.t. risk measure β does not converge at all, *i.e.*, there is no contraction property for $\mathcal{T}^*_\beta$. For simplicity and starting from the easiest case, in the rest of this paper, we mainly discuss deterministic dynamics, i.e., instead of $s' \sim \mathcal{P}(\cdot|s,a)$, we simply consider $s' = M(s,a)$ and thus $\mathcal{T}_\beta$ becomes:

$$\mathcal{T}^*_\beta Z(s,a) \stackrel{D}{:=} R(s,a) + \gamma Z(s', A'), A' \sim \pi_\beta(s') \ . \tag{10}$$

We already know that $\mathcal{T}^*_\beta$ cannot be a contraction in $\bar{d}_p$, but different from Lemma 3.1, even from the perspective of the risk measure β if β is not `mean`, it still cannot be realized as a "contraction" in L_∞; in other words, the risk measure of the value distribution $\beta[Z]$ is not guaranteed to converge to β of the value distribution $\beta[Z^*]$. Thus, $\mathcal{T}^*_\beta$ does not help to find an optimal solution to solve the RSRL objective Eq. (4).

Theorem 3.2. *Recursively applying risk-sensitive Bellman optimality operator $\mathcal{T}^*_\beta$ w.r.t. risk measure β does not solve the RSRL objective Eq. (4) and $\beta[Z_k]$ is not guaranteed to converge to $\beta[Z^*]$ if β is not `mean` or an affine in `mean`.*

The formal proof can be referred to Appendix C.2. The above theorem of no contraction indicates that optimizing



$\beta = \texttt{CVaR}(\eta = 0.1)$	s_0, a_0	s_0, a_1	s_1, a_0	s_1, a_1
R	$\begin{cases} 100, p = 0.9 \\ -10, p = 0.1 \end{cases}$	-5	$\begin{cases} 100, p = 0.9 \\ -10, p = 0.1 \end{cases}$	-5
Z^*	$\begin{cases} 200, p = 0.81 \\ 90, p = 0.18 \\ -20, p = 0.01 \end{cases}$	$\begin{cases} 95, p = 0.81 \\ -15, p = 0.1 \end{cases}$	$\begin{cases} 100, p = 0.9 \\ -10, p = 0.1 \end{cases}$	-5
$\beta(Z^*)$	7.9	-15	-10	-5
Z	0	0	$\begin{cases} 100, p = 0.9 \\ -10, p = 0.1 \end{cases}$	-5
$\beta(Z)$	0	0	-10	-5
$\mathcal{T}_\beta^* Z$	$\begin{cases} 95, p = 0.9 \\ -15, p = 0.1 \end{cases}$	-10	$\begin{cases} 100, p = 0.9 \\ -10, p = 0.1 \end{cases}$	-5
$\beta(\mathcal{T}_\beta^* Z)$	-15	-10	-10	-5

Figure 1: Undiscounted 3-state MDP for which the optimality operator $\mathcal{T}_\beta^*$ does not converge and obtain non-optimal result. We highlight the entries that are incorrectly updated.

towards $\mathcal{T}_\beta$ may lead to arbitrarily worse solutions than the optimal solution of the RSRL objective Eq. (4).

To better understand how the problem occurs, let us consider a naive contradictive example on a 3-state MDP (Fig. 1, left), where the agents have a constant reward of -5 for conducting a_1 and a binomial reward for a_0, as shown in the first row of Fig. 1 (right). In such context, we optimize towards $\beta = \texttt{CVaR}(\eta = 0.1)$. Now consider the initial value estimation Z to be accurate at s_1 (Fig. 1, right). We list the value of Z and its corresponding risk measure $\beta[Z]$; the results $\mathcal{T}_\beta^* Z$ when updating Z on s_0 using $\mathcal{T}_\beta^*$ and its corresponding risk measure. In this case, when updating $Z(s_0)$, $Z(s_1)$ will always indicate to use a_1, although this can lead to a worse risk measure evaluated along the whole trajectory starting from s_0, and prevent the agent from finding the optimality, *i.e.*, applying a_0 at both states.

History return distribution matters. The reason why the risk-sensitive Bellman optimality operator $\mathcal{T}_\beta^*$ diverges and optimizing following $\mathcal{T}_\beta^*$ does not lead to an optimal policy *w.r.t.* risk measure β comes from the fact that the risk measure over future return distributions cannot be maximized everywhere inside an MDP, hence it is not reasonable to use such a Bellman optimality operator to update the return distribution following the risk-sensitive optimal policy. In other words, updating π_β under $\mathcal{T}_\beta^*$ only ensures to improve the risk measure of the trajectory starting from s', *i.e.*, $\beta\left[Z^\pi(s')\right]$, which does not guarantee to move towards a better risk measure along the whole trajectory $\beta\left[Z^\pi(s_0)\right]$, and can be totally different from optimizing the RSRL objective Eq. (4). Therefore, to achieve unbiased optimization, at every state the policy should take into account the return distribution along the past trajectory starting from s_0.

4. Solving RSRL

To remedy the biased optimization issue of bellman-style update, we propose a novel algorithm that lies in a non-Markovian formulation without dynamic programming style optimization.

As we pointed out before, the key problem that leads the risk-sensitive optimal Bellman operator $\mathcal{T}_\beta^*$ into biased optimization is that *the risk measure over future return distributions cannot be maximized everywhere inside an MDP*. Thereafter, *the dynamic programming style optimization that only utilizes the information forward, i.e., in the future, does not help to find the policy that maximizes the risk measures along the whole trajectories as defined in Eq. (4)*. Thus, when we compute the value distribution at certain states, we must include information backward, *i.e.*, in the past, to help with modeling the risk measure along the whole trajectory. This motivates us to model the history-action value distribution $Z^\pi(h_t, a_t) \sim \mathcal{Z}$, called *historical return distribution*, instead of the state-action value distribution $Z^\pi(s_t, a_t)$, along with a history-based (non-Markovian) policy $A \sim \pi(\cdot|h)$:

$$
\begin{aligned}
Z^\pi(h_t, a) &\triangleq \sum_{i=0}^{t-1} \gamma^i R(s_i, a_i) + \gamma^t Z^\pi(\{s_t\}, a) \\
&= \sum_{i=0}^{t} \gamma^i R(s_i, a_i) + \gamma^{t+1} Z^\pi\left(\{s_{t+1}\}, A_{t+1}\right) \ ,
\end{aligned} \tag{11}
$$

where $A_{t+1} \sim \pi(\cdot|h_{t+1}), s_{t+1} = M(s_t, a_t), h_t = \{s_0, a_0, \cdots, s_t\} \in \mathcal{H}$ denotes the history sequence that happened before reaching (including) state s_t. Therefore, the history-action value $Z^\pi(h_t, a)$ just records the discounted return of the whole trajectory given history h_t backward and moves forward following policy π. Note that the policy is now Markovian under the history-based MDP, *i.e.*, the policy gives action only based on the current history.

4.1. Policy Evaluation

Similar to Bellman operators, we now define a new type of operator, named the history-relied (HR) operator, that defines the principle of updating the history-action value.

$$
\mathcal{T}_h^\pi Z(h_t, a) :\stackrel{D}{=} R_{0:t} + \gamma^{t+1} Z(\{s_{t+1}\}, A_{t+1}) \ , \tag{12}
$$

where $A_{t+1} \sim \pi(\cdot|h_{t+1}), s_{t+1} = M(s_t, a_t), R_{0:t} = \sum_{i=0}^{t} \gamma^i r_i$ is the discounted return accumulated before the timestep t. To continue, we show our first theoretical result, that the policy evaluation with HR operator converges in the metric of $\bar{d}_p$.

Theorem 4.1 (Policy Evaluation for $\mathcal{T}_h^\pi$)**.** *$\mathcal{T}_h^\pi : \mathcal{Z} \to \mathcal{Z}$ is a γ-contraction in the metric of the maximum form of p-Wasserstein distance $\bar{d}_p$.*

The proof of Theorem 4.1 can be referred to Appendix C.3. Using Theorem 4.1 and combining Banach's fixed point theorem, we can conclude that $\mathcal{T}_h^\pi$ has a unique fixed point. By inspection, this fixed point must be Z^π as defined in Eq. (11) since $\mathcal{T}_h^\pi Z^\pi = Z^\pi$.

4.2. Policy Improvement and (No) Value Iteration

So far, we have considered the value distribution of a fixed policy π and the convergence of policy evaluation. Now let's turn to the control setting and find out the optimal value distribution and its corresponding policy under the risk-sensitive context.

