Distributionally Robust Q-Learning

Zijian Liu [1] **Qinxun Bai** [2] **Jose Blanchet** [3] **Perry Dong** [4] **Wei Xu** [2] **Zhengqing Zhou** [5] **Zhengyuan Zhou** [1]

Abstract

Reinforcement learning (RL) has demonstrated remarkable achievements in simulated environments. However, carrying this success to real environments requires the important attribute of robustness, which the existing RL algorithms often lack as they assume that the future deployment environment is the same as the training environment (i.e. simulator) in which the policy is learned. This assumption often does not hold due to the discrepancy between the simulator and the real environment and, as a result, and hence renders the learned policy fragile when deployed.

In this paper, we propose a novel distributionally robust Q-learning algorithm that learns the best policy in the worst distributional perturbation of the environment. Our algorithm first transforms the infinite-dimensional learning problem (since the environment MDP perturbation lies in an infinite-dimensional space) into a finite-dimensional dual problem and subsequently uses a multi-level Monte-Carlo scheme to approximate the dual value using samples from the simulator. Despite the complexity, we show that the resulting distributionally robust Q-learning algorithm asymptotically converges to optimal worst-case policy, thus making it robust to future environment changes. Simulation results further demonstrate its strong empirical robustness.

1. Introduction

Reinforcement learning (RL) has demonstrated remarkable empirical success in simulated environments, with applications spanning domains such as robotics (Kober et al., 2013; Gu et al., 2017), computer vision (Sadeghi & Levine, 2016; Huang et al., 2017), finance (Li et al., 2009; Choi et al., 2009; Deng et al., 2017) and games (Silver et al., 2016; 2018). More recently, reinforcement learning has also been applied to economic domains such as personalized promotions in revenue management, inventory control in supply chains and scheduling in queueing networks. However, carrying this success to real environments requires an attribute that is often missing in the existing literature on policy learning: robustness, because existing RL algorithms often make the implicit assumption that the environment in which the policy is trained will be the same as the environment in which the policy will be deployed. In other words, a policy is evaluated in the same environment from which the training data has been generated.

While the above assumption is arguably a good starting point[1] for developing a rigorous understanding of algorithmic performance of policy learning, it does not capture the complexity of real-world applications because the discrepancy between training and deployment environments is often common and hard to anticipate (and hence account for) in advance. Two common sources of discrepancies are:

1. **Simulator model mis-specification**. In many reinforcement learning (RL) applications, a policy is often first trained in a simulator before being deployed in a real environment. However, simulator models often cannot capture the full complexity of the real environment, and hence will be mis-specified. Further, it is hard to know exactly how the real environment differs from the simulator (otherwise, the simulator would have been augmented/modified to account for that).
2. **Environment shifts**. The underlying environment may shift due to either non-stationarity or a different deployment environment (for the same task). As an example of the latter, a personalized content recommendation

[1]New York University, Stern School of Business [2]Horizon Robotics Inc., CA, USA [3]Department of Management Science and Engineering, Stanford University, CA, USA [4]Department of Electrical Engineering and Computer Sciences, UC Berkeley, CA, USA [5]Arena Technologies. Correspondence to: Zhengyuan Zhou <zzhou@stern.nyu.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

[1]Much like supervised learning was first developed under the same assumption, before adversarial training was recognized as a valuable tool to make empirical predictions robust; see, for instance, (Sinha et al., 2018; Goodfellow et al., 2014; Ganin et al., 2016; Zhang et al., 2019; Tramèr et al., 2017). The topic of domain adaptation (Shimodaira, 2000; Saerens et al., 2002; Ben-David et al., 2010; Courty et al., 2016; Ganin et al., 2016; Sun et al., 2020; Arjovsky et al., 2019; Wang & Deng, 2018; Sagawa et al., 2019) is another approach to address covariate shifts in supervised learning, but the new domain is often assumed to be known.

engine (learned from existing user browsing data collected in one region or market) may need to be deployed in a different region or market, with different population characteristics. Another example occurs in robotics, where a robot trained to perform certain maneuvers (such as walking (Schulman et al., 2013) or folding laundry (Maitin-Shepard et al., 2010)) in an environment can fail catastrophically (Drew, 2015) in a slightly different environment, where the terrain landscape (in walking) is slightly altered or the laundry object (in laundry folding) is positioned differently.

As a result, to learn an effective policy in practice, one must be cognizant of such discrepancies and take them into account during the training stage in order to learn a policy that is robust to the *unknown* environment changes that cannot be avoided and are difficult to know in advance. Whereas distributionally robust learning in the simpler supervised learning setting has been extensively studied in the past decade (see Section 1.1), an emerging literature has only recently been initiated to study this problem in the RL context (Si et al., 2020b; Zhou et al., 2021; Yang et al., 2021; Kishan & Kalathil, 2021). In particular, a common and natural formulation adopted therein is to consider distributional shifts for rewards and/or transition probabilities (defined via different divergence measures or distances between probability distributions), and to learn from data (collected from some generative model) the best policy under the worst-case distributional shift, which thus carries a certain level of robustness to the unknown environment shifts.

Currently, all the existing distributionally robust policy learning algorithms mentioned above (Si et al., 2020b;a; Zhou et al., 2021; Yang et al., 2021; Kishan & Kalathil, 2021) are model-based, which estimate the underlying MDP first before provisioning some policy from it. Although model-based methods are often more sample-efficient and easier to analyze, their drawbacks are also well-understood (Sutton & Barto, 2018; François-Lavet et al., 2018): they are computationally intensive; they require more memory to store MDP models and often do not generalize well to non-tabular RL settings. These issues limit the practical applicability of model-based algorithms, which stand in contrast to model-free algorithms that learn to select actions without first learning an MDP model. Such methods are often more computationally efficient, have less storage overhead, and better generalize to RL with function approximation. In particular, Q-learning (Watkins & Dayan, 1992), as the prototypical model-free learning algorithm, has widely been both studied theoretically and deployed in practical applications. However, Q-learning is not robust (as demonstrated in our simulations), and the policy learned by Q-learning in one environment can perform poorly in another under a worst-case shift (with bounded magnitude). As such, we are naturally led to the following research question:

Can we design a variant of Q-Learning that is distributionally robust?

1.1. Related Work

Robustness has been studied in several settings that are related to (but different from) our investigation. For instance, a rich literature has explored distributionally robust learning and optimization in *supervised learning* (Bertsimas & Sim, 2004; Delage & Ye, 2010; Hu & Hong, 2013; Shafieezadeh-Abadeh et al., 2015; Bayraksan & Love, 2015; Gao & Kleywegt, 2016; Namkoong & Duchi, 2016; Duchi et al., 2016; Staib & Jegelka, 2017; Shapiro, 2017; Lam & Zhou, 2017; Chen et al., 2018; Volpi et al., 2018; Lee & Raginsky, 2018; Nguyen et al., 2018; Yang, 2020; Mohajerin Esfahani & Kuhn, 2018; Zhao & Jiang, 2017; Abadeh et al., 2018; Zhao & Guan, 2018; Gao et al., 2018; Ghosh & Lam, 2019; Blanchet & Murthy, 2019; Duchi & Namkoong, 2018; Lam, 2019; Duchi et al., 2019), where the underlying testing distribution is still the same as the training distribution, and the learner merely uses the distributional uncertainty (around the empirical distribution) to guard against over-generalization due to lack of data. As such, the statistical learning results in this area focus on the setting where the distributional uncertainty decreases with the sample size (and as such, is part of the algorithm rather than the environment); further, it has been well recognized that under certain conditions, this approach is formally equivalent to regularization (Duchi & Namkoong, 2019; Duchi et al., 2016; Gao et al., 2017; Shafieezadeh-Abadeh et al., 2019), which prevents over-fitting in the small data regime.

On the other hand, learning predictive rules in testing distributions that are different from – and often perturbations of – training distributions have also been studied in (Sinha et al., 2018; Goodfellow et al., 2014; Ganin et al., 2016; Zhang et al., 2019; Tramèr et al., 2017), where the training procedure is robustified by first perturbing the original dataset with some synthesized noise before solving a empirical risk minimization problem, which has been observed to work well in a robust manner.

Going beyond the supervised learning setting, a related area to ours is distributionally robust Markov decision processes (MDPs) (González-Trejo et al., 2002; Iyengar, 2005; Xu & Mannor, 2010; El Ghaoui & Nilim, 2005; Wiesemann et al., 2013; Wolff et al., 2012; Mannor et al., 2016; Morimoto & Doya, 2005; Yang, 2020). This line of work has studied the known environment MDP setting (hence no learning) and has mainly focused on the computational issues[2]. Closest to

[2]For instance, they have characterized various types of uncertainty sets, and have shown that for almost all of them, the optimal distributionally robust policy is NP-hard to compute. For rectangular uncertainty sets (to which our formulation belongs), such

our work is the recent distributionally robust policy learning work already mentioned (Si et al., 2020b; Zhou et al., 2021; Yang et al., 2021; Kishan & Kalathil, 2021): (Si et al., 2020b) studies the special case of distributionally robust policy learning in contextual bandits (i.e. horizon is 1), while the other three study the infinite-horizon discounted RL setting.

1.2. Our Contributions

We design a distributionally robust Q-learning algorithm that has two features beyond the standard Q-learning algorithm. The first feature lies in the new values that the algorithm aims to learn: instead of Q-values, the algorithm now aims to estimate the distributionally robust Q-values. We achieve this by leveraging strong duality to transform the distributionally robust Q-values (an infinite-dimensional object since the distributional uncertainty set is infinite-dimensional) into a finite-dimensional quantity, also known as the distributionally robust Bellman operator. Second, we design a novel multi-level Monte-Carlo estimator to unbiasedly estimate the distributionally robust Bellman operator. Through a careful analysis of our estimator's bias and variance (see Theorem 3.7 and Theorem 3.8), we show that the distributionally robust Q-learning algorithm asymptotically converges to optimal distributionally robust policy (Theorem 3.10). As such, our results provide an initial affirmative answer to the open question raised above, and the convergence result provides a distributionally robust counterpart to the well-known asymptotic convergence of Q-learning (Jaakkola et al., 1994). Finally, we provide simulation results to demonstrate the robustness of the policy learned by the proposed distributionally-robust Q-learning algorithm.

1.3. Comparison with Existing Work

The distributionally robust Bellman equation was obtained before in (Iyengar, 2005). However, this is merely a tool for us to solve the problem in the dual space. Our main contribution is to design a distributionally robust Q-learning algorithm based on it. (Xu & Mannor, 2010) does not propose any new algorithm for learning a distributionally robust policy but only proved some properties of distributionally robust MDP. (Yang, 2020) considers the Wasserstein distance and uses a model-based algorithm that is totally different from ours. Our algorithm is the first model-free algorithm ever developed on this problem. Prior to our work, it is not clear at all whether Q-learning can be made robust, since all the existing algorithms in this area are model-based.

computation can be done efficiently in polynomial time, although depending on how such uncertainty sets are specified, some of the proposed algorithms require oracle access to solving an infinite-dimensional problem, and hence are infeasible.

2. Distributionally Robust Policy Learning with a Simulator

2.1. Standard Policy Learning

Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ be a tabular RL environment, where $\mathcal{S}$ and $\mathcal{A}$ are finite state space and action space respectively, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{P}(\mathbb{R}_{\geq 0})$ (the set of random variables that are supported on $\mathbb{R}_{\geq 0}$) is the randomized reward function, $\mathcal{P} = \{p_{s,a}(\cdot)\}_{(s,a)\in\mathcal{S}\times\mathcal{A}}$ is the transition model, and $\gamma \in (0,1)$ is the discount factor. We assume that the transition is Markovian, i.e., at each state $s \in \mathcal{S}$, if action $a \in \mathcal{A}$ is chosen, then the subsequent state is determined by the conditional distribution $p_{s,a}(\cdot) = p(\cdot|s,a)$. The decision maker will therefore receive a randomized reward $r(s,a)$. The value function $V^\pi(s)$ provides the expected cumulative discounted reward under the policy π with initial state $s \in \mathcal{S}$,

$$V^\pi(s) = \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r(s_t, a_t) \,\middle|\, s_1 = s\right].$$

Hence, the optimal value function is:

$$V^*(s) := \max_{\pi\in\Pi} V^\pi(s), \quad \forall s \in \mathcal{S},$$

where Π denotes the class of random policies. It is well known that the optimal value function is the solution of the following Bellman's equation:

$$V^*(s) = \max_{a\in\mathcal{A}} \left\{\mathbb{E}\left[r(s,a)\right] + \gamma \mathbb{E}_{p_{s,a}}\left[V^*(s')\right]\right\}, \forall s \in \mathcal{S}.$$

From here we define the optimal Q-function, Q^* as

$$\begin{aligned} Q^*(s,a) &= \mathbb{E}\left[r(s,a)\right] + \gamma \mathbb{E}_{p_{s,a}}\left[V^*(s')\right] \\ &= \mathbb{E}\left[r(s,a)\right] + \gamma \mathbb{E}_{p_{s,a}}\left[\max_{a'\in\mathcal{A}} Q^*(s',a')\right]. \end{aligned}$$

Throughout this paper, we impose the following assumption on the rewards.

Assumption 2.1. (Bounded rewards)
For any $(s,a) \in \mathcal{S} \times \mathcal{A}$, $r(s,a) \in [0, R_{\max}]$.

2.2. A Distributionally Robust Formulation

In reality, the transition model $\mathcal{P}$ and rewards $\mathcal{R}$ in $\mathcal{M}$ are subjected to change over time, this motivates us to learn a policy that is robust to certain perturbation in the environment. In particular, we consider the setting of distributionally robust RL, where both transition probabilities and rewards might be perturbed w.r.t. the Kullback-Leibler (KL) divergence $D_{\text{KL}}(P\|Q) = \int \log\left(\frac{dP}{dQ}\right) dP$ whenever $P \ll Q$ (P is absolutely continuous with respect to Q).

Remark 2.2. We pick KL divergence simply because it is one common divergence measure that is easy to analyze (given the already complicated nature of the problem). Our results can also be generalized to f-divergence.

