How Does Return Distribution in Distributional Reinforcement Learning Help Optimization?

Ke Sun, Bei Jiang, Linglong Kong[*]
Department of Mathematical and Statistical Sciences
University of Alberta
Edmonton, Canada
{ksun6,bei1,lkong}@ualberta.ca

Abstract

Distributional reinforcement learning, which focuses on learning the entire return distribution instead of only its expectation in standard RL, has demonstrated remarkable success in enhancing performance. Despite these advancements, our comprehension of how the return distribution within distributional RL still remains limited. In this study, we investigate the optimization advantages of distributional RL by utilizing its extra return distribution knowledge over classical RL within the Neural Fitted Z-Iteration (Neural FZI) framework. To begin with, we demonstrate that the distribution loss of distributional RL has desirable smoothness characteristics and hence enjoys stable gradients, which is in line with its tendency to promote optimization stability. Furthermore, the acceleration effect of distributional RL is revealed by decomposing the return distribution. It shows that distributional RL can perform favorably if the return distribution approximation is appropriate, measured by the variance of gradient estimates in each environment. Rigorous experiments validate the stable optimization behaviors of distributional RL and its acceleration effects compared to classical RL. Our research findings illuminate how the return distribution in distributional RL algorithms helps the optimization.

1 Introduction

Distributional reinforcement learning [3, 8, 7, 35, 25, 17, 30] characterizes the intrinsic randomness of returns within the framework of Reinforcement Learning (RL). When the agent interacts with the environment, the intrinsic uncertainty of the environment seeps into the stochasticity of rewards the agent receives and the inherently chaotic state and action dynamics of physical interaction, increasing the difficulty of the RL algorithm design. Distributional RL aims to represent the entire distribution of returns to capture more intrinsic uncertainty of the environment and, therefore, to use these return distributions to evaluate and optimize the policy. This is in stark contrast to the classical RL that only focuses on the expectation of the return distributions, such as temporal-difference (TD) learning [31] and Q-learning [33].

Despite the remarkable empirical success of distributional RL, the illumination of its theoretical advantages still needs to be studied. A distributional regularization effect [29] stemming from the additional return distribution knowledge has been characterized to explain the superiority of distributional RL over classical RL, but the benefit of the proposed regularization on the optimization of algorithms has not been further investigated. Such a gap inspires us to investigate the optimization impact of distributional RL by leveraging the full return distribution knowledge. However, existing literature [22, 28] that helps to analyze the optimization of RL learning may not apply to practical distributional RL algorithms as there still remains a gap between the theory and practice in RL.

---

[*]Corresponding Author

In this paper, we study the optimization advantages of distributional RL over classical RL. Within the Neural FZI framework, our optimization analysis can not only sufficiently characterize offline distributional RL behaviors but also approximate the online setting. Within this framework, we study the uniform stability of distributional loss based on categorical parameterization. Owing to the smoothness properties of distributional loss, distributional RL algorithms tend to satisfy the uniform stability in the optimization process, thus enjoying stable gradient behaviors in the input space. In addition to the optimization stability, we also elaborate on the acceleration effect of distributional RL algorithms based on the return density decomposition technique proposed recently. Distributional RL can speed up the convergence and perform favorably if the return distribution is approximated appropriately, measured by the gradient estimates' variance. Empirical results corroborate that distributional RL possesses stable gradient behaviors and acceleration effects by suggesting smaller gradient norms concerning the states and model parameters. Our study opens up many exciting research pathways in this domain through the lens of optimization, paving the way for future investigations to reveal more advantages of distributional RL. Our contributions can be summarized as follows:

- We specifically study the optimization advantage of practical distributional RL algorithms with the general function approximators. Within the Neural FZI framework, we can analyze the optimization properties of distributional RL by establishing its connection with supervised learning.

- We reveal the uniform stability of distributional RL thanks to the smoothness properties of distributional loss. By contrast, classical RL may not guarantee such a stable optimization property due to the sensitivity of the least squared loss.

- The acceleration effects of distributional RL have also been demonstrated through the return density decomposition. We show that distributional RL can speed up convergence if the parameterization error of the return distribution is appropriate.

2   Related Work

Interpretation of distributional RL. Interpreting the behavior difference between distributional and classical RL was initially studied using the coupled updates method in [18]. They conclude that both distributional and classical RL behave the same in the tabular and linear approximation settings and attribute the superiority of distributional RL to its non-linear approximation. However, the coupled methodology mainly investigated preserving the expectation of return distribution to measure the behavior differences, which rules out other factors, including the optimization effect due to the distributional loss [15]. An implicit risk-sensitive entropy regularization was then revealed in distributional RL by [29], without further analyzing its optimization benefits. Our work complements and extends their results through the lens of optimization.

Convergence and Acceleration in RL. Existing optimization analysis in RL is mainly based on the policy gradient framework. [22] shows that the policy gradient with a softmax parameterization converges at a $\mathcal{O}(1/t)$ rate, which significantly expands the existing asymptotic convergence results. Entropy regularization [12, 13] has gained increasing attention and [2] provides a fine-grained understanding of the impact of entropy on policy optimization and emphasizes that any strategy, such as entropy regularization, can only affect learning in one of two ways: either it reduces the noise in the gradient estimates or it changes the optimization landscape. The seemingly applicable analysis framework on value-based RL is PAC-MDP [28], which effectively analyzes the convergence of typical RL algorithms in the tabular setting. However, it is unclear whether this analysis applies to practical distributional RL algorithms. By contrast, our optimization is within a more interpretable Neural FZI framework and focuses on accelerating the distributional RL algorithm.

Stable Optimization. Stable optimization is one of the crucial properties for RL algorithms, and common strategies include Batch Normalization [27], Spectral Normalization [23], gradient penalty [11]. In RL, stable optimization techniques [10, 16] also benefit the training and the final performance. By contrast, we show that (categorical) distributional RL naturally enjoys stable optimization compared with classical RL.

3 Preliminary Knowledge

Classical RL. In a standard RL setting, the interaction between an agent and the environment is modeled as a Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, R, P, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote state and action spaces. P is the transition kernel dynamics, R is the reward measure and $\gamma \in (0, 1)$ is the discount factor. For a fixed policy π, the return, $Z^\pi = \sum_{t=0}^{\infty} \gamma^t R_t$, is a random variable representing the sum of discounted rewards observed along one trajectory of states while following the policy π. Classical RL focuses on the value and action-value functions, the expectation of returns Z^π. The action-value function $Q^\pi(s, a)$ is defined as $Q^\pi(s, a) = \mathbb{E}\left[Z^\pi(s, a)\right] = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$, where $s_0 = s$, $a_0 = a$, $s_{t+1} \sim P(\cdot|s_t, a_t)$, and $a_t \sim \pi(\cdot|s_t)$.

Distributional RL. Distributional RL, on the other hand, focuses on the action-state return distribution, the entire distribution of $Z^\pi(s, a)$ rather than only its expectation, i.e., $Q^\pi(s, a)$. Leveraging knowledge on the entire return distribution can better capture the uncertainty of returns and thus can be advantageous to explore the intrinsic uncertainty of the environment [7, 21]. Therefore, the scalar-based classical Bellman update is extended to the distributional Bellman update, which allows a flurry of distributional RL algorithms.

Categorical Distributional RL (CDRL). As the first successful distributional RL family, CDRL [3] approximates the action-state return distribution η by a categorical distribution $\hat{\eta} = \sum_{i=1}^{k} f_i \delta_{l_i}$ where $l_1, l_2, ..., l_k$ is a set of fixed supports and $\{f_i\}_{i=1}^{k}$ are learnable probabilities, normally parameterized by a neural network. A projection is also introduced to have the joint support with newly distributed target probabilities, equipped by a KL divergence to compute the distribution distance between the current and target return distribution within each Bellman update. In practice, C51 [3], an instance of CDRL with $k = 51$, performs favorably in various environments.

4 Optimization Analysis

Under Neural FZI established in Section 4.1, we analyze two optimization aspects of distributional RL based on the categorical parameterization, including the stable optimization from the loss function in Section 4.2, and its acceleration effect determined by the gradient estimate variance in Section 4.3.

4.1 Optimization Analysis for Distributional RL within Neural Fitted Z-Iteration

In classical RL, *Neural Fitted Q-Iteration* (Neural FQI) [9, 26] provides a statistical interpretation of DQN [24], capturing its two key features, i.e., the leverage of target network and experience replay:

$$Q_\theta^{k+1} = \operatorname*{argmin}_{Q_\theta} \frac{1}{n} \sum_{i=1}^{n} \left[y_i - Q_\theta^k(s_i, a_i)\right]^2, \tag{1}$$

where the target $y_i = r(s_i, a_i) + \gamma \max_{a \in \mathcal{A}} Q_{\theta^*}^k(s_i', a)$ is fixed within every T_{target} steps to update target network Q_{θ^*} by letting $\theta^* = \theta$. The experience buffer induces independent samples $\{(s_i, a_i, r_i, s_i')\}_{i \in [n]}$ and ideally without the optimization and TD approximation errors, Neural FQI is exactly the update under Bellman optimality operator [9]. Similarly, [29, 20] proposed *Neural Fitted Z-Iteration* (Neural FZI), a distributional version of Neural FQI based on the parameterization of Z_θ to characterize distributional RL:

$$Z_\theta^{k+1} = \operatorname*{argmin}_{Z_\theta} \frac{1}{n} \sum_{i=1}^{n} d_p(Y_i, Z_\theta^k(s_i, a_i)), \tag{2}$$

where the target $Y_i = R(s_i, a_i) + \gamma Z_{\theta^*}^k(s_i', \pi_Z(s_i'))$ is a random variable, whose distribution is also fixed within every T_{target} steps. The target follows a greedy policy rule, where $\pi_Z(s_i') = \operatorname{argmax}_{a'} \mathbb{E}\left[Z_{\theta^*}^k(s_i', a')\right]$ and d_p is the choice of distribution distance.