In the form above, we want to find the optimal risk-sensitive policy that maximizes the risk measure over the whole trajectory given the initial state distribution as defined in Eq. (4), which is equivalent,

$$\pi^*(h) \in \arg\max_{a\in\mathcal{A}} \mathbb{E}_{h\sim\mathcal{H},a\sim\pi}\left[\beta\left[Z^\pi(h,a)\right]\right] . \tag{13}$$

Suppose $\mathcal{H}$ and $\mathcal{A}$ are both finite, the solution of Eq. (13) will always exist (but may not be unique!). Denote the optimal risk-sensitive policy set is Π^*, where $\forall\pi_1^*,\pi_2^*\in\Pi^*$, we have their return distribution $\beta[Z_1^*]=\beta[Z_2^*]$ and they must satisfy the risk-sensitive HR optimality equation:

$$\beta[Z_1^*(h_t,a)] = \beta\left[R_{0:t} + \gamma^{t+1} Z_2^*(\{s_{t+1}\}, a_{t+1}^*)\right] \tag{14}$$

$$a_{t+1}^* \in \arg\max_{a\in\mathcal{A}} \beta\left[Z_2^*(h_{t+1},a)\right] . \tag{15}$$

We can prove Eq. (14) is also sufficient for Eq. (13), see Appendix C.5. Hereby, we define the risk-sensitive HR optimality operator $\mathcal{T}_{h,\beta}^*$:

$$\begin{aligned} &\mathcal{T}_{h,\beta}^* Z(h_t,a) \leftarrow R_{0:t} + \gamma^{t+1} Z(\{s_{t+1}\}, a_{t+1}) \\ &a_{t+1} = \pi'(h_{t+1}) = \arg\max_{a\in\mathcal{A}} \beta\left[Z(h_{t+1},a)\right] , \end{aligned} \tag{16}$$

where the policy is obtained by deterministically maximizing the history-action value under risk measure β. And Eq. (14) implies some "fixed" points for Eq. (4) or Eq. (13) from the perspective of risk measure β for $\mathcal{T}_{h,\beta}^*$.

Correspondingly, we can present our second theoretical result, that the policy improvement under HR optimality operator is also guaranteed to converge into the risk-sensitive optimal policy.

Theorem 4.2 (Policy Improvement for $\mathcal{T}_{h,\beta}^*$)**.** *For two deterministic policies π and π', if π' is obtained by $\mathcal{T}_{h,\beta}^*$:*

$$\pi'(h_t) \in \arg\max_{a\in\mathcal{A}} \beta\left[Z^\pi(h_t,a)\right] ,$$

then the following inequality holds

$$\beta\left[Z^\pi(h_t,\pi(h_t))\right] \le \beta\left[Z^{\pi'}(h_t,\pi'(h_t))\right] .$$

The formal proof can be referred to Appendix C.4. When the new greedy policy π', is as good as, but not better than, the old policy π in the sense of risk measures, we have that:

$$\beta[Z^\pi(h_t,a_t)] = \beta\left[\mathcal{T}_{h,\beta}^* Z^\pi(h_t,a_t)\right] ,$$

Unfolding the right side, we get:

$$\begin{aligned} &\beta[Z^\pi(h_t,a_t)] = \beta\left[R_{0:t} + \gamma^{t+1} Z^\pi(\{s_{t+1}\}, a_{t+1})\right] \\ &a_{t+1} = \pi'(h_{t+1}) \in \arg\max_{a\in\mathcal{A}} \beta\left[Z(h_{t+1},a)\right] , \end{aligned} \tag{17}$$

which is exactly the risk-sensitive HR optimality equation Eq. (14). Therefore, we conclude that utilizing $\mathcal{T}_{h,\beta}^*$ for policy improvement will give us a strictly better policy except when the original policy is already optimal.

In the sequel, we understand that if the optimal solution of Eq. (13) exists, there exists at least a sequence of distributional value function $\{Z_0, Z_1, \cdots, Z_n, Z_1^*, \cdots, Z_k^*\}$ induced by the sequence of policy $\{\pi_0,\pi_1,\cdots,\pi_n,\pi_1^*,\cdots,\pi_k^*\}$ such that $\beta[Z_1]\le\beta[Z_2]\le\cdots\le\beta[Z_n]\le\beta[Z_1^*]=\cdots=\beta[Z_k^*]$. However, starting from an arbitrary Z (which may not correspond to any policy), it is non-trivial to prove $\mathcal{T}_{h,\beta}^*$ converges to $\beta[Z_i^*]$.

Theorem 4.3. *For $Z_1, Z_2\in\mathcal{Z}$,* HR *optimality operator $\mathcal{T}_{h,\beta}^*$ has the following property:*

$$\|\beta[\mathcal{T}_{h,\beta}^* Z_1] - \beta[\mathcal{T}_{h,\beta}^* Z_2]\|_\infty \le \|\beta[Z_1]-\beta[Z_2]\|_\infty , \tag{18}$$

The proof is in Appendix C.6. Theorem 4.3 told us that the value iteration for $\mathcal{T}_{h,\beta}^*$ may not converge. Specifically, our proposed HR operator can be realized as a "nonexpensive mapping" from the perspective of risk measure β in L_∞. For our cases of limited spaces, we might expect there exists some "fixed" point Z^*, and the best we can hope is a pointwise convergence such that βZ converges to βZ^* after recursively applying HR optimality operator $\mathcal{T}_{h,\beta}^*$ w.r.t. risk measure β. However, from Theorem 4.3, we know that βZ_n is not assured to be converged to βZ^* at any speed, hence the starting from arbitrary value distribution Z_0, $\mathcal{T}_{h,\beta}^*$ does not necessarily solve the RSRL objective Eq. (4). As a result, βZ_n may possibly fall on a sphere around Z^*.

4.3. Trajectory Q-Learning

As discussed above, by estimating the historical return distribution and improving the policy accordingly, we can now derive our practical RSRL algorithm, namely Trajectory Q-Learning (TQL). Representing the policy π, the historical value function Q as neural networks parameterized by ϕ and θ respectively, and denoting the historical return distribution approximated by critics as

$$Z_\theta(h,a) = \frac{1}{N}\sum_{j=0}^{N-1} \mathrm{Dirac}\left[Q_\theta(h,a;\tau_j)\right] , \tag{19}$$

we optimize the following loss functions:

$$J_\pi(\phi) = \beta\left[Z_\theta(h, a)\right], \tag{20}$$

$$J_Q(\theta) = \mathbb{E}_{a' \sim \pi;\, \tau_i, \tau'_j \sim U([0,1])}\Big[\rho^\kappa_\tau\Big(\sum_{t=0}^{s_t=s} \gamma^t r(s, a) + \gamma \bar{Q}_{\theta'}\left(\{s'\}, a'; \tau'_j\right) - Q_\theta(h, a; \tau_i)\Big)\Big], \tag{21}$$

where ρ^κ_τ represents the quantile Huber loss (see Appendix A.2 for details). In practice, to accurately estimate $Z(\{s'\}, \cdot)$ which is just a normal state-based value function (Dabney et al., 2018a), we model $\bar{Q}_{\theta'}\left(\{s'\}, a'; \tau'_j\right)$ with an extra Markovian value function $Q_\psi(s, a; \tau)$, updated by

$$J_Q(\psi) = \mathbb{E}_{a' \sim \pi;\, \tau_i, \tau'_j \sim U([0,1])}\Big[\rho^\kappa_\tau\Big(r(s, a) + \gamma \bar{Q}_{\psi'}\left(s', a'; \tau'_j\right) - Q_\psi(s, a; \tau_i)\Big)\Big], \tag{22}$$

In total, the algorithm learns a policy π_ϕ, a history-based value function Z_θ, and a Markovian value function Z_ψ. At each timestep, Z_ψ and Z_θ are updated according to Eq. (22) and Eq. (21), and the policy π_ϕ is optimized with Eq. (20). For discrete control, we can omit ϕ and implement π by taking argmax from $\beta[Z(h, a)]$. We list the step-by-step algorithm in Algo. 1 (discrete) and Algo. 2 (continuous).

5. Related Work

5.1. Distributional Reinforcement Learning

Distributional RL considers the uncertainty by modeling the return distribution, enabling risk-sensitive policy learning. Bellemare et al. (2017) first studied the distributional perspective on RL and proposed C51, which approximates the return distribution with a categorical over fixed intervals. Dabney et al. (2018b) proposed QR-DQN, turning to learning the critic as quantile functions and using quantile regression to minimize the Wasserstein distance between the predicted and the target distribution. Dabney et al. (2018a) further proposed IQN, improving QR-DQN by quantile sampling and other techniques, which further investigate risk-sensitive learning upon various distortion risk measures.

