DISTRIBUTIONAL REINFORCEMENT LEARNING WITH DIFFUSION BRIDGE CRITICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in diffusion-based reinforcement learning (RL) methods have demonstrated promising results in a wide range of continuous control tasks. However, existing works in this field focus on the application of diffusion policies while leaving the diffusion critics unexplored. Actually, accurate value estimation is an important component for efficient RL, as policies are optimized by the values or gradients estimated by the critic model. Furthermore, given the stochasticity of most decision-making tasks, it has been confirmed that the critic is more appropriately depicted with a distributional model. Motivated by these points, we propose a novel distributional RL method with **D**iffusion **B**ridge **C**ritics (DBC). DBC directly models the inverse cumulative distribution function of return. This allows us to accurately capture the value distribution and prevents it from collapsing into a trivial Gaussian distribution owing to the strong distribution-matching capability of the diffusion bridge. Moreover, we further derive an analytic integral formula to address discretization errors in DBC, which is essential in value estimation. To our knowledge, DBC is the first work to employ the diffusion bridge model as the critic. Notably, DBC is also a plug-and-play component and can be integrated into most existing RL frameworks. Experimental results on MuJoCo robot control benchmarks and DMC benchmarks, demonstrating the superiority of DBC compared with previous distributional critic models.

1 INTRODUCTION

Diffusion models have recently been applied to reinforcement learning and demonstrate their superior performance in robot locomotion control and manipulation tasks Ren et al. (2024); Yang et al. (2023); Janner et al. (2022); Wang et al. (2022); Ajay et al. (2022); Chi et al. (2023) for their multimodality and powerful exploration capability. However, existing diffusion-based RL methods Yang et al. (2023); Ding et al. (2024; 2025); Celik et al. (2025); Wang et al. (2024a); Psenka et al. (2023) principally focus on leveraging diffusion models to enhance the policy expressiveness, while overlooking the potential of diffusion on modeling the distributional critic. In fact, accurate value estimation serves as the foundation for effective policy optimization. Without reliable estimates from the critic, the actor cannot be properly guided regardless of its architecture or expressiveness.

Distributional critics Bellemare et al. (2017) were originally proposed to better capture the inherent stochasticity in RL, thereby yielding more accurate value estimates for policy training. The core idea of existing works Kuznetsov et al. (2020); Dabney et al. (2018b) in this area is to represent the return distribution using finite discrete quantile levels to approximate the quantile values by neural networks. Although these methods can improve value estimation to some extent, it remains restricted by the discrete quantile representation and the limited expressiveness of neural networks. Besides, recent works such as Hu et al. (2025); Zhong et al. (2025) also attempt to apply diffusion models to depict the return distribution for its expressiveness. However, under the diffusion learning paradigm, the Bellman backup operator induces an accumulation of approximation errors, causing the diffusion critic to eventually collapse into a trivial Gaussian distribution.

To overcome these issues and obtain more accurate value estimation, we propose DBC, a novel distributional RL method that leverages expressive diffusion bridge models to depict the return distribution. Concretely, DBC learns the inverse cumulative distribution function (inverse CDF) of the return distribution. This formulation aligns naturally with quantile-based distributional reinforcement learning, while avoiding discrete quantile approximations and the Gaussian degradation problem

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.

Table 1: Comparison among distributional RL algorithms from different perspectives.

Method	Type	Non-Discretized	Expressiveness	Statistical Meaning	Non-Degradation
C51 Bellemare et al. (2017)	online	✗	✗	✓	✓
QR-DQN Dabney et al. (2018b)	online	✗	✗	✓	✓
IQN Dabney et al. (2018a)	online	✓	✗	✓	✓
DSAC Duan et al. (2022)	online	✓	✗	✗	✗
VD Hu et al. (2025)	online	✓	✓	✗	✗
VF Dong et al. (2025)	offline	✓	✓	✗	✗
DBC(*)	online	✓	✓	✓	✓

in vanilla diffusion critics. Besides, we also find that a key technical challenge in diffusion bridges for policy optimization lies in discretization errors arising from finite-time approximations. To address this challenge, we derive an analytic integral form that corrects the bias induced by discrete time steps, resulting in more accurate value estimation and stable policy improvement. Notably, DBC is a plug-and-play approach, which can be seamlessly integrated into existing RL algorithms without modifying the underlying policy architecture or optimization procedure. To summarize, our contribution is threefold:

- **Gaussian Degradation of Vanilla Diffusion Critic.** We identify and analyze the issue of directly employing diffusion models as critics in reinforcement learning, which leads to the degradation of itself to trivial Gaussian distributions with the Bellman backup operator. Consequently, it severely limits the expressiveness of the learned value distribution and undermines effective policy optimization.
- **Diffusion Bridge as Distributional Critic.** We propose DBC, a novel distributional RL method that employs diffusion bridge models to accurately learn the inverse CDF of the return distribution. With the anchor loss design and integral-consistent discretization technique, DBC is the first success that effectively incorporates diffusion bridge critics into distributional RL. We further establish finite-time and fixed-point guarantees for the anchor loss.
- **State-of-the-art Performance.** We evaluate DBC combined with SAC and TD3 on a range of MuJoCo robotic control tasks and DMC benchmarks and compare it with other representative distributional RL methods. Results demonstrate that DBC consistently achieves state-of-the-art performance, validating the effectiveness of diffusion bridge critics and highlighting the importance of an expressive diffusion-based critic in reinforcement learning.

2 Related Works

Diffusion-based Reinforcement Learning. Existing diffusion-based RL methods mostly focus on the technique of diffusion policy optimization. In offline RL, diffusion models Janner et al. (2022); Wang et al. (2022); Ajay et al. (2022) are employed to approximate the behavior policy in the offline dataset and enable policy optimization by sampling actions or trajectories that remain within the support of the diffusion model. In off-policy reinforcement learning, DIPO Yang et al. (2023) and QSM Psenka et al. (2023) optimize the diffusion policy by utilizing the gradient of the Q function. DACER Wang et al. (2024a) and DIME Celik et al. (2025) treat the diffusion policy as a black-box model and directly apply deterministic policy gradient with different techniques on entropy estimation. Besides, QVPO Ding et al. (2024), (SDAC) Ma et al. (2025a) perform the policy improvement with various choices on the weighted loss. Moreover, practical techniques for fine-tuning pretrained diffusion policies have also been proposed in DPPO Ren et al. (2024). Furthermore, FPO McAllister et al. (2025) and GenPO Ding et al. (2025) develop different tricks to approximate the log likelihood of the diffusion policy and try to optimize the diffusion policy with PPO loss Schulman et al. (2017). However, existing diffusion-based RL methods mostly focus on the diffusion policy and overlook how to effectively utilize the diffusion model as a powerful critic. In that case, our DBC is proposed to tackle the absence of efficient diffusion-based critics.

Distributional Critic Learning. Distributional reinforcement learning aims to learn the full probability distribution of value functions rather than merely estimating their expectations Bellemare et al. (2017; 2023). Early methods primarily rely on discretization or parametric assumptions. C51 discretizes the support of return distribution into a fixed set of atoms and learns a categorical distribution Bellemare et al. (2017). QR-DQN learns discrete quantile values via quantile regression Dabney et al. (2018b). IQN Dabney et al. (2018a) further introduces implicit quantile networks that map arbitrary quantile levels to corresponding quantile values; however, its simple MLP parameterization limits