In the original environment, let $\mathcal{P}^0 = \{p^0_{s,a}\}_{(s,a)\in\mathcal{S}\times\mathcal{A}}$ be the transition probabilities, ν^0 be the joint distribution of $\{r(s,a)\}_{(s,a)\in\mathcal{S}\times\mathcal{A}}$, where $r(s,a) \sim \nu^0_{s,a}$ (marginal distribution with respect to (s,a)). To construct the distributional uncertainty setting, for each $(s,a) \in \mathcal{S}\times\mathcal{A}$, we define robust KL balls that are centered at $p^0_{s,a}$ and $\nu^0_{s,a}$ by

$$\mathcal{P}_{s,a}(\delta) \coloneqq \left\{p_{s,a} \in \Delta_{|\mathcal{S}|} : D_{\mathrm{KL}}\left(p_{s,a}\|p^0_{s,a}\right) \le \delta\right\}$$

and

$$\mathcal{R}_{s,a}(\delta) \coloneqq \left\{r(s,a) \sim \nu_{s,a} : D_{\mathrm{KL}}(\nu_{s,a}\|\nu^0_{s,a}) \le \delta\right\}$$

respectively. Here $\delta > 0$ is the level of distributional robustness, and $\Delta_{|\mathcal{S}|}$ stands for the $|\mathcal{S}|-1$ dimensional probability simplex. Now we are able to build the uncertainty set $\mathcal{P}(\delta)$ by the Cartesian product of $\mathcal{P}_{s,a}(\delta)$ for each $(s,a) \in \mathcal{S}\times\mathcal{A}$. This type of uncertainty set is called $(s,a)-$rectangular set in standard literature (Wiesemann et al., 2013). Similarly we define $\mathcal{R}(\delta)$ by the Cartesian product of $\mathcal{R}_{s,a}(\delta)$ for each $(s,a) \in \mathcal{S}\times\mathcal{A}$. In the distributionally robust framework, the adversarial player is assumed to pick the worst-case transition model and rewards that minimize the expected cumulative discounted reward. To be clear, we define the distributionally robust value function as follows.

Definition 2.3. Given $\delta > 0$ and policy $\pi \in \Pi$, the distributionally robust value function $V^{\mathrm{rob},\pi}_\delta$ is defined as:

$$V^{\mathrm{rob},\pi}_\delta(s) \coloneqq \inf_{\mathbf{p}\in\mathcal{P}(\delta),\mathbf{r}\in\mathcal{R}(\delta)} \mathbb{E}_{\mathbf{p},\mathbf{r}}\left[\sum_{t=1}^{\infty}\gamma^{t-1}r(s_t,a_t)\middle| s_1 = s\right]. \tag{1}$$

Following the definition, $V^{\mathrm{rob},\pi}_\delta$ measures the quality of a policy π by computing its performance in the worst-case environment among the set of all possible environments that perturb the original transition $\mathcal{P}^0$ and reward distribution ν^0 under a δ-KL ball. The optimal distributionally robust value function is therefore defined by

$$V^{\mathrm{rob},*}_\delta(s) \coloneqq \max_{\pi\in\Pi} V^{\mathrm{rob},\pi}_\delta(s), \quad \forall s \in \mathcal{S}.$$

2.3. Strong Duality

Following the well known results in (Iyengar, 2005; Xu & Mannor, 2010), we can write down the distributionally robust dynamic programing for the distributionally robust value function $V^{\mathrm{rob},\pi}_\delta$ in Equation (1) as follows:

$$V^{\mathrm{rob},\pi}_\delta(s) = \inf_{\substack{p_{s,\pi(s)}\in\mathcal{P}_{s,\pi(s)}(\delta),\\ r\in\mathcal{R}_{s,\pi(s)}(\delta)}} \left\{\mathbb{E}[r(s,\pi(s))] + \gamma\mathbb{E}_{p_{s,\pi(s)}}\left[V^{\mathrm{rob},\pi}_\delta(s')\right]\right\}. \tag{2}$$

Moreover, we can write down the distributionally robust Bellman's equation for the optimal value function as follows:

$$V^{\mathrm{rob},*}_\delta(s) = \max_{a\in\mathcal{A}} \inf_{\substack{p_{s,a}\in\mathcal{P}_{s,a}(\delta),\\ r\in\mathcal{R}_{s,a}(\delta)}} \left\{\mathbb{E}\left[r(s,a)\right] + \gamma\mathbb{E}_{p_{s,a}}\left[V^{\mathrm{rob},*}_\delta(s')\right]\right\}. \tag{3}$$

Note that both Equation (2) and Equation (3) are in general computationally intractable since they involve infinite dimensional optimization. To address this issue, we introduce the following strong duality lemma from distributionally robust optimization under KL-perturbation.

Lemma 2.4 ((Hu & Hong, 2013), Theorem 1)**.** *Suppose $H(X)$ has finite moment generating function in the neighborhood of zero. Then for any $\delta > 0$,*

$$\sup_{P:D_{\mathrm{KL}}(P\|P_0)\le\delta} \mathbb{E}_P\left[H(X)\right] = \inf_{\alpha\ge 0}\left\{\alpha\log\left(\mathbb{E}_{P_0}\left[e^{H(X)/\alpha}\right]\right) + \alpha\delta\right\}.$$

By Lemma 2.4, we can transform the Equation (2) to the following equation.

$$\begin{aligned} V^{\mathrm{rob},\pi}_\delta(s) = &\sup_{\alpha\ge 0}\left\{-\alpha\log\left(\mathbb{E}_{\nu^0_{s,\pi(s)}}\left[e^{-r(s,\pi(s))/\alpha}\right]\right) - \alpha\delta\right\} + \\ &\gamma\sup_{\beta\ge 0}\left\{-\beta\log\left(\mathbb{E}_{p^0_{s,\pi(s)}}\left[e^{-V^{\mathrm{rob},\pi}_\delta(s')/\beta}\right]\right) - \beta\delta\right\}. \end{aligned} \tag{4}$$

As a direct consequence of the Equation (4) (note that the size of Π is finite in the tabular setting, see Theorem 3.2 in (Iyengar, 2005) for a standard proof), the optimal distributionally robust value function $V^{\mathrm{rob},*}_\delta$ in fact satisfies the following distributionally robust Bellman's equation.

$$\begin{aligned} V^{\mathrm{rob},*}_\delta(s) = &\max_{a\in\mathcal{A}}\left\{\sup_{\alpha\ge 0}\left\{-\alpha\log\left(\mathbb{E}_{\nu^0_{s,a}}\left[e^{-r(s,a)/\alpha}\right]\right) - \alpha\delta\right\} + \right. \\ &\left.\gamma\cdot\sup_{\beta\ge 0}\left\{-\beta\log\left(\mathbb{E}_{p^0_{s,a}}\left[e^{-V^{\mathrm{rob},*}_\delta(s')/\beta}\right]\right) - \beta\delta\right\}\right\}. \end{aligned} \tag{5}$$

3. Q-Learning in Distributionally Robust RL

3.1. Review of Q-Learning

The Q-learning algorithm determines the optimal Q-function using point samples. Let π be some random policy such that $\mathbb{P}(\pi(s) = a) > 0$ for all state-action pairs $(s,a) \in \mathcal{S}\times\mathcal{A}$. Suppose at time t, we draw a sample

(s_t, a_t, r_t, s'_t) from the environment according to the policy π. Then, Q-learning uses the following update rules

$$\begin{aligned}
&Q_{t+1}(s_t, a_t)\\
=&Q_t(s_t, a_t) + \alpha_t(s_t, a_t)\left[r_t + \gamma \max_{b\in\mathcal{A}} Q_t(s'_t, b) - Q_t(s_t, a_t)\right]\\
=&(1-\alpha_t(s_t, a_t))Q_t(s_t, a_t) + \alpha_t(s_t, a_t)\left[r_t + \gamma \max_{b\in\mathcal{A}} Q_t(s'_t, b)\right],
\end{aligned}$$

where the step-sizes α_t will be properly chosen. The key reason why Q-learning works is that both r_t and $\max_{b\in\mathcal{A}} Q_t(s'_t, b)$ are **unbiased** estimators of $\mathbb{E}[r(s_t, a_t)]$ and $\mathbb{E}_{p_{s_t,a_t}}[\max_{b\in\mathcal{A}} Q_t(s', b)]$. Therefore we can use the stochastic approximation theorem to show that Q_t converges to Q^* with careful choice of step-sizes. For more details, please read (Melo, 2001; Even-Dar et al., 2003).

3.2. Distributionally Robust Q-Learning

From Equation (3) and Equation (5), we know the optimal distributionally robust Q-function $Q_\delta^{\mathrm{rob},*}$ satisfies

$$\begin{aligned}
&Q_\delta^{\mathrm{rob},*}(s, a)\\
=&\inf_{\substack{p_{s,a}\in\mathcal{P}_{s,a}(\delta),\\ r\in\mathcal{R}_{s,a}(\delta)}} \left\{\mathbb{E}[r(s, a)] + \gamma\mathbb{E}_{p_{s,a}}\left[\max_{b\in\mathcal{A}} Q_\delta^{\mathrm{rob},*}(s', b)\right]\right\}\\
=&\sup_{\alpha\geq 0}\left\{-\alpha\log\left(\mathbb{E}_{\nu^0_{s,a}}\left[e^{-r(s,a)/\alpha}\right]\right) - \alpha\delta\right\} +\\
&\gamma\cdot\sup_{\beta\geq 0}\left\{-\beta\log\left(\mathbb{E}_{p^0_{s,a}}\left[e^{-\max_{b\in\mathcal{A}} Q_\delta^{\mathrm{rob},*}(s',b)/\beta}\right]\right) - \beta\delta\right\}.
\end{aligned}$$

By analogy with the Bellman Operator, we can define the δ-distributionally robust Bellman Operator as follows:

Definition 3.1. Given $\delta > 0$ and $Q \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$, the δ-distributionally robust Bellman Operator $\mathcal{T}_\delta^{\mathrm{rob}} : \mathbb{R}^{\mathcal{S}\times\mathcal{A}} \to \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ is defined as

$$\begin{aligned}
&\mathcal{T}_\delta^{\mathrm{rob}}(Q)(s, a) :=\\
&\sup_{\alpha\geq 0}\left\{-\alpha\log\left(\mathbb{E}_{\nu^0_{s,a}}\left[e^{-r(s,a)/\alpha}\right]\right) - \alpha\delta\right\} +\\
&\gamma\cdot\sup_{\beta\geq 0}\left\{-\beta\log\left(\mathbb{E}_{p^0_{s,a}}\left[e^{-\max_{b\in\mathcal{A}} Q(s',b)/\beta}\right]\right) - \beta\delta\right\}.
\end{aligned} \tag{6}$$

Remark 3.2. We define the δ-distributionally robust Bellman Operator by using its dual form. However, it may be more convenient to use the primal form

$$\begin{aligned}
&\mathcal{T}_\delta^{\mathrm{rob}}(Q)(s, a) =\\
&\inf_{\substack{p_{s,a}\in\mathcal{P}_{s,a}(\delta),\\ r\in\mathcal{R}_{s,a}(\delta)}} \left\{\mathbb{E}[r(s, a)] + \gamma\mathbb{E}_{p_{s,a}}\left[\max_{b\in\mathcal{A}} Q(s', b)\right]\right\}.
\end{aligned} \tag{7}$$

Our definition implies that $Q_\delta^{\mathrm{rob},*}$ is a fixed point of $\mathcal{T}_\delta^{\mathrm{rob}}$. Now suppose we have a simulator that allows us to sample (r, s') from $(\nu^0_{s,a}, p^0_{s,a})$, can we come up with a nice unbiased estimator of $\mathcal{T}_\delta^{\mathrm{rob}}(Q)(s, a)$? The plug-in estimator of $\mathcal{T}_\delta^{\mathrm{rob}}(Q)(s, a)$ is in fact **biased** (because of the non-linearity). For instance, if we just take one sample, say (s, a, r, s'). The corresponding plug-in estimator of $\mathcal{T}_\delta^{\mathrm{rob}}(Q)(s, a)$ is

$$\begin{aligned}
&\sup_{\alpha\geq 0}\left\{-\alpha\log\left(e^{-r/\alpha}\right) - \alpha\delta\right\} +\\
&\gamma\cdot\sup_{\beta\geq 0}\left\{-\beta\log\left(e^{-\max_{b\in\mathcal{A}} Q(s',b)/\beta}\right) - \beta\delta\right\}\\
=&r + \gamma\max_{b\in\mathcal{A}} Q(s', b),
\end{aligned}$$

which is the same as what we have in standard Q-learning. It is obviously not an unbiased estimator of $\mathcal{T}_\delta^{\mathrm{rob}}(Q)(s, a)$.

To address this issue, we propose a new estimator of $\mathcal{T}_\delta^{\mathrm{rob}}$ by introducing the multilevel Monte-Carlo method (Blanchet & Glynn, 2015; Blanchet et al., 2019). The formal description of our estimator is defined as follows:

Definition 3.3. Given $\delta > 0$, $\varepsilon \in (0, 0.5)$ and $Q \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$, the δ-distributionally robust estimator $\widehat{\mathcal{T}}_{\delta,\varepsilon}^{\mathrm{rob}}$ is defined as:

$$\widehat{\mathcal{T}}_{\delta,\varepsilon}^{\mathrm{rob}}(Q)(s, a) := \widehat{R}_\delta^{\mathrm{rob}}(s, a) + \gamma\widehat{T}_\delta^{\mathrm{rob}}(Q)(s, a). \tag{8}$$

For $\widehat{R}_\delta^{\mathrm{rob}}(s, a)$ and $\widehat{T}_\delta^{\mathrm{rob}}(s, a)$, we firstly sample $N \in \mathbb{N}$ from the distribution $\mathbb{P}(N = n) = p_n = \varepsilon(1-\varepsilon)^n$, then use the simulator to draw 2^{N+1} samples (r_i, s'_i) from $(\nu^0_{s,a}, p^0_{s,a})$. Finally we define

$$\widehat{R}_\delta^{\mathrm{rob}}(s, a) := r_1 + \frac{\Delta^r_{N,\delta}}{p_N}, . \tag{9}$$

$$\widehat{T}_\delta^{\mathrm{rob}}(Q)(s, a) := \max_{b\in\mathcal{A}} Q(s'_1, b) + \frac{\Delta^q_{N,\delta}(Q)}{p_N}. \tag{10}$$

where

$$\begin{aligned}
&\Delta^r_{N,\delta} :=\\
&\sup_{\alpha\geq 0}\left\{-\alpha\log\left(\frac{1}{2^{N+1}}\sum_{i=1}^{2^{N+1}} e^{-r_i/\alpha}\right) - \alpha\delta\right\} -\\
&\frac{1}{2}\sup_{\alpha\geq 0}\left\{-\alpha\log\left(\frac{1}{2^N}\sum_{i=1}^{2^N} e^{-r_{2i}/\alpha}\right) - \alpha\delta\right\} -\\
&\frac{1}{2}\sup_{\alpha\geq 0}\left\{-\alpha\log\left(\frac{1}{2^N}\sum_{i=1}^{2^N} e^{-r_{2i-1}/\alpha}\right) - \alpha\delta\right\}
\end{aligned} \tag{11}$$

and

$$\Delta^q_{N,\delta}(Q) :=$$
$$\sup_{\beta\geq 0}\left\{-\beta\log\left(\frac{1}{2^{N+1}}\sum_{i=1}^{2^{N+1}} e^{-\max_{b\in\mathcal{A}} Q(s_i',b)/\beta}\right)-\beta\delta\right\}-$$
$$\frac{1}{2}\sup_{\beta\geq 0}\left\{-\beta\log\left(\frac{1}{2^{N}}\sum_{i=1}^{2^{N}} e^{-\max_{b\in\mathcal{A}} Q(s_{2i}',b)/\beta}\right)-\beta\delta\right\}-$$
$$\frac{1}{2}\sup_{\beta\geq 0}\left\{-\beta\log\left(\frac{1}{2^{N}}\sum_{i=1}^{2^{N}} e^{-\max_{b\in\mathcal{A}} Q(s_{2i-1}',b)/\beta}\right)-\beta\delta\right\}. \tag{12}$$

Using the estimator $\widehat{\mathcal{T}}^{\mathrm{rob}}_{\delta,\varepsilon}$, our Distributionally Robust Q-Learning algorithm is summarized in Algorithm 1.