5.2. Risk in Reinforcement Learning

Risk management in RL towards real-world applications can be roughly divided into two categories, i.e., safe and constrained RL and distributional risk-sensitive RL. Safe and constrained RL formulates the risk as some kind of constraint to the policy optimization problem. For instance, Achiam et al. (2017) proposed a Lagrangian method which provides a theoretical bound on cost function while optimizing the policy; Dalal et al. (2018) built a safe layer to revise the action given by an unconstrained policy; Chow et al. (2018) used the Lyapunov approach to systematically transform dynamic programming and RL algorithms into their safe counterparts.

When the form of risks is either too complex or the constraints are hard to be explicitly defined, safe RL algorithms can be challenging to learn. In that case, distributional RL provides a way to utilize risk measures upon the return distributions for risk-sensitive learning. Among them, Tang et al. (2019) modeled the return distribution via its mean and variance and then learned an actor optimizing the `CVaR` of the return distribution; Keramati et al. (2020) proposed a novel optimistic version of the distributional Bellman operator that moves probability mass from the lower to the upper tail of the return distribution for sample-efficient learning of optimal policies in terms of `CVaR`; (Ma et al., 2020) modified SAC (Haarnoja et al., 2018) with distributional critics and discussed its application to risk-sensitive learning; Urpí et al. (2021) proposed their offline risk-averse learning scheme based on IQN (Dabney et al., 2018a) and BCQ (Fujimoto et al., 2019); Ma et al. (2021) proposed CODAC, which adapts distributional RL to the offline setting by penalizing the predicted quantiles of the return for out-of-distribution actions; recently, (Hong Lim and Malik, 2022) proposed a solution to resolve a similar issue specifically in optimizing policies towards `CVaR`, yet our TQL is general for any risk measure with theoretical guarantees. A more detailed comparison can be referred to Appendix A.3. In comparison, our proposed TQL is not only designed for a specific risk measure but is general for all kinds of risk measures.

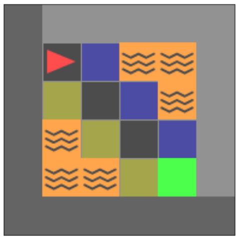
There are also several works (Bäuerle and Ott, 2011; Chow and Ghavamzadeh, 2014; Tamar et al., 2015) utilizing dynamic risk measures as their objective, which considers per-step risk instead of the static (trajectory-wise) risk in this paper. Dynamic risk has the advantage of time-consistency, but can be hard to estimate practically and short-sighted due to per-step optimization. We present a more detailed discussion on dynamic and static risk in Appendix A.4.

6. Experiments

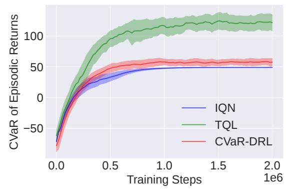
In this section, we design a series of experiments aimed to seek out: **RQ1**: Can our proposed TQL fit the ground-truth risk measures? **RQ2**: Can TQL find the optimal risk-sensitive policy and achieve better overall performance?

Environments. In order to examine the ability to optimize risk-sensitive policy, we design two specified environments for discrete and continuous control, respectively. For discrete actions, we design a risky mini-grid task shown in Fig. 2a; for continuous control, we augment extra risky penalties upon the continuous Mountain-Car environment (Moore, 1990), see Section 6.1 for more details.

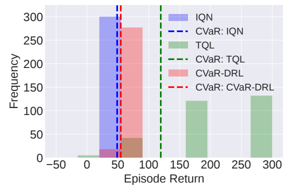
Implementation, baselines, and metric. For discrete action space, we implement TQL based on IQN (Dabney et al.,



(a) Designed mini-grid task.



(b) Learning curves.



(c) Final return distributions.

Figure 2: Mini-grid experiments designed for learning `CVaR` objective. (a) Illustration of risky mini-grid environment. The agent starts at the upper left corner of the grid (red triangle), and reaches the bottom right green grid to end the episode. At each timestep, the agent receives a constant penalty of -2. The yellow grids give a $+100$ bonus with the probability of $p = 0.75$ and 0 with the probability of $p = 0.25$, while the blue grids always give a reward of $+20$. Each yellow or blue grid can give its reward only once. The orange grids have a heavy penalty of -100 to avoid the agent from going there. (b-c) Experiment results on the task: (b) Vanilla IQN quickly converges to a sub-optimal solution; CVaR-DRL discovers a slightly better policy; TQL finds the optimal policy. (c) The return distributions of vanilla IQN and CVaR-DRL are more conservative, while that of TQL results in a higher `CVaR`.

2018a) to obtain the value distribution, and compare TQL with vanilla IQN and CVaR-DRL, a specific solution for learning a risk-sensitive policy towards better `CVaR`, proposed by (Hong Lim and Malik, 2022). For continuous control problems, we combine TD3 (Fujimoto et al., 2018) with IQN (Dabney et al., 2018a), named IQTD3, by replacing the critics in TD3 with distributional critics. We further build TQL upon IQTD3 and take IQTD3 as the baseline algorithm. For comparison, each algorithm is optimized towards various risk-sensitive objectives that are represented by different risk measures, including `mean`, `CVaR`, `POW`, and `Wang`, whose detailed description is in Appendix Section A.1; and the evaluation metrics are also those risk measures.

6.1. Results and Analysis

Value distribution analysis on 3-state MDP. In Section 3.2, we have illustrated in Fig. 1 that vanilla distributional RL is not able to reveal the global optimal risk-sensitive policy and its value. To validate, we learn the return distribution with a tabular version of vanilla IQN and TQL respectively, and visualize the learned return distribution in Fig. 3. The results show that vanilla distributional RL tends to learn $Z(s_1, \cdot)$ first as it is irrelevant to a_0, and thus $a_1 = 1$ will be chosen under s_1. However, when learning $Z(s_0, \cdot)$, Bellman update will use $Z(s_1, a_1)$ in target, ignoring all trajectories

Table 1: The action sequences of IQN and TQL policies at different training steps. IQN converges from back to front; CVaR-DRL leads to a slightly-better policy; TQL finds out the global optimum.

# Training steps	IQN	CVaR-DRL	TQL
2×10^4	$[\downarrow, \rightarrow, \downarrow, \rightarrow, \rightarrow, \downarrow]$	$[\rightarrow, \downarrow, \downarrow, \rightarrow, \rightarrow, \downarrow]$	$[\rightarrow, \downarrow, \downarrow, \rightarrow, \rightarrow, \downarrow]$
1×10^5	$[\downarrow, \rightarrow, \rightarrow, \downarrow, \rightarrow, \downarrow]$	$[\rightarrow, \downarrow, \downarrow, \rightarrow, \rightarrow, \downarrow]$	$[\downarrow, \rightarrow, \downarrow, \rightarrow, \rightarrow, \downarrow]$
2×10^5	$[\rightarrow, \downarrow, \rightarrow, \downarrow, \rightarrow, \downarrow]$	$[\downarrow, \rightarrow, \rightarrow, \downarrow, \rightarrow, \downarrow]$	$[\downarrow, \rightarrow, \downarrow, \rightarrow, \downarrow, \rightarrow]$

where $a_1 = 0$. On the contrary, TQL learns historical value distribution $\tilde{Z}$, which enforces the agent to consider all possible trajectories and thus reveal the optimal solution where $a_0 = a_1 = 0$.

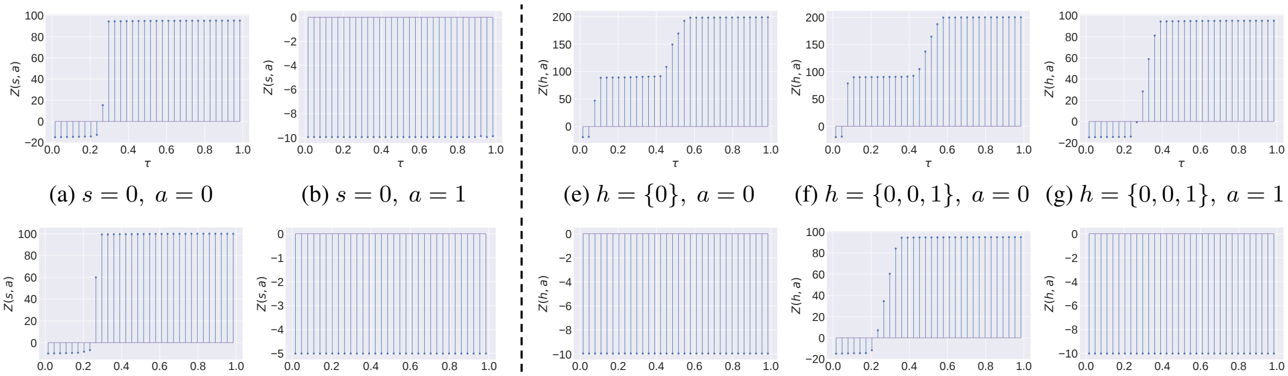
Discrete control evaluations. We first show the result of the discrete mini-grid task, which is designed for learning `CVaR` objective. We present the learning curves of TQL and vanilla IQN in Fig. 2b, which indicates that IQN consistently converges to the sub-optimal solution of visiting blue grids, similar to its behavior in the above-mentioned 3-state MDP. CVaR-DRL does improve the `CVaR` of historical return distribution to some extent, while it still produces sub-optimal policies (see Appendix A.3 for a more detailed analysis). In contrast, TQL is able to discover a better policy that achieves significantly higher `CVaR` than the vanilla IQN baseline. Furthermore, in Fig. 2c, we visualize the final policy's return distribution. The blue and green bars indicate the frequency of episode returns for vanilla IQN and TQL respectively, and the corresponding dashed lines show the `CVaR` of return distribution for two policies. TQL is very likely to obtain high positive returns with little risk of negative returns, while vanilla IQN's return is always negative due to its Markovian policy.