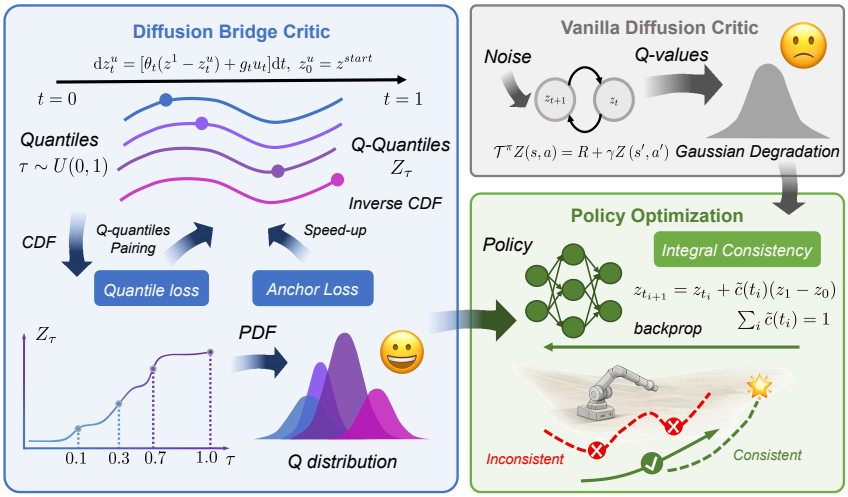


Figure 1: The training pipeline of Diffusion Bridge Critics. Compared with Vanilla Diffusion Critic, DBC explicitly models the inverse CDF of the return distribution and resolves the Gaussian Degradation Problem. Besides, the design of the integral-consistent discretization technique is also developed for accurate value estimation and stable policy optimization.

its ability to fit complex Q distributions. For continuous control, DSAC assumes Gaussian value distributions within SAC Duan et al. (2022; 2025), while TQC uses ensembles of quantile critics for stability at the cost of high parameter overhead Kuznetsov et al. (2020).

Very recently, diffusion and flow matching models have emerged as more expressive alternatives for value distribution learning. Diffusion models learn multimodal value distributions through iterative denoising Hu et al. (2025). Moreover, QDQ Zhang et al. (2024) utilizes consistency models to enable efficient uncertainty estimation via single-step sampling. Besides, flow matching models Agrawalla et al. (2026); Zhong et al. (2025); Chen et al. (2025); Dong et al. (2025) are applied to directly modeling return distributions generatively in offline or online RL settings. Notably, recent works Palenicek et al. (2025); Lee et al. (2025) on large-scale critic network architectures exhibit promising potential in RL, but this direction is outside the scope of this work: diffusion critics can also obtain better performance with larger networks.

Nevertheless, existing generative methods either rely on iterative sampling for scalar value estimation or introduce implicit distributional assumptions when generating return samples. In contrast, our method leverages diffusion bridges to learn the inverse CDF, enabling explicit value distribution modeling within a unified quantile framework. Table 1 provides a straightforward comparison that highlights the superiority of the proposed method across multiple aspects.

3 Preliminaries

3.1 Distributional Reinforcement Learning

Consider a Markov Decision Process (MDP) defined by the quintuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$. Here, $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, respectively, with state $\boldsymbol{s} \in \mathcal{S}$ and action $\boldsymbol{a} \in \mathcal{A}$ both represented as vectors. The state transition probability density is given by $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, $\mathcal{R}(\boldsymbol{s}, \boldsymbol{a})$ is a stochastic reward function, and $\gamma \in [0, 1)$ is the discount factor. A policy $\pi : \mathcal{S} \to \mathcal{A}$ maps states to actions. The random return $Z^\pi(\boldsymbol{s}, \boldsymbol{a})$ is defined as the discounted sum of future rewards $r_k \in \mathcal{R}$:

$$Z^\pi(\boldsymbol{s}, \boldsymbol{a}) = \sum_{k=0}^{\infty} \gamma^k r_k, \quad \boldsymbol{s}_0 = \boldsymbol{s},\ \boldsymbol{a}_0 = \boldsymbol{a}, \boldsymbol{a}_k \sim \pi(\cdot|\boldsymbol{s}_k),\ \boldsymbol{s}_{k+1} \sim \mathcal{P}(\cdot|\boldsymbol{s}_k, \boldsymbol{a}_k). \tag{1}$$