Algorithm 1 Distributionally Robust Q-Learning

Input: Uncertainty radius $\delta>0$, parameter $\varepsilon\in(0,0.5)$.
Initialization: $\widehat{Q}^{\mathrm{rob}}_{\delta,t}(s,a)\equiv 0,\ \forall(s,a)\in\mathcal{S}\times\mathcal{A}, t=1$.
repeat
  for every $(s,a)\in\mathcal{S}\times\mathcal{A}$ **do**
    Sample $N\in\mathbb{N}$ from $\mathbb{P}(N=n)=p_n=\varepsilon(1-\varepsilon)^n$.
    Draw 2^{N+1} samples (r_i, s_i') from the simulator.
    Compute $\Delta^r_{N,\delta}$ and $\Delta^q_{N,\delta}(\widehat{Q}^{\mathrm{rob}}_{\delta,t})$ through Equation (11) and Equation (12) respectively.

$$\widehat{R}^{\mathrm{rob}}_{\delta}(s,a)\leftarrow r_1+\frac{\Delta^r_{N,\delta}}{p_N}$$
$$\widehat{T}^{\mathrm{rob}}_{\delta}(\widehat{Q}^{\mathrm{rob}}_{\delta,t})(s,a)\leftarrow \max_{b\in\mathcal{A}}\widehat{Q}^{\mathrm{rob}}_{\delta,t}(s_1',b)+\frac{\Delta^q_{N,\delta}(\widehat{Q}^{\mathrm{rob}}_{\delta,t})}{p_N}$$
$$\widehat{\mathcal{T}}^{\mathrm{rob}}_{\delta,\varepsilon}(\widehat{Q}^{\mathrm{rob}}_{\delta,t})(s,a)\leftarrow \widehat{R}^{\mathrm{rob}}_{\delta}(s,a)+\gamma\widehat{T}^{\mathrm{rob}}_{\delta}(\widehat{Q}^{\mathrm{rob}}_{\delta,t})(s,a)$$
$$\widehat{Q}^{\mathrm{rob}}_{\delta,t+1}(s,a)\leftarrow (1-\alpha_t)\widehat{Q}^{\mathrm{rob}}_{\delta,t}(s,a)+\alpha_t\widehat{\mathcal{T}}^{\mathrm{rob}}_{\delta,\varepsilon}(\widehat{Q}^{\mathrm{rob}}_{\delta,t})(s,a)$$

  end for
  $t\leftarrow t+1$.
until $\widehat{Q}^{\mathrm{rob}}_{\delta,t}$ converges

Remark 3.4. In Algorithm 1, we can choose any α_t which satisfies the Robbins–Monro Condition, i.e., $\sum_{i=1}^{\infty}\alpha_t=\infty$, $\sum_{i=1}^{\infty}\alpha_t^2<\infty$.

Remark 3.5. δ is an exogenous variable that quantifies the level of conservatism, which we consider as pre-determined (and not part of the algorithm). That said, it can also be from past datasets if they are collected from different environments, in which case the shift can be estimated.

3.3. Theoretical Guarantee

First of all, we introduce Lemma 3.6 which plays a critical role in the proof of convergence of Algorithm 1.

Lemma 3.6. *Given $\delta>0$, $0<\gamma<1$, $\mathcal{T}^{\mathrm{rob}}_{\delta}$ is a γ-contraction map w.r.t. the infinity norm.*

Proof. Given any $Q_1, Q_2\in\mathbb{R}^{\mathcal{S}\times\mathcal{A}}$, let $V_i(s')=\max_{b\in\mathcal{A}}Q_i(s',b)$ for $i\in\{1,2\}$. Fix a pair $(s,a)\in\mathcal{S}\times\mathcal{A}$. By applying the primal form of $\mathcal{T}^{\mathrm{rob}}_{\delta}$ (Equation (7)), we have

$$\mathcal{T}^{\mathrm{rob}}_{\delta}(Q_1)(s,a)-\mathcal{T}^{\mathrm{rob}}_{\delta}(Q_2)(s,a)=$$
$$\gamma\left(\inf_{p_{s,a}\in\mathcal{P}_{s,a}(\delta)}\mathbb{E}_{p_{s,a}}[V_1(s')]-\inf_{p_{s,a}\in\mathcal{P}_{s,a}(\delta)}\mathbb{E}_{p_{s,a}}[V_2(s')]\right).$$

Consider any $\epsilon>0$, suppose $p(\epsilon)\in\mathcal{P}_{s,a}(\delta)$ makes $\mathbb{E}_{p(\epsilon)}[V_2(s')]\leq\inf_{p_{s,a}\in\mathcal{P}_{s,a}(\delta)}\mathbb{E}_{p_{s,a}}[V_2(s')]+\epsilon$. Then we know

$$\inf_{p_{s,a}\in\mathcal{P}_{s,a}(\delta)}\mathbb{E}_{p_{s,a}}[V_1(s')]-\inf_{p_{s,a}\in\mathcal{P}_{s,a}(\delta)}\mathbb{E}_{p_{s,a}}[V_2(s')]$$
$$\leq\mathbb{E}_{p(\epsilon)}[V_1(s')-V_2(s')]+\epsilon\leq\max_{s'\in\mathcal{S}}|V_1(s')-V_2(s')|+\epsilon$$
$$\leq\max_{(s',b)\in\mathcal{S}\times\mathcal{A}}|Q_1(s',b)-Q_2(s',b)|+\epsilon=\|Q_1-Q_2\|_\infty+\epsilon.$$

Combining the previous part, we have $\forall\epsilon>0$,

$$\mathcal{T}^{\mathrm{rob}}_{\delta}(Q_1)(s,a)-\mathcal{T}^{\mathrm{rob}}_{\delta}(Q_2)(s,a)\leq\gamma\|Q_1-Q_2\|_\infty+\gamma\epsilon,$$

which implies

$$\mathcal{T}^{\mathrm{rob}}_{\delta}(Q_1)(s,a)-\mathcal{T}^{\mathrm{rob}}_{\delta}(Q_2)(s,a)\leq\gamma\|Q_1-Q_2\|_\infty.$$

By a similar argument, we also have

$$\mathcal{T}^{\mathrm{rob}}_{\delta}(Q_2)(s,a)-\mathcal{T}^{\mathrm{rob}}_{\delta}(Q_1)(s,a)\leq\gamma\|Q_1-Q_2\|_\infty.$$

Hence, there is

$$|\mathcal{T}^{\mathrm{rob}}_{\delta}(Q_1)(s,a)-\mathcal{T}^{\mathrm{rob}}_{\delta}(Q_2)(s,a)|\leq\gamma\|Q_1-Q_2\|_\infty.$$

Note that the above result is true for any $(s,a)\in\mathcal{S}\times\mathcal{A}$, which implies

$$\|\mathcal{T}^{\mathrm{rob}}_{\delta}(Q_1)-\mathcal{T}^{\mathrm{rob}}_{\delta}(Q_2)\|_\infty\leq\gamma\|Q_1-Q_2\|_\infty.$$

□

Next, we give the key theorem of our estimator $\widehat{\mathcal{T}}^{\mathrm{rob}}_{\delta,\varepsilon}$.

Theorem 3.7. *Given $\delta>0$. If Assumption 2.1 holds, then for any $\varepsilon\in(0,0.5)$, $\widehat{\mathcal{T}}^{\mathrm{rob}}_{\delta,\varepsilon}$ is an unbiased estimator of $\mathcal{T}^{\mathrm{rob}}_{\delta}$, i.e., $\forall Q\in\mathbb{R}^{\mathcal{S}\times\mathcal{A}}, (s,a)\in\mathcal{S}\times\mathcal{A}$,*

$$\mathbb{E}\left[\widehat{\mathcal{T}}^{\mathrm{rob}}_{\delta,\varepsilon}(Q)(s,a)\right]=\mathcal{T}^{\mathrm{rob}}_{\delta}(Q)(s,a).$$

Due to the page limitation, we defer the whole proof of Theorem 3.7 into Appendix B and give a proof outline here. By the law of total probability, we first can get

$$\begin{aligned}&\mathbb{E}[\widehat{R}^{\mathrm{rob}}_{\delta}(s,a)]\\ &=\lim_{n\to\infty}\sup_{\alpha\geq 0}\left\{-\alpha\log\left(\frac{1}{2^n}\sum_{i=1}^{2^n}e^{-r_i/\alpha}\right)-\alpha\delta\right\},\end{aligned}$$

a similar result will hold for $\mathbb{E}[\widehat{T}^{\mathrm{rob}}_{\delta}(Q)(s,a)]$. One key technical point helping us to move on is to understand

$$\alpha^*=\operatorname*{argmax}_{\alpha\geq 0}\left\{-\alpha\log\left(\mathbb{E}_{\nu^0_{s,a}}\left[e^{-r(s,a)/\alpha}\right]\right)\right\},$$

and β^* defined in a similar way. By employing different events defined in Appendix B, we finally can derive

$$\begin{aligned}&\lim_{n\to\infty}\sup_{\alpha\geq 0}\left\{-\alpha\log\left(\frac{1}{2^n}\sum_{i=1}^{2^n}e^{-r_i/\alpha}\right)-\alpha\delta\right\}\\ =&\sup_{\alpha\geq 0}\left\{-\alpha\log\left(\mathbb{E}_{\nu^0_{s,a}}\left[e^{-r(s,a)/\alpha}\right]\right)\right\},\end{aligned}$$

which completes the unbiasedness of $\widehat{R}^{\mathrm{rob}}_{\delta}(s,a)$. Similar result will give us the unbiasedness of $\widehat{T}^{\mathrm{rob}}_{\delta}(Q)(s,a)$.

Besides, we introduce another key property of $\widehat{\mathcal{T}}^{\mathrm{rob}}_{\delta,\varepsilon}$.

Theorem 3.8. *Given $\delta>0$, $\varepsilon\in(0,0.5)$. If Assumption 2.1 holds, then there exists a constant $C(\delta,\varepsilon,\nu^0,\mathcal{P}^0)>0$ such that for any $Q\in\mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ and $(s,a)\in\mathcal{S}\times\mathcal{A}$,*

$$\mathsf{Var}\left[\widehat{\mathcal{T}}^{\mathrm{rob}}_{\delta,\varepsilon}(Q)(s,a)\right]\leq C(\delta,\varepsilon,\nu^0,\mathcal{P}^0)(1+\|Q\|^2_{\infty}).$$

The proof of Theorem 3.8 is also deferred to Appendix C.

Remark 3.9. For Theorem 3.8, our setting is different from (Blanchet & Glynn, 2015; Blanchet et al., 2019), which requires a careful analysis and a different proof strategy. One key difference is that expectation terms in Equation (6) are inside the log function. However, in (Blanchet & Glynn, 2015; Blanchet et al., 2019), expectations are always at the most outside. The non-linearity of the log function makes the analysis more difficult. Another technical point is we need to understand the optimizers of $\mathcal{T}^{\mathrm{rob}}_{\delta}$, i.e.,

$$\begin{aligned}\alpha^*&=\operatorname*{argmax}_{\alpha\geq 0}\left\{-\alpha\log\left(\mathbb{E}\left[e^{-r(s,a)/\alpha}\right]\right)-\alpha\delta\right\},\\ \beta^*&=\operatorname*{argmax}_{\beta\geq 0}\left\{-\beta\log\left(\mathbb{E}_{p^0_{s,a}}\left[e^{-\max_{b\in\mathcal{A}}Q(s',b)/\beta}\right]\right)-\beta\delta\right\},\end{aligned}$$

where proof techniques are different for $\alpha^*,\beta^*=0$ and $\alpha^*,\beta^*\neq 0$.

Finally, combing previous results, we are able to establish the following convergence guarantee for Algorithm 1.

Theorem 3.10. *Given $\delta>0$. If Assumption 2.1 holds, then for any $\varepsilon\in(0,0.5)$, $\widehat{Q}^{\mathrm{rob}}_{\delta,t}$ in Algorithm 1 will converge to $Q^{\mathrm{rob},*}_{\delta}$ as $t\to\infty$.*

Proof. Following the proof in (Melo, 2001; Even-Dar et al., 2003), we can rewrite the update rule of Algorithm 1 as

$$\begin{aligned}&D(\widehat{Q}^{\mathrm{rob}}_{\delta,t+1})(s,a)=\\ &(1-\alpha_t)D(\widehat{Q}^{\mathrm{rob}}_{\delta,t})(s,a)+\alpha_t(\widehat{\mathcal{T}}^{\mathrm{rob}}_{\delta,\varepsilon}(\widehat{Q}^{\mathrm{rob}}_{\delta,t})-Q^{\mathrm{rob},*}_{\delta})(s,a),\end{aligned}$$

where $D(Q):=Q-Q^{\mathrm{rob},*}_{\delta}$. Combining Lemma 3.6, Theorem 3.7 and the fact that $\mathcal{T}^{\mathrm{rob}}_{\delta}(Q^{\mathrm{rob},*}_{\delta})=Q^{\mathrm{rob},*}_{\delta}$, we have

$$\begin{aligned}&\left|\mathbb{E}\left[(\widehat{\mathcal{T}}^{\mathrm{rob}}_{\delta,\varepsilon}(\widehat{Q}^{\mathrm{rob}}_{\delta,t})-Q^{\mathrm{rob},*}_{\delta})(s,a)\right]\right|\\ =&\left|(\mathcal{T}^{\mathrm{rob}}_{\delta}(\widehat{Q}^{\mathrm{rob}}_{\delta,t})-Q^{\mathrm{rob},*}_{\delta})(s,a)\right|\\ =&\left|(\mathcal{T}^{\mathrm{rob}}_{\delta}(\widehat{Q}^{\mathrm{rob}}_{\delta,t})-\mathcal{T}^{\mathrm{rob}}_{\delta}(Q^{\mathrm{rob},*}_{\delta}))(s,a)\right|\\ \leq&\gamma\|D(\widehat{Q}^{\mathrm{rob}}_{\delta,t})\|_{\infty}.\end{aligned}$$

Next, by Theorem 3.8, we have

$$\begin{aligned}&\mathsf{Var}\left[(\widehat{\mathcal{T}}^{\mathrm{rob}}_{\delta,\varepsilon}(\widehat{Q}^{\mathrm{rob}}_{\delta,t})-Q^{\mathrm{rob},*}_{\delta})(s,a)\right]\\ =&\mathsf{Var}\left[\widehat{\mathcal{T}}^{\mathrm{rob}}_{\delta,\varepsilon}(\widehat{Q}^{\mathrm{rob}}_{\delta,t})(s,a)\right]\\ \leq&C(\delta,\varepsilon,\nu^0,\mathcal{P}^0)(1+\|\widehat{Q}^{\mathrm{rob}}_{\delta,t}\|^2_{\infty})\\ =&C(\delta,\varepsilon,\nu^0,\mathcal{P}^0)(1+\|D(\widehat{Q}^{\mathrm{rob}}_{\delta,t})+Q^{\mathrm{rob},*}_{\delta}\|^2_{\infty})\\ \leq&2(1+\|Q^{\mathrm{rob},*}_{\delta}\|^2_{\infty})C(\delta,\varepsilon,\nu^0,\mathcal{P}^0)(1+\|D(\widehat{Q}^{\mathrm{rob}}_{\delta,t})\|^2_{\infty}).\end{aligned}$$

Also note that α_t satisfies the Robbins–Monro Condition, these intermediate results then yield the convergence of Algorithm 1 by Theorem 1 of (Jaakkola et al., 1994). □

4. Numerical Experiments

We use a supply chain model to test Algorithm 1 in our numerical experiments. In the supply chain model, the state space $\mathcal{S}=[n]:=\{0,1,\cdots,n-1,n\}$ which represents the possible number of goods we have. The action space $\mathcal{A}=[n]$ represents the number of goods we can order. In every day $t\in\mathbb{N}_+$, the number of goods demanded by the market is an unobserved random variable $d_t\sim\mathrm{Uni}[n]$. We assume $\{d_t\}_{t\in\mathbb{N}_+}$ are multiple independent. At the start of every day t, suppose the number of goods is s_t, then we can determine the number of goods we want to order, i.e., our action $a_t\in[n-s_t]$. Besides, we assume there is no delay for us to receive our order. So the cost c_t at day t is

$$c_t=k\mathbb{1}[a_t>0]+h(s_t+a_t-d_t)^+ +p(d_t-s_t-a_t)^+,$$

h and p are pre-specified constants denoting holding cost for every single good and per unit lost sales penalty, respectively.