To better understand the difference in the optimization process, we further illustrate how the policy evolves during the training process in Tab. 1. In particular, we observe that vanilla IQN converges from the end of the episode to the beginning due to its updating mechanism of dynamic programming, and its property of Markovian prevents it from finding the global optimum; moreover, CVaR-DRL fails due to its approximation in `CVaR` estimation but leads to a slightly-better policy. However, TQL is always doing a global search and thus finally reveals the optimal policy.

Continuous control evaluations. To further learn on continuous risk-sensitive control problems with TQL, we design a risky penalty for the Mountain-Car environment:

$$R_{\text{risky}}(s, a) = \begin{cases} -c \cdot (2 - |a|), & p = \frac{1}{4 - 3|a|} \\ 0, & p = 1 - \frac{1}{4 - 3|a|} \end{cases} . \quad (23)$$

where $c \in [0, 1]$ is a scaling factor that controls the degree of risk related to the scale of actions. At each timestep, we augment the original reward with the risky penalty R_{risky}. Generally, actions close to 0 will result in higher expected accumulated rewards. However, to complete the task as fast as possible, the agent should choose larger actions that are close to 1, leading to more risky penalties. We compare


(a) $s=0,\ a=0$ (b) $s=0,\ a=1$ (e) $h=\{0\},\ a=0$ (f) $h=\{0,0,1\},\ a=0$ (g) $h=\{0,0,1\},\ a=1$

(c) $s=1,\ a=0$ (d) $s=1,\ a=1$ (h) $h=\{0\},\ a=1$ (i) $h=\{0,1,1\},\ a=0$ (j) $h=\{0,1,1\},\ a=1$

Figure 3: Predicted return distribution on different s or h and a input. The left 4 figures correspond to IQN: IQN first learns $Z(s_1,\cdot)$, see (c-d). It finds $a_1=1$ better and keeps this strategy when learning $Z(s_0,\cdot)$, leading to (a) and (b); the right 6 figures correspond to our proposed method TQL: (e) matches (f) as taking $a_1=0$ has better `CVaR` after taking $a_0=0$; (h) matches (j) as taking $a_1=1$ has better `CVaR` after taking $a_0=1$. Overall, the policy corresponds to (e) and (f), which achieve global optimum.

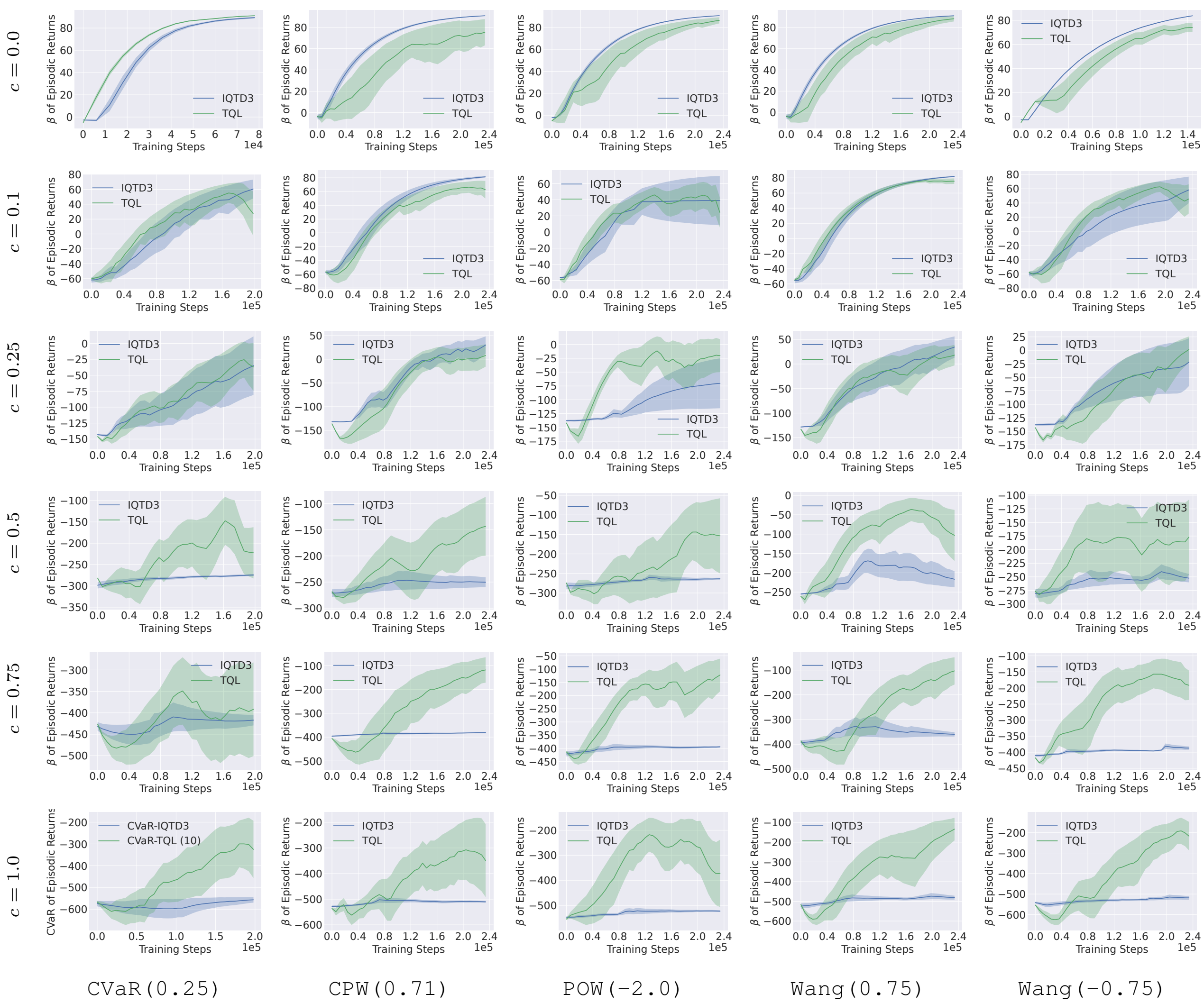


Figure 4: Learning curves on modified Mountain-Car environment with different risk measures as objective, measured by risk measures.

IQTD3 with the proposed TQL and show the results for various risk measures in Fig. 4.

Overall, when the potential risk is larger (i.e., larger risky penalty $c \in \{0.5, 0.75, 1.0\}$), TQL significantly outperforms IQTD3. The Markovian policy learned by IQTD3 can hardly find out how to complete the control task due to its short-sighted decision-making, while TQL consistently learns a better policy. When the risk is smaller, namely $c \in \{0.25, 0.1, 0.0\}$, the difference between TQL and IQTD3 becomes smaller, and both algorithms can learn an optimal risk-sensitive policy.

7. Conclusion and Future Work

In this paper, we present an in-depth analysis of the biased objective issue of the existing RSRL methods, and correspondingly propose Trajectory Q-Learning (TQL), a distributional RL algorithm for learning the optimal policy in RSRL. We justify the theoretical property of TQL and prove it converges to the optimal solution. Our experiments and the detailed analysis on both discrete and continuous control tasks validate the advantage of TQL in risk-sensitive settings. In future work, we plan to extend TQL to more complex tasks and real-world applications.

Acknowledgements

We thank Zhengyu Yang, Ming Zhou and Zheyuan Hu for their helpful discussions. The SJTU team is supported by "New Generation of AI 2030" Major Project (2018AAA0100900), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102) and National Natural Science Foundation of China (62076161). The author Minghuan Liu is also supported by Wu Wen Jun Honorary Doctoral Scholarship, AI Institute, Shanghai Jiao Tong University. We sincerely thank all anonymous reviewers for their helpful feedback to revise our first manuscript.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. 2017. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*. PMLR, 22–31.

