OFFLINE-TO-ONLINE DISTRIBUTIONAL REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Distributional reinforcement learning (DRL) has demonstrated strong performance and improved training stability by modeling the full distribution of returns. Despite this progress, we observe that in the offline-to-online setting, DRL suffers severe performance degradation at the onset of online finetuning. We trace this instability to quantile penalization during offline training, which may violate the intrinsic monotonic ordering of quantile values and induce substantially biased value estimates. To address this issue, we propose a monotonic quantile positional embedding and a monotonic quantile value network, which provably preserve quantile ordering under offline penalization. By enforcing this structural property, our method mitigates quantile estimation bias and stabilizes policy transfer from offline pretraining to online fine-tuning. We further introduce a quantile-guided exploration strategy that leverages the learned return distribution as an uncertainty signal to facilitate efficient online exploration. Experiments on the D4RL benchmark demonstrate that our method substantially reduces performance degradation at the onset of online interaction and consistently achieves superior final performance compared with existing offline-to-online reinforcement learning methods.

1 INTRODUCTION

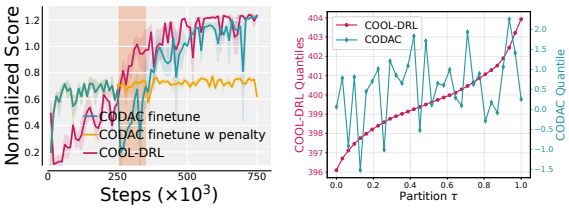
Offline reinforcement learning (RL) enables agents to learn policies from previously collected data without environment interaction (Fujimoto et al., 2019; Levine et al., 2020), making it particularly attractive in real-world applications (Zhou et al., 2023). However, their performance is fundamentally limited by the quality and coverage of the offline data (Jin et al., 2021). Recently, offline-to-online RL, which pretrains policies on the offline dataset and subsequently finetunes them through online interaction (Lee et al., 2021; Nakamoto et al., 2023), has emerged as a promising paradigm for improving final performance in practice. Despite its potential, fully unlocking the benefits of offline-to-online RL remains challenging, due to *instability* during the transfer from offline training to online finetuning, as well as the difficulty of effective *exploration* in the stages of interaction.

To mitigate instability in offline-to-online RL, prior work has explored different methods to reduce Q-value bias during the transition to online interaction. CalQL (Nakamoto et al., 2023) calibrates Q-value conservativeness using Monte Carlo return during offline training. SUF (Feng et al., 2024) prioritizes critic updates via a higher update-to-data ratio, while SO2 (Zhang et al., 2024) smooths ensembled Q-functions with perturbed value updates and increased critic update frequency. WSRL (Zhou et al., 2025) further leverages warm-up strategies to adapt Q-values to online dynamics. Nevertheless, these methods neither exploit Distributional RL (DRL), which has shown improved stability in online settings (Dabney et al., 2018a), nor explicitly incorporate effective exploration.

When considering distributional RL in the offline-to-online setting, we find that offline conservative penalties on quantiles, as employed by methods such as CODAC (Ma et al., 2021), can severely disrupt the intrinsic monotonicity of quantile estimates, posing unique challenges to stable online finetuning. In practice, this results in more severe performance degradation during the early stages of finetuning, as shown in Fig. 1(a). The inverse cumulative distribution function of CODAC's action-value estimates (cyan curve) in Fig. 1(b) clearly violates the required monotonicity and exhibits quantiles that are severely biased toward zero. Such distortions make the estimated value distributions unreliable, impairing action ranking and providing misleading signals for policy updates, ultimately leading to unstable policy improvement and performance degradation during early online finetuning.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To address this challenge, we propose **C**alibrated **O**ffline-to-**O**n**L**ine **D**istributional **R**einforcement **L**earning (**COOL-DRL**). We develop a monotonic quantile position embedding (MQPE) and a monotonic quantile value network (MQVN) to represent the quantile function, providing a theoretical guarantee of quantile monotonicity regardless of offline penalization. This design structurally calibrates value estimation bias, thereby mitigating performance degradation caused by severe disorder quantiles following offline penalty. During the online phase, we propose a quantile-based exploration strategy that exploits the uncertainty inherently encoded in quantiles, enabling more stable and effective finetuning. Overall, our contributions can be summarized as follows: (1) To our best knowledge, this work presents the first study of O2O DRL. (2) We propose MQPE and MQVN, which provide theoretical guarantees of quantile monotonicity and effectively calibrate structural bias. (3) We develop a quantile-based exploration strategy to facilitate effective online fine-tuning. (4) Extensive experiments demonstrate that our method consistently outperforms existing offline-to-online RL methods across a diverse range of tasks.



(a) (b)

Figure 1: (a) Offline-to-online performance (250K offline, 500K online) on walker2d-medium. DRL method CODAC (cyan) suffers a sharp performance drop at the start of online finetuning (orange area). Keeping its conservativeness (yellow) avoids the collapse but restricts final performance improvement. (b) The quantile estimates produced by CODAC (cyan) violate the required monotonic structure and exhibit severe value estimation bias.

2 Preliminaries

RL is typically formulated within the framework of Markov Decision Processes (MDPs) (Sutton & Barto, 2018; Mnih et al., 2015), defined by a tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $\mathcal{S}$ is state space, $\mathcal{A}$ is action space, $P(\cdot \mid s, a)$ is transition probability distribution, $R(\cdot \mid s, a)$ is reward distribution, and $\gamma \in (0, 1)$ is discount factor. At each timestep, the agent observes a state $s \in \mathcal{S}$, selects an action $a \in \mathcal{A}$ via a policy $\pi(\cdot \mid s)$, and transitions to a new state $s' \sim P(\cdot \mid s, a)$ while receiving a reward $r \sim R(\cdot \mid s, a)$. The goal is to learn a policy that maximizes the expected cumulative discounted return $Q^\pi(s, a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s, a_0 = a\right]$, generally called the action-value or Q-value.

2.1 Distributional RL

Traditional RL focuses on learning the expected return $Q^\pi(s, a)$. In contrast, DRL aims to model the full distribution of returns, $Z^\pi(s, a)$, which captures the randomness in cumulative rewards arising from the environment and policy. The distributional Bellman operator $\mathcal{T}^\pi$ is defined as:

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

where $r \sim R(\cdot \mid s, a)$, $s' \sim P(\cdot \mid s, a)$, $a' \sim \pi(\cdot \mid s')$, and $\overset{D}{=}$ implies both sides are equal in distribution. The fixed point of this operator, Z^π, represents the true return distribution. DRL provides richer training signals and enables stable decision-making. In practice, several approaches model the return distribution in distinct ways: C51 (Bellemare et al., 2017) approximates it using a fixed set of discrete probability masses; QR-DQN (Dabney et al., 2018b) learns value estimates at fixed quantile partitions; and IQN (Dabney et al., 2018a) dynamically generates quantile estimates by sampling random quantile fractions, thereby enhancing the expressiveness of the learned distribution.

2.2 Conservative Offline Distributional RL

Offline RL learns policies solely from a fixed dataset $\mathcal{D}$ without environment interaction. A key challenge is avoiding overestimation of out-of-distribution (OOD) action-values. For example, CODAC (Ma et al., 2021) addresses this by extending DRL to the offline setting with a conservative penalty on quantiles. The algorithm minimizes the following objective:

$$\min_Z \Big\{ \alpha \cdot \mathbb{E}_{\tau \sim U} \left[\mathbb{E}_{s \sim \mathcal{D}} \log \sum_a \exp(F_Z^{-1}(\tau)) - \mathbb{E}_{(s,a) \sim \mathcal{D}} F_Z^{-1}(\tau) \right] + \mathcal{L}_\kappa(\delta; \tau) \Big\} \tag{2}$$

where α is the penalty coefficient and U represents $\mathrm{Uniform}([0,1])$ distribution. The quantile function $F_Z^{-1}(\tau)$ is typically modeled by a neural network $F_\theta(\tau; s, a)$ with τ being the quantile partition. δ refers to the distributional temporal difference (TD) error, which in practice is computed as: $\delta = r + \gamma F_{\theta'}(\tau'; s', a') - F_\theta(\tau; s, a)$, where θ' represents the target network parameters. The τ-Huber quantile regression loss is:

$$\mathcal{L}_\kappa(\delta; \tau) = \begin{cases} |\tau - \mathbb{1}(\delta < 0)| \cdot \delta^2/(2\kappa), & |\delta| \le \kappa \\ |\tau - \mathbb{1}(\delta < 0)| \cdot (|\delta| - \kappa/2), & \text{otherwise.} \end{cases} \tag{3}$$

3 METHODOLOGY

In this section, we present COOL-DRL, a concise yet effective framework designed to improve training stability and final performance, establishing the first baseline for offline-to-online DRL. We first propose a monotonic quantile positional embedding and a monotonic quantile value network that provably enforce monotonicity in quantile estimates, thereby calibrating intrinsic value bias independently of offline penalization. Building on these components, we develop an uncertainty-aware exploration strategy that leverages quantile dispersion to guide online fine-tuning.

3.1 MONOTONIC QUANTILE VALUE NETWORK

When estimating the quantile value function $F_\theta(\tau; s, a)$, a desirable property is that the predicted values should be monotonically increasing with respect to the quantile partition τ. However, existing offline DRL methods, such as CODAC, often violate this property in practice. The quantile penalization applied during offline training further amplifies these inconsistencies, introducing severe bias into quantile estimation. As illustrated in Fig. 2, on the halfcheetah-medium task, CODAC's quantile estimates are heavily biased toward 0, while the corresponding Q-value learned by online SAC is approximately 713. Therefore, we propose a monotonicity-provable architecture that enforces intrinsic quantile monotonicity and reduce estimation bias.

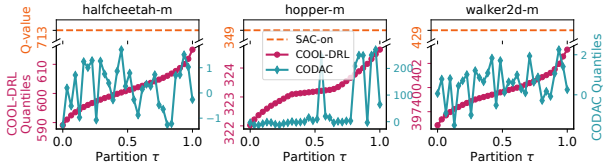


Figure 2: Quantile value estimates after offline training. Each subfigure compares the quantile curves produced by COOL-DRL and CODAC, together with the pure online SAC Q-value used as a reference. CODAC exhibits severely disordered and highly biased quantile estimates toward 0, whereas our method strictly preserves quantile monotonicity and substantially reduces estimation bias.

3.1.1 MONOTONIC QUANTILE POSITIONAL EMBEDDING

Unlike previous DRL methods (Dabney et al., 2018a;b; Bai et al., 2024) that learn the mapping $[\tau_0, \tau_1, \ldots, \tau_{N-1}] \mapsto [F_\theta(\tau_0), F_\theta(\tau_1), \ldots, F_\theta(\tau_{N-1})]$, which suffers from functional inconsistency across different input groups and lacks a global monotonicity guarantee with respect to τ, we directly construct a monotonic mapping $\tau \mapsto F_\theta(\tau)$ to address both issues. We first develop the Monotonic Quantile Positional Embedding (MQPE), which is designed to guarantee the monotonicity of quantile embeddings. Specifically, we propose a structured sinusoidal encoding that explicitly preserves the ordering of quantile indices. Formally, the embedding of a partition $\tau \in [0, 1]$ is defined as

$$\tau_{\text{emb}} = \left[\sin\left(c_0 + \tau \cdot \tfrac{2\pi}{d+1}\right) \ \ldots \ \sin\left(c_{d-1} + \tau \cdot \tfrac{2\pi}{d+1}\right)\right]^\top, \tag{4}$$

where d is the embedding dimension and the phase shift for the k-th dimension is given by

$$c_k = \frac{k\pi}{d+1} - \frac{\pi}{2}, \quad k = 0, 1, \ldots, d-1. \tag{5}$$

By selecting the offsets $\{c_k\}$ and restricting the effective projection domain to $[-\frac{\pi}{2}, \frac{\pi}{2}]$, this construction ensures that each embedding dimension evolves monotonically with τ. After obtaining the

sinusoidal embedding τ_{emb}, we further transform it into a latent representation using an exponentiated weighting scheme. Specifically, we define

$$h_{\text{Exp}}(\tau_{\text{emb}}) = \sigma(\exp(W_1)\tau_{\text{emb}} + b_1)\,, \quad (6)$$

where $W_1 \in \mathbb{R}^{n_1 \times d}$ and $b_1 \in \mathbb{R}^{n_1}$ are weight and bias parameters, resepectively, n_1 denotes the hidden dimension, and the activation function $\sigma(\cdot)$ is an element-wise operator and given by a clipped ReLU:

$$\text{ReLU}_u(x) = \begin{cases} u & \text{if } x \geq u, \\ x & \text{if } 0 \leq x < u, \\ 0 & \text{if } x < 0, \end{cases} \quad (7)$$

with $u > 0$ being a fixed upper bound. This completes the design of our MQPE. The representation faithfully encodes the inherent ordering of quantile levels, providing a monotone foundation for subsequent value estimation.

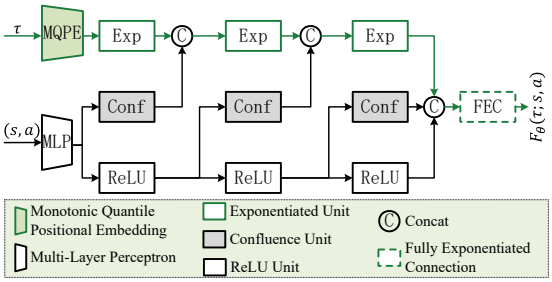


Figure 3: Overall architecture of the MQVN. The quantile partition τ is first passed through the MQPE module, which ensures that the embedded representation preserves quantile monotonicity. The result is further transformed into the exponentiated unit. The state–action pair (s, a) is processed by confluence unit and ReLU unit, where its representation is fused with the embedded τ.