Table 1. Distributionally robust policy for different ε

ε	$\widehat{\pi}^{\text{rob}}_{\delta,\varepsilon}(0)$	$\widehat{\pi}^{\text{rob}}_{\delta,\varepsilon}(1)$	$\widehat{\pi}^{\text{rob}}_{\delta,\varepsilon}(2)$	$\widehat{\pi}^{\text{rob}}_{\delta,\varepsilon}(3)$	$\widehat{\pi}^{\text{rob}}_{\delta,\varepsilon}(4)$	$\widehat{\pi}^{\text{rob}}_{\delta,\varepsilon}(s \geq 5)$
0.49	7	6	5	4	3	0
0.499	7	6	5	4	3	0
0.5	7	6	5	4	3	0
0.6	7	6	5	4	3	0

We can decompose c_t as two parts $h(s_t + a_t - d_t)^+$ and $p(d_t - s_t - a_t)^+$. A holding cost of $h(s_t + a_t - d_t)^+$ will appear on remaining goods and a lost sales penalty of $p(d_t - s_t - a_t)^+$ is incurred if the number of demand could not be served due to insufficient goods. k is a fixed ordering cost. In our experiments, due to the limit computation resources, we fix $n = 10$. Besides, we take $h = 1$, $p = 2$, $k = 3$ and set the discount factor $\gamma = 0.9$ with starting from $s_1 = 0$.

By using standard Q-learning, one can find the non distributionally robust, optimal policy π^* for the non perturbed problem, i.e., $\delta = 0$, satisfying

$$\pi^*(s) = \begin{cases} 8 - s & 0 \leq s \leq 2 \\ 0 & 3 \leq s \leq 10 \end{cases}.$$

For the distributionally robust setting, we can think the probability distribution of d_t is no longer a uniform distribution anymore, which is quite reasonable in reality.

In the simulation, we set $\delta = 1$ as the perturbation parameter. At the k-th step of Algorithm 1, we set the learning rate α_k be $\frac{1}{1+(1-\gamma)(k-1)}$ to satisfy the Robbins–Monro Condition. We will treat Algorithm 1 as having converged when the infinity norm of the difference between the updated value and the old value is no greater than $tolerance = 0.05$.

For the parameter ε used in our estimator, we consider $\varepsilon \in \{0.49, 0.499.0.5, 0.6\}$. Note that our Theorem 3.10 only ensures the convergence for $\varepsilon \in (0, 0.5)$, but we will test some values out of this range. Besides, the reason why we only choose $\{0.49, 0.499\}$ rather than adding some other smaller $\varepsilon \in (0.0.5)$ is that from the construction of our estimator, we can find smaller ε will lead to a huge number of samples with high probability, to avoid this problem, we only test ε close to 0.5. To reduce the effect of variance, for every ε, we will run 5 times and use the averaged value as the final output $\widehat{Q}^{\text{rob}}_{\delta,\varepsilon}$. Finally, we use $\widehat{Q}^{\text{rob}}_{\delta,\varepsilon}$ to find the greedy policy

$$\widehat{\pi}^{\text{rob}}_{\delta,\varepsilon}(s) = \underset{a \in [n-s]}{\operatorname{argmax}} \widehat{Q}^{\text{rob}}_{\delta,\varepsilon}(s,a), \quad \forall s \in \mathcal{S}$$

for every ε. The output greedy policy $\widehat{\pi}^{\text{rob}}_{\delta,\varepsilon}$ for every ε is listed in Table 1. We can see all policies are the same as

$$\widehat{\pi}^{\text{rob}}_{\delta}(s) = \begin{cases} 7 - s & 0 \leq s \leq 4 \\ 0 & 5 \leq s \leq 10 \end{cases}.$$

In Table 2, we list the averaged number of iterations and averaged number of samples used for every ε. Combing our results in Tables 1 and 2, it shows that Algorithm 1 may converge with even less samples when $\varepsilon \notin (0, 0.5)$.

Table 2. Averaged number of iterations and samples used for different ε.

ε	#ITERATIONS	#SAMPLES
0.49	1869.6	2959020.8
0.499	1815.4	2398951.2
0.5	1864.6	2478958.8
0.6	1864.8	820703.6

Now we define a perturbed uniform distribution with parameter m and b as follows:

$$\text{Uni}_{m,b}([n])(x) = \begin{cases} \frac{b+1}{n+1} & x \in \{m, m+1\} \\ \frac{n-1-2b}{n^2-1} & x \notin \{m, m+1\} \end{cases}.$$

With the perturbed distribution, we test our distributionally robuast policy $\widehat{\pi}^{\text{rob}}_{\delta}$ and non distributionally robuast policy π^* for $b \in \{1, 1.5, 2, 2.5\}$ (Note that b can not be too large, otherwise there will exist some pair (s,a) such that $D_{\text{KL}}(p_{s,a} \| p^0_{s,a}) > \delta$ or $D_{\text{KL}}(\nu_{s,a} \| \nu^0_{s,a}) > \delta$) and every $m \in [n-1]$. We report the total cost averaged over 2000 runs for different b in Figures 1 to 4.

With varying test probabilities, our distributionally robust policy performs better in worst cases when the probability distribution of demand is centered in $\{5, 6, 7\}$ instead of the uniform distribution, demonstrating again the effectiveness of our proposed distributionally robust formulations.

5. Conclusion

In this paper, we proposed a novel unbiased estimator of the distributionally robust Bellman Operator. By using the

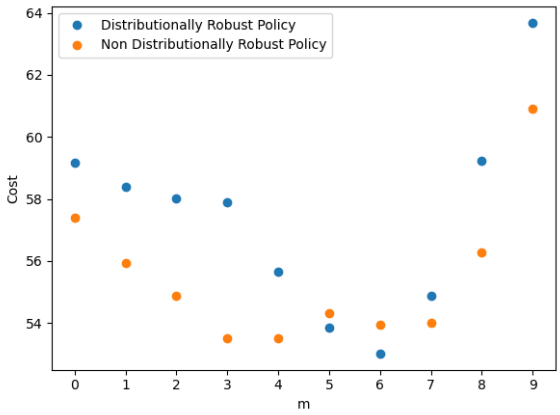

Figure 1. Total Cost for $b = 1$

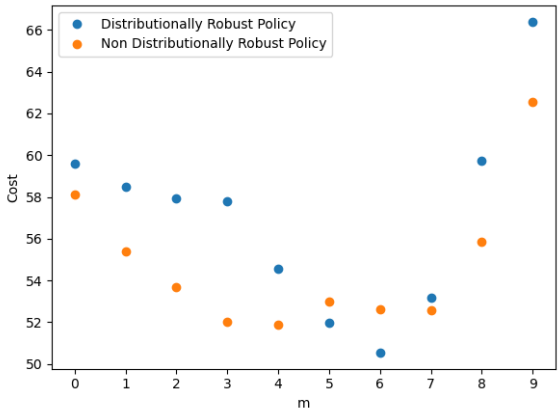

Figure 2. Total Cost for $b = 1.5$

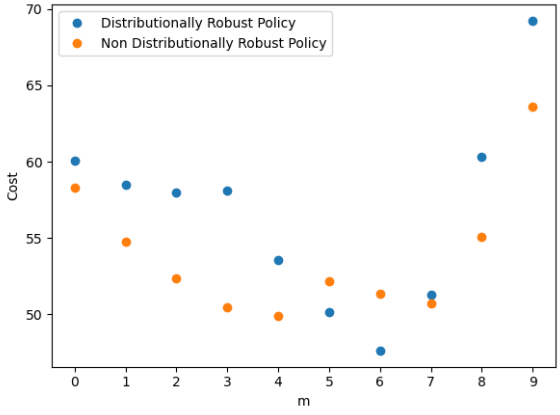

Figure 3. Total Cost for $b = 2$

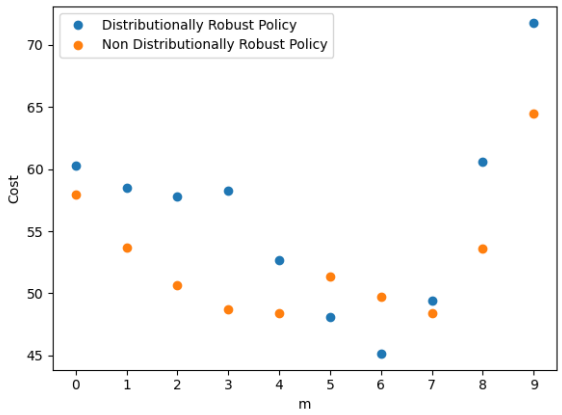

Figure 4. Total Cost for $b = 2.5$

estimator, we proposed a novel Q-learning algorithm, distributionally robust Q-learning, that is able to learn a robust Q value function under the KL-divergence perturbation of transition probabilities and rewards. We established the asymptotic convergence guarantee of the proposed distributionally robust Q-learning algorithm.

Several open problems suggest itself. First, although asymptotic convergence is desirable, it would also be interesting to obtain finite-sample guarantees for distributionally robust Q-learning algorithm. We believe such a result would require a completely different line of analysis, whose scope goes significantly beyond this paper. Second, this paper focused on the infinite-horizon discounted RL setting. A much more challenging but also highly useful setting to consider is the (infinite horizon) average reward RL (Dong et al., 2021). RL in this setting is less explored, and distributionally robust policy learning in this setting poses significant technical challenges. Finally, another important direction of research is to generalize the results to high-dimensional state settings (Ren & Zhou, 2020), where the intrinsic dimension is low. Data efficiency in such settings will be of particular importance. We look forward to these problems being addressed by the emerging distributionally robust reinforcement learning community.

Acknowledgements

This work is generously supported by the Horizon Robotics faculty award. This work is additionally supported by the Air Force Office of Scientific Research under award number FA9550-20-1-0397 and NSF grants 1915967 and 2118199. Zijian Liu would like to thank Junfu Yao for discussions.

References

Abadeh, S. S., Nguyen, V. A., Kuhn, D., and Esfahani, P. M. Wasserstein distributionally robust kalman filtering. In *Advances in Neural Information Processing Systems*, pp. 8483–8492, 2018.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Bayraksan, G. and Love, D. K. Data-driven stochastic programming using phi-divergences. In *The Operations Research Revolution*, pp. 1–19. Catonsville: Institute for Operations Research and the Management Sciences, 2015.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

Bertsimas, D. and Sim, M. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

Blanchet, J. and Murthy, K. Quantifying distributional model risk via optimal transport. *Mathematics of Operations Research*, 44(2):565–600, 2019. doi: 10.1287/moor.2018.0936.

Blanchet, J. H. and Glynn, P. W. Unbiased monte carlo for optimization and functions of expectations via multi-level randomization. In *2015 Winter Simulation Conference (WSC)*, pp. 3656–3667. IEEE, 2015.

Blanchet, J. H., Glynn, P. W., and Pei, Y. Unbiased multilevel monte carlo: Stochastic optimization, steady-state simulation, quantiles, and other applications. *arXiv preprint arXiv:1904.09929*, 2019.

Chen, Z., Kuhn, D., and Wiesemann, W. Data-driven chance constrained programs over wasserstein balls. *arXiv preprint arXiv:1809.00210*, 2018.

Choi, J. J., Laibson, D., Madrian, B. C., and Metrick, A. Reinforcement learning and savings behavior. *The Journal of finance*, 64(6):2515–2534, 2009.

Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865, 2016.

Delage, E. and Ye, Y. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations research*, 58(3):595–612, 2010.

Deng, Y., Bao, F., Kong, Y., Ren, Z., and Dai, Q. Deep direct reinforcement learning for financial signal representation and trading. *IEEE Transactions on Neural Networks and Learning Systems*, 28(3):653–664, 2017.

Dong, S., Van Roy, B., and Zhou, Z. Simple agent, complex environment: Efficient reinforcement learning with agent states. *arXiv preprint arXiv:2102.05261*, 2021.

Drew, K. California robot teaching itself to walk like a human toddler. *NBC News*, Dec 2015.

Duchi, J. and Namkoong, H. Learning models with uniform performance via distributionally robust optimization. *arXiv preprint arXiv:1810.08750*, 2018.

Duchi, J. and Namkoong, H. Variance-based regularization with convex objectives. *The Journal of Machine Learning Research*, 20(1):2450–2504, 2019.

Duchi, J., Glynn, P., and Namkoong, H. Statistics of robust optimization: A generalized empirical likelihood approach. *arXiv preprint arXiv:1610.03425*, 2016.

Duchi, J., Hashimoto, T., and Namkoong, H. Distributionally robust losses against mixture covariate shifts. *arXiv preprint arXiv:2007.13982*, 2019.

El Ghaoui, L. and Nilim, A. Robust solutions to markov decision problems with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

Even-Dar, E., Mansour, Y., and Bartlett, P. Learning rates for q-learning. *Journal of machine learning Research*, 5 (1), 2003.

Fournier, N. and Guillin, A. On the rate of convergence in wasserstein distance of the empirical measure. *Probability Theory and Related Fields*, 162(3-4):707–738, 2015.

François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., and Pineau, J. An introduction to deep reinforcement learning. *arXiv preprint arXiv:1811.12560*, 2018.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

Gao, R. and Kleywegt, A. J. Distributionally robust stochastic optimization with Wasserstein distance. *arXiv preprint arXiv:1604.02199*, 2016.