Approximate Supervised Learning within Neural FZI to Allow the Optimization Analysis. Previous optimization analysis focuses on either policy gradient methods [22, 1] or the sample complexity in the tabular setting [28]. However, there remains some gap between the theory and the practical neural network parameterized RL algorithm, and the previous results may not be directly attainable for the optimization analysis of distributional RL. By contrast, Neural FZI simplifies the

3

optimization problem in deep RL into an approximate iterative supervised learning on a local fixed offline dataset by leveraging experience buffer and target networks, allowing richer optimization analysis. It sufficiently characterizes the offline behaviors of practical distributional RL algorithms and can also approximate online algorithms. In particular, Neural FZI does not consider the exploration; the data distribution shift caused by exploration from an ϵ-greedy policy can be negligible in the online setting, *when the replay memory is sufficiently large or considering the short period.* Thus, the optimization in each phase of Neural FZI can be approximately viewed as supervised learning in contrast to PAC-MDP analysis [28] that explicitly involves the impact of exploration.

Two Key Factors. The Neural FZI framework offers new insights to analyze the optimization benefits for practical distributional RL algorithms, within which there are mainly two crucial components.

- **Factor 1: the choice of** d_p. On the one hand, d_p determines the convergence rate of distributional Bellman update, i.e., the speed of outer iterations in Neural FZI. For instance, distributional Bellman operator under Crámer distance is $\sqrt{\gamma}$-contractive [4], γ-contractive under Wasserstein distance [3]. Moreover, d_p also largely affects the continuous optimization problem concerning parameters θ in Z_θ within each iteration of Neural FZI.
- **Factor 2: the parameterization of** Z_θ. Given the same d_p, a more informative parameterization can approximate the true return distribution more reasonably, promoting the optimization within each phase of Neural FZI. For example, with a more expressiveness power on quantile functions, IQN [7] outperforms QR-DQN [8] on a wider range of environments.

Remark. We mainly attribute the optimization benefit of distributional RL to the choice of distributional loss d_p in Neural FZI relative to the least squared loss in Neural FQI based on the same categorical parameterization on Z_θ, despite the different convergence rates under them.

Categorical Pameterization Equipped with KL Divergence. To allow for theoretical analysis, we resort to the histogram function [32, 15] as the density estimator of Z_θ, a continuous version of categorical parameterization with their equivalent proof provided in [29]. After incorporating the projection to redistribute probabilities of target return distribution by the neighboring smoothing proposed in CDRL, the target, and current histogram function estimators inherit the joint supports, based on which we apply KL divergence as d_p. In particular, we denote the histogram density estimator as $f^{s,a}$ with k uniform partitions on the support, denote $\mathbf{x}(s)$ as the state feature on each state s. We let the support of $Z(s, a)$ be uniformly partitioned into k bins. The output dimension of $f^{s,\cdot}$ can be $|\mathcal{A}| \times k$, where we use the index a to focus on the function $f^{s,a}$. Hence, the function $f^{s,a} : \mathcal{X} \to [0,1]^k$ provides a k-dimensional vector $f^{s,a}(\mathbf{x}(s))$ of the coefficients, indicating the probability that the target is in this bin given the state feature $\mathbf{x}(s)$ and action a. Next, we use *softmax* based on the linear approximation $\mathbf{x}(s)^\top \theta_i$ to express $f^{s,a}$, i.e., $f_i^{s,a,\theta}(\mathbf{x}(s)) = \exp\left(\mathbf{x}(s)^\top \theta_i\right) / \sum_{j=1}^k \exp\left(\mathbf{x}(s)^\top \theta_j\right)$. For simplicity, we use $f_i^\theta(\mathbf{x}(s))$ to replace $f_i^{s,a,\theta}(\mathbf{x}(s))$.

Categorical Distributional Loss. Note that the form of $f^{s,a}$ is similar to that in Softmax policy gradient optimization [22, 31], but we focus on the value-based RL rather than the policy gradient RL. Our prediction probability $f_i^{s,a}$ is redefined as the probability in the i-th bin over the support of $Z(s, a)$, thus eventually serving as a density function. While the linear approximator is limited, this is the setting where, so far, the cleanest results can be firstly achieved, and understanding this setting is necessary for the first step towards bigger problems of understanding distributional RL algorithms. Under this categorical parameterization with KL divergence, the distributional objective function $\mathcal{L}_\theta(s, a)$ for the continuous optimization in each phase of Neural FZI (Eq. 2) can be expressed as:

$$\mathcal{L}_\theta(s, a) = -\sum_{i=1}^k \int_{z_i}^{z_i + w_i} p^{s,a}(y) \log \frac{f_i^\theta(\mathbf{x}(s))}{w_i} dy \propto -\sum_{i=1}^k p_i^{s,a} \log f_i^\theta(\mathbf{x}(s)), \tag{3}$$

where $\theta = \{\theta_1, ..., \theta_k\}$ and $p_i^{s,a}$ is the probability in the i-th bin of the true density function $p^{s,a}(x)$ for $Z(s, a)$ defined in Eq. 6. w_i is the width for the i-th bin $(z_i, z_{i+1}]$. The derivation of the categorical distributional loss under the categorical parameterization is given in Appendix A.

4.2 Stable Optimization Analysis under Uniform Stability

Optimization Properties. Our stable optimization conclusions are based on the smoothness properties of categorical distributional loss in Eq. 3. A similar histogram loss was also analyzed in

[15] along with a local Lipschitz constant analysis. By contrast, in Proposition 1, we extend their optimization results and further establish its connection with distributional RL.

Proposition 1. *(Properties of Categorical Distributional Loss) Assume the state features $\|\mathbf{x}(s)\|_2 \leq l$ for each state s, then $\mathcal{L}_\theta$ is kl-Lipschitz continuous, kl^2-smooth and convex w.r.t. the parameter θ.*

Please refer to Appendix B for the proof. The smoothness properties of categorical distributional loss d_p are the foundation for the stable optimization of distributional RL. In stark contrast, classical RL optimizes a least squared loss function [31] in Neural FQI. It is known that the least squared estimator has no bounded Lipschitz constant in general and is only $\lambda_{\max}$-smooth, where $\lambda_{\max}$ is the largest singular value of the data matrix. Specifically, we have $\|\nabla_\theta \mathcal{L}_\theta\| \leq kl$ for the categorical distributional loss in distributional RL. By contrast, the gradient norm in classical RL is $|y_i - Q_\theta^k(s,a)|\|\mathbf{x}(s)\|$, where $Q_\theta^k(s,a) = \sum_{i=1}^k (z_i + z_{i+1}) f_i^\theta(\mathbf{x}(s))/2w_i$ under the same categorical parameterization for a fair comparison. Clearly, $Q_\theta^k(s,a)$ can be sufficiently large if the support $[z_0, z_k]$ is specified to be large, which is common in environments with a high level of expected returns [3]. As such, $|y_i - Q_\theta^k(s,a)|$ can vary significantly larger than k and classical RL with the potentially larger upper bound of gradient norms is prone to the instability optimization issue.

Uniform Stability of Distributional RL. As an application of stable analysis in [14], we next show that distributional RL loss can naturally induce a uniform stability property under the desirable smoothness properties in Proposition 1, while classical RL can not. We first recap the definition of uniform stability for an algorithm while running *Stochastic Gradient Descent* (SGD) in Definition 1.

Definition 1. *(Uniform Stability) [14] Consider a loss function $g_w(e)$ parameterized by w encountered on the example e, a randomized algorithm $\mathcal{M}$ is uniformly stable if for all data sets $\mathcal{D}, \mathcal{D}'$ such that $\mathcal{D}, \mathcal{D}'$ differ in at most one example, we have*

$$\sup_e \mathbb{E}_\mathcal{M} \left[g_{\mathcal{M}(\mathcal{D})}(e) - g_{\mathcal{M}(\mathcal{D}')}(e) \right] \leq \epsilon_{stab}. \tag{4}$$

Remark: Rationale of Uniform Stability Analysis. One may be concerned whether the uniform stability analysis is applicable to the RL setting with a gradually varying experience replay buffer. Thanks to the Neural FZI framework, it can be viewed as an approximate supervised learning on a nearly fixed offline dataset $\mathcal{D}$ with each iteration of Neural FZI, as the experiment replay allows nearly independent sampling on a fixed data distribution in a short period when the reply memory is large enough [9]. As such, the loss difference by varying the dataset for at most one sample can serve as a surrogate to measure the uniform stability for an algorithm in each phase of Neural FZI.

Theorem 1. *(Uniform Stability for Distributional RL) Suppose that we run SGD under $\mathcal{L}_\theta$ in Eq. 3 with step sizes $\lambda_t \leq 2/kl^2$ for T steps. Assume $\|\mathbf{x}(s)\| \leq l$ for each state s and action a, then we have $\mathcal{L}_\theta$ satisfies the uniform stability in Definition 1 with $\epsilon_{stab} \leq \frac{4kT}{n}$, i.e.,*

$$\mathbb{E}\left| \mathcal{L}_{\theta_T}(s,a) - \mathcal{L}_{\theta_T'}(s,a) \right| \leq \frac{4kT}{n}, \tag{5}$$

where θ_T and θ_T' are the minimizers after T steps under the dataset $\mathcal{D}$ and $\mathcal{D}'$, respectively.

Please refer to the proof of Theorem 1 in Appendix C. Theorem 1 shows that while running SGD to solve the categorical distributional loss within each Neural FZI, the continuous optimization process in each iteration is ϵ_{stab}-uniformly stable with the stability errors shrinking at the rate of $O(n^{-1})$. The stable optimization has multiple advantages, including ϵ_{stab}-bounded generalization gap, a desirable local minimum in deep learning optimization literature [14], and improvement in performance in RL [5, 16]. By contrast, classical RL may not yield thestable optimization property without these smooth properties. For example, $\lambda_{\max}$-smooth may be of less help for the optimization given a bad conditional number of the design matrix where $\lambda_{\max}$ could be sufficiently large. Empirically, we validate the stable gradient behaviors, with smaller gradient norms in the input space, of CDRL compared with classical RL, and similar results are also observed in Quantile Regression distributional RL in Section 5.