3.1.2 Monotonic Quantile Value Network

We now propose the Monotonic Quantile Value Network (MQVN). Inspired by the scalable monotonic neural networks proposed by Kim & Lee (2024), we redesign the MLP architecture to fuse quantile embeddings with state-action features into a strictly monotonic quantile value network. The overall architecture is illustrated in Fig. 3. Specifically, MQVN is composed of three types of units:

(1) Exponentiated Unit. Consistent with Eq. (6), this unit ensures monotonicity with respect to the quantile embedding during fusion: $h_{\text{Exp}}(x) = \text{ReLU}_u\big(\exp(W)x + b\big)$.

(2) Confluence Unit. This unit transforms the state-action representation before fusion with the quantile embedding: $h_{\text{Conf}}(x) = \text{ReLU}_u\big(Wx + b\big)$.

(3) ReLU Unit. This unit applies a standard MLP transformation with ReLU activation, allowing non-monotonic transformations to enrich the state-action representation: $h_{\text{ReLU}}(x) = \text{ReLU}\big(Wx + b\big)$.

Together with the final fully exponentiated connection, these components constitute a flexible yet monotonic value network that strictly preserves quantile ordering while maintaining sufficient representational capacity. Notably, this proposed network architecture ensures theoretical monotonicity with respect to the quantile partition τ, which is formalized in the following theorem.

Theorem 3.1 *Let $F_\theta(\tau; s, a)$ be the proposed Monotonic Quantile Value Network in Fig. 3 and τ be the input quantile partition. For any $\tau \in (0, 1)$, we have*

$$\frac{\partial F_\theta(\tau; s, a)}{\partial \tau} \geq 0. \quad (8)$$

The essential linear layers and clipped ReLU operations in MQVN yield an L-Lipschitz quantile function, while its architectural constraints ensure monotonicity. These properties imply that the corresponding CDF is ζ-strongly monotone, with $\zeta = 1/L$. We leverage this property to establish the following quantile estimation error bound. All proofs are provided in the appendix.

Theorem 3.2 (Quantile Error Control under Monotonicity) *Consider a finite MDP with state and action spaces $\mathcal{S}$ and $\mathcal{A}$, and a fixed policy π. Assume that (i) transitions in $\mathcal{D}$ are independently sampled, with rewards and next states conditionally independent given (s, a); (ii) the empirical and true Bellman return distributions share a common compact support $\mathcal{X}_{s,a}$; and (iii) the true Bellman target admits a continuous, nondecreasing, and L-Lipschitz quantile representation of the form characterized by MQVN, with a uniform constant $0 < L < \infty$. Then, for any $\delta \in (0, 1)$, with*

probability at least $1-\delta$*, the following holds simultaneously for all observed* (s,a)*:*

$$\left\| F^{-1}_{\widehat{\mathcal{T}}^{\pi} Z(s,a)} - F^{-1}_{\mathcal{T}^{\pi} Z(s,a)} \right\|_{\infty} \leq \underbrace{\frac{1}{\zeta} \sqrt{\frac{5|\mathcal{S}|}{n(s,a)} \log \frac{4|\mathcal{S}||\mathcal{A}|}{\delta}}}_{\Delta(s,a)}, \tag{9}$$

where $\zeta = 1/L$ *and the supremum is taken over* $\tau \in (0,1)$*.*

3.2 Quantiles-based Uncertainty for Exploration

Leveraging the distributional nature of the value function, we quantify action-value uncertainty based on the dispersion of the predicted quantiles, thereby enabling an uncertainty-aware exploration mechanism during the online fine-tuning phase. To obtain statistically robust estimates of the mean and standard deviation from a limited set of quantiles, we compute the dispersion of the value distribution using the seven quantiles advocated by Selvidge (1980), corresponding to the quantile partitions $\{0.01,\ 0.1,\ 0.25,\ 0.5,\ 0.75,\ 0.9,\ 0.99\}$. Following Lau et al. (1998), the estimation formulas are as follows:

$$\hat{\mu} = 0.28871\,\hat{q}_{0.5} + 0.18584\,(\hat{q}_{0.25} + \hat{q}_{0.75}) + 0.13394\,(\hat{q}_{0.1} + \hat{q}_{0.9}) + 0.036128\,(\hat{q}_{0.01} + \hat{q}_{0.99}) \tag{10}$$

$$\hat{\sigma} = 0.06611\,(\hat{q}_{0.75} - \hat{q}_{0.25}) + 0.16815\,(\hat{q}_{0.9} - \hat{q}_{0.1}) + 0.10794\,(\hat{q}_{0.99} - \hat{q}_{0.01}), \tag{11}$$

where $\hat{q}_{\tau}$ denotes the τ-quantile of the action-value distribution and can be predicted by the proposed MQVN $F_{\theta}(\tau; s, a)$. Under appropriate assumption, we have the following error bound. A detailed theoretical proof of the error bound is provided in Appendix C.

Theorem 3.3 *Suppose the observations are i.i.d. from a distribution belonging to one of the identifiable parametric distribution families in the Pearson or Johnson systems. let* $\hat{q}_{\tau}$ *be the empirical quantile. The quantile-based estimators in Eqs. (10) and (11) achieve* $O_p(N^{-1/2})$ *estimation error, where* N *is the number of samples.*

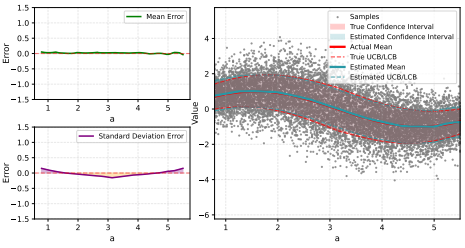


Figure 4: Validation of quantile-based estimators for normal distributions with sinusoidal mean and varying variance. The upper-left panel shows errors in mean estimation, while the lower-left panel presents errors in standard deviation estimation using formulas (10) and (11) via our MQVN. The right panel compares true (red) and estimated (cyan) one-standard-deviation confidence intervals overlaid on samples (gray points). Results demonstrate effective estimation of the UCB/LCB confidence interval.

To facilitate online exploration, we draw a base action a_0 from the current policy π and then construct N candidate actions by injecting Gaussian perturbations:

$$a_i = a_0 + \epsilon_i,\ \epsilon_i \sim \mathcal{N}(0, \epsilon^2 I),\ i = 1, \ldots, N, \tag{12}$$

where ϵ controls the exploration scale. For each candidate a_i, we compute its UCB value by

$$\widehat{\mathrm{UCB}}(s, a_i) = \hat{\mu}_i + \hat{\sigma}_i, \tag{13}$$

where $\hat{\mu}_i$ and $\hat{\sigma}_i$ are the estimated mean and standard deviation obtained via (10) and (11), using the quantiles predicted by $F_{\theta}(\tau; s, a_i)$. In Fig. 4, we further numerically validate the proposed quantile-based estimators in a two-dimensional example. For each action $a \in [\pi/4, 7\pi/4]$, the value distribution is modeled as a Gaussian distribution with mean $\sin(a)$ and variance that varies linearly between 0.5 and 1.5. We randomly sample actions and their corresponding returns, fit MQVN to these samples, and then combine the learned quantile function with Eqs. (10)-(11) to estimate the mean and standard deviation. The results demonstrate well-estimated confidence intervals.

The final exploratory action is sampled according to a softmax distribution with temperature β:

$$\pi_{\text{explore}}(a_i \mid s) = \frac{\exp\left(\widehat{\mathrm{UCB}}(s, a_i)/\beta\right)}{\sum_{j=1}^{N} \exp\left(\widehat{\mathrm{UCB}}(s, a_j)/\beta\right)}, \tag{14}$$

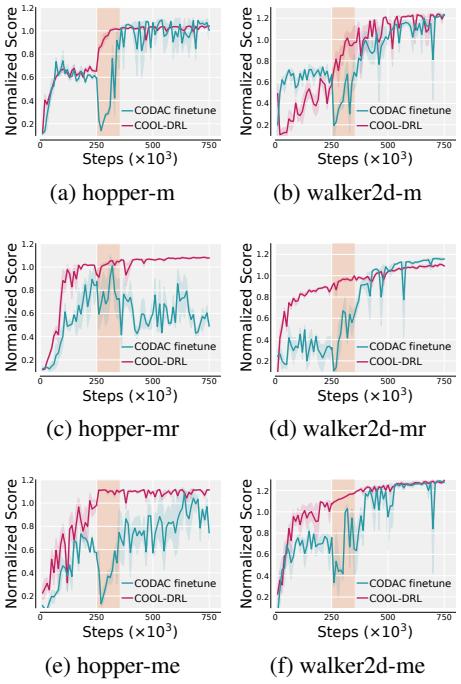
By combining quantile-based uncertainty estimations with the UCB-motivated sampling rule, our method systematically balances exploiting high-value actions and exploring nearby regions with high uncertainty. This leads to more stable and effective online finetuning.

4 EXPERIMENTS

Our experiments systematically evaluate the proposed method with respect to five questions: (1) **Stability during online finetuning.** Can our method mitigate instability at the onset of online adaptation? (2) **Overall performance improvement.** Does it outperform existing methods on the standard benchmark? (3) **Ablation on monotonic architecture.** What is the impact of removing the monotonicity guarantee in quantile estimation on stability and performance? (4) **Ablation on exploration.** How much performance degrades when the proposed quantile-based exploration is omitted? (5) **Kendall correlation analysis.** To what extent is the learned bias mitigated in value estimates? We evaluate Kendall's τ coefficient to (i) assess the rank correlation of Q-values across state-action pairs and (ii) verify the monotonicity of quantile estimates for the same state-action pair.

4.1 EXPERIMENTAL SETTINGS

Following the common practice in offline-to-online reinforcement learning, we evaluate our method COOL-DRL on the D4RL benchmark. Specifically, we first conduct experiments on the widely used Gym MuJoCo '-v2' domain, which includes three environments, each with medium (m), medium-replay (mr), and medium-expert (me) datasets. To further demonstrate the generality of our method, we additionally evaluate it on the AntMaze '-v2' domain across three difficulty levels: umaze, medium, and large. Evaluation metrics and detailed hyperparameter configurations of our method and the baseline algorithms, are provided in Appendix D.



(a) hopper-m (b) walker2d-m

(c) hopper-mr (d) walker2d-mr

(e) hopper-me (f) walker2d-me

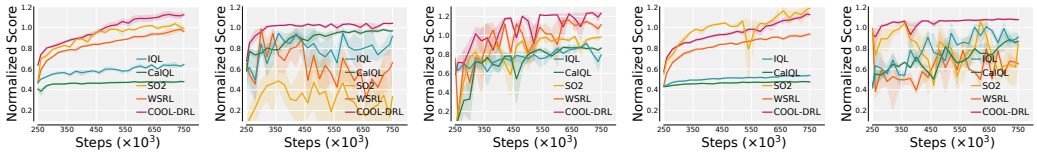
Figure 5: Normalized score curves of COOL-DRL and CODAC on the hopper and walker2d tasks. The shaded region around each curve indicates one standard deviation. The results show that COOL-DRL achieves stable offline-to-online transfer, whereas CODAC suffers a substantial performance drop, particularly at the beginning of online finetuning (orange region).

4.2 STABLE ONLINE FINETUNING

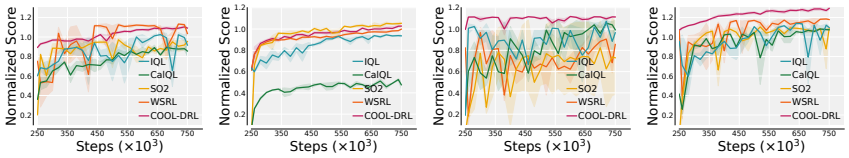
Prior to benchmarking against SOTA methods, we first investigate the stability of COOL-DRL during the critical offline-to-online transfer, as illustrated in Fig. 5. Since halfcheetah tasks typically exhibit stable online finetuning transfer regardless of the underlying algorithm, we focus our analysis on the more challenging hopper and walker2d environments to better discriminate performance stability. The experimental training steps consists of an offline pretraining phase (0–250K steps) followed immediately by an online finetuning phase (250K–750K steps).

As observed in the shaded orange region of Fig. 5, the distributional baseline, CODAC, suffers a significant performance decline at the onset of online finetuning. This instability stems from the offline conservative penalties, which distort the quantile values and violate the monotonicity requirement. Consequently, the model must aggressively correct these distorted values during the early online phase, resulting in severe fluctuation. In extreme cases, such as that shown in Fig. 5(c), the inability to finally rectify these values leads to gradual policy collapse.

In contrast, COOL-DRL employs a monotonic quantile value network, which theoretically guarantees the orderliness of quantile values regardless of the offline penalties. This structural constraint mitigates the potential bias and ensures a smooth and stable transition from offline to online learning.



(a) halfcheetah-m (b) hopper-m (c) walker2d-m (d) halfcheetah-mr (e) hopper-mr



(f) walker2d-mr (g) halfcheetah-me (h) hopper-me (i) walker2d-me

Figure 6: Normalized score curves of COOL-DRL on MuJoCo tasks. The shaded area around each curve denotes one standard deviation across five random seeds.

Furthermore, we observe that COOL-DRL maintains high stability throughout the entire online phase. This is attributed to our online exploration, which facilitates robust trial-and-error learning on perturbed behaviors. This mechanism enhances the stability of performance gains and mitigates late-stage fluctuation, a finding further corroborated by our ablation studies on online exploration.