Nicole Bäuerle and Jonathan Ott. 2011. Markov decision processes with average-value-at-risk criteria. *Mathematical Methods of Operations Research* 74 (2011), 361–379.

Marc G Bellemare, Will Dabney, and Rémi Munos. 2017. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*. PMLR, 449–458.

Marc G Bellemare, Will Dabney, and Mark Rowland. 2023. *Distributional reinforcement learning*. MIT Press.

Yinlam Chow and Mohammad Ghavamzadeh. 2014. Algorithms for CVaR optimization in MDPs. In *Proceedings of the 28th Advances in Neural Information Processing Systems (NeurIPS)*. 3509–3517.

Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. 2018. A lyapunov-based approach to safe reinforcement learning. *Proceedings of the 32nd Advances in Neural Information Processing Systems (NeurIPS)* (2018), 8103–8112.

Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. 2018a. Implicit quantile networks for distributional reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*. PMLR, 1096–1105.

Will Dabney, Mark Rowland, Marc G Bellemare, and Rémi Munos. 2018b. Distributional reinforcement learning with quantile regression. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*.

Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, Todd Hester, Cosmin Paduraru, and Yuval Tassa. 2018. Safe exploration in continuous action spaces. *arXiv preprint arXiv:1801.08757* (2018).

Scott Fujimoto, Herke Hoof, and David Meger. 2018. Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*. PMLR, 1587–1596.

Scott Fujimoto, David Meger, and Doina Precup. 2019. Off-policy deep reinforcement learning without exploration. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*. PMLR, 2052–2062.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*. PMLR, 1861–1870.

Shiau Hong Lim and Ilyas Malik. 2022. Distributional Reinforcement Learning for Risk-Sensitive Policies. In *Proceedings of the 36th Advances in Neural Information Processing Systems (NeurIPS)*.

Peter J Huber. 1964. Robust estimation of a location parameter. *The Annuals of Mathematical Statistics* 1 (1964), 73–101.

Ramtin Keramati, Christoph Dann, Alex Tamkin, and Emma Brunskill. 2020. Being optimistic to be conservative: Quickly learning a cvar policy. In *Proceedings of the*

34th AAAI Conference on Artificial Intelligence (AAAI). 4436–4443.

Roger Koenker and Kevin F Hallock. 2001. Quantile regression. *Journal of economic perspectives* 15, 4 (2001), 143–156.

Xiaoteng Ma, Li Xia, Zhengyuan Zhou, Jun Yang, and Qianchuan Zhao. 2020. DSAC: Distributional Soft Actor Critic for Risk-Sensitive Reinforcement Learning. *arXiv preprint arXiv:2004.14547* (2020).

Yecheng Jason Ma, Dinesh Jayaraman, and Osbert Bastani. 2021. Conservative Offline Distributional Reinforcement Learning. In *Proceedings of the 35th Advances in Neural Information Processing Systems (NeurIPS)*. 19235–19247.

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. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533.

Andrew William Moore. 1990. *Efficient Memory-based Learning for Robot Control*. Technical Report. University of Cambridge.

Alfred Müller. 1997. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability* 29, 2 (1997), 429–443.

Marc Rigter, Bruno Lacerda, and Nick Hawes. 2022. One Risk to Rule Them All: A Risk-Sensitive Perspective on Model-Based Offline Reinforcement Learning. *arXiv preprint arXiv:2212.00124* (2022).

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of go without human knowledge. *Nature* 550, 7676 (2017), 354–359.

Richard S Sutton and Andrew G Barto. 2018. Reinforcement learning: An introduction. (2018).

Aviv Tamar, Yinlam Chow, Mohammad Ghavamzadeh, and Shie Mannor. 2015. Policy gradient for coherent risk measures. In *Proceedings of the 29th Advances in Neural Information Processing Systems (NIPS)*, Vol. 28.

Yichuan Charlie Tang, Jian Zhang, and Ruslan Salakhutdinov. 2019. Worst Cases Policy Gradients. In *Proceedings of the 3rd Annual Conference on Robot Learning (CoRL)*, Vol. 100. 1078–1093.

Amos Tversky and Daniel Kahneman. 1992. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and Uncertainty* 5 (1992), 297–323.

Núria Armengol Urpí, Sebastian Curi, and Andreas Krause. 2021. Risk-Averse Offline Reinforcement Learning. *arXiv preprint arXiv:2102.05371* (2021).

Shaun Wang. 1996. Premium calculation by transforming the layer premium density. *ASTIN Bulletin: The Journal of the IAA* 26, 1 (1996), 71–92.

George Wu and Richard Gonzalez. 1996. Curvature of the probability weighting function. *Management science* 42, 12 (1996), 1676–1690.

Guan Yang, Minghuan Liu, Weijun Hong, Weinan Zhang, Fei Fang, Guangjun Zeng, and Yue Lin. 2022. PerfectDou: Dominating DouDizhu with Perfect Information Distillation. *arXiv preprint arXiv:2203.16406* (2022).

A. Additional Backgrounds

A.1. Typical Distortion Risk Measures

Distortion risk measure is a large family including various measures. Here we briefly introduce how we derive those used in our experiments from its framework and their properties.

- `Mean` is obtained by an identity distortion function:

$$h^{-1}_{\texttt{mean}}(\tau) = \tau, \quad \forall \tau \in [0,1] \ . \tag{24}$$

  `Mean` treats each quantile equally and serves as a risk-neutral measure, which indicates the unconditioned overall performance of a random variable.

- η-`CVaR` is obtained by a linear projection of fractions:

$$h^{-1}_{\eta-\texttt{CVaR}}(\tau) = \eta\tau \ . \tag{25}$$

  `CVaR` is always risk-averse, and smaller η makes it more conservative.

- η-`Wang` is a simple measure that controls risk preference by translating the c.d.f. of standard Gaussian:

$$h^{-1}_{\eta-\texttt{Wang}}(\tau) = \Phi(\Phi^{-1}(\tau) + \eta) \ . \tag{26}$$

  where Φ is the c.d.f. of standard Gaussian distribution. Positive η corresponds to risk-seeking, and negative η corresponds to risk-aversity.

- η-`CPW` is introduced in Tversky and Kahneman (1992), it is neither globally risk-averse nor risk-seeking.

$$h^{-1}_{\eta-\texttt{CPW}}(\tau) = \Phi(\Phi^{-1}(\tau) + \eta) \ . \tag{27}$$

  Wu and Gonzalez (1996) proposed that $\eta = 0.71$ matches human subjects well.

- η-`POW` is proposed in Dabney et al. (2018a). It is a simple power formula for risk-averse ($\eta < 0$) or risk-seeking ($\eta > 0$) policies:

$$h^{-1}_{\eta-\texttt{POW}}(\tau) = \begin{cases} \tau^{\frac{1}{1+|\eta|}}, & \text{if } \eta \geq 0 \\ 1-(1-\tau)^{\frac{1}{1+|\eta|}} & \text{otherwise} \end{cases} \ . \tag{28}$$

A.2. Quantile Regression and Quantile Huber Loss

Quantile regression. Dabney et al. (2018b) first to approximate the return distribution with quantiles and optimize the value function with quantile regression (Koenker and Hallock, 2001), which uses the quantile regression loss to estimate quantile functions of a distribution.

The quantile regression loss is an asymmetric convex loss function that penalizes overestimation and underestimation errors with different weights. For a distribution Z with c.d.f. $F_Z(z)$, and a given quantile τ, the value of quantile function $F_Z^{-1}(\tau)$ can be characterized as the minimizer of the following *quantile regression loss*:

$$\begin{aligned} \mathcal{L}^{\tau}_{\mathrm{QR}}(\theta) &:= \mathbb{E}_{\hat{z}\sim Z}\left[\rho_\tau(\hat{z}-\theta)\right] \ , \text{ where} \\ \rho_\tau(u) &= u(\tau - \delta_{u<0}) \ , \ \forall u \in \mathbb{R} \end{aligned} \tag{29}$$

Quantile Huber loss. Dabney et al. (2018b) found that the non-smoothness at zero could limit performance when using non-linear function approximation, hence proposed a modified quantile loss, namely the *quantile Huber loss*.