Remark: Limitations. The potential optimization instability for classical RL can be used to partially explain its inferiority to distributional RL in most environments, although it may not explain why distributional RL could not perform favorably in certain games [6]. We leave the comprehensive explanation as future works.

Remark: Non-linear Categorical Parameterization. Although the stability above optimization conclusions are established on the linear categorical parameterization on Z^π, similar conclusions with a non-linear categorical parameterization can be naturally expected by non-convex optimization techniques proposed in [14]. We empirically validate our theoretical conclusions by directly applying practical neural network parameterized distributional RL algorithms.

4.3 Acceleration Effect of distributional RL

To characterize the acceleration effect of distributional RL, we additionally leverage the recently proposed *return density function decomposition* [29].

Return Density Function Decomposition. To decompose the optimization impact of return distribution into its expectation and the remaining distribution part, we apply the return density function decomposition to decompose the target histogram density function $p^{s,a}$. This decomposition was successfully applied to derive the distributional regularization effect of distributional RL and was rigorously justified in [29]. Based on the categorical parameterization, we denote Δ_E as the interval that $\mathbb{E}\left[Z\pi(s,a)\right]$ falls into, i.e., $\mathbb{E}\left[Z\pi(s,a)\right] \in \Delta_E$, and the categorical parameterized $p^{s,a}(x) = \sum_{i=1}^{N} f_i^\theta \mathbb{1}(x \in \Delta_i)/\Delta$ can be decomposed as

$$p^{s,a}(x) = (1-\epsilon)p_E^{s,a} + \epsilon\mu^{s,a} = (1-\epsilon)\mathbb{1}(x \in \Delta_E)/\Delta + \epsilon\sum_{i=1}^{N} p_i^\mu \mathbb{1}(x \in \Delta_i)/\Delta, \tag{6}$$

where $p^{s,a}$ is decomposed into a single-bin histogram density $p_E^{s,a}$ and an induced one $\widehat{\mu}^{s,a}$ with each bin probability p_i^μ determined by f_i^θ. The pre-specified ϵ measures the impact of remaining distribution $\widehat{\mu}^{s,a}$ *independent of* its expectation $\mathbb{E}\left[Z^\pi(s,a)\right]$. It shows optimizing the first term in Eq. 6 is equivalent to the classical RL loss in Neural FQI [29], which we provide the proof in Appendix G for completeness. Therefore, this decomposition allows us to conduct the acceleration effect of distributional RL loss as opposed to classical RL.

Measuring the Variance of Gradient Estimates. Within Neural FZI, our goal is to minimize $\frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_\theta(s_i, a_i)$. We rewrite $\mathcal{L}_\theta(s,a)$ as $\mathcal{L}_\theta(g^{s,a}, f_\theta^{s,a})$, where the target density function $g^{s,a}$ can be $p^{s,a}$, $\mu^{s,a}$ or $p_E^{s,a}$, and $f^{s,a,\theta}$ is rewritten as $f_\theta^{s,a}$ for conciseness. We denote $G^k(\theta) = \mathbb{E}\left[\mathcal{L}_\theta(p_E^{s,a}, f_\theta^{s,a})\right]$ and use $G(\theta)$ for $G^k(\theta)$ for simplicity. Based on Proposition 1 in Section 4.2, the appealing optimization properties concerning the parameter θ in f_θ still hold for $G(\theta)$. Although $p_E^{s,a}$ is a single-bin density without non-zero joint support as $f_\theta^{s,a}$, thanks to the leverage of target networks, the KL-based $\mathcal{L}_\theta$ would degrade to the cross-entropy loss, on which $\mathcal{L}_\theta$ is still well-defined. As the KL divergence has unbiased gradient estimates, we let the variance of its stochastic gradient over the expectation-related term $p_E^{s,a}$ be bounded, i.e.,

$$\mathbb{E}_{(s,a)\sim\rho^\pi}\left[\|\nabla\mathcal{L}_\theta(p_E^{s,a}, f_\theta^{s,a})) - \nabla G(\theta)\|^2\right] = \sigma^2. \tag{7}$$

Next, following the similar label smoothing analysis in [34], we further characterize the approximation degree of $f_\theta^{s,a}$ to the target return distribution $\mu^{s,a}$ by measuring its variance as $\kappa\sigma^2$:

$$\mathbb{E}_{(s,a)\sim\rho^\pi}\left[\|\nabla\mathcal{L}_\theta(\mu^{s,a}, f_\theta^{s,a})) - \nabla G(\theta)\|^2\right] = \hat{\sigma}^2 := \kappa\sigma^2. \tag{8}$$

Notably, κ can be used to measure the approximation error between $f_\theta^{s,a}$ and $\mu^{s,a}$ and we do not assume $\hat{\sigma}^2$ to be bounded as κ can be arbitrarily large. This expression $\kappa\sigma^2$ for $\hat{\sigma}^2$ allows us to utilize κ to characterize different acceleration effects for distributional RL given different κ. Concretely, a favorable approximation of $f_\theta^{s,a}$ to $\mu^{s,a}$, which coincides with the role of the Z_θ parameterization, will lead to a small κ, contributing to the acceleration effect of distributional RL as shown in Theorem 2.

Proposition 2. *Based on the return density decomposition in Eq. 6, and Eq. 8, we have:*

$$\mathbb{E}_{(s,a)\sim\rho^\pi}\left[\|\nabla\mathcal{L}_\theta(p^{s,a}, f_\theta^{s,a})) - \nabla G(\theta)\|^2\right] \leq (1-\epsilon)^2\sigma^2 + \epsilon^2\kappa\sigma^2. \tag{9}$$

Proposition 2 reveals the upper bound of gradient estimate variance for the whole target density function $p^{s,a}$, with proof in Appendix D. Before comparing the sample complexity in optimizing both classical and distributional RL, we define the first-order τ-stationary point.

Definition 2. *(First-order τ-Stationary Point) When $\min_\theta G(\theta)$, the updated parameters θ_T after T steps is a first-order τ-stationary point if $\|\nabla_\theta G(\theta_T)\| \leq \tau$.*

Based on Definition 2, we formally characterize the acceleration effects for distributional RL in Theorem 2 that depends upon approximation errors between $\mu^{s,a}$ and $f_\theta^{s,a}$ measured by κ.

Theorem 2. *(Sample Complexity and Acceleration Effects of Distributional RL) While running SGD to minimize $\mathcal{L}_\theta$ in Eq. 6 within Neural FZI, we assume the step size $\lambda \leq \frac{1}{kl^2} \min\{1, \frac{\tau^2}{2\sigma^2}\}$, $\epsilon = 1/(1+\kappa)$, and the sample is uniformly drawn from T samples. Denote $G(\theta_0)$ as initialization.*

*(1) (**Classical RL**) The sample complexity $T = \frac{4G(\theta_0)}{\lambda\tau^2} = O(\frac{1}{\tau^4})$ when minimizing $\mathcal{L}_\theta(p_E^{s,a}, f_\theta^{s,a})$, such that $\mathcal{L}_\theta$ converges to a τ-stationary point in expectation.*

*(2) (**Distributional RL**) The sample complexity $T = O(\frac{1}{\tau^2})$ when minimizing $\mathcal{L}_\theta(p^{s,a}, f_\theta^{s,a})$, such that $\mathcal{L}_\theta$ converges to a $\max\{\tau, 2\sigma\kappa\}$-stationary point in expectation.*

The proof is provided in Appendix E. Theorem 2 is inspired by the intuitive connection between the return distribution in distributional RL and the label distribution in label smoothing [34].

Interpretation of Theorem 2. Theorem 2 demonstrates that optimizing the categorical distributional loss of distributional RL can speed up the convergence with the sample complexity from $O(\frac{1}{\tau^4})$ to $O(\frac{1}{\tau^2})$, if the distribution approximation error is favorable. In particular, when the agnostic κ determined by the environment satisfies $2\kappa\sigma \leq \tau$, the distributional RL algorithm has an effective return distribution parameterization for Z_θ with a smaller approximation error between $f_\theta^{s,a}$ and $\mu^{s,a}$ ($p^{s,a}$). In this case, the acceleration effect of distributional RL over classical RL can be guaranteed. However, it is not vice versa. When $2\kappa\sigma > \tau$, it is unclear whether the required sample complexity for distributional RL is higher than classical RL, as classical RL will require a lower sample complexity than $O(\frac{1}{\tau^4})$ to achieve a $2\kappa\sigma$-stationary point in this case. These theoretical results also coincide with past empirical observations [8, 6], where distributional RL algorithms outperform classical RL in most cases, but are inferior in certain environments. Based on our results in Theorem 2, we contend that these certain environments have much intrinsic uncertainty, the distribution parameterization error between Z_θ and the true return distribution under the distributional TD approximation is still too large ($\kappa > \frac{\tau}{2\sigma}$) to guarantee an acceleration effect as revealed in Theorem 2.