4.3 Performance Comparison

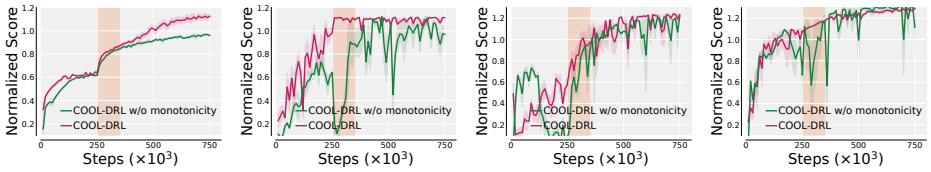
We benchmark COOL-DRL against existing offline-to-online methods including IQL (Kostrikov et al., 2022), CalQL (Nakamoto et al., 2023), SO2 (Zhang et al., 2024), and WSRL (Zhou et al., 2025) on the MuJoCo task suite. To ensure a fair comparison, we re-evaluate all baselines using official implementations under the same setting: 250K steps for offline pre-training followed by 500K steps for online finetuning. The online learning curves are presented in Fig. 6. COOL-DRL consistently outperforms IQL and CalQL across all tasks. For instance, on the halfcheetah-m task, our method achieves a final score of 1.12, significantly surpassing IQL (0.64) and CalQL (0.48). While they maintain stability during online finetuning, their performance turns to plateaus prematurely due to conservatism inherited from the offline phase, preventing further improvement.

Figure 7: The performance comparison on the AntMaze domain. Both COOL-DRL and WSRL achieve saturated performance (~ 1.00), significantly outperforming other methods. SO2 fails because its complete reliance on the ensemble technique for offline initialization prevents it from establishing a viable initial policy, thereby rendering subsequent online exploration ineffective.

Methods	umaze-d	medium-d	large-d
IQL	$0.75_{\pm 0.12} \rightarrow 0.73_{\pm 0.13}$	$0.73_{\pm 0.09} \rightarrow 0.78_{\pm 0.08}$	$0.12_{\pm 0.08} \rightarrow 0.47_{\pm 0.02}$
CalQL	$0.47_{\pm 0.10} \rightarrow \mathbf{0.98}_{\pm 0.02}$	$0.43_{\pm 0.12} \rightarrow 0.92_{\pm 0.02}$	$0.02_{\pm 0.02} \rightarrow 0.32_{\pm 0.45}$
SO2	$0.00_{\pm 0.00} \rightarrow 0.00_{\pm 0.00}$	$0.00_{\pm 0.00} \rightarrow 0.00_{\pm 0.00}$	$0.00_{\pm 0.00} \rightarrow 0.00_{\pm 0.00}$
WSRL	$0.27_{\pm 0.15} \rightarrow \mathbf{1.00}_{\pm 0.00}$	$0.33_{\pm 0.10} \rightarrow \mathbf{1.00}_{\pm 0.00}$	$0.42_{\pm 0.10} \rightarrow \mathbf{0.97}_{\pm 0.02}$
COOL-DRL	$0.25_{\pm 0.07} \rightarrow \mathbf{1.00}_{\pm 0.00}$	$0.50_{\pm 0.15} \rightarrow \mathbf{1.00}_{\pm 0.00}$	$0.35_{\pm 0.07} \rightarrow \mathbf{0.95}_{\pm 0.00}$

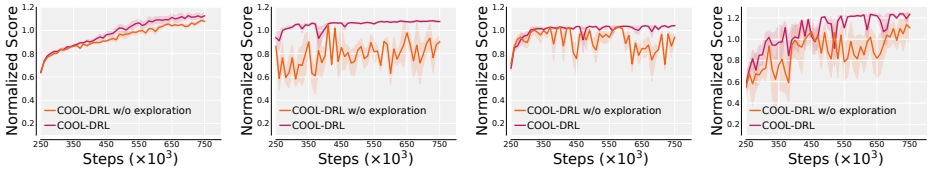
Regarding ensemble-based methods, SO2 achieves marginal gains over our method on halfcheetah-mr and halfcheetah-me, but underperforms across the broader benchmark tasks. As an ensemble-based method, SO2 exhibits sensitivity to the ensemble size; notably, it yields suboptimal performance on hopper tasks despite utilizing 10 critic networks. Similarly, WSRL, which also relies on ensemble mechanisms, approaches our performance on several tasks but mirrors the similar limitations on the hopper environment. Overall, these results demonstrate that COOL-DRL achieves superior performance through the combination of monotonicity-guaranteed quantile estimation and robust online exploration, without relying on ensemble-based critics.

We further evaluate our method on the diverse AntMaze datasets, with results summarized in Table 7. Given the sparse rewards and slower convergence, all methods are trained for 500K steps in both the offline and online phases. Among the baselines, IQL struggles to learn effective policies, while CalQL achieves competitive performance on the umaze and medium datasets. In contrast, SO2, which uses the ensemble-based EDAC (An et al., 2021) for offline initialization, fails completely



(a) halfcheetah-m (b) hopper-me (c) walker2d-m (d) walker2d-me

Figure 8: Ablation study of the MQVN, confirming the its crucial role in transfer stability.



(a) halfcheetah-m (b) hopper-mr (c) hopper-m (d) walker2d-m

Figure 9: Ablation study demonstrating the impact of the online exploration strategy. Incorporating the exploration mechanism improves the maximum asymptotic performance of the policy.

on AntMaze; without a viable initial policy, effective online exploration is impossible, resulting in near-zero performance. Performance on these tasks also appears to saturate at a normalized score of approximately 1.00 (near-100% success), as achieved by the WSRL baseline. COOL-DRL matches this state-of-the-art performance, demonstrating its broad applicability to AntMaze tasks.

4.4 Ablation Study

Ablation on Monotonicity. To validate the efficacy of the MQVN structure, we conduct an ablation study by replacing it with a standard MLP, while preserving our quantile-based online exploration strategy (see Fig. 8). The results indicate that, except for the typically stable halfcheetah environment, the removal of the monotonicity guarantee leads to a pronounced performance degradation immediately following the offline-to-online transfer. Particularly, in the walker2d-m task, policy collapse is observed as early as the end of the offline pre-training phase, which is visible in Fig. 8(c). Despite this initial instability, the ablated policy's final performance after online finetuning progressively recovers and approaches the performance of the full model. This robust recovery also implicitly confirms the effectiveness of our online exploration strategy.

Ablation on Exploration. To isolate the contribution of online exploration, we remove this component while retaining the MQVN structure. As shown in Fig. 9, the ablated model maintains stable initial offline-to-online transfer without significant performance degradation, further confirming that MQVN is primarily responsible for preventing performance drops during the transition. However, the overall online finetuning process reveals two key benefits of quantile-value uncertainty-based exploration: (1) Reduced training instability: By proactively exploring uncertain behaviors and incorporating the resulting data into training, the strategy reduces fluctuations during online policy improvement and promotes more robust and consistent behavior selection. (2) Enhanced final performance: Online exploration improves the policy's final converged performance.

4.5 Sensitivity to Online Exploration Scale

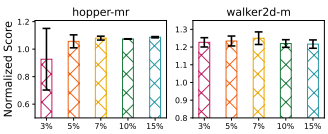
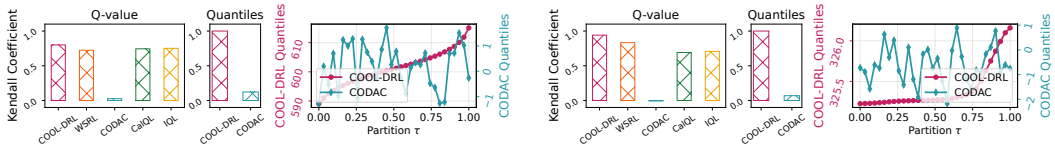


Figure 10: Sensitivity analysis of exploration scale ϵ.

We analyze the sensitivity of COOL-DRL to the exploration scale ϵ, where ϵ corresponds to the standard deviation of Gaussian action noise given the unit action bound. We vary $\epsilon \in \{0.03, 0.05, 0.07, 0.10, 0.15\}$ on hopper-mr and walker2d-m shown in Fig. 10, with each configuration evaluated over three random seeds. On hopper-mr, performance is suboptimal when ϵ is small, likely due to insufficient exploration that hinders the discovery of suboptimal behaviors, whereas increasing ϵ leads to improved and stable performance. In contrast, walker2d-m exhibits consistently strong performance



(a) halfcheetah-m (b) hopper-me

Figure 11: Kendall's τ analysis of Q-value and quantile estimates. Left: COOL-DRL better preserves Q-value ordering than CODAC. Center: for quantiles, COOL-DRL achieves 1.00 due to provable monotonicity, whereas CODAC exhibits significant crossing. Right: corresponding quantile curves, showing strictly monotonic behavior for COOL-DRL versus disordered quantiles for CODAC.

with minor fluctuations across all settings, indicating robustness to the choice of exploration scale. Overall, these confirm that the proposed exploration strategy is stable across a broad range of ϵ values.

4.6 Further Kendall Correlation Analysis on Value Estimations

Prior work (Zhang et al., 2024) has leveraged the Kendall's τ coefficient to assess the quality of Q-value estimation, as this metric measures the rank correlation between two vectors. Intuitively, if an action a_1 is better than a_2, a well-estimated value function should satisfy $Q(s, a_1) > Q(s, a_2)$. By evaluating the orderliness of value estimates, we gain insight into their overall fidelity.

4.6.1 Q-Value Orderliness

We estimate the Kendall's τ coefficient for the Q-value estimates obtained from various methods. To do this, we randomly sample 1,000 state-action pairs, (s, a), from the dataset and use each method's offline-trained critic network to compute the estimated Q-values. Since the true optimal Q-values are unattainable, we adopt the Q-values evaluated by the critic network of the best-performance method after full online finetuning as the proxy ground truth. The calculated coefficients for the Q-value vectors versus the proxy ground truth are presented in the left portion of the subplots in Fig. 11. The results indicate that the Q-value ordering of the classical distributional offline RL algorithm CODAC is more severely compromised by conservatism than that of non-distributional methods, whereas our method effectively mitigates this value bias and obtains higher Kendall correlation coefficient.

4.6.2 Quantile Value Monotonicity

We assess the intrinsic orderliness of distributional methods via Kendall's τ for their quantile estimates (COOL-DRL vs. CODAC), using the intrinsic rank order of the quantile partitions as ground truth. Averaged over 1,000 sampled state-action pairs, τ is shown in the center portion of Fig. 11. Due to COOL-DRL's theoretically guaranteed strict monotonicity, its Kendall coefficient is consistently 1.00, whereas CODAC's remains notably low, indicating severe quantile crossing. The right portion of Fig. 11 confirms this: COOL-DRL's quantile-value curves strictly preserve monotonicity, while CODAC's are chaotic and lack structural meaning. This provides stronger evidence that conservative constraints distort CODAC's value estimation more severely than non-distributional counterparts.

5 Conclusion

The degradation of quantile estimate monotonicity caused by conservative penalization severely compromises policy stability and final performance in offline-to-online DRL. Our proposed method, COOL-DRL, resolves this through two key contributions. First, the MQVN module theoretically guarantees correct quantile rank ordering, validated by a perfect Kendall's τ correlation, thereby mitigating value estimation bias and ensuring robust stability during the offline-to-online transfer. Second, we leverage the inherent uncertainty quantified by these quantiles to devise an effective online exploration strategy that minimizes training fluctuation and maximizes asymptotic performance. Empirically, COOL-DRL demonstrates substantial advantages on the D4RL benchmark, achieving superior overall performance while notably avoiding the need for ensemble techniques by other existing methods, thus offering an effective solution to the offline-to-online problem.

AI USE STATEMENT

AI tools were used solely to improve the clarity, grammar, and expression of the manuscript. The authors remain fully responsible for the content, analysis, interpretation, and conclusions presented in this work. No AI tools were used to generate the research results or to replace the authors' own analysis and judgment.

ETHICS STATEMENT

This work does not involve human participants, personal data, or animal subjects. The research focuses on methodological development for offline-to-online reinforcement learning and does not raise any specific ethical concerns beyond those generally associated with machine learning research.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed method, experimental setup, and implementation in the main paper and appendix. Additional implementation details, including an anonymous link to the source code, are also provided in the appendix to facilitate the reproduction of our results.

REFERENCES

Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified Q-ensemble. In *Advances in Neural Information Processing Systems*, volume 34, pp. 7436–7447, 2021.

Chenjia Bai, Ting Xiao, Zhoufan Zhu, Lingxiao Wang, Fan Zhou, Animesh Garg, Bin He, Peng Liu, and Zhaoran Wang. Monotonic quantile network for worst-case offline reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 35(7):8954–8968, 2024.

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

Liting Chen, Jie Yan, Zhengdao Shao, Lu Wang, Qingwei Lin, Saravanakumar Rajmohan, Thomas Moscibroda, and Dongmei Zhang. Conservative state value estimation for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 35064–35083, 2023.

Tianyu Chen, Zhendong Wang, and Mingyuan Zhou. Diffusion policies creating a trust region for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 1–22, 2024.

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

Will Dabney, Mark Rowland, Marc Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 1–10, 2018b.

Jiaheng Feng, Mingxiao Feng, Haolin Song, Wengang Zhou, and Houqiang Li. Suf: Stabilized unconstrained fine-tuning for offline-to-online reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 11961–11969, 2024.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 20132–20145, 2021.