Gao, R., Chen, X., and Kleywegt, A. J. Wasserstein distributional robustness and regularization in statistical learning. *arXiv e-prints*, pp. arXiv–1712, 2017.

Gao, R., Xie, L., Xie, Y., and Xu, H. Robust hypothesis testing using wasserstein uncertainty sets. In *Advances in Neural Information Processing Systems*, pp. 7902–7912, 2018.

Ghosh, S. and Lam, H. Robust analysis in stochastic simulation: Computation and performance guarantees. *Operations Research*, 2019.

González-Trejo, J., Hernández-Lerma, O., and Hoyos-Reyes, L. F. Minimax control of discrete-time stochastic systems. *SIAM Journal on Control and Optimization*, 41 (5):1626–1659, 2002.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Gu, S., Holly, E., Lillicrap, T., and Levine, S. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 3389–3396. IEEE, 2017.

Hu, Z. and Hong, L. J. Kullback-leibler divergence constrained distributionally robust optimization. *Available at Optimization Online*, 2013.

Huang, C., Lucey, S., and Ramanan, D. Learning policies for adaptive tracking with deep feature cascades. *ICCV*, pp. 105–114, 2017.

Iyengar, G. N. Robust dynamic programming. *Mathematics of Operations Research*, 30(2):257–280, 2005.

Jaakkola, T., Jordan, M. I., and Singh, S. P. On the convergence of stochastic iterative dynamic programming algorithms. *Neural computation*, 6(6):1185–1201, 1994.

Kishan, P. and Kalathil, D. Sample complexity of robust reinforcement learning with a generative model. *arXiv preprint arXiv:2112.01506*, 2021.

Kober, J., Bagnell, J. A., and Peters, J. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

Lam, H. Recovering best statistical guarantees via the empirical divergence-based distributionally robust optimization. *Operations Research*, 67(4):1090–1105, 2019.

Lam, H. and Zhou, E. The empirical likelihood approach to quantifying uncertainty in sample average approximation. *Operations Research Letters*, 45(4):301–307, 2017.

Lee, J. and Raginsky, M. Minimax statistical learning with wasserstein distances. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 2692–2701, USA, 2018. Curran Associates Inc.

Li, Y., Szepesvari, C., and Schuurmans, D. Learning exercise policies for american options. In *Artificial Intelligence and Statistics*, pp. 352–359, 2009.

Maitin-Shepard, J., Cusumano-Towner, M., Lei, J., and Abbeel, P. Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *2010 IEEE International Conference on Robotics and Automation*, pp. 2308–2315, 2010.

Mannor, S., Mebel, O., and Xu, H. Robust mdps with k-rectangular uncertainty. *Mathematics of Operations Research*, 41(4):1484–1509, 2016.

Melo, F. S. Convergence of q-learning: A simple proof. *Institute Of Systems and Robotics, Tech. Rep*, pp. 1–4, 2001.

Mohajerin Esfahani, P. and Kuhn, D. Data-driven distributionally robust optimization using the Wasserstein metric: performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1):115–166, Sep 2018. ISSN 1436-4646. doi: 10.1007/s10107-017-1172-1.

Morimoto, J. and Doya, K. Robust reinforcement learning. *Neural computation*, 17(2):335–359, 2005.

Namkoong, H. and Duchi, J. C. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 2216–2224. Red Hook: Curran Associates Inc., 2016.

Nguyen, V. A., Kuhn, D., and Esfahani, P. M. Distributionally robust inverse covariance estimation: The Wasserstein shrinkage estimator. *arXiv preprint arXiv:1805.07194*, 2018.

Ren, Z. and Zhou, Z. Dynamic batch learning in high-dimensional sparse linear contextual bandits. *arXiv preprint arXiv:2008.11918*, 2020.

Sadeghi, F. and Levine, S. Cad2rl: Real single-image flight without a single real image. *arXiv preprint arXiv:1611.04201*, 2016.

Saerens, M., Latinne, P., and Decaestecker, C. Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural computation*, 14(1):21–41, 2002.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Schulman, J., Ho, J., Lee, A. X., Awwal, I., Bradlow, H., and Abbeel, P. Finding locally optimal, collision-free trajectories with sequential convex optimization. In *Robotics: science and systems*, volume 9, pp. 1–10. Citeseer, 2013.

Shafieezadeh-Abadeh, S., Esfahani, P., and Kuhn, D. Distributionally robust logistic regression. In *Advances in Neural Information Processing Systems 28*, pp. 1576–1584. 2015.

Shafieezadeh-Abadeh, S., Kuhn, D., and Esfahani, P. M. Regularization via mass transportation. *Journal of Machine Learning Research*, 20(103):1–68, 2019.

Shapiro, A. Distributionally robust stochastic programming. *SIAM Journal on Optimization*, 27(4):2258–2275, 2017.

Shimodaira, H. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

Si, N., Zhang, F., Zhou, Z., and Blanchet, J. Distributional robust batch contextual bandits. *arXiv preprint arXiv:2006.05630*, 2020a.

Si, N., Zhang, F., Zhou, Z., and Blanchet, J. Distributionally robust policy evaluation and learning in offline contextual bandits. In *International Conference on Machine Learning (ICML)*, 2020b.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

Sinha, A., Namkoong, H., and Duchi, J. Certifiable distributional robustness with principled adversarial training. In *International Conference on Learning Representations*, 2018.

Staib, M. and Jegelka, S. Distributionally robust deep learning as a generalization of adversarial training. In *NIPS workshop on Machine Learning and Computer Security*, 2017.

Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A., and Hardt, M. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, pp. 9229–9248. PMLR, 2020.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., and McDaniel, P. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

Volpi, R., Namkoong, H., Sener, O., Duchi, J., Murino, V., and Savarese, S. Generalizing to unseen domains via adversarial data augmentation. *arXiv preprint arXiv:1805.12018*, 2018.

Wang, M. and Deng, W. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

Watkins, C. J. and Dayan, P. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

Wiesemann, W., Kuhn, D., and Rustem, B. Robust markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, 2013.

Wolff, E. M., Topcu, U., and Murray, R. M. Robust control of uncertain markov decision processes with temporal logic specifications. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pp. 3372–3379. IEEE, 2012.

Xu, H. and Mannor, S. Distributionally robust markov decision processes. In *Advances in Neural Information Processing Systems*, pp. 2505–2513, 2010.

Yang, I. Wasserstein distributionally robust stochastic control: A data-driven approach. *IEEE Transactions on Automatic Control*, 2020.

Yang, W., Zhang, L., and Zhang, Z. Towards theoretical understandings of robust markov decision processes: Sample complexity and asymptotics. *arXiv preprint arXiv:2105.03863*, 2021.

Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., and Jordan, M. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pp. 7472–7482. PMLR, 2019.

Zhao, C. and Guan, Y. Data-driven risk-averse stochastic optimization with Wasserstein metric. *Operations Research Letters*, 46(2):262 – 267, 2018. ISSN 0167-6377. doi: https://doi.org/10.1016/j.orl.2018.01.011.

Zhao, C. and Jiang, R. Distributionally robust contingency-constrained unit commitment. *IEEE Transactions on Power Systems*, 33(1):94–102, 2017.

Zhou, Z., Zhou, Z., Bai, Q., Qiu, L., Blanchet, J., and Glynn, P. Finite-sample regret bound for distributionally robust offline tabular reinforcement learning. In Banerjee, A. and Fukumizu, K. (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 3331–3339. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/zhou21d.html.

A. Technical Lemmas

First of all, we introduce two ancillary concentration inequalities.

Lemma A.1 ((Fournier & Guillin, 2015), Concentration inequality for Wasserstein distance). *For $\mu \in \mathcal{P}(\mathbb{R})$, we consider an i.i.d. sequence $(X_k)_{k\geq 1}$ of μ-distributed random variables and, for all $n \geq 1$, the empirical measure*

$$\mu_n := \frac{1}{n}\sum_{k=1}^{n} \delta_{X_k}.$$

Assume that there exists $\gamma > 0$ such that $\mathcal{E}_{2,\gamma}(\mu) := \int_{\mathbb{R}} \exp(\gamma|x|^2)\mu(dx) < \infty$. Then for all $n \geq 1$, all $x > 0$,

$$\mathbb{P}\left(\mathcal{W}(\mu_n, \mu) \geq x\right) \leq Ce^{-cnx^2},$$

where the Wasserstein distance $\mathcal{W}(\mu_n, \mu)$ is defined by

$$\mathcal{W}(\mu_n, \mu) := \inf_{\pi\in\Pi(\mu_n,\mu)} \left\{\int |x - y|\pi(dx, dy)\right\},$$

and the positive constant C and c depends only on γ and $\mathcal{E}_{2,\gamma}(\mu)$.

Lemma A.2 (Hoeffding's inequality)**.** *Let $X_1, \cdots, X_n$ be independent random variables such that $X_i \in [a_i, b_i]$ almost surely for all $i = 1, 2, \cdots, n$. Then for every $t > 0$,*

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n}(X_i - \mathbb{E}X_i)\right| \geq t\right) \leq 2\exp\left(-\frac{2n^2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right).$$

B. Missing Proof of Theorem 3.7

Proof. Fix a pair $(s, a) \in \mathcal{S} \times \mathcal{A}$, we first prove $\widehat{R}^{\text{rob}}_{\delta}(s, a)$ defined in Equation (9) is an unbiased estimator of $\sup_{\alpha\geq 0}\left\{-\alpha\log\left(\mathbb{E}_{\nu^0_{s,a}}\left[e^{-r(s,a)/\alpha}\right]\right) - \alpha\delta\right\}$ which forms the first part of our δ-distributionally robust Bellman Operator (see Equation (6)). Let

$$g(\alpha) = -\alpha\log\left(\mathbb{E}_{\nu^0_{s,a}}\left[e^{-r(s,a)/\alpha}\right]\right) - \alpha\delta.$$

Denote $\alpha^* = \text{argmax}_{\alpha\geq 0}\, g(\alpha)$. Given 2^N samples $\{r_i \sim \nu^0_{s,a}\}$, let $\widehat{\nu}_{2^N} = \frac{1}{2^N}\sum_{i=1}^{2^N}\delta_{r_i}$. Similarly, $\widehat{\nu}_{2^N_O}$ (resp. $\widehat{\nu}_{2^N_E}$) is defined by using the subset of 2^N samples in which the index of every element is odd (resp. even). We use g_{2^N} to represent the function defined by replacing $\nu^0_{s,a}$ by $\widehat{\nu}_{2^N}$ in g and $\alpha^*_{2^N} = \text{argmax}_{\alpha\geq 0}\, g_{2^N}(\alpha)$. Similarly, we also have $g_{2^N_O}$, $g_{2^N_E}$, $\alpha^*_{2^N_O}$ and $\alpha^*_{2^N_E}$. Now we have

$$\begin{aligned}
\mathbb{E}\left[\widehat{R}^{\text{rob}}_{\delta}(s, a)\right] &= \mathbb{E}\left[r_1 + \frac{\Delta^r_{N,\delta}}{p_N}\right] \\
&= \mathbb{E}\left[r_1\right] + \mathbb{E}\left[\frac{\Delta^r_{N,\delta}}{p_N}\right] \\
&= \mathbb{E}\left[g_{2^0}(\alpha^*_{2^0})\right] + \sum_{n=0}^{\infty}\mathbb{E}\left[\frac{\Delta^r_{N,\delta}}{p_N}\middle| N = n\right]p_n \\
&= \mathbb{E}\left[g_{2^0}(\alpha^*_{2^0})\right] + \sum_{n=0}^{\infty}\mathbb{E}\left[\Delta^r_{n,\delta}\right] \\
&= \mathbb{E}\left[g_{2^0}(\alpha^*_{2^0})\right] + \sum_{n=0}^{\infty}\mathbb{E}\left[g_{2^{n+1}}(\alpha^*_{2^{n+1}}) - \frac{g_{2^{n+1}_O}(\alpha^*_{2^{n+1}_O}) + g_{2^{n+1}_E}(\alpha^*_{2^{n+1}_E})}{2}\right] \\
&= \mathbb{E}\left[g_{2^0}(\alpha^*_{2^0})\right] + \sum_{n=0}^{\infty}\mathbb{E}\left[g_{2^{n+1}}(\alpha^*_{2^{n+1}}) - g_{2^n}(\alpha^*_{2^n})\right] \\
&= \lim_{n\to\infty}\mathbb{E}\left[g_{2^n}(\alpha^*_{2^n})\right].
\end{aligned}$$

We only need to show

$$\lim_{n\to\infty} \mathbb{E}\left[g_{2^n}(\alpha^*_{2^n})\right] = g(\alpha^*). \tag{13}$$

Based on our Assumption 2.1, we can give an upper bound of α^* by observing

$$\begin{aligned} 0 &\le \operatorname{ess\,inf} r(s,a) \\ &= g(0) \\ &\le g(\alpha^*) \\ &= -\alpha^* \log\left(\mathbb{E}_{\nu^0_{s,a}}\left[e^{-r(s,a)/\beta^*}\right]\right) - \alpha^*\delta \\ &\le R_{\max} - \alpha^*\delta \\ \Rightarrow \alpha^* &\le R_{\max}/\delta. \end{aligned}$$

By a similar argument, $\alpha^*_{2^n} \le R_{\max}/\delta$ also holds. Besides, from above derivation, we can find

$$0 \le \operatorname{ess\,inf} r(s,a) \le g(\alpha^*) \le R_{\max}. \tag{14}$$

Thus

$$0 \le g_{2^n}(\alpha^*_{2^n}) \le R_{\max} \tag{15}$$

holds by a similar argument. Starting from here, we split the proof into two cases: $\alpha^* = 0$ and $\alpha^* > 0$.