Smaller Gradient Norms in the Weight Space. The acceleration effect of distributional RL in Theorem 2 also implies that distributional RL tends to have smaller gradient norms concerning parameters than classical RL at the same training step, according to the definition of Lipschitz constant in terms of the first-order stationary point. The small gradient norms we analyze here are *in the weight space*, commonly used and directly linked with the convergence rate analysis. In contrast, the uniform stability analyzed in Section 4.2 is defined on the bounded loss difference that is strongly correlated to the gradient norms *in the input space*. Similar works include Spectral Normalization to stabilize the training of Generative Adversarial Networks [23] and RL [10], which normalizes the
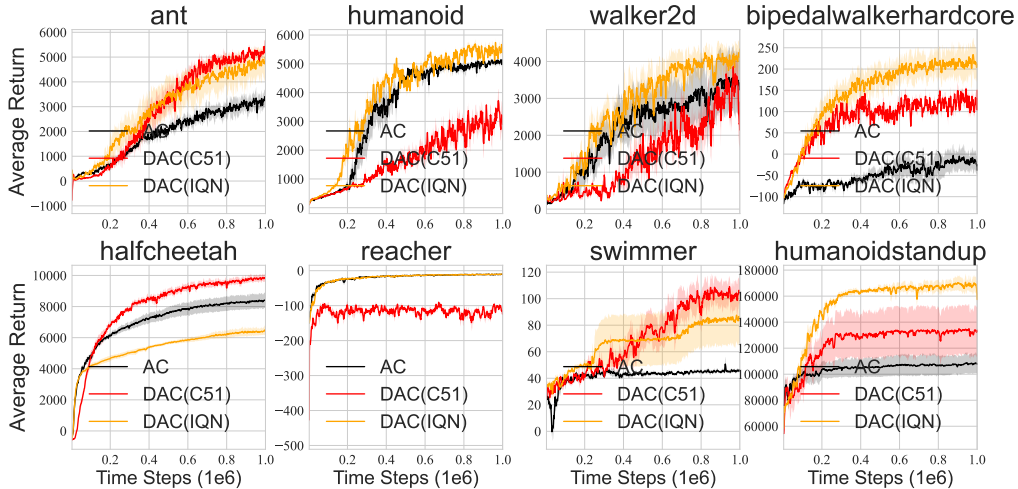

Figure 1: **Performance.** Learning curve of AC, DAC (C51), and DAC (IQN) over five seeds with smooth size five across eight MuJoCo games.

spectral norm of the weight matrix in each layer to lead to a one-valued Lipschitz constant concerning the input. We empirically demonstrate both of them in Section 5.

5 Experiments

Our experiments focus on the online distributional RL algorithms on continuous control Mujoco environments to demonstrate their stable gradient behaviors and acceleration effects.

Implementation. Our implementation is based Soft Actor Critic (SAC) [13] and distributional Soft Actor Critic [19]. We eliminate the optimization impact of entropy regularization in these algorithm implementations, and thus, we denote the resulting algorithms as Actor Critic (AC) and Distributional Actor Critic (DAC) for conciseness. For DAC, we first perform a categorical parameterized C51 critic loss from the classical least-squared critic loss dubbed DAC (C51), which coincides with our theoretical analysis in Sections 4.2 and 4.3. We further apply our experiments on Quantile Regression distributional RL, i.e., Implicit Quantile Network (IQN), denoted as DAC (IQN), to heuristically extend our conclusion in broader algorithm classes. More implementation details are provided in Appendix F.

5.1 Performance and Uniform Stability

Figure 1 suggests both DAC (IQN) and DAC (C51) excel at the classical RL counterpart, i.e., AC (black lines), in most environments, which allows our further optimization analysis.

Proxy: Gradient Norms in the Input Space. We demonstrate the advantage of uniform optimization stability for distributional RL over classical RL. According to Theorem 1, the stable optimization of distribution loss within Neural FZI is described as a bounded loss difference for a neighboring dataset regarding each state s and action a. In other words, the error bound holds by taking the supreme over each state and action pair. To measure this algorithm stability, while far from perfect, we consider leveraging *the average gradient norms concerning the state feature* $\mathbf{x}(s)$ in the whole optimization process as the proxy. This is because the gradient magnitude in the input space could measure the sensitivity of the loss function regarding each state and action pair.

Results. Figure 2 suggests that both DAC (C51) and DAC (IQN) entail a much smaller gradient norm magnitude than the classical AC (black lines) across all environments, corroborating the advantage of uniform stability for distributional RL over classical RL analyzed in Theorem 1. As analyzed in Section 4.2, this result provides empirical evidence to interpret behaviors of distributional RL.
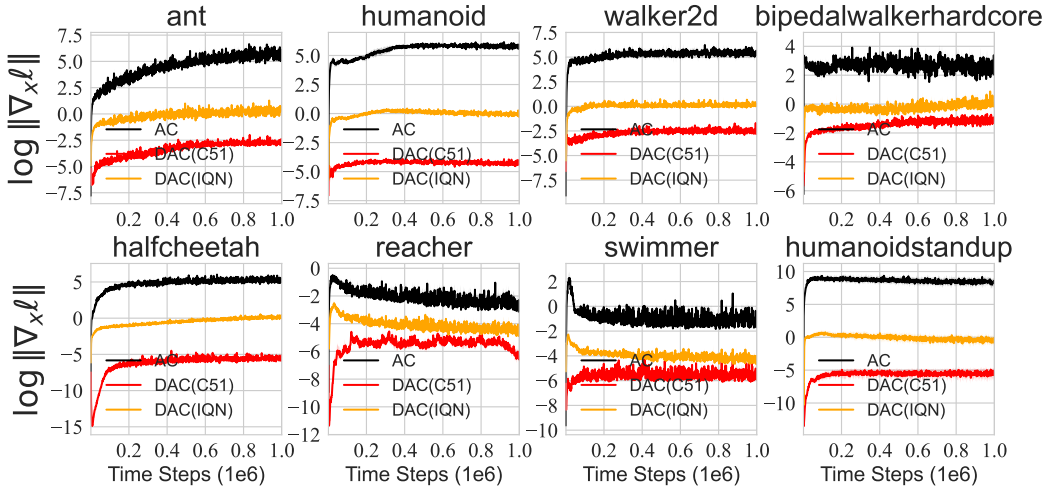

Figure 2: **Uniform Stability.** The critic gradient norms in the logarithmic scale regarding **the state** during the training of AC, DAC (C51), DAC (IQN) over 5 seeds on eight MuJoCo environments.
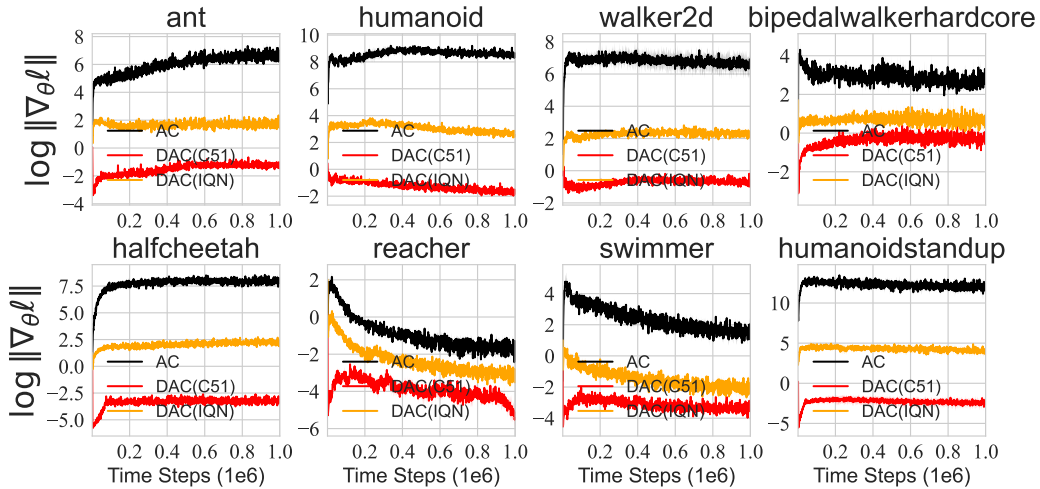
Figure 3: **Acceleration Effect.** The critic gradient norms in the logarithmic scale regarding **network parameters** in the training of AC, DAC (C51), DAC (IQN) over 5 seeds on MuJoCo environments.

5.2 Acceleration Effect of Distributional RL

Proxy: Gradient Norms in the Weight Space. Theorem 2 implies that if the distribution parameterization is appropriate, distributional RL can speed up the convergence and thus can achieve better first-order stationary point, corresponding to smaller gradient norms given the time step in the learning process. To demonstrate it, we take the same step size for both DAC and AC, and evaluate the ℓ_2-norms of gradients *concerning network parameters* of their critics. A direct comparison between vanilla AC and DAC algorithm is given in Figure 3, despite the slight difference in the network architecture in the last layer. For an *apple-to-apple comparison*, we keep the same DAC architecture while implementing a variant AC by optimizing the expectation of represented return distribution. We also find a similar result in Appendix H.

Results. Figure 3 showcases that both DAC (C51) and DAC (IQN) have smaller gradient norms in terms of network parameters θ compared with AC in the whole optimization process. This result also validates that distributional RL loss tends to enjoy smoothness properties in Proposition 1. Moreover, it turns out that DAC (IQN) tends to have smaller gradient norms than DAC (C51). Given the fact that DAC (IQN) outperforms DAC (C51) in most environments in Figure 1, we hypothesize that DAC (IQN) may have a better acceleration effect than DAC (C51), contributing to explaining its superiority. Moreover, the more expressive parameterization of IQN over C51 is also helpful in interpreting both the acceleration and the improvement in the final performance. Lastly, according to Theorem 2, the access to the agnostic κ can serve as a sufficient condition to discriminate whether a specific distributional RL algorithm can accelerate the training in a given environment. However, a precise evaluation of κ is tricky, which we leave as valuable future work.

6 Conclusion, Limitations and Future Work

In our paper, we answer the question: *how does return distribution in distributional RL help the optimization* from perspectives of the uniform stability and acceleration effect in the optimization. Our conclusions are made within a new Neural FZI framework that connects the optimization results in supervised learning with practical deep RL algorithms.

Limitations and Future Work. Our optimization analysis of distributional RL is based on categorical parameterization, and therefore, some optimization properties, such as uniform stability, may not directly apply to other distributional RL families. The alternative analysis on distributional RL algorithms based on Wasserstein distance is also an integral and valuable complement to our conclusions, which we leave as future work.

9

Acknowledgements

References

[1] Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. Optimality and approximation with policy gradient methods in markov decision processes. In *Conference on Learning Theory*, pages 64–66. PMLR, 2020.