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

Xiao Huang, Xu Liu, Enze Zhang, Tong Yu, and Shuai Li. Offline-to-online reinforcement learning with classifier-free diffusion generation. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 1–12, 2025.

Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline rl? In *Proceedings of the 38th International Conference on Machine Learning*, pp. 5084–5096, 2021.

Hyunho Kim and Jong-Seok Lee. Scalable monotonic neural networks. In *The Twelfth International Conference on Learning Representations*, pp. 1–17, 2024.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations*, pp. 1–11, 2022.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1179–1191, 2020.

Hon-Shiang Lau, Amy Hing-Ling Lau, and Chrwan-Jyh Ho. Improved moment-estimation formulas using more than three subjective fractiles. *Management Science*, 44(3):346–351, 1998.

Seunghyun Lee, Younggyo Seo, Kimin Lee, Pieter Abbeel, and Jinwoo Shin. Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble. *Proceedings of Machine Learning Research*, 164:1702–1712, 2021.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems, 2020.

Tenglong Liu, Yang Li, Yixing Lan, Hao Gao, Wei Pan, and Xin Xu. Adaptive advantage-guided policy regularization for offline reinforcement learning. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 1–19, 2024.

Wenhui Liu, Zhijian Wu, Jingchao Wang, Dingjiang Huang, and Shuigeng Zhou. Imagination-limited q-learning for offline reinforcement learning. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 5770–5778, 2025.

Jiafei Lyu, Xiaoteng Ma, Xiu Li, and Zongqing Lu. Mildly conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 1711–1724, 2022.

Yecheng Ma, Dinesh Jayaraman, and Osbert Bastani. Conservative offline distributional reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 19235–19247, 2021.

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.

J. J. A. Moors, L. W. G. Strijbosch, and W. J. H. van Groenendaal. Estimating mean and variance through quantiles: An experimental comparison of different methods. CentER Discussion Paper 2002-67, CentER, Tilburg University, Tilburg, 2002.

Mitsuhiko Nakamoto, Simon Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning. In *Advances in Neural Information Processing Systems*, pp. 62244–62269, 2023.

Kwanyoung Park, Seohong Park, Youngwoon Lee, and Sergey Levine. Scalable offline model-based RL with action chunks. In *The Fourteenth International Conference on Learning Representations*, pp. 1–25, 2026.

J Selvidge. Assessing the extremes of probability distributions by the fractile method. *Decision Sciences*, 11(3):493–502, 1980.

Robert J. Serling. *Approximation Theorems of Mathematical Statistics*. Wiley, 1980.

Yongjae Shin, Jeonghye Kim, Whiyoung Jung, Sunghoon Hong, Deunsol Yoon, Youngsoo Jang, Geon-Hyeong Kim, Jongseong Chae, Youngchul Sung, Kanghoon Lee, and Woohyung Lim. Online pre-training for offline-to-online reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 1–23, 2025.

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

Aad W Van der Vaart. *Asymptotic statistics*. Cambridge university press, 1998.

Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *The Eleventh International Conference on Learning Representations*, pp. 1–17, 2023.

Derek Yang, Li Zhao, Zichuan Lin, Tao Qin, Jiang Bian, and Tie-Yan Liu. Fully parameterized quantile function for distributional reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 32, pp. 1–10, 2019.

Haichao Zhang, Wei Xu, and Haonan Yu. Policy expansion for bridging offline-to-online reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Yinmin Zhang, Jie Liu, Chuming Li, Yazhe Niu, Yaodong Yang, Yu Liu, and Wanli Ouyang. A perspective of q-value estimation on offline-to-online reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 16908–16916, 2024.

Fan Zhou, Jianing Wang, and Xingdong Feng. Non-crossing quantile regression for distributional reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 15909–15919, 2020.

Gaoyue Zhou, Liyiming Ke, Siddhartha Srinivasa, Abhinav Gupta, Aravind Rajeswaran, and Vikash Kumar. Real world offline reinforcement learning with realistic data source. In *International Conference on Robotics and Automation*, pp. 7176–7183, 2023.

Zhiyuan Zhou, Andy Peng, Qiyang Li, Sergey Levine, and Aviral Kumar. Efficient online reinforcement learning fine-tuning need not retain offline data. In *The Thirteenth International Conference on Learning Representations*, 2025.

A PROOF OF THEOREM 3.1: MONOTONICITY

We proceed by direct computation of the derivative through the network layers. For brevity, let $f(\tau)$ denote the output of the monotonic quantile value network $F_\theta(\tau; s, a)$ at quantile level $\tau \in (0, 1)$.

First, we compute the derivative of the quantile embedding layer. Let

$$\alpha = \frac{2\pi}{d+1} > 0,$$

the embedding vector τ_{emb} has components

$$\tau_{\text{emb}}^{(k)}(\tau) = \sin\big(c_k + \alpha\tau\big), \qquad k = 0, \dots, d-1,$$

with $c_k = \frac{k\pi}{d+1} - \frac{\pi}{2}$. Hence

$$\frac{\mathrm{d}\tau_{\text{emb}}^{(k)}}{\mathrm{d}\tau} = \alpha \cos\big(c_k + \alpha\tau\big),$$

or in vector form

$$\frac{\mathrm{d}\tau_{\text{emb}}}{\mathrm{d}\tau} = \alpha\Big[\cos(c_0 + \alpha\tau) \quad \dots \quad \cos(c_{d-1} + \alpha\tau)\Big]^\top.$$

For notational convenience, we denote the embedding vector τ_{emb} by h_0. We examine the forward pass through the hidden layers and their derivatives. For $i = 1, 2, 3, 4$, we have

$$a_i = \exp(W_i)h_{i-1} + b_i, \qquad h_i = \mathrm{ReLU}_u(a_i),$$

where W_i and b_i are the weight and bias of the i-th exponentiated unit, respectively. In particular, W_1 and b_1 denote the parameters introduced by the MQPE module, whereas the subsequent W_i and b_i correspond to the exponentiated units used in the remainder of the MQVN architecture. The exponentiation is applied element-wise to W_i, and ReLU_u is also applied element-wise to each component of a_i.

The Jacobian of the element-wise ReLU_u operation is a diagonal matrix. Specifically,

$$R_i = \nabla\,\mathrm{ReLU}_u(a_i) = \mathrm{diag}\big(r_i^{(1)}, \dots, r_i^{(n_i)}\big), \tag{15}$$

where for each k,

$$r_i^{(k)} = \begin{cases} 0, & a_i^{(k)} < 0 \text{ or } a_i^{(k)} > u, \\ 1, & 0 < a_i^{(k)} < u. \end{cases}$$

At boundaries, we may take any subgradient value in $[0, 1]$. Therefore, $r_i^{(k)} \in [0, 1]$ holds in all cases.

Applying the chain rule, the derivative of h_i with respect to τ is

$$\frac{\mathrm{d}h_i}{\mathrm{d}\tau} = R_i \cdot \frac{\mathrm{d}a_i}{\mathrm{d}\tau} = R_i \cdot \exp(W_i)\frac{\mathrm{d}h_{i-1}}{\mathrm{d}\tau}.$$

For layer $i = 1$, noting that $h_0 \in \mathbb{R}^d$ and $W_1 \in \mathbb{R}^{n_1 \times d}$, we have

$$\frac{\mathrm{d}h_1}{\mathrm{d}\tau} = R_1 \cdot \exp(W_1)\frac{\mathrm{d}h_0}{\mathrm{d}\tau}.$$

For layers $i = 2, 3, 4$, the weight matrices $W_i \in \mathbb{R}^{n_i \times n_{i-1}}$, so

$$\frac{\mathrm{d}h_2}{\mathrm{d}\tau} = R_2 \cdot \exp(W_2)\frac{\mathrm{d}h_1}{\mathrm{d}\tau}, \qquad \frac{\mathrm{d}h_3}{\mathrm{d}\tau} = R_3 \cdot \exp(W_3)\frac{\mathrm{d}h_2}{\mathrm{d}\tau}, \qquad \frac{\mathrm{d}h_4}{\mathrm{d}\tau} = R_4 \cdot \exp(W_4)\frac{\mathrm{d}h_3}{\mathrm{d}\tau}.$$

The final fully exponentiated connection layer gives $f(\tau) = \exp(w_5)^\top h_4 + b_5$, we hence have

$$\frac{\mathrm{d}f}{\mathrm{d}\tau} = \exp(w_5)^\top \frac{\mathrm{d}h_4}{\mathrm{d}\tau}.$$

Unrolling this recursion, we obtain

$$\frac{\mathrm{d}f}{\mathrm{d}\tau} = \exp(w_5)^\top R_4 \exp(W_4) R_3 \exp(W_3) R_2 \exp(W_2) R_1 \exp(W_1)\frac{\mathrm{d}h_0}{\mathrm{d}\tau}. \tag{16}$$

By substituting the expression for $\frac{\mathrm{d}h_0}{\mathrm{d}\tau}$ into Eq. (16), we obtain

$$\frac{\mathrm{d}f}{\mathrm{d}\tau} = \alpha \, \exp(w_5)^\top R_4 \exp(W_4) R_3 \exp(W_3) R_2 \exp(W_2) R_1 \exp(W_1) \begin{bmatrix} \cos(c_0 + \alpha\tau) \\ \cos(c_1 + \alpha\tau) \\ \vdots \\ \cos(c_{d-1} + \alpha\tau) \end{bmatrix}. \quad (17)$$

We now analyze the sign of these expressions. We first simplify the trigonometric term in Eq. (17). Since

$$c_k + \alpha\tau = \frac{k\pi}{d+1} - \frac{\pi}{2} + \frac{2\pi}{d+1}\tau = \frac{\pi}{d+1}\big(k + 2\tau\big) - \frac{\pi}{2},$$

and using $\cos\big(x - \frac{\pi}{2}\big) = \sin x$, we have

$$\cos\big(c_k + \alpha\tau\big) = \sin\left(\frac{\pi}{d+1}(k + 2\tau)\right).$$

Let us define the vector

$$v = \Big[\sin\big(\frac{\pi}{d+1}(0 + 2\tau)\big) \; \ldots \; \sin\big(\frac{\pi}{d+1}(d - 1 + 2\tau)\big)\Big]^\top.$$

For $\tau \in (0, 1)$ and $k = 0, \ldots, d-1$, each component satisfies

$$\frac{\pi}{d+1}(0 + 2 \cdot 0) < \frac{\pi}{d+1}(k + 2\tau) < \frac{\pi}{d+1}(d - 1 + 2) = \pi.$$

Since $\sin\theta > 0$ for all $\theta \in (0, \pi)$, every component of v is strictly positive.

Now consider the sequence of linear transformations applied to v. All matrices involved have nonnegative entries:

- $\exp(w_5)$ or $\exp(W_i), i = 1, \cdots, 4$ (element-wise exponentiation) gives vector or matrices with strictly positive entries;
- R_i are diagonal matrices with diagonal entries in $[0, 1]$;
- The product of nonnegative matrices is nonnegative.

Since v has all positive components and is premultiplied by matrices with nonnegative entries, the result of $R_1 \exp(W_1)v$ is a vector with nonnegative components. This property propagates through each subsequent multiplication by $R_i \exp(W_i),\ i = 2, 3, 4$. Finally, $\alpha \exp(w_5)^\top$ (a row vector with positive entries) takes a nonnegative inner product, yielding a nonnegative scalar, i.e.,

$$\frac{\mathrm{d}f(\tau)}{\mathrm{d}\tau} \geq 0 \quad \forall\tau \in (0, 1).$$

The function f is continuous and piecewise linear with nondecreasing segments. Hence $f(\tau)$ is monotone non-decreasing in τ on $[0, 1]$.

In the above derivation, we omit the representation part of the state-action pair (s, a) for brevity. We now show that incorporating concatenated state-action latent representations does not affect the monotonicity of the quantile value function with respect to τ. Specifically, let z_{i-1} denote the state-action representation at layer $i-1$. The weight matrix is partitioned as $W_i = [\, W_i^h \;\; W_i^z \,]$, where W_i^h corresponds to the weights associated with the quantile-dependent hidden representation h_{i-1}, and W_i^z corresponds to the weights associated with the state-action representation z_{i-1}. Therefore, the h_i can be written as

$$a_i = \exp\big([\, W_i^h \;\; W_i^z \,]\big) \begin{bmatrix} h_{i-1} \\ z_{i-1} \end{bmatrix} + b_i = \exp(W_i^h)\, h_{i-1} + \exp(W_i^z)\, z_{i-1} + b_i, \quad h_i = \mathrm{ReLU}_u(a_i).$$

Since the state-action representation z_{i-1} is independent of the quantile level τ, its contribution vanishes when taking derivatives with respect to τ through the linear transformation. Specifically, we have

$$\frac{\mathrm{d}a_i}{\mathrm{d}\tau} = \exp(W_i^h)\, \frac{\mathrm{d}h_{i-1}}{\mathrm{d}\tau}.$$

The subsequent activation $h_i = \mathrm{ReLU}_u(a_i)$ introduces a Jacobian matrix R_i as in Eq. (15), which is diagonal with non-negative entries determined by the activation status of each unit. Although a_i depends on z_{i-1}, this dependence only affects the values of the diagonal entries in R_i and does not introduce negative elements. Consequently, the Jacobian remains element-wise non-negative.

Therefore, the confluence of the state–action representation does not affect the non-negativity of the derivative $\frac{\mathrm{d}h_i}{\mathrm{d}\tau}$, and the monotonicity of the overall function $F_\theta(\tau; s, a)$ with respect to τ is preserved.