- $\alpha^* = 0$. By proposition 2 in (Hu & Hong, 2013), $\alpha^* = 0$ implies $\lambda \coloneqq \nu^0_{s,a}(r(s,a) = \operatorname{ess\,inf} r(s,a)) > 0$. We first introduce the following two events:

$$\begin{aligned} G_{2^n} &\coloneqq \{\exists r'_i = \operatorname{ess\,inf} r(s,a)\} \cap \{\forall r'_i \ge \operatorname{ess\,inf} r(s,a)\}, \\ Z_{2^n} &\coloneqq \{\alpha^*_{2^n} = 0\}. \end{aligned}$$

  Note that we have the following result for G_{2^n}:

$$\begin{aligned} \mathbb{P}[G^c_{2^n}] &= 1 - \mathbb{P}[G_{2^n}] \\ &= 1 - \mathbb{P}[\{\exists r'_i = \operatorname{ess\,inf} r(s,a)\}] \\ &= \mathbb{P}[\{\forall r'_i \ne \operatorname{ess\,inf} r(s,a)\}] \\ &= (1-\lambda)^{2^n}. \end{aligned}$$

  By Lemma 4 in (Zhou et al., 2021), $\forall \epsilon > 0$, there exists a constant $N(\epsilon,\delta)$, such that $\forall n \ge N(\epsilon,\delta)$, $\mathbb{P}[Z^c_{2^n}] \le \epsilon$. Now we choose $\epsilon > 0$ arbitrarily, when $n \ge N(\epsilon)$,

$$\begin{aligned} \mathbb{E}\left[|g_{2^n}(\alpha^*_{2^n}) - g(\alpha^*)|\right] &= \mathbb{E}\left[|g_{2^n}(\alpha^*_{2^n}) - g(\alpha^*)|\mathbb{1}_{G_{2^n}\cap Z_{2^n}}\right] + \mathbb{E}\left[|g_{2^n}(\alpha^*_{2^n}) - g(\alpha^*)|\mathbb{1}_{G^c_{2^n}\cup Z^c_{2^n}}\right] \\ &= \mathbb{E}\left[|g_{2^n}(\alpha^*_{2^n}) - g(\alpha^*)|\mathbb{1}_{G^c_{2^n}\cup Z^c_{2^n}}\right] \\ &\le R_{\max}(\mathbb{P}[G^c_{2^n}] + \mathbb{P}[Z^c_{2^n}]) \\ &\le R_{\max}((1-\lambda)^{2^n} + \epsilon), \end{aligned}$$

  where the first inequality is by Equation (14) and Equation (15). Hence we know

$$\lim_{n\to\infty} \mathbb{E}\left[|g_{2^n}(\alpha^*_{2^n}) - g(\alpha^*)|\right] \le \epsilon R_{\max},$$

  which implies

$$\lim_{n\to\infty} \mathbb{E}\left[|g_{2^n}(\alpha^*_{2^n}) - g(\alpha^*)|\right] = 0.$$

  Using above result, Equation (13) holds immediately in this case.

- $\alpha^* > 0$. We define the following event in this case,

$$NZ_{2^n} := \left\{ \frac{\alpha^*}{2} \le \alpha^*_{2^n} \right\}.$$

By Lemma 4 in (Zhou et al., 2021), $\forall \epsilon > 0$, there exists a constant $N'(\epsilon)$, such that once $n \ge N'(\epsilon)$, we have $\mathbb{P}[NZ^c_{2^n}] \le \epsilon$. Now we choose $\epsilon > 0$ arbitrarily, when $n \ge N'(\epsilon)$

$$\begin{aligned}\mathbb{E}\left[|g_{2^n}(\alpha^*_{2^n}) - g(\alpha^*)|\right] &\le \mathbb{E}\left[|g_{2^n}(\alpha^*_{2^n}) - g(\alpha^*)|\mathbb{1}_{NZ_{2^n}}\right] + \mathbb{E}\left[|g_{2^n}(\alpha^*_{2^n}) - g(\alpha^*)|\mathbb{1}_{NZ^c_{2^n}}\right] \\ &\le \mathbb{E}\left[\sup_{\alpha \in \left[\frac{\alpha^*}{2}, \frac{R_{\max}}{\delta}\right]} |g_{2^n}(\alpha) - g(\alpha)|\mathbb{1}_{NZ_{2^n}}\right] + R_{\max}\epsilon.\end{aligned}$$

Observe that $\mathbb{E}_{\nu^0_{s,a}}\left[e^{-r(s,a)/\alpha}\right] \ge e^{-R_{\max}/\alpha}$ and $\mathbb{E}_{\widehat{\nu}_{2^n}}\left[e^{-r(s,a)/\alpha}\right] \ge e^{-R_{\max}/\alpha}$ hold, combining the Lipschitz property of $\log x$ when x is bounded below, we know

$$\begin{aligned}\frac{|g_{2^n}(\alpha) - g(\alpha)|}{\alpha} &= \left|\log\left(\mathbb{E}_{\nu^0_{s,a}}\left[e^{-r(s,a)/\alpha}\right]\right) - \log\left(\mathbb{E}_{\widehat{\nu}_{2^n}}\left[e^{-r(s,a)/\alpha}\right]\right)\right| \\ &\le \frac{\left|\mathbb{E}_{\nu^0_{s,a}}\left[e^{-r(s,a)/\alpha}\right] - \mathbb{E}_{\widehat{\nu}_{2^n}}\left[e^{-r(s,a)/\alpha}\right]\right|}{e^{-R_{\max}/\alpha}}.\end{aligned}$$

Then we have

$$\begin{aligned}\sup_{\alpha \in \left[\frac{\alpha^*}{2}, \frac{R_{\max}}{\delta}\right]} |g_{2^n}(\alpha) - g(\alpha)|\mathbb{1}_{NZ_{2^n}} &\le \sup_{\alpha \in \left[\frac{\alpha^*}{2}, \frac{R_{\max}}{\delta}\right]} |g_{2^n}(\alpha) - g(\alpha)| \\ &= \sup_{\alpha \in \left[\frac{\alpha^*}{2}, \frac{R_{\max}}{\delta}\right]} \frac{|g_{2^n}(\alpha) - g(\alpha)|}{\alpha}\alpha \\ &\le \sup_{\alpha \in \left[\frac{\alpha^*}{2}, \frac{R_{\max}}{\delta}\right]} \left|\mathbb{E}_{\nu^0_{s,a}}\left[e^{-r(s,a)/\alpha}\right] - \mathbb{E}_{\widehat{\nu}_{2^n}}\left[e^{-r(s,a)/\alpha}\right]\right| \times \frac{R_{\max}e^{2R_{\max}/\alpha^*}}{\delta}.\end{aligned}$$

For any $\alpha \in \left[\frac{\alpha^*}{2}, \frac{R_{\max}}{\delta}\right]$, the function $e^{-x/\alpha}$ is a Lipschitz function on $[0, \infty)$, and the Lipschitz constant is bounded by $2/\alpha^*$. Hence, by the dual representation of Wasserstein distance, we have

$$\sup_{\alpha \in \left[\frac{\alpha^*}{2}, \frac{R_{\max}}{\delta}\right]} |g_{2^n}(\alpha) - g(\alpha)|\mathbb{1}_{NZ_{2^n}} \le \frac{2R_{\max}e^{2R_{\max}/\alpha^*}}{\delta\alpha^*}\mathcal{W}(\nu^0_{s,a}, \widehat{\nu}_{2^n}), \tag{16}$$

where the Wasserstein distance $\mathcal{W}(\nu^0_{s,a}, \widehat{\nu}_{2^n})$ is defined by

$$\mathcal{W}(\nu^0_{s,a}, \widehat{\nu}_{2^n}) := \inf_{\pi \in \Pi(\nu^0_{s,a}, \widehat{\nu}_{2^n})} \left\{ \int |x - y| \mathrm{d}\pi(x, y) \right\}.$$

Now we know

$$\mathbb{E}\left[|g_{2^n}(\alpha^*_{2^n}) - g(\alpha^*)|\right] \le \frac{2R_{\max}e^{2R_{\max}/\alpha^*}}{\delta\alpha^*}\mathbb{E}\left[\mathcal{W}(\nu^0_{s,a}, \widehat{\nu}_{2^n})\right] + R_{\max}\epsilon.$$

By Lemma A.1 (note that we assume r is bounded, so the condition for Lemma A.1 is satisfied automatically), there exists $c, C > 0$ such that $\mathbb{P}[\mathcal{W}(\nu^0_{s,a}, \widehat{\nu}_{2^n}) \ge t] \le Ce^{-2^n ct^2}$. We have

$$\begin{aligned}\mathbb{E}\left[\mathcal{W}(\nu^0_{s,a}, \widehat{\nu}_{2^n})\right] &= \int_0^\infty \mathbb{P}[\mathcal{W}(\nu^0_{s,a}, \widehat{\nu}_{2^n}) \ge t]\mathrm{d}t \\ &= \int_0^{R_{\max}} \mathbb{P}[\mathcal{W}(\nu^0_{s,a}, \widehat{\nu}_{2^n}) \ge t]\mathrm{d}t \\ &\le \int_0^{R_{\max}} Ce^{-2^n ct^2}\mathrm{d}t \\ &\le \frac{C\sqrt{\pi}}{\sqrt{c}2^{\frac{n+2}{2}}}.\end{aligned}$$

This implies

$$\lim_{n\to\infty} \mathbb{E}\left[|g_{2^n}(\alpha^*_{2^n}) - g(\alpha^*)|\right] \le R_{\max}\epsilon.$$

Note that we choose $\epsilon > 0$ arbitrarily, so there is

$$\lim_{n\to\infty} \mathbb{E}\left[|g_{2^n}(\alpha^*_{2^n}) - g(\alpha^*)|\right] = 0$$

Hence we know Equation (13) holds in this case.

Now we can conclude that $\widehat{R}^{\text{rob}}_\delta(s,a)$ is an unbiased estimator of $\sup_{\alpha\ge 0}\left\{-\alpha\log\left(\mathbb{E}_{\nu^0_{s,a}}\left[e^{-r(s,a)/\alpha}\right]\right) - \alpha\delta\right\}$.

From here we give the proof that $\widehat{T}^{\text{rob}}_\delta(Q)(s,a)$ defined in Equation (10) is an unbiased estimator of $\sup_{\beta\ge 0}\left\{-\beta\log\left(\mathbb{E}_{p^0_{s,a}}\left[e^{-\max_{b\in\mathcal{A}} Q(s',b)/\beta}\right]\right) - \beta\delta\right\}$ which constructs the remaining part of our δ-distributionally robust Bellman Operator. Given $Q \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ and $(s,a)\in\mathcal{S}\times\mathcal{A}$. We define $V(s') = \max_{b\in\mathcal{A}} Q(s',b)$ for any $s'\in\mathcal{S}$ and

$$\mathcal{S}^* = \operatorname*{argmin}_{s'\in\mathcal{S}, p^0_{s,a}(s')>0} V(s').$$

Let

$$f(\beta) = -\beta\log\left(\mathbb{E}_{p^0_{s,a}}\left[e^{-V(s')/\beta}\right]\right) - \beta\delta.$$

Denote $\beta^* = \operatorname{argmax}_{\beta\ge 0} f(\beta)$. Given 2^N samples $\{s'_i \sim p^0_{s,a}\}$, let $\widehat{p}_{2^N}(s) = \frac{\sum_{i=1}^{2^N}\mathbb{1}[s'_i=s]}{2^N}$. Similarly, $\widehat{p}_{2^N_O}$ (resp. $\widehat{p}_{2^N_E}$) is defined by using the subset of 2^N samples in which the index of every element is odd (resp. even). We use f_{2^N} to represent the function defined by replacing $p^0_{s,a}$ by $\widehat{p}_{2^N}$ in f and $\beta^*_{2^N} = \operatorname{argmax}_{\beta\ge 0} f_{2^N}(\beta)$. Similarly, we also have $f_{2^N_O}$, $f_{2^N_E}$, $\beta^*_{2^N_O}$ and $\beta^*_{2^N_E}$. Following the similar proof for g used in the previous part, we will have

$$\mathbb{E}\left[\widehat{T}^{\text{rob}}_\delta(Q)(s,a)\right] = \lim_{n\to\infty}\mathbb{E}\left[f_{2^n}(\beta^*_{2^n})\right].$$

Hence we only need to show

$$\lim_{n\to\infty}\mathbb{E}\left[f_{2^n}(\beta^*_{2^n})\right] = f(\beta^*). \tag{17}$$

We can observe that both β^* and $\beta^*_{2^n}$ are no bigger than $2\|Q\|_\infty/\delta$. For β^*, this is because

$$\begin{aligned}
\min_{\substack{s'\in\mathcal{S}\\ p^0_{s,a}(s')>0}} V(s') =& f(0)\\
\le& f(\beta^*)\\
=& -\beta^*\log\left(\mathbb{E}_{p^0_{s,a}}\left[e^{-V(s')/\beta^*}\right]\right) - \beta^*\delta\\
\le& \max_{\substack{s'\in\mathcal{S}\\ p^0_{s,a}(s')>0}} V(s') - \beta^*\delta\\
\Rightarrow \beta^* \le& \left(\max_{\substack{s'\in\mathcal{S}\\ p^0_{s,a}(s')>0}} V(s') - \min_{\substack{s'\in\mathcal{S}\\ p^0_{s,a}(s')>0}} V(s')\right)/\delta\\
\le& 2\|Q\|_\infty/\delta.
\end{aligned}$$

If we apply the similar argument to $\beta^*_{2^n}$, we can get the same result. Besides, from the above derivation, we can see

$$-\|Q\|_\infty \le \min_{s'\in\mathcal{S}, p^0_{s,a}(s')>0} V(s') \le f(\beta^*) \le \max_{s'\in\mathcal{S}, p^0_{s,a}(s')>0} V(s') \le \|Q\|_\infty. \tag{18}$$

Besides,

$$-\|Q\|_\infty \le f_{2^n}(\beta^*_{2^n}) \le \|Q\|_\infty \tag{19}$$

also holds with a similar reason. Now we define the following two key events,

$$E_{2^n} \coloneqq \left\{p^0_{s,a}(s'_i) > 0, \forall 1 \le i \le 2^n\right\},$$
$$T_{2^n} \coloneqq \left\{\sum_{s'\in\mathcal{S}^*} \widehat{p}_{2^n}(s') \ge \frac{1}{2}\sum_{s'\in\mathcal{S}^*} p^0_{s,a}(s')\right\}.$$