[2] Zafarali Ahmed, Nicolas Le Roux, Mohammad Norouzi, and Dale Schuurmans. Understanding the impact of entropy on policy optimization. In *International conference on machine learning*, pages 151–160. PMLR, 2019.

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

[4] Marc G Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos. The cramer distance as a solution to biased wasserstein gradients. *arXiv preprint arXiv:1705.10743*, 2017.

[5] Johan Bjorck, Carla P Gomes, and Kilian Q Weinberger. Towards deeper deep reinforcement learning. *Advances in neural information processing systems (NeurIPS)*, 2021.

[6] Johan Samir Obando Ceron and Pablo Samuel Castro. Revisiting rainbow: Promoting more insightful and inclusive deep reinforcement learning research. In *International Conference on Machine Learning*, pages 1373–1383. PMLR, 2021.

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

[8] Will Dabney, Mark Rowland, Marc G Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. *Association for the Advancement of Artificial Intelligence (AAAI)*, 2018.

[9] Jianqing Fan, Zhaoran Wang, Yuchen Xie, and Zhuoran Yang. A theoretical analysis of deep q-learning. In *Learning for Dynamics and Control*, pages 486–489. PMLR, 2020.

[10] Florin Gogianu, Tudor Berariu, Mihaela C Rosca, Claudia Clopath, Lucian Busoniu, and Razvan Pascanu. Spectral normalisation for deep reinforcement learning: an optimisation perspective. In *International Conference on Machine Learning*, pages 3734–3744. PMLR, 2021.

[11] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[12] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*, pages 1352–1361. PMLR, 2017.

[13] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

[14] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pages 1225–1234. PMLR, 2016.

[15] Ehsan Imani and Martha White. Improving regression performance with distributional losses. In *International Conference on Machine Learning*, pages 2157–2166. PMLR, 2018.

[16] Alexander Li and Deepak Pathak. Functional regularization for reinforcement learning via learned fourier features. *Advances in Neural Information Processing Systems*, 34, 2021.

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

[18] Clare Lyle, Marc G Bellemare, and Pablo Samuel Castro. A comparative analysis of expected and distributional reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4504–4511, 2019.

[19] Xiaoteng Ma, Li Xia, Zhengyuan Zhou, Jun Yang, and Qianchuan Zhao. Dsac: Distributional soft actor critic for risk-sensitive reinforcement learning. *arXiv preprint arXiv:2004.14547*, 2020.

[20] Yecheng Jason Ma, Dinesh Jayaraman, and Osbert Bastani. Conservative offline distributional reinforcement learning. *arXiv preprint arXiv:2107.06106*, 2021.

[21] Borislav Mavrin, Shangtong Zhang, Hengshuai Yao, Linglong Kong, Kaiwen Wu, and Yaoliang Yu. Distributional reinforcement learning for efficient exploration. *International Conference on Machine Learning (ICML)*, 2019.

[22] Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In *International Conference on Machine Learning*, pages 6820–6829. PMLR, 2020.

[23] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *International Conference on Learning Representations*, 2018.

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

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

[26] Martin Riedmiller. Neural fitted q iteration–first experiences with a data efficient neural reinforcement learning method. In *European conference on machine learning*, pages 317–328. Springer, 2005.

[27] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? *Advances in neural information processing systems*, 31, 2018.

[28] Alexander L Strehl, Lihong Li, and Michael L Littman. Reinforcement learning in finite mdps: Pac analysis. *Journal of Machine Learning Research*, 10(11), 2009.

[29] Ke Sun, Yingnan Zhao, Yi Liu, Shi Enze, Wang Yafei, Yan Xiaodong, Bei Jiang, and Linglong Kong. Interpreting distributional reinforcement learning: A regularization perspective. *arXiv preprint arXiv:2110.03155*, 2021.

[30] Ke Sun, Yingnan Zhao, Yi Liu, Bei Jiang, and Linglong Kong. Distributional reinforcement learning via sinkhorn iterations. *arXiv preprint arXiv:2202.00769*, 2022.

[31] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An Introduction*. MIT press, 2018.

[32] Larry Wasserman. *All of nonparametric statistics*. Springer Science & Business Media, 2006.

[33] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[34] Yi Xu, Yuanhong Xu, Qi Qian, Hao Li, and Rong Jin. Towards understanding label smoothing. *arXiv preprint arXiv:2006.11653*, 2020.

[35] Derek Yang, Li Zhao, Zichuan Lin, Tao Qin, Jiang Bian, and Tie-Yan Liu. Fully parameterized quantile function for distributional reinforcement learning. *Advances in neural information processing systems*, 32:6193–6202, 2019.

A Derivation of Categorical Distributional Loss

We show the derivation details of the Categorical distribution loss starting from KL divergence between p and q_θ. p_i is the cumulative probability increment of target distribution $\{Y_i\}_{i\in[n]}$ within the i-th bin, and q_θ corresponds to a (normalized) histogram, and has density values $\frac{f_i^\theta(\mathbf{x}(s))}{w_i}$ per bin. Thus, we have:

$$
\begin{aligned}
D_{\mathrm{KL}}\left(p^{s,a}, q_\theta^{s,a}\right) &= \int_a^b p^{s,a}(y)\log p^{s,a}(y)dy - \int_a^b p^{s,a}(y)\log q_\theta^{s,a}(y)dy \\
&\propto -\int_a^b p^{s,a}(y)\log q_\theta^{s,a}(y)dy \\
&= -\sum_{i=1}^k \int_{z_i}^{z_i+w_i} p^{s,a}(y)\log \frac{f_i^\theta(\mathbf{x}(s))}{w_i}dy \\
&= -\sum_{i=1}^k \log \frac{f_i^\theta(\mathbf{x}(s))}{w_i}\underbrace{\left(F^{s,a}\left(z_i+w_i\right) - F^{s,a}\left(z_i\right)\right)}_{p_i^{s,a}} \\
&\propto -\sum_{i=1}^k p_i^{s,a}\log f_i^\theta(\mathbf{x}(s))
\end{aligned}
\tag{10}
$$

where the first $\propto$ results from the fixed target $p^{s,a}$ in the Neural FZI framework. The second equality is based on the categorical parameterization for the density function $q_\theta^{s,a}$. The last $\propto$ holds because the width parameter w_i can be ignored for this minimization problem.

B Proof of Proposition 1

Proof. For the Categorical distributional loss below,

$$
\mathcal{L}_\theta(s,a) = -\sum_{i=1}^k p_i^{s,a}\log f_i^\theta(\mathbf{x}(s)), \text{ where } f_i^\theta(\mathbf{x}(s)) = \frac{\exp\left(\mathbf{x}(s)^\top \theta_i\right)}{\sum_{j=1}^k \exp\left(\mathbf{x}(s)^\top \theta_j\right)}
$$

(1) Convexity. Note that $-\log \frac{\exp\left(\mathbf{x}(s)^\top \theta_i\right)}{\sum_{j=1}^k \exp\left(\mathbf{x}(s)^\top \theta_j\right)} = \log\sum_{j=1}^k \exp\left(\mathbf{x}(s)^\top \theta_j\right) - \mathbf{x}(s)^\top \theta_i$, the first term is Log-sum-exp, which is convex (see Convex optimization by Boyd and Vandenberghe), and the second term is affine function. Thus, $\mathcal{L}_\theta(s,a)$ is convex.

(2) $\mathcal{L}_\theta(s,a)$ is kl-Lipschitz continuous. We compute the gradient of the Histogram distributional loss regarding θ_i:

$$
\begin{aligned}
\frac{\partial}{\partial \theta_i}\sum_{j=1}^k p_j^{s,a}\log f_j^\theta(\mathbf{x}(s)) &= \sum_{j=1}^k p_j^{s,a}\frac{1}{f_j^\theta(\mathbf{x}(s))}\nabla_{\theta_i} f_j^\theta(\mathbf{x}(s)) \\
&= \sum_{j=1}^k p_j^{s,a}\frac{1}{f_j^\theta(\mathbf{x}(s))}f_i^\theta(\mathbf{x}(s))(\delta_{ij} - f_j^\theta(\mathbf{x}(s)))\mathbf{x}(s) \\
&= \left(p_i^{s,a}(1 - f_i^\theta(\mathbf{x}(s))) - \sum_{j\neq i}^k p_j^{s,a} f_i^\theta(\mathbf{x}(s))\right)\mathbf{x}(s) \\
&= \left(p_i^{s,a} - p_i^{s,a} f_i^\theta(\mathbf{x}(s)) - (1 - p_i^{s,a})f_i^\theta(\mathbf{x}(s))\right)\mathbf{x}(s) \\
&= \left(p_i^{s,a} - f_i^\theta(\mathbf{x}(s))\right)\mathbf{x}(s)
\end{aligned}
\tag{11}
$$

where $\delta_{ij} = 1$ if $i = j$, otherwise 0. Then, as we have $\|\mathbf{x}(s)\| \leq l$, we bound the norm of its gradient

$$
\begin{aligned}
\|\frac{\partial}{\partial \theta} \sum_{j=1}^{k} p_j \log f_j^\theta(\mathbf{x}(s))\| &\leq \sum_{i=1}^{k} \|\frac{\partial}{\partial \theta_i} \sum_{j=1}^{k} p_j \log f_j^\theta(\mathbf{x}(s))\| \\
&= \sum_{i=1}^{k} \| \left(p_i^{s,a} - f_i^\theta(\mathbf{x}(s)) \right) \mathbf{x}(s)\| \\
&\leq \sum_{i=1}^{k} |p_i^{s,a} - f_i^\theta(\mathbf{x}(s))|\|\mathbf{x}(s)\| \\
&\leq kl
\end{aligned}
\tag{12}
$$

The last equality satisfies because $|p_i - f_i^\theta(\mathbf{x}(s))|$ is less than 1 and even smaller. Therefore, we obtain that $\mathcal{L}_\theta$ is kl-Lipschitz.