In contrast to traditional reinforcement learning, which estimates the expected return $Q^\pi(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}[Z^\pi]$, distributional reinforcement learning aims to model the full distribution of the one-dimensional random variable Z^π. The distributional Bellman operator $\mathcal{T}^\pi$ is defined as:

$$\mathcal{T}^\pi Z(\boldsymbol{s}, \boldsymbol{a}) \overset{D}{=} \mathcal{R}(\boldsymbol{s}, \boldsymbol{a}) + \gamma Z(\boldsymbol{s}', \pi(\boldsymbol{s}')),\ \boldsymbol{s}' \sim \mathcal{P}(\cdot|\boldsymbol{s}, \boldsymbol{a}), \tag{2}$$

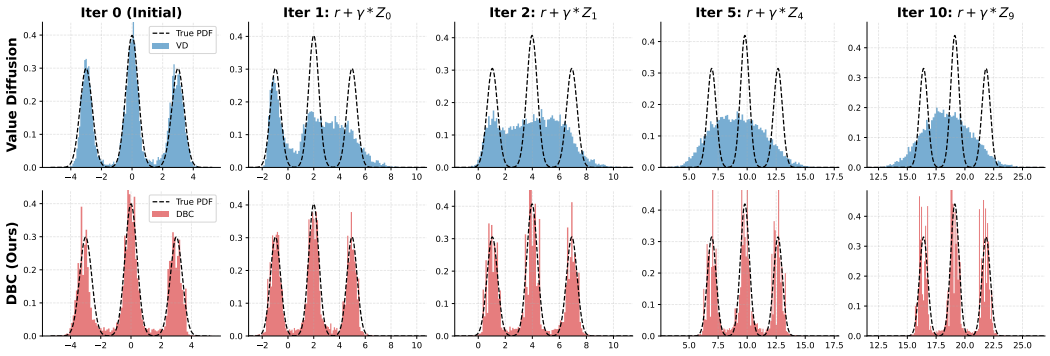


Figure 2: Top row: Value diffusion fits the iterative target $r + \gamma \cdot Z_{k-1}^{\mathrm{FM}}$. Bottom row: DBC fits the iterative target $r + \gamma \cdot Z_{k-1}^{\mathrm{DBC}}$. DBC introduces quantile conditioning τ, which consequently preserves multimodality of the distribution under iterative Bellman drift $Z_k \leftarrow r + \gamma Z_{k-1}$. We takes 100 inner training steps to fit Z_k for each method in each iteration and 10000 steps to fit the initial distribution in iteration 0.

where $\overset{D}{=}$ denotes equality in distribution. Under the p-Wasserstein distance, $\mathcal{T}^\pi$ is a γ-contraction operator Bellemare et al. (2017), which guarantees the existence and uniqueness of its distributional fixed point. We characterize the distribution via its quantile function $F^{-1}_{Z(\boldsymbol{s},\boldsymbol{a})}(\tau)$, where $\tau \in [0,1]$ is the quantile level.

$$\mathcal{L}_{\mathrm{QR}}(\theta) = \mathbb{E}_{(\boldsymbol{s},\boldsymbol{a})\sim\mathcal{D},\ \tau\sim U([0,1])}\left[\rho_\tau^\kappa\big(y - f_\theta(\boldsymbol{s},\boldsymbol{a},\tau)\big)\right], \quad \rho_\tau^\kappa(u) = |\tau - \mathbb{I}(u<0)| \cdot \mathcal{L}_\kappa^{\mathrm{Huber}}(u), \tag{3}$$

where y is the target sample and $\mathcal{L}_\kappa^{\mathrm{Huber}}$ is the huber loss with parameter κ (we set $\kappa = 1$ in our experiments). Because ρ_τ^κ is asymmetric, the order of its arguments matters; below we therefore write the losses in the two-argument form $\ell_\tau^\kappa(p,z) := \rho_\tau^\kappa(z-p)$, which always takes the predicted quantile value p first and the regression target z second.