B Proof of Theorem 3.2: Quantile control under monotonicity

The proof proceeds in three steps. First, we establish the Lipschitz continuity of the MQVN quantile function and the resulting strong monotonicity of its induced CDF. Second, we derive a uniform concentration bound for the empirical Bellman target CDF. Finally, we translate this CDF bound into a uniform quantile estimation error bound.

Step 1: Lipschitz continuity and strong monotonicity induced by MQVN. The MQVN quantile function is constructed using linear layers and clipped ReLU activations.

Consider a sequence of K linear layers

$$f_k(x) = W_k x + b_k, \qquad k = 1, \dots, K, \tag{18}$$

interleaved with clipped ReLU activations.

Each linear layer satisfies

$$\begin{aligned} \|f_k(u) - f_k(v)\|_2 &= \|W_k(u - v)\|_2 \\ &\le \|W_k\|_2 \|u - v\|_2, \end{aligned} \tag{19}$$

where $\|W_k\|_2$ denotes the spectral norm.

The clipped ReLU activation

$$\sigma(x) = \min\{\max\{x, 0\}, c\}, \qquad c > 0, \tag{20}$$

is non-expansive:

$$\|\sigma(u) - \sigma(v)\|_2 \le \|u - v\|_2. \tag{21}$$

Consequently, by the compositional property of Lipschitz functions, the resulting quantile network satisfies

$$|Q_\theta(\tau_2) - Q_\theta(\tau_1)| \le L|\tau_2 - \tau_1|, \tag{22}$$

where a valid Lipschitz constant for the sequential network is

$$L = \prod_{k=1}^{K} \|W_k\|_2. \tag{23}$$

For a piecewise quantile architecture, the same argument applies to each partition. If the quantile function is continuous across partition boundaries, a global Lipschitz constant is given by the maximum of the partition-wise constants, including any input normalization and output scaling factors.

Together with the architectural monotonicity constraints, these properties ensure that Q_θ is continuous, nondecreasing, and L-Lipschitz.

Let F_θ denote its induced CDF, and let $[a, b]$ be its support interval.

For any $a \le x_1 < x_2 \le b$, define

$$\tau_i = \sup\{\tau \in [0, 1] : Q_\theta(\tau) \le x_i\}, \qquad i = 1, 2. \tag{24}$$

Since Q_θ is continuous and nondecreasing, it follows that

$$Q_\theta(\tau_i) = x_i, \qquad F_\theta(x_i) = \tau_i. \tag{25}$$

Therefore,

$$\begin{aligned} x_2 - x_1 &= Q_\theta(\tau_2) - Q_\theta(\tau_1) \\ &\le L(\tau_2 - \tau_1) \\ &= L\big(F_\theta(x_2) - F_\theta(x_1)\big). \end{aligned} \tag{26}$$

Rearranging yields

$$F_\theta(x_2) - F_\theta(x_1) \ge \zeta(x_2 - x_1), \qquad \zeta = \frac{1}{L}. \tag{27}$$

Thus, the CDF induced by MQVN is ζ-strongly monotone on its support interval.

By the realizability assumption in the theorem, this property also holds for the true Bellman target $\mathcal{T}^\pi Z(s,a)$.

Step 2: Bounding the CDF estimation error. Fix an observed state-action pair (s,a), and write $n = n(s,a)$.

Let $R(\cdot \mid s,a)$ and $P(\cdot \mid s,a)$ denote the true reward and transition distributions, respectively. Let $\widehat{R}(\cdot \mid s,a)$ and $\widehat{P}(\cdot \mid s,a)$ denote their empirical counterparts.

Under the conditional independence assumption, the true Bellman target CDF can be written as

$$\begin{aligned} &F_{\mathcal{T}^\pi Z(s,a)}(x) \\ &= \sum_{s',a'} P(s' \mid s,a)\pi(a' \mid s') F_{R(s,a)+\gamma Z(s',a')}(x). \end{aligned} \tag{28}$$

By the definition of the product-model empirical Bellman operator,

$$\begin{aligned} &F_{\widehat{\mathcal{T}}^\pi Z(s,a)}(x) \\ &= \sum_{s',a'} \widehat{P}(s' \mid s,a)\pi(a' \mid s') F_{\widehat{R}(s,a)+\gamma Z(s',a')}(x). \end{aligned} \tag{29}$$

Here, the reward and continuation-return distributions are combined through independent convolution.

Adding and subtracting

$$\sum_{s',a'} \widehat{P}(s' \mid s,a)\pi(a' \mid s') F_{R(s,a)+\gamma Z(s',a')}(x) \tag{30}$$

gives

$$\begin{aligned} &F_{\widehat{\mathcal{T}}^\pi Z(s,a)}(x) - F_{\mathcal{T}^\pi Z(s,a)}(x) \\ &= \sum_{s',a'} \widehat{P}(s' \mid s,a)\pi(a' \mid s') \Big[F_{\widehat{R}(s,a)+\gamma Z(s',a')}(x) - F_{R(s,a)+\gamma Z(s',a')}(x) \Big] \\ &\quad + \sum_{s',a'} \big[\widehat{P}(s' \mid s,a) - P(s' \mid s,a) \big] \pi(a' \mid s') F_{R(s,a)+\gamma Z(s',a')}(x). \end{aligned} \tag{31}$$

We bound the two terms separately.

For the first term, convolution with the same independent continuation-return distribution is non-expansive under the CDF supremum norm.

Indeed, for any independent random variable Y,

$$\begin{aligned} &|F_{\widehat{R}+Y}(x) - F_{R+Y}(x)| \\ &= \big| \mathbb{E}_Y \left[F_{\widehat{R}}(x-Y) - F_R(x-Y) \right] \big| \\ &\le \|F_{\widehat{R}} - F_R\|_\infty. \end{aligned} \tag{32}$$

Consequently, because the mixture weights are nonnegative and sum to one, the absolute value of the first term in Eq. equation 31 is bounded by

$$\left\| F_{\widehat{R}(s,a)} - F_{R(s,a)} \right\|_\infty. \tag{33}$$

For the second term, define

$$h_x(s') = \sum_{a'} \pi(a' \mid s') F_{R(s,a)+\gamma Z(s',a')}(x). \tag{34}$$

Since $h_x(s')$ is a convex combination of CDF values,

$$0 \le h_x(s') \le 1. \tag{35}$$

It follows that

$$\begin{aligned} &\left| \sum_{s'} \big[\widehat{P}(s' \mid s,a) - P(s' \mid s,a)\big] h_x(s') \right| \\ &\le \left\| \widehat{P}(\cdot \mid s,a) - P(\cdot \mid s,a) \right\|_1 . \end{aligned} \tag{36}$$

Combining the two bounds and taking the supremum over x yields

$$\begin{aligned} &\left\| F_{\widehat{\mathcal{T}}^\pi Z(s,a)} - F_{\mathcal{T}^\pi Z(s,a)} \right\|_\infty \\ &\le \left\| F_{\widehat{R}(s,a)} - F_{R(s,a)} \right\|_\infty \\ &\quad + \left\| \widehat{P}(\cdot \mid s,a) - P(\cdot \mid s,a) \right\|_1 . \end{aligned} \tag{37}$$

We next control these two estimation errors.

Let

$$N = |\mathcal{S}||\mathcal{A}|, \qquad m = |\mathcal{S}|. \tag{38}$$

By the Dvoretzky–Kiefer–Wolfowitz inequality, for every $t > 0$,

$$\Pr\left(\left\| F_{\widehat{R}(s,a)} - F_{R(s,a)} \right\|_\infty > t \right) \le 2e^{-2nt^2}. \tag{39}$$

Applying a union bound over the observed state-action pairs, with probability at least $1 - \delta/2$,

$$\left\| F_{\widehat{R}(s,a)} - F_{R(s,a)} \right\|_\infty \le \sqrt{\frac{1}{2n(s,a)} \log \frac{4N}{\delta}} \tag{40}$$

simultaneously for all observed (s,a).

For the transition distribution, the multinomial ℓ_1 concentration inequality gives

$$\Pr\left(\left\| \widehat{P}(\cdot \mid s,a) - P(\cdot \mid s,a) \right\|_1 > t \right) \le 2^m e^{-nt^2/2}. \tag{41}$$

Setting

$$t = \sqrt{\frac{2m}{n(s,a)} \log \frac{4N}{\delta}} \tag{42}$$

and applying a union bound gives, with probability at least $1 - \delta/2$,

$$\left\| \widehat{P}(\cdot \mid s,a) - P(\cdot \mid s,a) \right\|_1 \le \sqrt{\frac{2|\mathcal{S}|}{n(s,a)} \log \frac{4N}{\delta}} \tag{43}$$

simultaneously for all observed (s,a).

Combining Eqs. equation 37, equation 40, and equation 43, another union bound implies that, with probability at least $1 - \delta$,

$$\begin{aligned} &\left\| F_{\widehat{\mathcal{T}}^\pi Z(s,a)} - F_{\mathcal{T}^\pi Z(s,a)} \right\|_\infty \\ &\le \left(\frac{1}{\sqrt{2}} + \sqrt{2|\mathcal{S}|} \right) \sqrt{\frac{1}{n(s,a)} \log \frac{4N}{\delta}}. \end{aligned} \tag{44}$$

Since $|\mathcal{S}| \geq 1$,

$$\frac{1}{\sqrt{2}} + \sqrt{2|\mathcal{S}|} \leq \sqrt{5|\mathcal{S}|}. \tag{45}$$

We therefore obtain

$$\left\| F_{\widehat{\mathcal{T}}^\pi Z(s,a)} - F_{\mathcal{T}^\pi Z(s,a)} \right\|_\infty \leq \varepsilon(s,a), \tag{46}$$

where

$$\varepsilon(s,a) = \sqrt{\frac{5|\mathcal{S}|}{n(s,a)} \log \frac{4|\mathcal{S}||\mathcal{A}|}{\delta}}. \tag{47}$$

This bound holds simultaneously for all observed state-action pairs with probability at least $1-\delta$.

Step 3: Translating the CDF bound into a quantile estimation bound. Fix an observed (s,a) and write

$$F = F_{\mathcal{T}^\pi Z(s,a)}, \qquad G = F_{\widehat{\mathcal{T}}^\pi Z(s,a)}. \tag{48}$$

By Eq. equation 46, on the high-probability event,

$$\|F - G\|_\infty \leq \varepsilon(s,a). \tag{49}$$

We first establish the support relation needed to control the extreme quantiles.

Let $[a,b]$ denote the support interval of the true Bellman target. Such an interval exists and is compact because its quantile representation is continuous on $[0,1]$.

Almost surely, every observed reward belongs to the support of the true reward distribution, and every observed next state with positive empirical probability also has positive probability under the true transition distribution.

Under the conditional independence assumption, every reward and next-state combination assigned positive probability by the empirical product model is therefore also supported by the true product model.

Since the same continuation-return distributions and policy are used by both Bellman operators, this implies

$$\operatorname{supp}(G) \subseteq \operatorname{supp}(F) = [a,b] \tag{50}$$

almost surely.

Next, by the strong monotonicity of F established in Step 1, for $a \leq x_1 < x_2 \leq b$,

$$F(x_2) - F(x_1) \geq \zeta(x_2 - x_1). \tag{51}$$

Equivalently, its generalized inverse is $1/\zeta$-Lipschitz:

$$|F^{-1}(u) - F^{-1}(v)| \leq \frac{|u-v|}{\zeta}, \qquad u,v \in (0,1). \tag{52}$$

By continuity, this bound also holds when the quantile function is extended to 0 and 1 using the endpoints a and b.

The uniform CDF bound gives

$$F(x) - \varepsilon(s,a) \leq G(x) \leq F(x) + \varepsilon(s,a). \tag{53}$$

For any $\tau \in (0,1)$, define

$$\tau_- = \max\{0, \tau - \varepsilon(s,a)\}, \tag{54}$$

and

$$\tau_+ = \min\{1, \tau + \varepsilon(s,a)\}. \tag{55}$$

By the monotonicity of generalized inverses and the support inclusion in Eq. equation 50,

$$F^{-1}(\tau_-) \leq G^{-1}(\tau) \leq F^{-1}(\tau_+). \tag{56}$$

The support inclusion ensures that this inequality remains valid when $\tau_- = 0$ or $\tau_+ = 1$.

Applying the inverse Lipschitz bound,

$$\begin{aligned} F^{-1}(\tau_+) - F^{-1}(\tau) &\leq \frac{\tau_+ - \tau}{\zeta} \\ &\leq \frac{\varepsilon(s,a)}{\zeta}, \end{aligned} \tag{57}$$

and similarly,

$$\begin{aligned} F^{-1}(\tau) - F^{-1}(\tau_-) &\leq \frac{\tau - \tau_-}{\zeta} \\ &\leq \frac{\varepsilon(s,a)}{\zeta}. \end{aligned} \tag{58}$$

Combining these inequalities with Eq. equation 56 yields

$$|G^{-1}(\tau) - F^{-1}(\tau)| \leq \frac{\varepsilon(s,a)}{\zeta}. \tag{59}$$

Taking the supremum over $\tau \in (0,1)$, we obtain

$$\begin{aligned} &\left\| F^{-1}_{\widehat{\mathcal{T}}^\pi Z(s,a)} - F^{-1}_{\mathcal{T}^\pi Z(s,a)} \right\|_\infty \\ &\leq \frac{1}{\zeta} \sqrt{\frac{5|\mathcal{S}|}{n(s,a)} \log \frac{4|\mathcal{S}||\mathcal{A}|}{\delta}} \\ &= \Delta(s,a). \end{aligned} \tag{60}$$

Because the concentration event in Step 2 holds simultaneously for all observed state-action pairs, the quantile error bound also holds simultaneously for all such pairs with probability at least $1 - \delta$. This completes the proof.