(3) $\mathcal{L}_\theta$ is kl^2-Lipschitz smooth. A lemma is that $\log(1 + \exp(x))$ is $\frac{1}{4}$-smooth as its second-order gradient is bounded by $\frac{1}{4}$, and if $g(w)$ is β-smooth w.r.t. w, then $g(\langle x, w \rangle)$ is $\beta\|x\|^2$-smooth. Based on this knowledge, we firstly focus on the 1-dimensional case of the function $\log f_j^\theta(z)$, where $f_j^\theta(z) = \frac{\exp z_j}{\sum_{i=1}^{k} \exp z_i}$. As we have derived, we know that $\frac{\partial}{\partial \theta_i} \log f_j^\theta(z_j) = \delta_{ij} - f_i^\theta(z_i)$. Then the second-order gradient is $\frac{\partial^2 \log f_j^\theta(z)}{\partial \theta_i \partial \theta_k} = -f_i^\theta(z)(\delta_{ik} - f_k^\theta(z)) = f_i^\theta(z)(f_k^\theta(z) - 1)$ if $i = k$, otherwise $f_i^\theta(z)f_k^\theta(z)$. Clearly, $|\frac{\partial^2 \log f_j^\theta(z)}{\partial \theta_i \partial \theta_k}| \leq 1$, which implies that $\log f_j^\theta(z)$ is 1-smooth. Thus, $\log f_j^\theta(\langle x, \theta_i \rangle)$ is $\|x\|^2$-smooth, or l^2-smooth. Further, $\sum_{j=1}^{k} p_j^{s,a} \log f_j^\theta(\mathbf{x}(s))$ is also l^2-smooth as we have

$$
\begin{aligned}
\|\nabla_{\theta_i} \sum_{j=1}^{k} p_j^{s,a} \log f_j^\mu(\mathbf{x}(s)) &- \nabla_{\theta_i} \sum_{j=1}^{k} p_j^{s,a} \log f_j^\nu(\mathbf{x}(s))\| \\
&\leq \sum_{j=1}^{k} p_j^{s,a} \|\nabla_{\theta_i} \log f_j^\mu(\mathbf{x}(s)) - \nabla_{\theta_i} \log f_j^\nu(\mathbf{x}(s))\| \\
&\leq \sum_{j=1}^{k} p_j^{s,a} \cdot l^2 \|\mu - \nu\| \\
&= l^2 \|\mu - \nu\|
\end{aligned}
\tag{13}
$$

for each parameter μ and ν. Therefore, we further have

$$
\begin{aligned}
\|\nabla_\theta \sum_{j=1}^{k} p_j^{s,a} \log f_j^\mu(\mathbf{x}(s)) &- \nabla_\theta \sum_{j=1}^{k} p_j^{s,a} \log f_j^\nu(\mathbf{x}(s))\| \\
&\leq \sum_{i=1}^{k} \|\nabla_{\theta_i} \sum_{j=1}^{k} p_j^{s,a} \log f_j^\mu(\mathbf{x}(s)) - \nabla_{\theta_i} \sum_{j=1}^{k} p_j^{s,a} \log f_j^\nu(\mathbf{x}(s))\| \\
&\leq \sum_{i=1}^{k} l^2 \|\mu - \nu\| \\
&= kl^2 \|\mu - \nu\|
\end{aligned}
\tag{14}
$$

Finally, we conclude that $\mathcal{L}_\theta(s, a)$ is kl^2-smooth.

$\square$

C Proof of Theorem 1

Proof. Consider the stochastic gradient descent rule as $G_{\lambda,\mathcal{L}}(\theta) = \theta - \lambda\nabla_\theta\mathcal{L}_\theta$. Firstly, we provide two definitions about $\mathcal{L}_\theta$ for the following proof.

Definition 3. *(σ-bounded) An update rule is σ-bounded if $\sup_\theta \|\theta - \lambda\nabla_\theta\mathcal{L}_\theta\| \leq \sigma$.*

Definition 4. *(η-expansive) An update rule is η-expansive if $\sup_{v,w} \frac{\|G_{\lambda,\mathcal{L}}(v) - G_{\lambda,\mathcal{L}}(w)\|}{\|u-w\|} \leq \eta$.*

Lemma 1. *(Grow Recursion, Lemma 2.5 [14]) Fix an arbitrary sequence of updates $G_1, ..., G_T$ and another sequence $G'_1, ..., G'_T$. Let $\theta_0 = \theta'_0$ be the starting point and define $\delta_t = \|\theta'_i - \theta_t\|$, where θ_t and θ'_t are defined recursively through*

$$\theta_{t+1} = G_{\lambda,\mathcal{L}}(\theta_t), \ \theta'_{t+1} = G'_{\lambda,\mathcal{L}}(\theta'_t)$$

Then we have the recurrence relation:

$$\delta_{t+1} \leq \begin{cases} \eta\delta_t & G_t = G'_t \text{ is } \eta\text{-expansive} \\ \min(\eta, 1)\delta_t + 2\sigma_t & G_t \text{ and } G'_t \text{ are } \sigma\text{-bounded}, G_t \text{ is } \eta \text{ expansive} \end{cases}$$

Lemma 2. *(Lipschitz Continuity) Assume $\mathcal{L}_\theta$ is L-Lipschitz, the gradient update $G_{\lambda,\mathcal{L}}$ is (λL)-bounded.*

Proof. $\|\theta - G_{\lambda,\mathcal{L}}(\theta)\| = \|\lambda\nabla_\theta\mathcal{L}_\theta\| \leq \lambda L$ $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ □

Lemma 3. *(Lipschitz Smoothness and Convex) Assume $\mathcal{L}_\theta$ is β-smooth and convex, then for any $\lambda \leq \frac{2}{\beta}$, the gradient update $G_{\lambda,\mathcal{L}}$ is 1-expansive.*

Proof. Please refer to Lemma 3.7 in [14] for the proof. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ □

Based on all the results above, we start to prove Theorem 1. Our proof is largely based on [14], but it is applicable in distributional RL settings and considering desirable properties of histogram distributional loss. According to Proposition 1, we attain that $\mathcal{L}_\theta$ is kl-Lipschitz as well as kl^2-smooth, and thus based on Lemma 2 and Lemma 3, we have $G_{\lambda,\mathcal{L}}$ is (λkl)-bounded, and 1-expansive if $\lambda \leq \frac{2}{kl^2}$. In the step t, SGD selects samples that are both in $\mathcal{D}$ and $\mathcal{D}'$, with probability $1 - \frac{1}{n}$. In this case, $G_t = G'_t$, and thus $\delta_{t+1} \leq \delta_t$ as G_t is 1-expansive based on Lemma 1. The other case is that samples selected are different with probability $\frac{1}{n}$, where $\delta_{t+1} \leq \delta_t + 2\lambda_t kl$ based on Lemma 1. Thus, if $\lambda_t \leq \frac{2}{kl^2}$, for each state s and action a, we have:

$$\mathbb{E}\left|\mathcal{L}_{\theta_T}(s,a) - \mathcal{L}_{\theta'_T}(s,a)\right| \leq kl\mathbb{E}\left[\delta_T\right], \text{ where } \delta_T = \|\theta_T - \theta'_T\|$$

$$\leq kl\left((1 - \frac{1}{n})\mathbb{E}\left[\delta_{T-1}\right] + \frac{1}{n}\mathbb{E}\left[\delta_{T-1}\right] + \frac{2\lambda_{T-1}kl}{n}\right)$$

$$= kl\left(\mathbb{E}\left[\delta_{T-1}\right] + \frac{2\lambda_{T-1}kl}{n}\right)$$

$$= kl\left(\mathbb{E}\left[\delta_0\right] + \sum_{t=0}^{T-1}\frac{2\lambda_t kl}{n}\right) \qquad (15)$$

$$\leq \frac{2k^2l^2}{n}\sum_{t=0}^{T-1}\frac{2}{kl^2}$$

$$= \frac{4kT}{n}$$

Since this bound holds for all $\mathcal{D}$, $\mathcal{D}'$ and s, a, we attain the uniform stability in Definition 1 for our categorical distributional loss applied in distributional RL.

Define the population risk as:

$$R[\theta] = \mathbb{E}_x\mathcal{L}_\theta(s,a)$$

and the empirical risk as:

$$R_S[\theta] = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_\theta(s_i, a_i)$$

According to Theorem 2.2 in [14], if an algorithm $\mathcal{M}$ is ϵ_{stab}-uniformly stable, then the generalization gap is ϵ_{stab}-bounded, i.e.,

$$|\mathbb{E}_{S,A}\left[R_S[\mathcal{M}(\mathcal{D})] - R[\mathcal{M}(\mathcal{D}')]\right]| \leq \epsilon_{\text{stab}}$$

$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ □

D Proof of Proposition 2

$$\mathbb{E}_{(s,a)\sim\rho^{\pi}}\left[\|\nabla\mathcal{L}_{\theta}(p^{s,a},f_{\theta}^{s,a}))-\nabla G(\theta)\|^{2}\right]\leq(1-\epsilon)^{2}\sigma^{2}+\epsilon^{2}\kappa\sigma^{2}. \tag{16}$$

Proof. As we know that $p^{s,a}(x)=(1-\epsilon)p_{E}^{s,a}+\epsilon\mu^{s,a}(x)$ and we use KL divergence in $\mathcal{L}_{\theta}$, then we have:

$$\nabla\mathcal{L}_{\theta}(p^{s,a},f_{\theta}^{s,a})=(1-\epsilon)\nabla\mathcal{L}_{\theta}(p_{E}^{s,a},f_{\theta}^{s,a})+\epsilon\nabla\mathcal{L}_{\theta}(\mu^{s,a},f_{\theta}^{s,a})$$