3.2 Diffusion Bridge

Diffusion bridge models have achieved significant success across a range of applications, including image translation Liu et al. (2023); Li et al. (2023); Zhou et al. (2023); Albergo et al. (2023), image restoration Luo et al. (2023); Yue et al. (2024); Zhu et al. (2025); Wang et al. (2025), and video generation Wang et al. (2024b), which overcome the limitation of Gaussian prior in standard diffusion models. UniDB Zhu et al. (2025) is a unified framework demonstrating that *h*-transform-based models Zhou et al. (2023); Yue et al. (2024) can be unified through the lens of Stochastic Optimal Control (SOC) theory. To model the transition starting from a sample z^{start} in prior distribution to z^{end} in data distribution, UniDB Zhu et al. (2025) constructed the following Linear Quadratic Optimal Control problem Kappen (2008); Chen et al. (2023) as

$$\min_{u_t} \int_0^1 \frac{1}{2}\|u_t\|_2^2 \mathrm{d}t + \frac{\omega}{2}\|z_1^u - z^{\mathrm{end}}\|_2^2 \quad \text{s.t. } \mathrm{d}z_t^u = [\beta_t(z^{\mathrm{end}} - z_t^u) + g_t u_t]\mathrm{d}t,\ z_0^u = z^{\mathrm{start}}, \tag{4}$$

where z_t^u is the controlled diffusion process, u_t is the controller, $\int_0^1 \frac{1}{2}\|u_t\|_2^2 \mathrm{d}t$ is the transient cost, $\frac{\omega}{2}\|z_1^u - z^{\mathrm{end}}\|_2^2$ is the terminal cost with its penalty coefficient ω, β_t and g_t are two scalar-valued functions with a relationship $g_t^2 = 2\lambda^2\beta_t$ where the steady variance level λ^2 is a given constant. In this paper, the proposed DBC directly considers the deterministic optimal control problem Eq. (4) with fixed endpoints ($\omega \to \infty$ to make the controlled dynamics precisely converge to z^{end}) Chen et al. (2023) in UniDB. Therefore, we can explicitly model the mapping from the quantile level τ to the quantile value Z_τ, resulting in an output structure with clear statistical meaning. According to Pontryagin Maximum Principle Kappen (2008), UniDB provided the solution to the SOC problem Eq. (4) and the corresponding transition z_t between z^{start} and z^{end} as

$$u_t^* = g_t \frac{e^{-2\bar{\beta}_{t:1}}}{\bar{\sigma}_{t:1}^2}(z^{\mathrm{end}} - z_t^u), \text{ where } z_t^u = \xi(t)z^{\mathrm{start}} + (1-\xi(t))z^{\mathrm{end}},\ \xi(t) = e^{-\bar{\beta}_{0:t}}\frac{\bar{\sigma}_{t:1}^2}{\bar{\sigma}_{0:1}^2}, \tag{5}$$

where $\bar{\beta}_{t_1:t_2} = \int_{t_1}^{t_2} \beta_{t'}\mathrm{d}t'$ and $\bar{\sigma}_{t_1:t_2}^2 = \lambda^2(1 - e^{-2\bar{\beta}_{t_1:t_2}})$. For more details, please refer to UniDB Zhu et al. (2025).

4 DIFFUSION BRIDGE CRITIC

In this section, we first reveal the Gaussian degradation problem in vanilla diffusion critics. To address this issue, we propose the **D**iffusion **B**ridge **C**ritic, a distributional critic built upon the standard actor-critic framework. Unlike vanilla diffusion critics that directly model the raw return distribution, DBC models the quantile function $Z_\tau = F_Z^{-1}(\tau)$, a representation that is inherently more stable under iterative Bellman updates. Specifically, with the quantile loss, DBC learns a diffusion bridge that maps the quantile level τ to its corresponding return quantile Z_τ. To stabilize training, we design an **anchor loss** that provides direct supervision from sample quantiles. We further prove that, despite its biased target, the anchor loss reduces the finite-time fitting error when combined with the quantile loss (Theorem C.5), and that the anchored Bellman operator remains a γ-contraction whose fixed point deviates from the unanchored one by only $O(wK_{\text{tgt}}^{-1/2})$ (Theorem C.9). Furthermore, since both critic training and policy optimization rely on accurate endpoint values from the diffusion bridge, we propose an **integral-consistent discretization** technique that eliminates the endpoint bias introduced by naive Euler discretization, thereby providing unbiased value estimates for the entire training pipeline. The full pipeline of DBC is illustrated in Figure 1, and we detail these three components in the following subsections. We provide a brief overview of our method in Algorithm 1.