Corollary B.1 (Expected Value Error) *Under the assumptions of Theorem 3.2, with probability at least $1 - \delta$, simultaneously for all observed (s,a),*

$$\begin{aligned} &\left| \mathbb{E}[\widehat{\mathcal{T}}^\pi Z(s,a)] - \mathbb{E}[\mathcal{T}^\pi Z(s,a)] \right| \\ &\leq \Delta(s,a). \end{aligned} \tag{61}$$

For any integrable distribution, its expectation equals the integral of its quantile function:

$$\mathbb{E}[X] = \int_0^1 F_X^{-1}(\tau)\, d\tau. \tag{62}$$

Therefore,

$$\begin{aligned} &\left| \mathbb{E}[\widehat{\mathcal{T}}^\pi Z(s,a)] - \mathbb{E}[\mathcal{T}^\pi Z(s,a)] \right| \\ &= \left| \int_0^1 \left(F^{-1}_{\widehat{\mathcal{T}}^\pi Z(s,a)}(\tau) - F^{-1}_{\mathcal{T}^\pi Z(s,a)}(\tau) \right) d\tau \right| \\ &\leq \int_0^1 \left| F^{-1}_{\widehat{\mathcal{T}}^\pi Z(s,a)}(\tau) - F^{-1}_{\mathcal{T}^\pi Z(s,a)}(\tau) \right| d\tau \\ &\leq \Delta(s,a). \end{aligned} \tag{63}$$

The result follows directly from Theorem 3.2.

C Proof of Theorem 3.3: Error Bound of Quantile-Based Estimators

Let $X_1, \ldots, X_N$ be i.i.d. random variables drawn from a distribution with cumulative distribution function F and density f. We assume that F belongs to one of the identifiable parametric distribution families contained in the Pearson or Johnson systems (Moors et al., 2002). For any $\tau \in (0,1)$, denote the population quantile by $q_\tau = F^{-1}(\tau)$ and the corresponding empirical quantile by $\hat{q}_\tau$. We consider the quantile-based estimator of the mean defined by

$$\hat{\mu} = \sum_{k=1}^{K} a_k \hat{q}_{\tau_k},$$

where the quantile levels $\tau_k \in (0,1)$ are fixed and the coefficients a_k are deterministic. The coefficients in (10) are chosen from (Lau et al., 1998) such that $\sum_{k=1}^{K} a_k = 1$ and $\sum_{k=1}^{K} a_k q_{\tau_k} = \mu$. We now consider that the asymptotic distribution and error bound of

$$\hat{\mu} - \mu = \sum_{k=1}^{K} a_k (\hat{q}_{\tau_k} - q_{\tau_k}).$$

Obviously, the asymptotic behavior of the above error is governed by that of the difference between quantile q_{τ_k} and empirical quantile $\hat{q}_{\tau_k}$. Let

$$F_N(x) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{X_i \le x\}$$

denote the empirical cumulative distribution function. According to the asymptotic property (Serling, 1980; Van der Vaart, 1998), we have $\sqrt{N}(\hat{q}_\tau - q_\tau) \xrightarrow{d} \mathcal{N}(0, \tau(1-\tau)/[f(q_\tau)]^2)$ for each fixed $\tau \in (0,1)$, where $\mathcal{N}$ represents a normal distribution. Furthermore, the empirical quantile admits the Bahadur's representation

$$\hat{q}_\tau - q_\tau = -\frac{F_N(q_\tau) - \tau}{f(q_\tau)} + o_p(N^{-1/2}). \tag{64}$$

The empirical cumulative distribution function has an exact covariance structure: for any $x, y \in \mathbb{R}$,

$$\begin{aligned}
&\mathrm{Cov}(F_N(x), F_N(y)) \\
&= \mathbb{E}\left[F_N(x) F_N(y)\right] - \mathbb{E}\left[F_N(x)\right] \mathbb{E}\left[F_N(y)\right] \\
&= \mathbb{E}\left[\frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \mathbf{1}\{X_i \le x\} \mathbf{1}\{X_j \le y\}\right] - F(x)F(y) \\
&= \frac{1}{N^2}\left(\sum_{i \ne j} \mathbb{E}[\mathbf{1}\{X_i \le x\}] \mathbb{E}[\mathbf{1}\{X_j \le y\}] + \mathbb{E}\sum_{i=1}^{N} \mathbf{1}\{X_i \le x\} \mathbf{1}\{X_i \le y\}\right) - F(x)F(y) \\
&= \frac{1}{N^2}\big(N(N-1)F(x)F(y) + NF(\min(x,y))\big) - F(x)F(y) \\
&= \frac{1}{N}\big(F(\min(x,y)) - F(x)F(y)\big).
\end{aligned} \tag{65}$$

Combining this identity together with the Bahadur's representation Eq. (64) yields, for any fixed $\tau_i, \tau_j \in (0,1)$,

$$\mathrm{Cov}(\hat{q}_{\tau_i}, \hat{q}_{\tau_j}) = \frac{1}{N} \frac{\min(\tau_i, \tau_j) - \tau_i \tau_j}{f(q_{\tau_i}) f(q_{\tau_j})} + o_p(N^{-1}).$$

Since $\hat{\mu} - \mu$ is a linear combination of $\hat{q} - q$, its variance follows as

$$\begin{aligned}
\mathrm{Var}(\hat{\mu} - \mu) &= \mathrm{Cov}\left(\sum_{i=1}^{K} a_i (\hat{q}_{\tau_i} - q_{\tau_i}), \sum_{j=1}^{K} a_j (\hat{q}_{\tau_j} - q_{\tau_j})\right) \\
&= \frac{1}{N} \sum_{i,j} a_i a_j \frac{\min(\tau_i, \tau_j) - \tau_i \tau_j}{f(q_{\tau_i}) f(q_{\tau_j})} + o_p(N^{-1}).
\end{aligned} \tag{66}$$

As we mentioned above, the joint asymptotic normality of the empirical cumulative distribution function implies the joint asymptotic normality of the empirical quantiles, i.e. $\sqrt{N}(\hat{q}_\tau - q_\tau) \xrightarrow{d} \mathcal{N}(0, \tau(1-\tau)/[f(q_\tau)]^2)$. As $\hat{\mu} - \mu$ is the linear combination of the corresponding quantiles, we hence have

$$\sqrt{N}(\hat{\mu} - \mu) \xrightarrow{d} \mathcal{N}\left(0, \sum_{i,j} a_i a_j \frac{\min(\tau_i, \tau_j) - \tau_i \tau_j}{f(q_{\tau_i}) f(q_{\tau_j})}\right). \tag{67}$$

Therefore, we can derive an asymptotic probability error bound

$$\mathbb{P}\left((\hat{\mu} - \mu) \leq \sqrt{\frac{\sum_{i,j} a_i a_j \frac{\min(\tau_i, \tau_j) - \tau_i \tau_j}{f(q_{\tau_i}) f(q_{\tau_j})}}{N}} t\right) \longrightarrow \Phi(t), \quad N \to \infty, \tag{68}$$

Hence,

$$\mathbb{P}\left(|\hat{\mu} - \mu| > \sqrt{\frac{\sum_{i,j} a_i a_j \frac{\min(\tau_i, \tau_j) - \tau_i \tau_j}{f(q_{\tau_i}) f(q_{\tau_j})}}{N}} t\right) \longrightarrow 2(1 - \Phi(t)), \tag{69}$$

Let $\delta \in (0, 1)$ be a target error probability, and define $z_{1-\delta/2} = \Phi^{-1}(1 - \delta/2)$. By choosing $t = z_{1-\delta/2}$, we can obtain that

$$\mathbb{P}\left(|\hat{\mu} - \mu| \leq z_{1-\delta/2} \sqrt{\frac{\sum_{i,j} a_i a_j \frac{\min(\tau_i, \tau_j) - \tau_i \tau_j}{f(q_{\tau_i}) f(q_{\tau_j})}}{N}}\right) \longrightarrow 1 - \delta, \tag{70}$$

We thus conclude that the estimation error satisfies $|\hat{\mu} - \mu| = O_p(N^{-1/2})$. These results establish that the quantile-based estimator $\hat{\mu}$ is root-N consistent and asymptotically normal, with an explicit asymptotic variance determined by the chosen quantile levels and weights. An analogous analysis can be carried out for the standard deviation estimator $\hat{\sigma}$.

D Experimental Details

The pseudocode is summarized in Algorithm 1. The basic hyperparameter configuration, which is consistently applied across both offline pretraining and online finetuning, is detailed in Table 1, including network architecture, optimization settings, and learning rates. Our code is available at: `https://anonymous.4open.science/r/COOLDRL-ECE6`. All evaluations are conducted using the Normalized Score metric. For the MuJoCo domain, it is computed as:

$$\text{Normalized Score} = \frac{\text{Learned Score} - \text{Base Score}}{\text{Expert Score} - \text{Base Score}},$$

where both the *Expert Score* and *Base Score* are obtained directly from the D4RL benchmark, while the *Learned Score* denotes the raw return achieved by the algorithm during evaluation. For the AntMaze domain, the *Normalized Score* can be regarded as the task *success rate* (divided by 100).

D.1 Hyperparameter Settings for Online Finetuning

The core hyperparameter settings for COOL-DRL during the online finetuning phase are determined by the proposed exploration strategy, as the monotonic quantile value network architecture does not introduce new manual tuning parameters. The key exploration parameters are the perturbation scale ϵ and the temperature β used for candidate action sampling. For perturbation scale ϵ: We tested two perturbation magnitudes, set to 5% and 10% of the maximum action value. Since actions are normalized to the range $[-1, 1]$, these correspond to 0.05 and 0.1 perturbation, respectively. The specific hyperparameter choices across different environments are detailed in Table 2. For sampling temperature β: We set 2 for halfcheetah-me and hopper-mr/me, 5 for walker2d-mr, and 1 for all other tasks.

Additionally, we set the ratio of offline data usage to 0.5. However, following the established practice of WSRL for stable environments, the halfcheetah environment exclusively utilizes online data.

Algorithm 1 COOL-DRL

Require: The offline dataset $\mathcal{D}_{\text{off}}$, number of offline iterations N_{off}, discount factor γ, target network update rate τ, Critic and Actor learning rate, offline penalty coefficient α and threshold ζ

// Offline pretraining

$F_\theta^{\text{off}}, \pi_\phi^{\text{off}} \leftarrow$ Train via CODAC updates (Ma et al., 2021) with the proposed MQVN architecture in Fig. 3

Require: Offline trained $F_\theta^{\text{off}}, \pi_\phi^{\text{off}}$, online replay buffer $\mathcal{R}$, number of online iterations N_{on}

$F_\theta \leftarrow F_\theta^{\text{off}}, \pi_\phi \leftarrow \pi_\phi^{\text{off}}, \mathcal{R} \leftarrow \emptyset$

// Online finetuning

while step $\leq N_{\text{on}}$ **do**

  Obtain candidate actions according to (12)

  Compute the mean and standard deviation of the candidate actions via (10) and (11)

  Sample action a via (14) and interact with the environment

  $(s, a, s', r) \leftarrow \text{interact}(a, \text{environment})$

  $\mathcal{R} \leftarrow \mathcal{R} \cup (s, a, s', r)$

  Sample a mini-batch from $\mathcal{D}_{\text{off}} \cup \mathcal{R}$

  $F_\theta \leftarrow$ CODAC quantile update without penalty, i.e. $\mathcal{L}_\kappa(\delta; \tau)$ in (2)

  $\pi_\phi \leftarrow$ CODAC policy update

end while

Table 1: Basic hyperparameter settings.

Hyperparameters	Value
hidden dim	256
Actor & Critic hidden layers	2 for MuJoCo, 4 for Antmaze
Optimizer	Adam (Kingma & Ba, 2014)
Batch size	256
Critic Learning rate	3×10^{-4}
Actor Learning rate	1×10^{-4}
Entropy	True
Discount factor γ	0.99
Target update rate τ	0.005

Furthermore, due to the slower convergence typical of the AntMaze tasks, we uniformly set the critic update frequency to 4 times, aligning with the settings used for existing methods like WSRL and SO2. For MuJoCo tasks, we use a single critic update per environment step.

Table 2: Main hyperparameter during online finetuning.

Task	Exploration scale ϵ	Task	Exploration scale ϵ
halfcheetah-m	10%	halfcheetah-me	10%
hopper-m	5%	hopper-me	5%
walker2d-m	5%	walker2d-me	10%
halfcheetah-mr	5%	antmaze-ud	5%
hopper-mr	10%	antmaze-md	5%
walker2d-mr	5%	antmaze-ld	5%

D.2 Hyperparameter Settings for Offline Training

For the offline pre-training phase, COOL-DRL utilized a tuning strategy similar to CODAC for the primary conservatism hyperparameters, including the quantile penalization coefficient α and the threshold parameter ζ. The specific offline hyperparameter settings are listed in Table 3. A key difference is that while CODAC required differential learning rates and entropy tuning for its critic network, our method consistently employed a fixed learning rate across all tasks and did not utilize entropy backup in the value function updates. In addition, owing to the enforced quantile monotonicity of COOL-DRL, we restrict conservative penalization during offline training to quantile

values with τ greater than 0.5 for the hopper and walker2d-m/me tasks. This design choice mitigates overly aggressive value suppression while preserving the intended regularization effect.

Table 3: Main hyperparameter during offline training.