Therefore,

$$\begin{aligned}&\mathbb{E}_{(s,a)\sim\rho^{\pi}}\left[\|\nabla\mathcal{L}_{\theta}(p^{s,a},f_{\theta}^{s,a}))-\nabla G(\theta)\|^{2}\right]\\&\leq\mathbb{E}_{(s,a)\sim\rho^{\pi}}\left[(1-\epsilon)^{2}\|\nabla\mathcal{L}_{\theta}(p_{E}^{s,a},f_{\theta}^{s,a}))-\nabla G(\theta)\|^{2}+\epsilon^{2}\|\nabla\mathcal{L}_{\theta}(\mu^{s,a},f_{\theta}^{s,a}))-\nabla G(\theta)\|^{2}\right]\quad(17)\\&=(1-\epsilon)^{2}\sigma^{2}+\epsilon^{2}\kappa\sigma^{2},\end{aligned}$$

where the first inequality uses the triangle inequality of norm, i.e., $\|(1-\epsilon)\mathbf{a}+\epsilon\mathbf{b}\|^{2}\leq(1-\epsilon)^{2}\|\mathbf{a}\|^{2}+\epsilon^{2}\|\mathbf{b}\|^{2}$, and the last equality uses the definition of the variance of $\mathcal{L}_{\theta}(p_{E}^{s,a},f_{\theta}^{s,a})$ and $\mathcal{L}_{\theta}(\mu^{s,a},f_{\theta}^{s,a})$. $\qquad\square$

E Proof of Theorem 2

Proof. **Classical RL** (1) If we only consider the expectation of $Z^{\pi}(s,a)$, we use the information $p_{E}^{s,a}$ to construct the loss function. As $\mathcal{L}_{\theta}(p_{E}^{s,a},q_{\theta}^{s,a})$ is kl^{2}-smooth, we have

$$\begin{aligned}G(\theta_{t+1})-G(\theta_{t})&\leq\langle\nabla G(\theta_{t}),\theta_{t+1}-\theta_{t}\rangle+\frac{kl^{2}}{2}\|\theta_{t+1}-\theta_{t}\|^{2}\\&=-\lambda\langle\nabla G(\theta_{t}),\nabla\mathcal{L}_{\theta}(p_{E}^{s,a},f_{\theta}^{s,a})\rangle+\frac{kl^{2}\lambda^{2}}{2}\|\nabla\mathcal{L}_{\theta}(p_{E}^{s,a},f_{\theta}^{s,a})\|^{2}\end{aligned}\tag{18}$$

where the inequality is according to the definition of Lipschitz-smoothness, and the last equation is based on the updating rule of θ. Next, we take the expectation on both sides,

$$\begin{aligned}&\mathbb{E}\left[G(\theta_{t+1})-G(\theta_{t})\right]\\&\leq-\lambda\mathbb{E}\left[\|\nabla G(\theta_{t})\|^{2}\right]+\frac{kl^{2}\lambda^{2}}{2}\mathbb{E}\left[\|\nabla\mathcal{L}_{\theta}(p_{E}^{s,a},f_{\theta}^{s,a})-\nabla G(\theta_{t})+\nabla G(\theta_{t})\|^{2}\right]\\&\leq-\lambda\mathbb{E}\left[\|\nabla G(\theta_{t})\|^{2}\right]+\frac{kl^{2}\lambda^{2}}{2}\mathbb{E}\left[\|\nabla\mathcal{L}_{\theta}(p_{E}^{s,a},f_{\theta}^{s,a})-\nabla G(\theta_{t})\|^{2}\right]+\frac{kl^{2}\lambda^{2}}{2}\mathbb{E}\left[\|\nabla G(\theta_{t})\|^{2}\right]\quad(19)\\&=\frac{\lambda(kl^{2}\lambda-2)}{2}\mathbb{E}\left[\|\nabla G(\theta_{t})\|^{2}\right]+\frac{kl^{2}\lambda^{2}}{2}\sigma^{2}\\&\leq-\frac{\lambda}{2}\mathbb{E}\left[\|\nabla G(\theta_{t})\|^{2}\right]+\frac{kl^{2}\lambda^{2}}{2}\sigma^{2}\end{aligned}$$

where the first two inequalities hold because $\nabla G(\theta)=\mathbb{E}\left[\nabla\mathcal{L}_{\theta}\right]$ and the last inequality comes from $\lambda\leq\frac{1}{kl^{2}}$. Through the summation, we obtain that

$$\mathbb{E}\left[G(\theta_{T})-G(\theta_{0})\right]\leq-\frac{\lambda}{2}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla G(\theta_{t})\|^{2}\right]+\frac{kl^{2}\lambda^{2}T}{2}\sigma^{2}$$

We let $\mathbb{E}\left[G(\theta_{T})\right]=0$, we have

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla G(\theta_{t})\|^{2}\right]\leq\frac{2G(\theta_{0})}{\lambda T}+kl^{2}\lambda\sigma^{2}$$

By setting $\lambda\leq\frac{\tau^{2}}{2kl^{2}\sigma^{2}}$ (simultaneously $\lambda\leq\frac{1}{kl^{2}}$, i.e., $\lambda\leq\frac{1}{kl^{2}}\min\{1,\frac{\tau^{2}}{2\sigma^{2}}\}$) and $T=\frac{4G(\theta_{0})}{\lambda\tau^{2}}$, we can have $\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla G(\theta_{t})\|^{2}\right]\leq\tau^{2}$, implying that the degenerated loss function based on the expectation $p_{E}^{s,a}$ can achieve τ-stationary point if the sample complexity $T=O(\frac{1}{\tau^{4}})$.

Distributional RL (2). We are still based on the kl^2-smoothness of $\mathcal{L}(p^{s,a}, f_\theta^{s,a})$.

$$G(\theta_{t+1}) - G(\theta_t) \leq \langle \nabla G(\theta_t), \theta_{t+1} - \theta_t \rangle + \frac{kl^2}{2} \|\theta_{t+1} - \theta_t\|^2$$

$$= -\lambda \langle \nabla G(\theta_t), \nabla \mathcal{L}_\theta(p^{s,a}, f_\theta^{s,a}) \rangle + \frac{kl^2\lambda^2}{2} \|\nabla \mathcal{L}_\theta(p^{s,a}, f_\theta^{s,a})\|^2$$

$$= -\frac{\lambda}{2} \|\nabla G(\theta_t)\|^2 + \frac{\lambda}{2} \|\nabla G(\theta_t) - \nabla \mathcal{L}_\theta(p^{s,a}, f_\theta^{s,a})\|^2 + \frac{\lambda(kl^2\lambda - 1)}{2} \|\nabla \mathcal{L}_\theta(p^{s,a}, f_\theta^{s,a})\|^2$$

$$\leq -\frac{\lambda}{2} \|\nabla G(\theta_t)\|^2 + \frac{\lambda}{2} \|\nabla G(\theta_t) - \nabla \mathcal{L}_\theta(p^{s,a}, f_\theta^{s,a})\|^2$$

$$\tag{20}$$

where the second equation is based on $\langle \mathbf{a}, -\mathbf{b} \rangle = \frac{1}{2}\left(\|\mathbf{a} - \mathbf{b}\|^2 - \|\mathbf{a}\|^2 - \|\mathbf{b}\|^2\right)$, and the last inequality is according to $\lambda \leq \frac{1}{kl^2}$. After taking the expectation, we have

$$\mathbb{E}\left[G(\theta_{t+1}) - G(\theta_t)\right] \leq -\frac{\lambda}{2}\mathbb{E}\left[\|\nabla G(\theta_t)\|^2\right] + \frac{\lambda}{2}\mathbb{E}\left[\|\nabla G(\theta_t) - \nabla \mathcal{L}_\theta(p^{s,a}, f_\theta^{s,a})\|^2\right]$$

$$\leq -\frac{\lambda}{2}\mathbb{E}\left[\|\nabla G(\theta_t)\|^2\right] + \frac{\lambda}{2}\left((1 - \epsilon)^2\sigma^2 + \epsilon^2\kappa\sigma^2\right) \tag{21}$$

where the last inequality is based on Proposition 2. We take the summation, and therefore,

$$\mathbb{E}\left[G(\theta_T) - G(\theta_0)\right] \leq -\frac{\lambda}{2}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla G(\theta_t)\|^2\right] + \frac{T\lambda}{2}\left((1 - \epsilon)^2\sigma^2 + \epsilon^2\kappa\sigma^2\right)$$

We let $\mathbb{E}\left[G(\theta_T)\right] = 0$ and $\epsilon = \frac{1}{1+\kappa}$, then,

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla G(\theta_t)\|^2\right] \leq \frac{2G(\theta_0)}{\lambda T} + (1 - \epsilon)^2\sigma^2 + \epsilon^2\kappa\sigma^2$$

$$= \frac{2G(\theta_0)}{\lambda T} + \frac{2\kappa^2}{(1 + \kappa)^2}\sigma^2 \tag{22}$$

$$\leq \frac{2G(\theta_0)}{\lambda T} + 2\kappa^2\sigma^2$$

If $\kappa \leq \frac{\tau}{2\sigma}$ and let $T = \frac{4G(\theta_0)}{\lambda\tau^2}$, this leads to $\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla G(\theta_t)\|^2\right] \leq \tau^2$, i.e., τ-stationary point, with the sample complexity as $O(\frac{1}{\tau^2})$. If $\kappa > \frac{\tau}{2\sigma}$, we set $T = \frac{G(\theta_0)}{\lambda\kappa^2\sigma^2}$. This implies that $\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla G(\theta_t)\|^2\right] \leq 4\kappa^2\sigma^2$, which can only achieve $2\kappa\sigma$-stationary point. Putting two cases together, we conclude that distributional RL can achieve $\max\{\tau, 2\kappa\sigma\}$-stationary point (since τ can be pre-given, while $2\kappa\sigma$ is determined by the environment.) $\qquad\square$

F   Implementation Details

Our implementation is directly adapted from the source code in [19]. For DAC (IQN), we consider the quantile regression for the distribution estimation on the critic loss. Instead of using fixed quantiles in QR-DQN [8], we leverage the quantile fraction generation based on IQN [7] that uniformly samples quantile fractions in order to approximate the full quantile function. In particular, we fix the number of quantile fractions as N and keep them ascending. Besides, we adapt the sampling as $\tau_0 = 0, \tau_i = \epsilon_i / \sum_{i=0}^{N-1}$, where $\epsilon_i \in U[0,1], i = 1, ..., N$.