Note the facts $\mathbb{P}\left[E_2^n\right] = 1$ and

$$\begin{aligned}
\mathbb{P}\left[T^c_{2^n}\right] &= \mathbb{P}\left[\sum_{s'\in\mathcal{S}^*} \widehat{p}_{2^n}(s') < \frac{1}{2}\sum_{s'\in\mathcal{S}^*} p^0_{s,a}(s')\right] \\
&= \mathbb{P}\left[\sum_{s'\in\mathcal{S}^*} p^0_{s,a}(s') - \sum_{s'\in\mathcal{S}^*} \widehat{p}_{2^n}(s') > \frac{1}{2}\sum_{s'\in\mathcal{S}^*} p^0_{s,a}(s')\right] \\
&= \mathbb{P}\left[\sum_{s'\in\mathcal{S}^*} p^0_{s,a}(s') - \sum_{s'\in\mathcal{S}^*}\sum_{i=1}^{2^n} \frac{\mathbb{1}[s'_i = s']}{2^n} > \frac{1}{2}\sum_{s'\in\mathcal{S}^*} p^0_{s,a}(s')\right] \\
&= \mathbb{P}\left[\sum_{s'\in\mathcal{S}^*} p^0_{s,a}(s') - \frac{1}{2^n}\sum_{i=1}^{2^n}\sum_{s'\in\mathcal{S}^*} \mathbb{1}[s'_i = s'] > \frac{1}{2}\sum_{s'\in\mathcal{S}^*} p^0_{s,a}(s')\right] \\
&\le e^{-2^{n-1}\left(\sum_{s'\in\mathcal{S}^*} p^0_{s,a}(s')\right)^2},
\end{aligned}$$

where the last inequality is right because of Lemma A.2. With the above proposition, we can start to show that Equation (17) is true, we consider the following decomposition

$$\begin{aligned}
\mathbb{E}\left[\left|f_{2^n}(\beta^*_{2^n}) - f(\beta^*)\right|\right] &= \mathbb{E}\left[\left|f_{2^n}(\beta^*_{2^n}) - f(\beta^*)\right|\mathbb{1}_{E_{2^n}}\right] \\
&= \mathbb{E}\left[\left|f_{2^n}(\beta^*_{2^n}) - f(\beta^*)\right|\mathbb{1}_{E_{2^n}\cap T_{2^n}}\right] + \mathbb{E}\left[\left|f_{2^n}(\beta^*_{2^n}) - f(\beta^*)\right|\mathbb{1}_{E_{2^n}\cap T^c_{2^n}}\right] \\
&\le \mathbb{E}\left[\left|f_{2^n}(\beta^*_{2^n}) - f(\beta^*)\right|\mathbb{1}_{T_{2^n}\cap E_{2^n}}\right] + 2\|Q\|_\infty \mathbb{P}\left[T^c_{2^n}\cap E_{2^n}\right],
\end{aligned}$$

where the last inequality holds due to Equation (18) and Equation (19). Now denote $\min_{s'\in\mathcal{S}, p^0_{s,a}(s')>0} V(s')$ as v. Note that we have shown $0 \le \beta^*, \beta^*_{2^n} \le 2\|Q\|_\infty/\delta$. Observe that under T_{2^n} and E_{2^n}, $\forall \beta \in [0, 2\|Q\|_\infty/\delta]$, there are

$$\begin{aligned}
\mathbb{E}_{p^0_{s,a}}\left[e^{(-V(s')+v)/\beta}\right] &= \sum_{s'\in\mathcal{S}} p^0_{s,a}(s') e^{(-V(s')+v)/\beta} \ge \sum_{s'\in\mathcal{S}^*} p^0_{s,a}(s') e^{(-V(s')+v)/\beta} \\
&= \sum_{s'\in\mathcal{S}^*} p^0_{s,a}(s') \ge \frac{1}{2}\sum_{s'\in\mathcal{S}^*} p^0_{s,a}(s')
\end{aligned}$$

and

$$\begin{aligned}
\mathbb{E}_{\widehat{p}_{2^n}}\left[e^{(-V(s')+v)/\beta}\right] &= \sum_{s'\in\mathcal{S}} \widehat{p}_{2^n}(s') e^{(-V(s')+v)/\beta} \ge \sum_{s'\in\mathcal{S}^*} \widehat{p}_{2^n}(s') e^{(-V(s')+v)/\beta} \\
&= \sum_{s'\in\mathcal{S}^*} \widehat{p}_{2^n}(s') \ge \frac{1}{2}\sum_{s;\in\mathcal{S}^*} p^0_{s,a}(s').
\end{aligned}$$

Then under T_{2^n} and E_{2^n}, we know

$$\begin{aligned}
\left|f_{2^n}(\beta^*_{2^n}) - f(\beta^*)\right| &\le \sup_{\beta\in\left[0,\frac{2\|Q\|_\infty}{\delta}\right]} \beta\left|\log\left(\mathbb{E}_{p^0_{s,a}}\left[e^{-V(s')/\beta}\right]\right) - \log\left(\mathbb{E}_{\widehat{p}_{2^n}}\left[e^{-V(s')/\beta}\right]\right)\right| \\
&\le \frac{2\|Q\|_\infty}{\delta} \sup_{\beta\in\left[0,\frac{2\|Q\|_\infty}{\delta}\right]} \left|\log\left(\mathbb{E}_{p^0_{s,a}}\left[e^{(-V(s')+v)/\beta}\right]\right) - \log\left(\mathbb{E}_{\widehat{p}_{2^n}}\left[e^{(-V(s')+v)/\beta}\right]\right)\right| \\
&\le \frac{2\|Q\|_\infty}{\delta} \sup_{\beta\in\left[0,\frac{2\|Q\|_\infty}{\delta}\right]} \frac{\left|\mathbb{E}_{p^0_{s,a}}\left[e^{(-V(s')+v)/\beta}\right] - \mathbb{E}_{\widehat{p}_{2^n}}\left[e^{(-V(s')+v)/\beta}\right]\right|}{\frac{1}{2}\sum_{s'\in\mathcal{S}^*} p^0_{s,a}(s')} \\
&\le \frac{4\|Q\|_\infty}{\sum_{s'\in\mathcal{S}^*} p^0_{s,a}(s')} \|p^0_{s,a} - \widehat{p}_{2^n}\|_1.
\end{aligned} \tag{20}$$

The third inequality is right is by the Lipschitz property for $\log x$ when x is bounded from below. Finally we have

$$\begin{aligned}\mathbb{E}\left[|f_{2^n}(\beta^*_{2^n}) - f(\beta^*)|\right] &\le \mathbb{E}\left[|f_{2^n}(\beta^*_{2^n}) - f(\beta^*)|\mathbb{1}_{T_{2^n}\cap E_{2^n}}\right] + 2\|Q\|_\infty \mathbb{P}\left[T^c_{2^n}\cap E_{2^n}\right] \\ &\le \frac{4\|Q\|_\infty}{\sum_{s'\in\mathcal{S}^*} p^0_{s,a}(s')}\mathbb{E}\left[\|p^0_{s,a} - \widehat{p}_{2^n}\|_1\right] + \|Q\|_\infty e^{-2^{n-1}\left(\sum_{s'\in\mathcal{S}^*} p^0_{s,a}(s')\right)^2}.\end{aligned}$$

Note

$$\begin{aligned}\mathbb{E}\left[\|p^0_{s,a} - \widehat{p}_{2^n}\|_1\right] &= \int_0^\infty \mathbb{P}\left[\|p^0_{s,a} - \widehat{p}_{2^n}\|_1 \ge t\right] \mathrm{d}t \\ &= \int_0^2 \mathbb{P}\left[\|p^0_{s,a} - \widehat{p}_{2^n}\|_1 \ge t\right] \mathrm{d}t \\ &\le \int_0^2 \sum_{s'\in\mathcal{S}} \mathbb{P}\left[|p^0_{s,a}(s') - \widehat{p}_{2^n}(s')| \ge \frac{t}{|\mathcal{S}|}\right] \mathrm{d}t \\ &\le |\mathcal{S}| \int_0^2 2e^{-2^{n+1}t^2/|\mathcal{S}|^2} \mathrm{d}t \\ &\le \frac{|\mathcal{S}|^2\sqrt{\pi}}{2^{\frac{n+1}{2}}},\end{aligned}$$

where the second inequality is by Lemma A.2. Hence we have

$$\mathbb{E}\left[|f_{2^n}(\beta^*_{2^n}) - f(\beta^*)|\right] \le \frac{4\|Q\|_\infty |\mathcal{S}|^2\sqrt{\pi}}{\left(\sum_{s'\in\mathcal{S}^*} p^0_{s,a}(s')\right) 2^{\frac{n+3}{2}}} + 2\|Q\|_\infty e^{-2^{n-1}\left(\sum_{s'\in\mathcal{S}^*} p^0_{s,a}(s')\right)^2}.$$

This is enough to conclude

$$\lim_{n\to\infty} \mathbb{E}\left[|f^n_2(\beta^*_{2^n}) - f(\beta^*)|\right] = 0. \tag{21}$$

Note that Equation (21) implies that Equation (17) is true. Now we complete our partial proof, i.e., $\widehat{T}^{\mathrm{rob}}_\delta(Q)(s,a)$ is an unbiased estimator of $\sup_{\beta\ge 0}\left\{-\beta\log\left(\mathbb{E}_{p^0_{s,a}}\left[e^{-\max_{b\in\mathcal{A}} Q(s',b)/\beta}\right]\right) - \beta\delta\right\}$.

Finally we complete the proof that $\widehat{\mathcal{T}}^{\mathrm{rob}}_{\delta,\varepsilon}(Q)(s,a) \coloneqq \widehat{R}^{\mathrm{rob}}_\delta(s,a) + \gamma\widehat{T}^{\mathrm{rob}}_\delta(Q)(s,a)$ is an unbiased estimator of $\mathcal{T}^{\mathrm{rob}}_\delta(Q)(s,a)$ for any $Q\in\mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ and $(s,a)\in\mathcal{S}\times\mathcal{A}$. □

C. Missing Proof of Theorem 3.8

Proof. Following the same notations defined in the proof of Theorem 3.7 (cf. Appendix B), we have already known for any $\varepsilon\in(0,0.5)$, $Q\in\mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ and $(s,a)\in\mathcal{S}\times\mathcal{A}$,

$$\begin{aligned}\mathbb{E}[\widehat{\mathcal{T}}^{\mathrm{rob}}_{\delta,\varepsilon}(Q)(s,a)] &= \mathbb{E}[\widehat{R}^{\mathrm{rob}}_\delta(s,a) + \gamma\widehat{T}^{\mathrm{rob}}_\delta(Q)(s,a)] \\ &= \sup_{\alpha\ge 0}\left\{-\alpha\log\left(\mathbb{E}_{\nu^0_{s,a}}\left[e^{-r(s,a)/\alpha}\right]\right) - \alpha\delta\right\} + \gamma\cdot\sup_{\beta\ge 0}\left\{-\beta\log\left(\mathbb{E}_{p^0_{s,a}}\left[e^{-\max_{b\in\mathcal{A}} Q(s',b)/\beta}\right]\right) - \beta\delta\right\} \\ &= g(\alpha^*) + \gamma f(\beta^*).\end{aligned}$$

By the definition of variance we have

$$\mathsf{Var}[\widehat{\mathcal{T}}^{\mathrm{rob}}_{\delta,\varepsilon}(Q)(s,a)] \le 2\mathsf{Var}[\widehat{R}^{\mathrm{rob}}_\delta(s,a)] + 2\gamma^2\mathsf{Var}[\widehat{T}^{\mathrm{rob}}_\delta(Q)(s,a)].$$

We first analysis the term $\mathsf{Var}[\widehat{R}_\delta^{\mathrm{rob}}(s,a)]$ by noticing

$$\begin{aligned}\mathsf{Var}[\widehat{R}_\delta^{\mathrm{rob}}(s,a)] &= \mathbb{E}[(\widehat{R}_\delta^{\mathrm{rob}}(s,a) - g(\alpha^*))^2] \\ &\le \mathbb{E}\left[\left(r_1 + \frac{\Delta^r_{N,\delta}}{p_N}\right)^2\right] \\ &\le 2\mathbb{E}[(r_1)^2] + 2\mathbb{E}\left[\left(\frac{\Delta^r_{N,\delta}}{p_N}\right)^2\right] \\ &\le 2R_{\max}^2 + 2\sum_{n=0}^{\infty}\frac{\mathbb{E}[(\Delta^r_{n,\delta})^2]}{p_n}.\end{aligned}$$

Now let us bound the term $\mathbb{E}[(\Delta^r_{n,\delta})^2]$. Like the proof of Theorem 3.7, we also consider following two cases

- $\alpha^* = 0$. By Proposition 2 in (Hu & Hong, 2013), $\alpha^* = 0$ if and only if $\lambda := \nu^0_{s,a}(r(s,a) = \operatorname{ess\,inf} r(s,a)) > 0$ and $\lambda \ge e^{-\delta}$. Since δ is chosen by us, we can ignore the edge case $\lambda = e^{-\delta}$ by introducing randomness on δ. Now we define the following three events:

$$\begin{aligned}C_{2^n_O} &:= \{\widehat{\nu}_{2^n_O}(\operatorname{ess\,inf} r(s,a)) > e^{-\delta}\}, \\ C_{2^n_E} &:= \{\widehat{\nu}_{2^n_E}(\operatorname{ess\,inf} r(s,a)) > e^{-\delta}\}, \\ C_{2^n} &:= \{\widehat{\nu}_{2^n}(\operatorname{ess\,inf} r(s,a)) > e^{-\delta}\}.\end{aligned}$$

  When the event $C_{2^n_O} \cap G_{2^n_O}$ (recall $G_{2^n_O} := \{\exists r'_i = \operatorname{ess\,inf} r(s,a), i \in \{1,3,\cdots,2^n-1\}\} \cap \{\forall r'_i \ge \operatorname{ess\,inf} r(s,a), i \in \{1,3,\cdots,2^n-1\}\}$) holds, again, Proposition 2 in (Hu & Hong, 2013) implies that we have $\alpha^*_{2^n_O} = 0$ and $g_{2^n_O}(\alpha^*_{2^n_O}) = \operatorname{ess\,inf} r(s,a)$. The same argument can be applied to the subscript 2^n_E and 2^n. Besides, note that

$$C_{2^n_O} \cap C_{2^n_E} \subseteq C_{2^n}$$

  and

$$G_{2^n_O} \cap G_{2^n_E} \subseteq G_{2^n},$$

  which implies $C_{2^n_O} \cap C_{2^n_E} \cap G_{2^n_O} \cap G_{2^n_E} \subseteq C_{2^n} \cap G_{2^n}$. Then we have the following result

$$\begin{aligned}\mathbb{E}[(\Delta^r_{n,\delta})^2] &= \mathbb{E}\left[\left(g_{2^{n+1}}(\alpha^*_{2^{n+1}}) - \frac{g_{2^{n+1}_O}(\alpha^*_{2^{n+1}_O}) + g_{2^{n+1}_E}(\alpha^*_{2^{n+1}_E})}{2}\right)^2\right] \\ &= \mathbb{E}\left[\left(g_{2^{n+1}}(\alpha^*_{2^{n+1}}) - \frac{g_{2^{n+1}_O}(\alpha^*_{2^{n+1}_O}) + g_{2^{n+1}_E}(\alpha^*_{2^{n+1}_E})}{2}\right)^2 \mathbb{1}_{C_{2^{n+1}_O}\cap C_{2^{n+1}_E}\cap G_{2^{n+1}_O}\cap G_{2^{n+1}_E}}\right] \\ &\quad + \mathbb{E}\left[\left(g_{2^{n+1}}(\alpha^*_{2^{n+1}}) - \frac{g_{2^{n+1}_O}(\alpha^*_{2^{n+1}_O}) + g_{2^{n+1}_E}(\alpha^*_{2^{n+1}_E})}{2}\right)^2 \mathbb{1}_{(C_{2^{n+1}_O}\cap C_{2^{n+1}_E}\cap G_{2^{n+1}_O}\cap G_{2^{n+1}_E})^c}\right] \\ &= \mathbb{E}\left[\left(g_{2^{n+1}}(\alpha^*_{2^{n+1}}) - \frac{g_{2^{n+1}_O}(\alpha^*_{2^{n+1}_O}) + g_{2^{n+1}_E}(\alpha^*_{2^{n+1}_E})}{2}\right)^2 \mathbb{1}_{(C_{2^{n+1}_O}\cap C_{2^{n+1}_E}\cap G_{2^{n+1}_O}\cap G_{2^{n+1}_E})^c}\right] \\ &\le 2R_{\max}^2(\mathbb{P}[C^c_{2^n}] + \mathbb{P}[G^c_{2^n}]).\end{aligned}$$

  Note the following two bounds

$$\mathbb{P}[C^c_{2^n}] \le e^{-2^{n+1}(\lambda - e^{-\delta})^2}, \quad \mathbb{P}[G^c_{2^n}] \le (1-\lambda)^{2^n}.$$

  The first bound is due to Lemma A.2 again. Finally we have

$$\mathbb{E}[(\Delta^r_{n,\delta})^2] \le 2R_{\max}^2(e^{-2^{n+1}(\lambda - e^{-\delta})^2} + (1-\lambda)^{2^n}).$$

Then if we choose $p_n = \varepsilon(1-\varepsilon)^n$, we know

$$\mathsf{Var}[\widehat{R}_\delta^{\mathrm{rob}}(s,a)] \le 2R_{\max}^2 + 4R_{\max}^2 \sum_{n=0}^{\infty} \frac{e^{-2^{n+1}(\lambda - e^{-\delta})^2} + (1-\lambda)^{2^n}}{p_n} = K_1'(\lambda, \delta, \varepsilon, \nu^0) < \infty.$$

Note that λ depends on (s,a). However, the number of pair (s,a) is finite, hence we can find a uniform constant bound $K_1(\delta, \varepsilon, \nu^0) \ge K_1'(\lambda, \delta, \varepsilon, \nu^0)$ for any $(s,a) \in \mathcal{S} \times \mathcal{A}$ which makes $\alpha^* = 0$.