The Huber loss is given by Huber (1964):

$$\mathcal{L}_\kappa(u) = \begin{cases} \frac{1}{2}u^2 & \text{if } |u| \leq \kappa \\ \kappa(|u| - \frac{1}{2}\kappa) & \text{otherwise} \end{cases} \ , \tag{30}$$

and the quantile Huber loss is then an asymmetric version of the Huber loss:

$$\rho^\kappa_\tau(u) = |\tau - \delta_{u<0}|\mathcal{L}_\kappa(u) \ . \tag{31}$$

A.3. Comparison of Our Work to (Hong Lim and Malik, 2022)

The recent work of (Hong Lim and Malik, 2022) imposes a similar issue of optimizing `CVaR` in previous RSRL methods, and proposes modifications to existing algorithm methods that use a moving threshold for episodic `CVaR` estimation and include a new distributional Bellman operator. They show that the optimal `CVaR` policy corresponds to the fixed point of their proposed distributional Bellman operator.

However, (Hong Lim and Malik, 2022) does not show the convergence property of their proposed distributional Bellman operator, and their solution is available specifically for `CVaR` policy optimization. Furthermore, their proposed moving threshold is approximated with the learned return distribution, which also impacts the ability to find the optimal policy. In contrast, our proposed TQL is provided with convergence property and enables optimization under all kinds of risk measures. Experiments in Section 6.1 also show the advantage of TQL over (Hong Lim and Malik, 2022) in practical tasks.

A.4. Static Risk vs. Dynamic Risk

In this paper, we consider static risk measures β over the whole trajectory. However, some other works (Bäuerle and Ott, 2011; Chow and Ghavamzadeh, 2014; Tamar et al., 2015) consider dynamic risk measures ρ, which is defined recursively over a trajectory $\tau = (s_0, a_0, s_1, a_1, \cdots)$:

$$\rho[\tau] = \rho\left[R(s_0, a_0) + \gamma\rho\left[R(s_1, a_1) + \gamma\rho[R(s_2, a_2) + \cdots]\right]\right] \tag{32}$$

Dynamic risk measures have the advantage of time consistency, and (Bäuerle and Ott, 2011; Chow and Ghavamzadeh, 2014; Tamar et al., 2015) has conducted an in-depth analysis of the policy optimization based on dynamic risk measures. However, dynamic risk measures are more short-sighted and hard to estimate in practical tasks due to their per-step definition, and in the real-world people care more about the overall risk over the whole decision process. Therefore, static risk measures are most commonly used for evaluation in fields such as finance and medical treatments. Furthermore, the control problem over static risk measures have been shown to be non-Markovian and non-stationary (Bellemare et al., 2023), hence it can be hardly resolved by DP style RSRL algorithms, which motivates the use of episodic return distribution and history-dependent policy in our TQL algorithm.

B. Algorithms

Discrete control. We present the step-by-step algorithm of TQL for the policy with discrete actions in Algo. 1.

Algorithm 1 Trajectory Q-Learning (TQL) (Discrete)

```
Input: Parameter vectors θ, ψ, and Risk Measure β
Initialize parameters θ, ψ, and replay buffer D.
for each iteration do
  for each environment step do
    a_t ~ argmax_a β[Z(h_t, a)]
    s_{t+1} ~ p(s_{t+1} | s_t, a_t)
    D ← D ∪ {(h_t, a_t, r(s_t, a_t), s_{t+1})}
  end for
  for each gradient step do
    ψ ← ψ − ∇_ψ J_Q(ψ)    (Eq. (22))
    θ ← θ − ∇_θ J_Q(θ)    (Eq. (21))
  end for
end for
```

Continuous control. We present the step-by-step algorithm of TQL for continuous control in Algo. 2.

Algorithm 2 Trajectory Q-Learning (TQL) (Continuous)

Input: Parameter vectors ϕ, θ, ψ, and Risk Measure β
Initialize parameters ϕ, θ, ψ, and replay buffer $\mathcal{D}$.
for each iteration **do**
  for each environment step **do**
    $a_t \sim \pi_\phi\left(a_t|h_t\right)$
    $s_{t+1} \sim p\left(s_{t+1}|s_t, a_t\right)$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \left\{\left(h_t, a_t, r\left(s_t, a_t\right), s_{t+1}\right)\right\}$
  end for
  for each gradient step **do**
    $\psi \leftarrow \psi - \nabla_\psi J_Q(\psi)$ (Eq. (22))
    $\theta \leftarrow \theta - \nabla_\theta J_Q(\theta)$ (Eq. (21))
    $\phi \leftarrow \phi - \nabla_\phi J_\pi(\phi)$ (Eq. (20))
  end for
end for

C. Proof

C.1. Proof of Lemma 3.1

Lemma C.1 (Value iteration theorem)**.** *Recursively applying the distributional Bellman optimality operator $\mathcal{T}^* Z_{k+1} = Z_k$ on arbitrary value distribution Z_0 solves the objective Eq. (4) when β is exactly* `mean` *where the optimal policy is obtained via Eq. (5), and for $Z_1, Z_2 \in \mathcal{Z}$, we have:*

$$\|\mathbb{E}\mathcal{T}^* Z_1 - \mathbb{E}\mathcal{T}^* Z_2\|_\infty \leq \gamma \|\mathbb{E}Z_1 - \mathbb{E}Z_1\|_\infty \ , \tag{33}$$

and in particular $\mathbb{E}Z_k \to \mathbb{E}Z^*$ exponentially quickly.

Proof. The RSRL objective for `mean` policy is

$$\underset{A_0,\cdots,A_{T-1}}{\arg\max} \ \mathbb{E}\left[Z(S_0, A_0)\right] \ . \tag{34}$$

From the Bellman equation, we can unfold $Z(S_0, A_0)$ and get an equivalent form:

$$\underset{A_0,\cdots,A_{T-1}}{\arg\max} \ \mathbb{E}\left[R(S_0, A_0) + \gamma Z(S_1, A_1)\right] \ , \text{ where } S_1 \sim \mathcal{P}(S_1|S_0, A_0). \tag{35}$$

Notice that `mean` is linearly additive, hence Eq. (35) can be further divided into two parts of optimization:

$$\underset{A_0}{\arg\max} \ \mathbb{E}\left[R(S_0, A_0) + \gamma \max_{A_1,\cdots,A_{T-1}} \mathbb{E}\left[Z(S_1, A_1)\right]\right] \ . \tag{36}$$

Repeating this process on $Z(S_1, A_1)$ and further, we will obtain the dynamic programming objective as following:

$$\begin{aligned} & \underset{A_0}{\arg\max} \ \mathbb{E}\left[R(S_0, A_0) + \gamma \max_{A_1,\cdots,A_{T-1}} \mathbb{E}\left[Z(S_1, A_1)\right]\right] \\ \Leftrightarrow \ & \underset{A_0}{\arg\max} \ \mathbb{E}\Bigg[R(S_0, A_0) + \gamma Z\Bigg(S_1, \underset{A_1}{\arg\max}\, \mathbb{E}\Big[R(S_1, A_1) \\ & \quad + \ \gamma \max_{A_2,\cdots,A_{T-1}} \mathbb{E}\left[Z(S_2, A_2)\right]\Big]\Bigg)\Bigg] \\ \Leftrightarrow \ & \cdots \end{aligned} \tag{37}$$

which will finally get to the single-step target.

For $Z_1, Z_2 \in \mathcal{Z}$, using the linearly additive property of `mean`, we have

$$\begin{aligned}\|\mathbb{E}\, \mathcal{T}_D^* Z_1 - \mathbb{E}\, \mathcal{T}_D^* Z_2\|_\infty &= \|\mathcal{T}_E^*\, \mathbb{E} Z_1 - \mathcal{T}_E^*\, \mathbb{E} Z_2\|_\infty \\ &\leq \gamma \|\mathbb{E} Z_1 - \mathbb{E} Z_2\|_\infty \end{aligned} \tag{38}$$

where $\mathcal{T}_D^*$ denotes the distributional operator and $\mathcal{T}_E^*$ denotes the usual operator. □

C.2. Proof of Theorem 3.2

Theorem C.2. *Recursively applying risk-sensitive Bellman optimality operator $\mathcal{T}_\beta^*$ w.r.t. risk measure β does not solve the RSRL objective Eq. (4) and $\beta[Z_k]$ is not guaranteed to converge to $\beta[Z^*]$ if β is not* `mean` *or an affine in* `mean`*.*

Proof. We want to show that

$$\left\|\beta\left[\mathcal{T}_\beta^* Z_1(s,a)\right] - \beta\left[\mathcal{T}_\beta^* Z_2(s,a)\right]\right\|_\infty \leq \gamma \left\|\beta\left[Z_1(s,a)\right] - \beta\left[Z_2(s,a)\right]\right\|_\infty$$

is not guaranteed to be true. We prove this by a counterexample. Given a risk measure β, consider two value distributions Z_1 and Z_2 where $Z_1(s,a) = Z_2(s,a),\ \forall s \in \mathcal{S}, a \in \mathcal{A}$. Assume there are more than one optimal actions $a \in \mathcal{A}_1^* \subset \mathcal{A}$ ($|\mathcal{A}_1^*| > 1$) at state s_1 in terms of risk measure $\beta[Z_i(s_1, \cdot)],\ i \in \{0,1\}$, i.e.,

$$\forall\, a^* \in \mathcal{A}_1^*,\ a' \in \mathcal{A},\ \beta[Z_i(s_1, a^*)] \geq \beta[Z_i(s_1, a')]\ .$$

and these optimal actions correspond to different value distributions, i.e.