F.1   Hyper-parameters and Network structure

We adopt the same hyper-parameters listed in Table 1 and network structure as in the original distributional SAC paper [19].

F.2   Best l_k for DAC (C51)

As suggested in Table 1, after a line search for the hyperparameter tuning, we select l_k as 500, 10,000, 15,000, 160, 50, 5,000, 500, 500 for ant, halfcheetah, humanoidstand, swimmer, bipedalwalkerhardcore, humanoid, walker2d and reacher, respectively.

Table 1: Hyper-parameters Sheet.

Hyperparameter	Value
Shared	
Policy network learning rate	3e-4
(Quantile / Categorical) Value network learning rate	3e-4
Optimization	Adam
Discount factor	0.99
Target smoothing	5e-3
Batch size	256
Replay buffer size	1e6
Minimum steps before training	1e4
DAC (IQN)	
Number of quantile fractions (N)	32
Quantile fraction embedding size	64
Huber regression threshold	1
DAC (C51)	
Number of Atoms (k)	51

Hyperparameter	l_k for C51	Max episode lenght
Walker2d-v2	500	1000
Swimmer-v2	160	1000
Reacher-v2	500	1000
Ant-v2	500	1000
HalfCheetah-v2	10,000	1000
Humanoid-v2	5,000	1000
HumanoidStandup-v2	15,000	1000
BipedalWalkerHardcore-v2	50	2000

G   Equivalence between the loss function in Theorem 2 and mean squared loss in Neural FQI

Proposition 3. *(Equivalence between the first term in Decomposed Neural FZI and Neural FQI) In Neural FZI, if the function class $\{Z_\theta : \theta \in \Theta\}$ is sufficiently large such that it contains the target $\{Y_i\}_{i=1}^n$. As $\Delta \to 0$, minimizing **the first term** in implies*

$$P(Z_\theta^{k+1}(s,a) = \mathcal{T}^{opt} Q_{\theta^*}^k(s,a)) = 1, \quad \forall k. \tag{23}$$

Proof. Firstly, we define the distributional Bellman optimality operator $\mathfrak{T}^{\text{opt}}$ as follows:

$$\mathfrak{T}^{\text{opt}} Z(s,a) \stackrel{D}{=} R(s,a) + \gamma Z(S', a^*), S' \sim P(\cdot \mid s,a), \quad a^* = \underset{a'}{\operatorname{argmax}} \mathbb{E}\left[Z(S', a')\right] \tag{24}$$

If $\{Z_\theta : \theta \in \Theta\}$ is sufficiently large enough such that it contains $\mathfrak{T}^{\text{opt}} Z_{\theta^*}$, then optimizing Neural FZI in Eq. 2 leads to $Z_\theta^{k+1} = \mathfrak{T}^{\text{opt}} Z_{\theta^*}$.

We apply the action-value density function decomposition on the target histogram function $\widehat{p}^{s,a}(x)$. Consider the parameterized histogram density function h_θ and denote h_θ^E / Δ as the bin height in the bin Δ_E, under the KL divergence between the first histogram function $\mathbb{1}(x \in \Delta_E)$ with $h_\theta(x)$, the objective function is simplified as

$$D_{\text{KL}}(\mathbb{1}(x \in \Delta_E)/\Delta, h_\theta(x)) \propto -\int_{x \in \Delta_E} \frac{1}{\Delta} \log \frac{h_\theta^E}{\Delta} dx \propto -\log h_\theta^E \tag{25}$$

Since $\{Z_\theta : \theta \in \Theta\}$ is sufficiently large enough, the KL minimizer would be $\widehat{h}_\theta = \mathbb{1}(x \in \Delta_E)/\Delta$ in expectation. Then, $\arg\min_{h_\theta} \lim_{\Delta \to 0} D_{\mathrm{KL}}(\mathbb{1}(x \in \Delta_E)/\Delta, h_\theta(x)) = \delta_{\mathbb{E}[Z^{\mathrm{target}}(s,a)]}$, where $\delta_{\mathbb{E}[Z^{\mathrm{target}}(s,a)]}$ is a Dirac Delta function centered at $\mathbb{E}[Z^{\mathrm{target}}(s,a)]$ and can be viewed as a generalized probability density function. This also applies from h_θ to Z_θ. In Neural FZI, we have $Z^{\mathrm{target}} = \mathfrak{T}^{\mathrm{opt}} Z_{\theta^*}$. According to the definition of the Dirac function, as $\Delta \to 0$, we attain

$$P(Z_\theta^{k+1}(s,a) = \mathbb{E}\left[\mathfrak{T}^{\mathrm{opt}} Z_{\theta^*}^k(s,a)\right]) = 1 \tag{26}$$

Due to the linearity of expectation analyzed in Lemma 4 of [3], we have

$$\mathbb{E}\left[\mathfrak{T}^{\mathrm{opt}} Z_{\theta^*}^k(s,a)\right] = \mathfrak{T}^{\mathrm{opt}} \mathbb{E}\left[Z_{\theta^*}^k(s,a)\right] = \mathcal{T}^{\mathrm{opt}} Q_{\theta^*}^k(s,a) \tag{27}$$

Finally, we obtain:

$$P(Z_\theta^{k+1}(s,a) = \mathcal{T}^{\mathrm{opt}} Q_{\theta^*}^k(s,a)) = 1 \quad \text{as } \Delta \to 0 \tag{28}$$

$\square$

H   Experimental Results on Acceleration Effects of Distributional RL

Same Architecture.   For a fair comparison, we keep the same DAC network architecture and evaluate the gradient norms of DAC (C51) and a variant of AC, which is optimized based on the expectation of the represented value distribution within the DAC implementation framework. Figure 4 suggests DAC (C51) still enjoys smaller gradient norms than AC in this fair comparison setting.
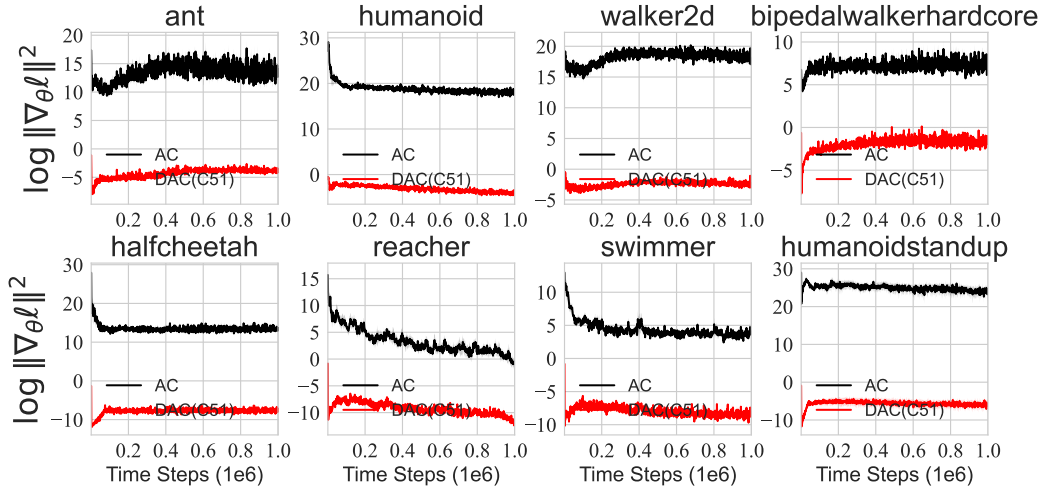

Figure 4: The critic gradient norms in the logarithmic scale during the training of AC and DAC (C51) over five seeds on three MuJoCo games. **We keep the same DAC network architecture and evaluate based on the expectation of the represented value distribution**.

Results under Return Density Decomposition   We also provide gradient norms of both expectation and distribution based on the Return Density Function decomposition in Eq. 6. Similar results can still be observed in Figure 5.
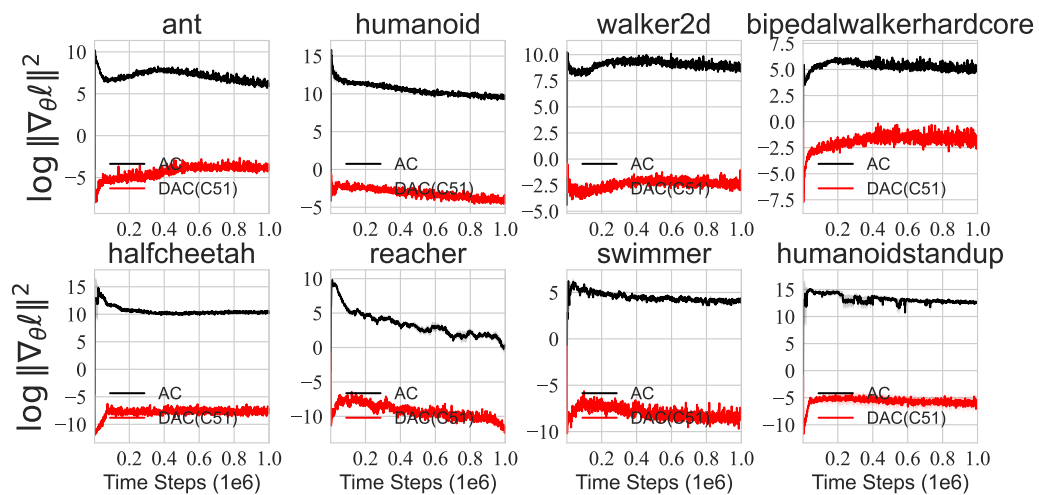
Figure 5: The critic gradient norms in the logarithmic scale during the training of AC and DAC (C51) over five seeds on three MuJoCo games. **Results of AC is the expectation part calculated via the Return Density Function Decomposition**.