- $\alpha^* > 0$. In this case, we follow the way proposed by (Zhou et al., 2021). Define

$$\tau := \min\left\{ \underline{\alpha} \log\left(\mathbb{E}_{\nu^0_{s,a}}\left[e^{-X/\underline{\alpha}}\right]\right) + \underline{\alpha}\delta, \bar{\alpha}\log\left(\mathbb{E}_{\nu^0_{s,a}}\left[e^{-X/\bar{\alpha}}\right]\right) + \bar{\alpha}\delta \right\} - \left(\alpha^* \log\left(\mathbb{E}_{\nu^0_{s,a}}\left[e^{-X/\alpha^*}\right]\right) + \alpha^*\delta\right),$$

where $\underline{\alpha} = \alpha^*/2$ and $\overline{\alpha} = R_{\max}/\delta$. Besides, we introduce the following event

$$F_{2^n}(\alpha) := \left\{ \left| \mathbb{E}_{\widehat{\nu}_{2^n}}\left[e^{-r(s,a)/\alpha}\right] - \mathbb{E}_{\nu^0_{s,a}}\left[e^{-r(s,a)/\alpha}\right] \right| < \frac{\tau}{2}\left(2\alpha e^{R_{\max}/\alpha}\right)^{-1} \right\}.$$

Note that by Lemma 4 in (Zhou et al., 2021), under $F_{2^n}(\underline{\alpha}) \cap F_{2^n}(\overline{\alpha}) \cap F_{2^n}(\alpha^*)$, we have $\alpha^*_{2^n} \in [\underline{\alpha}, \overline{\alpha}]$. Let $F_{2^n} = F_{2^n}(\underline{\alpha}) \cap F_{2^n}(\overline{\alpha}) \cap F_{2^n}(\alpha^*)$. Note that $F_{2^n_O} \cap F_{2^n_E} \subset F_{2^n}$, thus we have

$$\begin{aligned}
\mathbb{E}[(\Delta^r_{n,\delta})^2] &= \mathbb{E}\left[\left(g_{2^{n+1}}(\alpha^*_{2^{n+1}}) - \frac{g_{2^{n+1}_O}(\alpha^*_{2^{n+1}_O}) + g_{2^{n+1}_E}(\alpha^*_{2^{n+1}_E})}{2}\right)^2\right] \\
&= \mathbb{E}\left[\left(g_{2^{n+1}}(\alpha^*_{2^{n+1}}) - \frac{g_{2^{n+1}_O}(\alpha^*_{2^{n+1}_O}) + g_{2^{n+1}_E}(\alpha^*_{2^{n+1}_E})}{2}\right)^2 \mathbb{1}_{F_{2^{n+1}_O} \cap F_{2^{n+1}_E}}\right] \\
&\quad + \mathbb{E}\left[\left(g_{2^{n+1}}(\alpha^*_{2^{n+1}}) - \frac{g_{2^{n+1}_O}(\alpha^*_{2^{n+1}_O}) + g_{2^{n+1}_E}(\alpha^*_{2^{n+1}_E})}{2}\right)^2 \mathbb{1}_{(F_{2^{n+1}_O} \cap F_{2^{n+1}_E})^c}\right] \\
&\le \mathbb{E}\left[\left(g_{2^{n+1}}(\alpha^*_{2^{n+1}}) - \frac{g_{2^{n+1}_O}(\alpha^*_{2^{n+1}_O}) + g_{2^{n+1}_E}(\alpha^*_{2^{n+1}_E})}{2}\right)^2 \mathbb{1}_{F_{2^{n+1}_O} \cap F_{2^{n+1}_E}}\right] + 2R_{\max}^2 \mathbb{P}[F^c_{2^n}] \\
&\le 2\mathbb{E}\left[\left(g_{2^{n+1}}(\alpha^*_{2^{n+1}}) - g(\alpha^*)\right)^2 \mathbb{1}_{F_{2^{n+1}}}\right] + 2\mathbb{E}\left[\left(g_{2^n}(\alpha^*_{2^n}) - g(\alpha^*)\right)^2 \mathbb{1}_{F_{2^n}}\right] + 2R_{\max}^2 \mathbb{P}[F^c_{2^n}] \\
&\le \frac{16R_{\max}^2 e^{4R_{\max}/\alpha^*}}{(\delta\alpha^*)^2}\mathbb{E}[\mathcal{W}^2(\nu^0_{s,a}, \widehat{\nu}_{2^{n+1}})] + \frac{16R_{\max}^2 e^{4R_{\max}/\alpha^*}}{(\delta\alpha^*)^2}\mathbb{E}[\mathcal{W}^2(\nu^0_{s,a}, \widehat{\nu}_{2^n})] + 2R_{\max}^2 \mathbb{P}[F^c_{2^n}],
\end{aligned}$$

where the last inequality holds by a similar argument for Equation (16). Using a similar calculation used in the proof of Theorem 3.7, we can find

$$\begin{aligned}
\mathbb{E}[\mathcal{W}^2(\nu^0_{s,a}, \widehat{\nu}_{2^{n+1}})] &= O(2^{-n}), \\
\mathbb{E}[\mathcal{W}^2(\nu^0_{s,a}, \widehat{\nu}_{2^n})] &= O(2^{-n}).
\end{aligned}$$

Besides, by Lemma 4 in (Zhou et al., 2021), we know $\mathbb{P}[F^c_{2^n}] = O(e^{-2^n})$, Besides, note $p_n = \varepsilon(1-\varepsilon)^n$ for $\varepsilon \in (0, 0.5)$. Then we know

$$\begin{aligned}
&\mathbb{E}[(\widehat{R}_\delta^{\mathrm{rob}}(s,a) - g(\alpha^*))^2] \\
\le& 2R_{\max}^2 + \sum_{n=0}^{\infty} \frac{16R_{\max}^2 e^{4R_{\max}/\alpha^*}}{(\delta\alpha^*)^2}\frac{\mathbb{E}[\mathcal{W}^2(\nu^0_{s,a}, \widehat{\nu}_{2^{n+1}})]}{p_n} + \frac{16R_{\max}^2 e^{4R_{\max}/\alpha^*}}{(\delta\alpha^*)^2}\frac{\mathbb{E}[\mathcal{W}^2(\nu^0_{s,a}, \widehat{\nu}_{2^n})]}{p_n} + 2R_{\max}^2\frac{\mathbb{P}[F^c_{2^n}]}{p_n} \\
=& K_2'(\alpha^*, \delta, \varepsilon, \nu^0) \\
<& \infty.
\end{aligned}$$

Since we are in the tabular setting ,we can find a constant $K_2(\delta, \varepsilon, \nu^0) > K_2'(\alpha^*, \delta, \varepsilon, \nu^0)$ for any (s,a) which makes $\alpha^* > 0$.

By the previous two cases, we know $\mathbb{E}[(\widehat{R}^{\rm rob}_\delta(s,a)-g(\alpha^*))^2]\le \max(K_1(\delta,\varepsilon,\nu^0),K_2(\delta,\varepsilon,\nu^0))$. Now we start to analysis $\mathsf{Var}[\widehat{T}^{\rm rob}_\delta(Q)(s,a)]=\mathbb{E}[(\widehat{T}^{\rm rob}_\delta(Q)(s,a)-f(\beta^*))^2]$. By the similar approach, we can find

$$\begin{aligned}\mathbb{E}[(\widehat{T}^{\rm rob}_\delta(Q)(s,a)-f(\beta^*))^2]&=\mathbb{E}[(\max_{b\in\mathcal{A}}Q(s_1',b)+\frac{\Delta^q_{N,\delta}}{p_N}-f(\beta^*))^2]\\&\le \mathbb{E}[(\max_{b\in\mathcal{A}}Q(s_1',b)+\frac{\Delta^q_{N,\delta}}{p_N})^2]\\&\le 2\mathbb{E}[(\max_{b\in\mathcal{A}}Q(s_1',b))^2]+2\mathbb{E}\left[\left(\frac{\Delta^q_{N,\delta}}{p_N}\right)^2\right]\\&\le 2\|Q\|^2_\infty+2\sum_{n=0}^\infty\frac{\mathbb{E}[(\Delta^q_{n,\delta})^2]}{p_n}.\end{aligned}$$

Use the same notations we defined in the proof of Theorem 3.7, we can find

$$\begin{aligned}&\mathbb{E}[(\Delta^q_{n,\delta})^2]\\=&\mathbb{E}\left[\left(f_{2^{n+1}}(\beta^*_{2^{n+1}})-\frac{f_{2^{n+1}_O}(\beta^*_{2^{n+1}_O})+f_{2^{n+1}_E}(\beta^*_{2^{n+1}_E})}{2}\right)^2\right]\\=&\mathbb{E}\left[\left(f_{2^{n+1}}(\beta^*_{2^{n+1}})-\frac{f_{2^{n+1}_O}(\beta^*_{2^{n+1}_O})+f_{2^{n+1}_E}(\beta^*_{2^{n+1}_E})}{2}\right)^2\mathbb{1}_{E_{2^{n+1}_O}\cap E_{2^{n+1}_E}\cap T_{2^{n+1}_O}\cap T_{2^{n+1}_E}}\right]\\&+\mathbb{E}\left[\left(f_{2^{n+1}}(\beta^*_{2^{n+1}})-\frac{f_{2^{n+1}_O}(\beta^*_{2^{n+1}_O})+f_{2^{n+1}_E}(\beta^*_{2^{n+1}_E})}{2}\right)^2\mathbb{1}_{(E_{2^{n+1}_O}\cap E_{2^{n+1}_E}\cap T_{2^{n+1}_O}\cap T_{2^{n+1}_E})^c}\right]\\\le&\mathbb{E}\left[\left(f_{2^{n+1}}(\beta^*_{2^{n+1}})-\frac{f_{2^{n+1}_O}(\beta^*_{2^{n+1}_O})+f_{2^{n+1}_E}(\beta^*_{2^{n+1}_E})}{2}\right)^2\mathbb{1}_{E_{2^{n+1}_O}\cap E_{2^{n+1}_E}\cap T_{2^{n+1}_O}\cap T_{2^{n+1}_E}}\right]+8\|Q\|^2_\infty(\mathbb{P}[E^c_{2^n}]+\mathbb{P}[T^c_{2^n}])\\\le&2\mathbb{E}\left[\left(f_{2^{n+1}}(\beta^*_{2^{n+1}})-f(\beta^*)\right)^2\mathbb{1}_{E_{2^{n+1}}\cap T_{2^{n+1}}}\right]+2\mathbb{E}\left[\left(f_{2^n}(\beta^*_{2^n})-f(\beta^*)\right)^2\mathbb{1}_{E_{2^n}\cap T_{2^n}}\right]+8\|Q\|^2_\infty(\mathbb{P}[E^c_{2^n}]+\mathbb{P}[T^c_{2^n}])\\\le&\frac{32\|Q\|^2_\infty}{(\sum_{s'\in\mathcal{S}^*}p^0_{s,a}(s'))^2}\mathbb{E}[\|p^0_{s,a}-\widehat{p}_{2^{n+1}}\|_1^2]+\frac{32\|Q\|^2_\infty}{(\sum_{s'\in\mathcal{S}^*}p^0_{s,a}(s'))^2}\mathbb{E}[\|p^0_{s,a}-\widehat{p}_{2^n}\|_1^2]+8\|Q\|^2_\infty(\mathbb{P}[E^c_{2^n}]+\mathbb{P}[T^c_{2^n}]),\end{aligned}$$

where the last inequality follows a similar argument like Equation (20). By a similar calculation in the proof of Theorem 3.7, we will have $\mathbb{E}[\|p^0_{s,a}-\widehat{p}_{2^{n+1}}\|_1^2]=O(2^{-n})$ and $\mathbb{E}[\|p^0_{s,a}-\widehat{p}_{2^n}\|_1^2]=O(2^{-n})$. We have already known $\mathbb{P}[T^c_{2^n}]=O(e^{-2^n})$ and $\mathbb{P}[E^c_{2^n}]=0$ from the proof of Theorem 3.7. Note that $p_n=\varepsilon(1-\varepsilon)^n$ for $\varepsilon\in(0,0.5)$, we have

$$\sum_{n=0}^\infty\frac{\mathbb{E}[(\Delta^q_{n,\delta})^2]}{p_n}=\|Q\|^2_\infty K_3'(s,a,\varepsilon,\delta,\mathcal{P}^0)<\infty.$$

Now we can find a uniform bound $K_3(\delta,\varepsilon,\mathcal{P}^0)>0$ such that

$$\begin{aligned}\mathbb{E}[(\widehat{T}^{\rm rob}_\delta(Q)(s,a)-f(\beta^*))^2]&\le 2\|Q\|^2_\infty+2\sum_{n=0}^\infty\frac{\mathbb{E}[(\Delta^q_{n,\delta})^2]}{p_n}\\&\le(2+2K_3'(s,a,\varepsilon,\delta,\mathcal{P}^0))\|Q\|^2_\infty\\&\le K_3(\delta,\varepsilon,\mathcal{P}^0)(1+\|Q\|^2_\infty),\end{aligned}$$

Thues we can see

$$\begin{aligned}\mathsf{Var}[\widehat{\mathcal{T}}^{\rm rob}_{\delta,\varepsilon}(Q)(s,a)]&\le 2\max(K_1(\delta,\varepsilon,\nu^0),K_2(\delta,\varepsilon,\nu^0))+2\gamma^2K_3(\delta,\varepsilon,\mathcal{P}^0)(1+\|Q\|^2_\infty)\\&\le C(\delta,\varepsilon,\nu^0,\mathcal{P}^0)(1+\|Q\|^2_\infty),\end{aligned}$$

where $C(\delta,\varepsilon,\nu^0,\mathcal{P}^0)>0$ is some uniform constant. Hence we complete the proof. □