$$Z_i(s_1, a) \overset{D}{\neq} Z_i(s, a'),\ \forall\, a, a' \in \mathcal{A}_1^*,\ i \in \{0,1\}\ .$$

In case where we do not have a strict preference over $\mathcal{A}_1^*$, the update formula for $Z_i(s_0,a_0),\ i \in \{0,1\}$ where $M(s_0,a_0) = s_1$ will be

$$Z_i(s_0,a_0) \leftarrow R(s_0,a_0) + \gamma Z_i(s_1, a),\ \forall a \in \mathcal{A}_1^*\ . \tag{39}$$

When Eq. (39) uses $a = a_1 \in \mathcal{A}_1^*$ in the right-hand side, it holds that

$$Z_i(s_0,a_0) \overset{D}{\neq} R(s_0,a_0) + \gamma Z_i(s_1, a_1'),\ a_1' \in \mathcal{A}_1^* \text{ and } a_1' \neq a_1\ .$$

Although $\beta[Z_i(s_1,a)],\ \forall\, a \in \mathcal{A}_1^*,\ i \in \{0,1\}$ are all the same, consider one update as follows:

$$\begin{aligned} Z_1'(s_0,a_0) &\leftarrow R(s_0,a_0) + \gamma Z_1(s_1,a),\ a \in \mathcal{A}_1^*\ , \\ Z_2'(s_0,a_0) &\leftarrow R(s_0,a_0) + \gamma Z_2(s_1,a'),\ a' \in \mathcal{A}_1^* \text{ and } a' \neq a\ . \end{aligned}$$

Notice that $\|\beta[Z_1] - \beta[Z_2]\|_\infty = 0$. Since $R(s_0,a_0)$ and $Z_i(s_1,\cdot)$ can be arbitrary distributions, we can always find such $R(s_0,a_0)$ and $Z_i(s_1,\cdot)$ that $Z_1' \neq Z_2'$ when β is not `mean` or an affine in `mean`, as we show an example for `CVaR` as follows.

$\beta = \texttt{CVaR}(\eta = 0.1)$	s_0, a_0	s_1, a_0	s_1, a_1
R	$\begin{cases} 100, p = 0.9 \\ -10, p = 0.1 \end{cases}$	$\begin{cases} 100, p = 0.9 \\ -10, p = 0.1 \end{cases}$	-10
Z_i	$\begin{cases} 90, p = 0.9 \\ -20, p = 0.1 \end{cases}$	$\begin{cases} 100, p = 0.9 \\ -10, p = 0.1 \end{cases}$	-10
$\beta[Z_i]$	-20	-10	-10
$Z_1'(s_0,a_0) \leftarrow R(s_0,a_0) + \gamma Z_1(s_1,a_0),\quad Z_2'(s_0,a_0) \leftarrow R(s_0,a_0) + \gamma Z_2(s_1,a_1)$			
Z_1'	$\begin{cases} 200, p = 0.81 \\ 90, p = 0.18 \\ -20, p = 0.01 \end{cases}$	$\begin{cases} 100, p = 0.9 \\ -10, p = 0.1 \end{cases}$	-10
Z_2'	$\begin{cases} 90, p = 0.9 \\ -20, p = 0.1 \end{cases}$	$\begin{cases} 100, p = 0.9 \\ -10, p = 0.1 \end{cases}$	-10
$\beta[Z_1']$	79	-10	-10
$\beta[Z_2']$	-20	-10	-10

In this case $\|\beta[Z'_1] - \beta[Z'_2]\|_\infty > \gamma\|\beta[Z_1] - \beta[Z_2]\|_\infty = 0$. Therefore, $\|\beta[Z'_1] - \beta[Z'_2]\|_\infty \leq \gamma\|\beta[Z_1] - \beta[Z_2]\|_\infty$ is not guaranteed to be true.

Finally, a straight forward deduction is that, repeating using the following two formula stochastically to update Z will never reach a convergence.

$$
\begin{aligned}
&Z'(s_0, a_0) \leftarrow R(s_0, a_0) + \gamma Z(s_1, a),\ a \in \mathcal{A}_1^*\ , \\
&Z'(s_0, a_0) \leftarrow R(s_0, a_0) + \gamma Z(s_1, a'),\ a' \in \mathcal{A}_1^* \text{ and } a' \neq a\ .
\end{aligned}
$$

□

C.3. Proof of Theorem 4.1

Theorem C.3 (Policy Evaluation for $\mathcal{T}_h^\pi$)**.** *$\mathcal{T}_h^\pi : \mathcal{Z} \to \mathcal{Z}$ is a γ-contraction in the metric of the maximum form of p-Wasserstein distance $\bar{d}_p$.*

Proof. Let's first define the transition operator $P^\pi : \mathcal{Z} \to \mathcal{Z}$

$$
\begin{aligned}
&P^\pi Z(h, a) \triangleq Z(h', A') \\
&\qquad\qquad h' = M(h, a) = M(s, a),\ A' \sim \pi(\cdot|H'),
\end{aligned}
$$

where h' is actually s' that also falls in the space of $\mathcal{H}$. Then, we have:

$$
\begin{aligned}
&\ d_p\left(\mathcal{T}_h^\pi Z_1(h, a), \mathcal{T}_h^\pi Z_2(h, a)\right) \\
=&\ d_p\left(R_{0:t} + \gamma^{t+1} P^\pi Z_1(h, a), R_{0:t} + \gamma^{t+1} P^\pi Z_2(h, a)\right) \\
\leq&\ \gamma^{t+1} d_p\left(P^\pi Z_1(h, a), P^\pi Z_2(h, a)\right) \\
\leq&\ \gamma^{t+1} \sup_{h', a'} d_p\left(P^\pi Z_1(h', a'), P^\pi Z_2(h', a')\right) \\
\leq&\ \gamma \sup_{h', a'} d_p\left(P^\pi Z_1(h', a'), P^\pi Z_2(h', a')\right)\ .
\end{aligned}
\tag{40}
$$

Then it is easy to see

$$
\begin{aligned}
&\bar{d}_p\left(\mathcal{T}_h^\pi Z_1(h, a), \mathcal{T}^\pi Z_2(h, a)\right) \\
=&\sup_{h, a} d_p\left(\mathcal{T}_h^\pi Z_1(h, a), \mathcal{T}_h^\pi Z_2(h, a)\right) \\
\leq&\gamma \sup_{h', a'} d_p\left(P^\pi Z_1(h', a'), P^\pi Z_2(h', a')\right) \\
=&\gamma \bar{d}_p(Z_1, Z_2)\ .
\end{aligned}
\tag{41}
$$

□

C.4. Proof of Theorem 4.2

Theorem C.4 (Policy Improvement for $\mathcal{T}_{h,\beta}^*$)**.** *For two deterministic policies π and π', if π' is obtained by $\mathcal{T}_{h,\beta}^*$:*

$$
\pi'(h_t) = \arg\max_{a \in \mathcal{A}}\ \beta\left[Z^\pi(h_t, a)\right]\ ,
$$

then the following inequality holds

$$
\beta\left[Z^\pi(h_t, \pi(h_t))\right] \leq \beta\left[Z^{\pi'}(h_t, \pi'(h_t))\right]\ .
$$

Proof. As π' is a greedy policy w.r.t. $\beta\left[Z^\pi\right]$, we have

$$
\beta\left[Z^\pi(h_t, \pi(h_t))\right] \leq \beta\left[Z^\pi(h_t, \pi'(h_t))\right]\ .
$$