Task	Penalization coefficient α	Threshold parameter ζ
halfcheetah-m	10	10
hopper-m	5	10
walker2d-m	5	5
halfcheetah-mr	1	10
hopper-mr	1	5
walker2d-mr	5	10
halfcheetah-me	5	10
hopper-me	10	15
walker2d-me	10	5
antmaze-ud	10	10
antmaze-md	5	10
antmaze-ld	5	10

D.3 Implementation Details for Baselines

We implement CODAC (Ma et al., 2021) within our JAX-based framework in the offline stage, faithfully following the algorithmic design described in the original paper. Hyperparameters are tuned according to the recommendations provided by the authors to ensure strong offline performance. During the online finetuning phase, the conservative penalty is removed, consistent with the procedure used in COOL-DRL. As a result, the CODAC finetune baseline follows the same offline-to-online transition protocol and differs only in that it does not incorporate our proposed monotonic quantile architecture or quantile-based exploration strategy. For the experimental results involving CalQL (Nakamoto et al., 2023), IQL (Kostrikov et al., 2022), and WSRL (Zhou et al., 2025), we utilized the official code repository provided by the WSRL for reproducibility, ensuring that their original, established optimal parameter settings were retained. The results for SO2 (Zhang et al., 2024) were derived using its official parameter settings. Since SO2 and WSRL both belong to the ensemble category of algorithms, we maintained the same critic network ensemble size of 10 for both methods where applicable. Additionally, the update frequency for the critic networks in both SO2 and WSRL was uniformly set to 4 updates per environment step for all tasks.

D.4 Computing Infrastructure

Performance associated experiments are conducted on a computing platform equipped with an NVIDIA A100 GPU with 40 GB of memory, Intel(R) Xeon(R) Gold 6240R CPU @ 2.40GHz, on the Ubuntu 22.04 LTS system.

E Additional Experimental Results

Additional quantile curve visualizations. We provide additional quantile curve visualizations as shown in Fig. 12, comparing COOL-DRL and CODAC across all medium-expert tasks, with the corresponding online SAC Q-values shown as a reference. CODAC produces severely disordered and highly biased quantile estimates, whereas our method strictly preserves quantile monotonicity and substantially reduces estimation bias.

Additional sensitivity analyses. We further examine the effect of the exploration scale $\epsilon \in \{0.03, 0.05, 0.07, 0.10, 0.15\}$ on hopper-m and walker2d-me, with each configuration evaluated over three random seeds. On hopper-m, performance remains stable for moderate exploration scales, while overly large ϵ values lead to gradual performance degradation and increased variance, indicating that excessive exploration can disrupt stable policy refinement. In contrast, walker2d-me exhibits consistently strong performance with low variance across all settings, demonstrating robustness to the choice of ϵ. Typically, $\epsilon = 0.07$ provides consistently strong performance across all evaluated

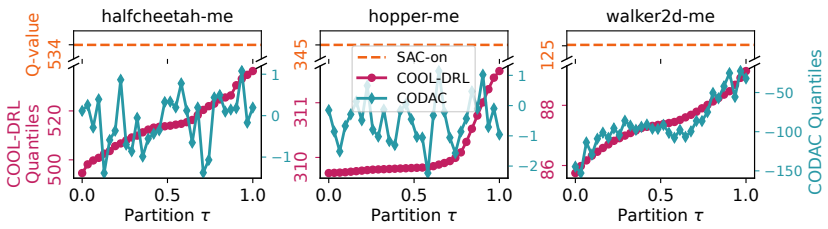


Figure 12: Quantile value estimates on medium-expert tasks after offline training.

tasks. These results further confirm that the proposed exploration strategy maintains stable learning dynamics and is not overly sensitive to the exploration scale within a broad and practical range.

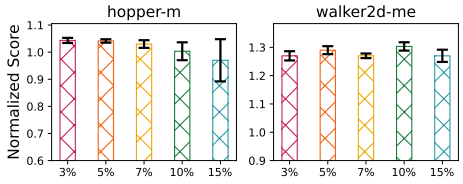


Figure 13: Sensitivity analysis of exploration scale on hopper-m and walker2d-me.

We additionally study the sensitivity of COOL-DRL to the exploration temperature β, as shown in Fig. 14. Overall, we find that the method is robust to a wide range of values, and setting $\beta \in \{1, 2\}$ yields consistently strong performance across all tasks. When $\beta = 0.5$, the uncertainty term is amplified with $1/\beta$, leading to overly aggressive exploration that destabilizes value-guided policy updates and may results in noticeable performance degradation. In contrast, larger values such as $\beta = 5$ can sometimes further improve performance, particularly on medium-replay datasets, where increased exploration diversity is beneficial. These results suggest that moderate temperatures provide a reliable default, yielding consistently stable performance across tasks.

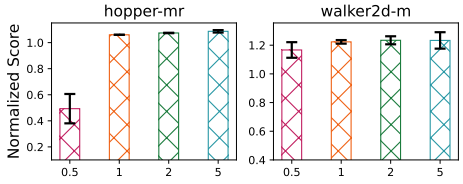


Figure 14: Sensitivity analysis of exploration temperature on hopper-mr and walker2d-m.

Additional comparison on other AntMaze tasks. In the experimental section, we report results on the AntMaze diverse tasks due to space limitations. To provide a more comprehensive evaluation, we present additional results on other AntMaze tasks in Table 4. Although WSRL achieves strong performance on the large-play task, its best results rely on ensemble methods, and its performance degrades significantly without them. Overall, COOL-DRL maintains competitive performance and stability across these additional tasks.

Performance under fixed exploration scale. According to the sensitivity analysis, the exploration scale remains stable within the range of 5%–10%. In practice, a fixed exploration scale of 5% already yields strong performance. Accordingly, as a practical guideline, we apply a fixed exploration scale of 5% across all tasks (the AntMaze results have already been reported using this fixed scale). The results are shown in Table 5.

Table 4: The performance comparison on other AntMaze tasks.

Methods	umaze	medium-p	large-p
IQL	$0.97_{\pm 0.02}$	$0.83_{\pm 0.06}$	$0.38_{\pm 0.06}$
CalQL	$0.90_{\pm 0.05}$	$0.92_{\pm 0.04}$	$0.83_{\pm 0.07}$
SO2	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$
WSRL	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$
WSRL w/o ensemble	$0.97_{\pm 0.04}$	$0.92_{\pm 0.06}$	$0.60_{\pm 0.15}$
COOL-DRL	$1.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.88_{\pm 0.02}$

Table 5: The performance under fixed exploration scale for all tasks.

	IQL	CalQL	SO2	WSRL	COOL-DRL (fixed)
halfcheetah-m	0.64	0.48	0.99	0.96	$1.11_{\pm 0.05}$
hopper-m	0.92	0.97	0.33	0.66	$1.04_{\pm 0.01}$
walker2d-m	0.80	0.87	0.98	1.11	$1.23_{\pm 0.03}$
halfcheetah-mr	0.54	0.48	1.19	0.94	$1.13_{\pm 0.04}$
hopper-mr	0.91	0.87	0.84	0.65	$1.06_{\pm 0.05}$
walker2d-mr	0.92	0.86	0.95	1.04	$1.09_{\pm 0.02}$
halfcheetah-me	0.94	0.48	1.05	1.00	$1.02_{\pm 0.03}$
hopper-me	0.96	0.99	0.73	0.86	$1.11_{\pm 0.00}$
walker2d-me	1.10	1.07	1.09	1.18	$1.28_{\pm 0.01}$

F Further Discussion on Stability

The offline penalty applied across quantiles in CODAC disrupts quantile monotonicity and introduces significant bias. The monotonicity design of our method COOL-DRL is intended to mitigate this value bias, thereby avoiding large corrections during online finetuning and improving stability.

First, we directly examine the learned quantile values. As shown in Fig. 2 and Fig. 11, the quantile estimates produced by our method are consistently closer to the reference Q-values from online SAC, i.e., requiring smaller corrections at the onset of online finetuning, whereas CODAC exhibits severely biased value ranges, as summarized in Table 6.

Table 6: Range of quantile values at the end of offline training.

Task	Quantile (CODAC)	Quantile (Ours)	Q-value (SAC-on)
halfcheetah-m	-2 $\sim$ 2	585 $\sim$ 615	713
hopper-m	0 $\sim$ 250	320 $\sim$ 326	349
walker2d-m	-2 $\sim$ 2	394 $\sim$ 405	429

The second evidence can be observed from Kendall correlation coefficient of Q-value ordering, as proposed in SO2, which measures value bias in rank consistency. Intuitively, if an action a_1 is better than a_2, a well-estimated value function should satisfy $Q(s, a_1) > Q(s, a_2)$. As shown in Fig. 11, CODAC yields Kendall coefficients close to 0, indicating nearly random ordering, whereas our method achieves significantly higher Kendall correlation.

Together, these indicate that enforcing monotonicity not only intrinsically reduces quantile misordering, but also alleviates value magnitude and ranking bias, thereby mitigating instability. Our ablation studies on monotonicity also support this claim.

G Monotonicity and Flexibility

The quantile function is inverse cumulative distribution function (ICDF), and its monotonicity is an inherent property. Enforcing monotonicity does not reduce approximation flexibility since the

ground-truth itself is monotonic. Instead, the proposed architecture prevents invalid (non-monotonic) solutions that may arise. It restricts the hypothesis space to valid functions, which can reduce noise introduced by non-monotonic approximations.

To further validate this, we conduct purely online experiments by applying both our MQVN and CODAC (MLP-based) architecture within same DRL framework, while keeping other hyperparameters same. Experiments are run for 500K steps. The results show that our architecture achieves better performance. Intuitively, the structure constrains the search space to where it should lie, rather than limiting its expressivity.

Table 7: Comparison between MQVN and an MLP-based architecture in purely online experiments.

Task	MLP-based (CODAC)	MQVN (Ours)
halfcheetah	$0.58_{\pm 0.23}$	$0.74_{\pm 0.06}$
hopper	$0.66_{\pm 0.25}$	$0.89_{\pm 0.09}$
walker2d	$1.08_{\pm 0.06}$	$1.09_{\pm 0.04}$

H Computation Cost

We report the computational cost on hopper-m, measured over 100 training steps on a V100 GPU. The average running times are 1.0s for IQL, 2.1s for CalQL, 3.6s for WSRL, 3.1s for CODAC, and 3.7s for our method. Compared with CODAC, our method introduces modest overhead from the monotonic positional embedding and exponential weighting design. In both CODAC and our method, the primary computational cost arises from the sampling procedure used for the quantile penalty term, while the relatively high cost of WSRL mainly stems from its ensemble-based design.

I Additional Experimental Results

In this section, we present additional experiments to further investigate the architectural components, generalization, computational efficiency, and hyperparameter sensitivity of COOL-DRL.

I.1 Additional Architectural and Exploration Ablations

We conduct additional ablation studies to examine the respective contributions of the monotonic quantile positional embedding (MQPE), the monotonic architecture, and the exploration strategy. Besides final performance, we evaluate stability during the offline-to-online (O2O) transition to better understand the role of each component.

Effect of MQPE versus the IQN cosine embedding. To isolate the contribution of MQPE, we replace it with the standard cosine embedding used in Implicit Quantile Networks (IQN), while retaining the remaining MQVN architecture. Unlike MQPE, the IQN cosine embedding does not preserve the theoretical monotonicity guarantee of the complete quantile network with respect to the quantile fraction τ, although the subsequent MQVN layers retain their monotonicity constraints.

As shown in Table 8, MQPE achieves higher final performance on both tasks, with a more pronounced improvement on hopper-me. Although the differences in final performance are relatively modest, replacing MQPE with the IQN cosine embedding produces an O2O performance drawdown of approximately 0.4 on hopper-me. These observations suggest that MQPE contributes to both final performance and stability during online adaptation.

Joint ablation of monotonicity and exploration. We further investigate the respective contributions of the monotonicity constraints and exploration strategy by comparing COOL-DRL with variants that remove either component. For a unified comparison, we report final performance and O2O drawdown on hopper-me and walker2d-me. O2O drawdown denotes the largest performance decline observed during the initial offline-to-online transition, with negative values indicating a decrease in performance.

Table 8: Comparison between MQPE and the standard IQN cosine embedding. Results are reported as mean $\pm$ standard deviation.

Task	MQVN with MQPE	MQVN with IQN
hopper-me	1.113 ± 0.003	1.057 ± 0.043
walker2d-m	1.234 ± 0.028	1.211 ± 0.005

Table 9: Joint ablation of monotonicity and exploration. The score measures final performance, while O2O drawdown measures the largest performance decline during the initial offline-to-online transition.

Task	Method	Score	O2O drawdown
hopper-me	COOL-DRL	1.113	0
	w/o Monotonicity	0.971	-0.52
	w/o Exploration	0.932	0
walker2d-me	COOL-DRL	1.294	-0.09
	w/o Monotonicity	1.114	-0.15
	w/o Exploration	1.097	-0.03

As shown in Table 9, removing either component reduces final performance on both tasks. However, their effects on O2O stability differ. Removing the monotonicity constraints results in larger performance drawdowns on both tasks, whereas removing the exploration strategy primarily reduces final performance without consistently increasing transition drawdown. These findings suggest that the monotonic architecture contributes to stable offline-to-online adaptation, while the exploration strategy plays an important role in improving final performance. The two components therefore provide complementary benefits.

I.2 Extended Benchmark Evaluation

To evaluate the generality of COOL-DRL beyond the 15 MuJoCo and AntMaze tasks considered in the original evaluation, we conduct additional experiments on expert-level MuJoCo datasets and Adroit manipulation tasks.