Since π' is a deterministic policy, we can denote $a'_t = \pi'(h_t)$, and unfold $Z^\pi(h_t, a'_t)$, we have

$$\begin{aligned} Z^\pi(h_t, a'_t) &= R_{0:t-1} + \gamma^t R(s_t, a'_t) \\ &\quad + \gamma^{t+1} Z^\pi\left(\{\tilde{s}_{t+1}\}, \pi(\cdot|h_t \cup \{a'_t, \tilde{s}_{t+1}\})\right) \\ &= Z^\pi\left(h_t \cup \{a'_t, \tilde{s}_{t+1}\}, \pi(\cdot|h_t \cup \{a'_t, \tilde{s}_{t+1}\})\right) \ , \end{aligned}$$

where $\tilde{s}_{t+1} = M(s_t, a'_t)$. Denoting $\tilde{h}_{t+1} = h_t \cup \{a'_t, \tilde{s}_{t+1}\}$, $\tilde{a}_{t+1} = \pi(\tilde{h}_{t+1})$ and $\tilde{a}'_{t+1} = \pi'(\tilde{h}_{t+1})$, we further have

$$\begin{aligned} & Z^\pi(h_t, a'_t) = Z^\pi\left(\tilde{h}_{t+1}, \tilde{a}_{t+1})\right) \\ & \beta\left[Z^\pi(\tilde{h}_{t+1}, A_{t+1}))\right] \leq \beta\left[Z^\pi(\tilde{h}_{t+1}, \tilde{a}'_{t+1})\right] \ . \end{aligned}$$

Putting them together, we obtain

$$\begin{aligned} \beta\left[Z^\pi(h_t, a_t))\right] \leq\ & \beta\left[Z^\pi(h_t, a'_t)\right] \\ =\ & \beta\left[Z^\pi\left(\tilde{h}_{t+1}, \tilde{a}_{t+1})\right)\right] \\ \leq\ & \beta\left[Z^\pi(\tilde{h}_{t+1}, \tilde{a}'_{t+1})\right] \\ =\ & \beta\left[Z^\pi(\tilde{h}_{t+1} \cup \{a'_{t+1}, \tilde{s}_{t+2}\}, \tilde{a}'_{t+2})\right] \\ \leq\ & \cdots\cdots \\ \leq\ & \beta\left[Z^{\pi'}(h_t, a'_t)\right] \end{aligned} \tag{42}$$

□

C.5. Proof of Eq. (14) implying Eq. (13)

Proof. We only consider episodic tasks[1], where each episode terminates at a certain termination state, say, at time step $T+1$. Hereby, we show we can obtain Eq. (13) given Eq. (14) by induction:

- Considering timestep T, where $\forall \pi,\ Z^\pi(h_T, a_T) = R_{0:T}$, Eq. (13) holds, i.e., $\beta[Z^*(h_T, a_T)] = \beta[Z^{\pi^*}(h_T, a_T)]$.
- Given that Eq. (13) holds at timestep $t+1$, namely $\beta[Z^*(h_{t+1}, \cdot)] = \beta[Z^{\pi^*}(h_{t+1}, \cdot)]$. Consider Eq. (14) at timestep t:

$$\begin{aligned} \beta[Z^*(h_t, a_t)] &= \beta\left[R_{0:t} + \gamma^{t+1} Z^*(\{s_{t+1}\}, a^*_{t+1})\right] \\ &= \beta\left[Z^*(h_t \cup \{a_t, s_{t+1}\}, a^*_{t+1})\right] \\ &= \beta\left[Z^{\pi^*}(h_t \cup \{a_t, s_{t+1}\}, a^*_{t+1})\right] \\ &= \beta[Z^{\pi^*}(h_t, a_t)] \ . \end{aligned}$$

  This indicates Eq. (13) still holds at timestep t.
- Given the two statements above, Eq. (14) is sufficient for Eq. (13).

□

C.6. Proof of Theorem 4.3

Theorem C.5. *For $Z_1, Z_2 \in \mathcal{Z}_h$,* HR *optimality operator $\mathcal{T}^*_{h,\beta}$ has the following property:*

$$\|\beta[\mathcal{T}^* Z_1] - \beta[\mathcal{T}^* Z_2]\|_\infty \leq \|\beta[Z_1] - \beta[Z_1]\|_\infty \ , \tag{43}$$

[1]Note such tasks can also be represented in a uniform notation of infinite horizon by adding an absorbing state after termination states, see Sutton and Barto (2018).

Proof. $\forall s \in \mathcal{S}, a \in \mathcal{A}, s' = M(s,a)$,

$$
\begin{aligned}
& \|\beta\left[\mathcal{T}_\beta^* Z_1(h,a)\right] - \beta\left[\mathcal{T}_\beta^* Z_2(h,a)\right]\|_\infty \\
= & \|\max_{a'} \beta\left[R_{0:t} + \gamma^{t+1} Z_1(\{s'\}, a')\right] \\
& \quad - \max_{a'} \beta\left[R_{0:t} + \gamma^{t+1} Z_2(\{s'\}, a')\right]\|_\infty \\
\le & \|\beta\left[R_{0:t} + \gamma^{t+1} Z_1(\{s'\}, a')\right] \\
& \quad - \beta\left[R_{0:t} + \gamma^{t+1} Z_2(\{s'\}, a')\right]\|_\infty \\
= & \|\beta\left[Z_1(h,a)\right] - \beta\left[Z_2(h,a)\right]\|_\infty
\end{aligned}
\tag{44}
$$

□

D. Implementation Details

D.1. Practical Modeling of Episodic Value Distribution

We propose to involve the history information in Section 4, which enables the learning of episodic value distribution. In practice, the summarization of history information can be achieved via any sequence model, e.g. GRU, LSTM, and Transformer.

For our experiments on both MiniGrid and MountainCar tasks, we use a single-layer GRU network for simplicity. We first encode the observations $\{s_t\}$ and $\{a_t\}$ into representations of the same dimension d with two encoders $\mathrm{enc}_s : \mathcal{S} \to \mathbb{R}^d$ and $\mathrm{enc}_a : \mathcal{A} \to \mathbb{R}^d$. Then we concatenate the representations $\mathrm{enc}_s(s_t)$ and $\mathrm{enc}_a(a_t)$ into $[\mathrm{enc}_s(s_t), \mathrm{enc}_a(a_t)] \in \mathbb{R}^{2d}$, and stack the representations until current timestep as the input for the GRU network.

The GRU network produces the summarization of history h_t as

$$
repr_t = \mathrm{GRU}\left([[\mathrm{enc}_s(s_0), \mathrm{enc}_a(a_0)], \cdots, [\mathrm{enc}_s(s_t), \mathrm{enc}_a(a_t)]]\right)
$$

and we finally model the episodic value distribution and history-dependent policy via MLPs with two hidden layers, taking $repr_t$ and a_t as input. It is worth noting that, as conditioning the policy and value function on history will inflate the searching space, we use a rolling window of fixed length $L = 10$ on the history trajectory when computing $repr_t$ on continuous Mountain-Car tasks, i.e.

$$
repr_t = \mathrm{GRU}\left([[\mathrm{enc}_s(s_{t-L+1}), \mathrm{enc}_a(a_{t-L+1})], \cdots, [\mathrm{enc}_s(s_t), \mathrm{enc}_a(a_t)]]\right) \ ,
$$

which will significantly reduce the computation cost and complexity.

D.2. Hyperparameters

We list the hyperparameters of TQL for discrete MiniGrid tasks in Tab. 2.

We list the hyperparameters of TQL for continuous Mountain-Car tasks in Tab. 3.

E. Limitations of This Work

In this paper, we are the first to pose the biased objective problem in the optimization of previous risk-sensitive reinforcement learning (RSRL) algorithms. Accordingly, we propose Trajectory Q-Learning (TQL), as a general solution for RSRL, and verify its effectiveness, both theoretically and empirically. Although we believe this solution is general to a large set of RSRL problems and can be inspiring to the following study on RSRL, it still has limitations in many aspects.

First, our solution assumes deterministic environmental dynamics, which implies a limitation the uncertainty only comes from the randomness of the reward function. In some cases, it is possible to translate a problem with stochastic dynamics structures into one with deterministic dynamics and random rewards. However, RSRL with stochastic dynamics is valuable and worth studying. Regarding this, we point out that a potential solution can be building a transition model and sampling transitions for policy optimization similar to (Rigter et al., 2022), combined with the proposed method in this work. Due to the workload and the value of the problem, we leave it as future work and make this work a good start for a more general solution for RSRL.

Table 2: Hyperparameters on Discrete MiniGrid

ϵ_{init}	0.25
ϵ_{final}	0.001
ϵ_{test}	0.0
Buffer size	3e5
Batch size	32
Learning rate	1e-3
γ	0.99
Sample size	128
Online sample size	64
Target sample size	64
Hidden sizes	[512]
Start timesteps	5000
Target update frequency	500

Table 3: Hyperparameters on Continuous Mountain-Car

Buffer size	5e4
Batch size	128
Policy lr	3e-4
γ	0.99
Sample size	32
Online sample size	64
Target sample size	64
Hidden sizes	[256, 256]
Start timesteps	2.5e4
Soft update weight	0.005

Beyond deterministic dynamics, this work is also limited to evaluating the method on simple and toy environments and lacks the results on large-scale domains and real-world problem settings. We will try our best to test the method on more complex ad realistic problems and report the progress we will have made.