Expert-level MuJoCo datasets. We additionally evaluate COOL-DRL on three expert-level MuJoCo datasets. The results are summarized in Table 10.

COOL-DRL achieves scores above 1.0 on all three expert-level datasets, demonstrating its performance in settings with high-quality offline demonstrations. These datasets were not prioritized in the original evaluation because offline policies trained on expert demonstrations can already attain high performance, leaving relatively limited room for measuring improvements from subsequent online interaction. Instead, the original experiments included medium-expert datasets, which provide high-quality offline data while retaining meaningful opportunities for online adaptation. The additional expert-level results complement the original evaluation by covering these higher-quality data settings.

Adroit manipulation tasks. We further extend the evaluation to two Adroit manipulation tasks, pen-human-v0 and pen-cloned-v0, to examine performance beyond the original benchmark domains. The IQL and SO2 results on pen-human-v0 are taken from the SO2 paper. Since an IQL result on pen-cloned-v0 was unavailable in the referenced results, we reproduced it using the corresponding implementation.

As shown in Table 11, COOL-DRL achieves a higher reported score than both IQL and SO2 on pen-human-v0. On pen-cloned-v0, however, its score is lower than that of IQL, indicating that the performance improvements observed on pen-human-v0 do not extend uniformly to all Adroit datasets. Together, these experiments provide a broader assessment of COOL-DRL across locomotion and dexterous manipulation tasks.

Table 10: Performance of COOL-DRL on additional expert-level MuJoCo datasets. Results are reported as mean ± standard deviation.

Task	COOL-DRL
halfCheetah-expert	1.014 ± 0.008
hopper-expert	1.113 ± 0.013
walker2d-expert	1.273 ± 0.036

Table 11: Performance comparison on additional Adroit manipulation tasks. A dash indicates an unavailable result.

Task	IQL	SO2	COOL-DRL
pen-human-v0	0.607	0.706	0.915 ± 0.115
pen-cloned-v0	0.544 ± 0.099	–	0.430 ± 0.026

I.3 Additional Sensitivity to the number of action samples.

Multiple action samples are commonly used in offline RL algorithms such as BEAR, CQL, and CODAC, with $n = 10$ being a frequently adopted setting. To examine the sensitivity of COOL-DRL to this choice, we compare $n = 5$ and $n = 10$ on three medium-level MuJoCo tasks.

As shown in Table 12, the performance differences between the two settings are relatively small across all three tasks. Although $n = 10$ achieves slightly higher mean scores, reducing the number of action samples to $n = 5$ retains comparable performance. These results suggest that COOL-DRL is not highly sensitive to the number of action samples within the evaluated range, allowing a trade-off between sampling cost and performance.

I.4 Additional Performance under an equal computation budget.

O2O methods are typically compared using the same number of environment-interaction steps. However, differences in training time per step can result in substantially different computational costs. To provide an additional comparison, we evaluate IQL, CalQL, WSRL, and COOL-DRL under an equal wall-clock computation budget.

Based on the measured training time per step, we convert the common computation budget into the corresponding number of online steps for each method. Under this budget, IQL, CalQL, WSRL, and COOL-DRL are allocated 500K, 250K, 140K, and 135K online steps, respectively, where K denotes one thousand environment-interaction steps.

As shown in Table 13, COOL-DRL achieves the highest reported score on seven of the nine tasks. On walker2d-m, WSRL achieves a slightly higher score, while on HalfCheetah-ME, IQL achieves a marginally higher score. Despite receiving substantially fewer online interaction steps than IQL and CalQL, COOL-DRL maintains competitive performance across the evaluated tasks. These results complement the standard interaction-based evaluation by providing additional evidence of performance under a fixed computation budget.

I.5 Additional Hyperparameter Sensitivity on offline hyperparameters

We additionally examine the sensitivity of COOL-DRL to the offline conservative-constraint hyperparameters α and ζ. The coefficient α controls the strength of the conservative penalty, whereas ζ determines its threshold. Both parameters are used exclusively during offline training, and the corresponding constraints are removed during online fine-tuning. Therefore, these hyperparameters are not directly associated with the MQVN architecture or exploration strategy.

We first investigate the effect of α on halfcheetah-mr, hopper-m, and walker2d-me by comparing $\alpha = 5$ and $\alpha = 10$.

As shown in Table 14, both settings yield broadly comparable performance. While $\alpha = 5$ produces higher mean scores on halfcheetah-mr and hopper-m, $\alpha = 10$ achieves a slightly higher mean score

Table 12: Sensitivity to the number of action samples. Results are reported as mean ± standard deviation.

Task	$n = 5$	$n = 10$
halfCheetah-m	1.12 ± 0.05	1.13 ± 0.06
hopper-m	1.01 ± 0.05	1.04 ± 0.01
walker2d-m	1.19 ± 0.09	1.23 ± 0.03

Table 13: Performance comparison under an equal wall-clock computation budget. The number in parentheses indicates the corresponding number of online environment-interaction steps. COOL-DRL results are reported as mean ± standard deviation.

Task	IQL (500K)	CalQL (250K)	WSRL (140K)	COOL-DRL (135K)
halfCheetah-m	0.64	0.47	0.80	0.90 ± 0.02
hopper-m	0.92	0.94	0.74	1.02 ± 0.01
walker2d-m	0.80	0.79	0.95	0.92 ± 0.33
halfCheetah-mr	0.54	0.46	0.79	0.90 ± 0.01
hopper-mr	0.91	0.82	0.45	0.95 ± 0.16
walker2d-mr	0.92	0.76	0.91	0.97 ± 0.03
halfCheetah-me	0.94	0.46	0.91	0.93 ± 0.02
hopper-me	0.96	0.76	0.56	1.05 ± 0.09
walker2d-me	1.10	1.05	1.00	1.20 ± 0.06

on walker2d-me. These results indicate that the effect of α varies across tasks without producing a consistent performance advantage for either setting.

We further evaluate sensitivity to the threshold ζ on hopper-m.

As shown in Table 15, the two threshold settings produce comparable performance, with a modestly higher mean score at $\zeta = 10$. Together, these experiments suggest that COOL-DRL exhibits moderate robustness to variations in its offline conservative-constraint hyperparameters within the evaluated ranges.

J ADEQUATE COMPARISON WITH PRIOR NON-CROSSING METHODS

Mathematically, both the pure online method NC-QR-DQN Zhou et al. (2020) and the offline method CQR Bai et al. (2024) compute quantile values using an architecture that maps a discrete set of quantile fractions $[\tau_0, \tau_1, ..., \tau_{N-1}]$ to quantile estimates $[F_\theta(\tau_0), F_\theta(\tau_1), ..., F_\theta(\tau_{N-1})]$, where the non-crossing mechanism is enforced through *cumulative summation*. This design has two limitations:

(1) Lack of functional consistency. For example, consider two input fraction sets [0.1, 0.2, 0.3, 0.4] and [0.2, 0.4, 0.6, 0.8]. The predicted value $F_\theta(0.2)$ may differ across these inputs, indicating that $F_\theta(\tau)$ is actually not a well-defined quantile function over τ.

(2) Lack of global and theoretical monotonicity guarantees. Monotonicity is enforced only within each input fraction set, e.g., $F_\theta(0.4) \geq F_\theta(0.3) \geq F_\theta(0.2) \geq F_\theta(0.1)$, but this does not guarantee $F_\theta(0.2) \geq F_\theta(0.15)$ since 0.15 is not included in the input set.

In contrast, our method models a continuous mapping $\tau \mapsto F_\theta(\tau)$ and enforces monotonicity via the proposed MQPE and MQVN architectural constraints, with the provable guarantee $\partial F_\theta(\tau)/\partial \tau \geq 0, \forall \tau \in (0, 1)$. This ensures globally consistent and monotonic quantile estimates.

Empirically, we incorporate additional experiments comparing our method with CQR quantile enforcement under the same training protocol. The results are shown in the Table 16, demonstrating the superiority of our method.

Table 14: Sensitivity to the conservative penalty coefficient α. Results are reported as mean $\pm$ standard deviation.

Task	$\alpha = 5$	$\alpha = 10$
halfCheetah-mr	0.549 ± 0.120	0.503 ± 0.005
hopper-m	0.682 ± 0.066	0.637 ± 0.061
walker2d-me	1.064 ± 0.024	1.091 ± 0.017

Table 15: Sensitivity to the conservative-constraint threshold ζ on hopper-m. Results are reported as mean $\pm$ standard deviation.

Task	$\zeta = 10$	$\zeta = 15$
hopper-m	0.682 ± 0.066	0.643 ± 0.082

Table 16: Performance comparison with the non-crossing method.

Task	CQR	COOL-DRL (Ours)
halfcheetah-m	$0.79_{\pm 0.01}$	$1.13_{\pm 0.06}$
hopper-m	$0.96_{\pm 0.15}$	$1.04_{\pm 0.01}$
walker2d-m	$0.95_{\pm 0.16}$	$1.23_{\pm 0.03}$
halfcheetah-mr	$0.79_{\pm 0.01}$	$1.13_{\pm 0.04}$
hopper-mr	$0.88_{\pm 0.09}$	$1.08_{\pm 0.01}$
walker2d-mr	$1.10_{\pm 0.01}$	$1.09_{\pm 0.02}$
halfcheetah-me	$0.98_{\pm 0.01}$	$1.03_{\pm 0.02}$
hopper-me	$1.07_{\pm 0.01}$	$1.11_{\pm 0.00}$
walker2d-me	$1.15_{\pm 0.07}$	$1.29_{\pm 0.04}$

K Related Work

Offline RL. The direct challenge in offline RL (Levine et al., 2020) is mitigating optimistic overestimation of out-of-distribution action values, which arises primarily from distributional shift between the behavior policy and the learned policy. Solutions of model-free methods typically fall into two categories: policy constraint methods (e.g., BCQ (Fujimoto et al., 2019), TD3+BC (Fujimoto & Gu, 2021), Diffusion-QL (Wang et al., 2023), DTQL (Chen et al., 2024), A2PR (Liu et al., 2024)), which restrict the learned policy close to the behavior policy; and value regularization methods (e.g., CQL (Kumar et al., 2020), MCQ (Lyu et al., 2022), CSVE (Chen et al., 2023), ILQ (Liu et al., 2025)), which actively suppress OOD value estimates. We omit discussion of model-based methods (Park et al., 2026), as they are beyond the scope of our method. CODAC (Ma et al., 2021), a distributional offline reinforcement learning method, also employs value regularization by penalizing quantile estimates. However, uniformly penalizing all quantiles introduces stronger pessimism than conservative Q-learning, which limits the effective transfer of the well-known advantages of distributional reinforcement learning from online to offline settings. In particular, our experiments exhibit that the degree of conservatism required for offline training severely degrades the structural integrity of quantile value estimates, breaking their inherent monotonicity. This excessive bias substantially impairs performance during the early stages of offline-to-online finetuning, undermining the potential benefits of distributional value representations. In fact, online distributional reinforcement learning methods such as IQN (Dabney et al., 2018a) and FQF (Yang et al., 2019) also suffer from the quantile crossing issue; however, in the absence of explicit quantile penalization, this problem is typically less pronounced. Although Zhou et al. (Zhou et al., 2020) and CQR (Bai et al., 2024) later proposed non-crossing quantile regression to enforce monotonicity over fixed quantile partitions, this approach does not fundamentally guarantee global monotonicity of quantile value estimates. Therefore, we propose COOL-DRL, which theoretically guarantees quantile monotonicity regardless of the strength of offline penalization, thereby reducing bias in quantile value estimation during offline training.

Offline-to-Online RL. Offline-to-online RL aims to further optimize an offline-trained policy with limited online interaction. Early methods like Lee et al. (2021) focused on mitigating bootstrap

error through offline and online replay buffer balance and ensemble methods. CalQL addresses biased Q-values resulting from offline excessive conservatism by using maximum return estimates for calibration (Nakamoto et al., 2023), thereby alleviating initial performance dips. Other research explores continuous policy expansion or value-policy alignment, such as PEX (Zhang et al., 2023) and SO2 (optimistic ensemble Q-learning) (Zhang et al., 2024). WSRL discussed the effectiveness of offline dataset pruning during online finetuning (Zhou et al., 2025). Other methods address Q-value bias by introducing an online pretraining stage to learn a new Q-value function (Shin et al., 2025), or by synthesizing offline and online data to improve performance (Huang et al., 2025). Our method distinguishes itself by approaching offline-to-online transfer stability from a distributional value learning perspective. Unlike prior work focusing on policy expansion or Q-value calibration, our approach fundamentally prevents intrinsic quantile distortion by guaranteeing monotonicity. This ensures stable transfer, and we further leverage the resulting quantile-based uncertainty to design an effective online exploration strategy that maximizes asymptotic performance.

L Limitations and Discussion

While COOL-DRL is designed to improve stability during the offline-to-online transition, emphasizing stable transfer may, in some cases, reduce policy plasticity. As illustrated in Fig. 5(d), the CODAC finetune exhibits a performance drop at the onset of online finetuning, yet it is eventually able to refactor the policy more aggressively and achieve slightly higher asymptotic performance. This observation may highlights an inherent trade-off between *stability* and *plasticity* in offline-to-online reinforcement learning. Developing principled mechanisms to better balance stable transfer with adaptive plasticity remains an important direction for future research.

M Broader Impacts

This work focuses on improving value estimation in offline-to-online reinforcement learning. It is not expected to introduce direct risks related to physical harm, security vulnerabilities, discrimination, or human rights violations. This work may improve the reliability of reinforcement learning systems, enabling more efficient decision-making in applications such as robotics.