4.1 LIMITATION OF VANILLA DIFFUSION CRITICS

As illustrated in Figure 2, the distribution learned by the diffusion critic gradually collapses to a Gaussian as training progresses. This occurs because, despite the expressive power of diffusion models, the critic cannot be fully optimized within each limited update step under the RL bootstrapping paradigm. The accumulated approximation errors eventually cause the learned distribution to degenerate into a Gaussian. We provide a theoretical proof of this degeneration based on the Central Limit Theorem.

Theorem 4.1 (Gaussian Degradation of Diffusion Critics)**.** *Diffusion Critics f_θ finally degrades into a Gaussian distribution $\mathcal{N}(Q(s,a),\sigma^2)$ under the Bellman backup operator:*

$$\mathcal{T}^\pi Z(s,a) \overset{D}{=} \mathcal{R}(s,a)+\gamma Z(s',a'),\ Z \sim f_\theta(s,a). \tag{6}$$

The proof can be referred to Appendix B.1. Moreover, as shown in Figure 2, DBC preserves the multimodality of the return distribution throughout iterative updates, effectively addressing the Gaussian degradation problem observed in diffusion critics. This advantage stems from the fact that DBC explicitly models the relationship between the quantile level τ and the corresponding return distribution Z_τ. A detailed discussion is provided in Appendix B.1. A complete description of the DBC algorithm will be presented in the following subsections.

Algorithm 1 Training for DBC

Input: replay buffer $\mathcal{D}$, online parameters θ, target parameters ϕ, number of critic heads H, discount γ, target particles K_{tgt} and online quantile levels K_{onl}.

for $k = 1$ **to** N **do**
  Sample $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}', m) \sim \mathcal{D}$, and sample $\boldsymbol{a}' \sim \pi(\cdot \mid \boldsymbol{s}')$
  Generate target values $\{z_j\}_{j=1}^{K_{\text{tgt}}}$ by Eq. (17)
  Build the target particle set $\mathcal{Y} = \{y_j\}_{j=1}^{K_{\text{tgt}}}$ by Eq. (8)
  Sample quantile levels $\{\tau_i\}_{i=1}^{K_{\text{onl}}} \sim U([0,1])$ and set $z^{\text{start}}(\tau_i) \leftarrow \tau_i$
  Compute $\{\hat{y}_{\tau_i}\}_{i=1}^{K_{\text{onl}}}$ from $\mathcal{Y}$ by Eq. (10)
  for $h = 1$ **to** H **do**
    Calculate $\mathcal{L}^{(h)}$ according to Eq. (12)
  end for
  Update θ using $\frac{1}{H}\nabla_\theta \sum_{h=1}^{H} \mathcal{L}^{(h)}$
  Soft update for target network
end for

4.2 TRAINING PROCESS OF DBC

To resolve the Gaussian degradation problem, we propose diffusion bridge critics, which adopt a quantile-function parameterization for the distributional critic. Given a state-action pair $(\boldsymbol{s}, \boldsymbol{a})$, a quantile level $\tau \in (0,1)$, and a bridge time $t \in [0,1]$, the online network directly predicts the data endpoint:

$$\hat{z}_{\tau,t} = f_\theta(z_t, t, \tau, \boldsymbol{s}, \boldsymbol{a}), \tag{7}$$

where z_t is an intermediate variable along the diffusion bridge trajectory Eq. (5). For each critic update, the target network generates a set of particles $\{z_j^{\text{target}}\}_{j=1}^{K_{\text{tgt}}}$ via the bridge ODE, and the

Bellman transform yields:

$$\mathcal{Y} = \{y_j\}_{j=1}^{K_{\text{tgt}}}, \quad y_j := r + \gamma z_j^{\text{target}}. \tag{8}$$

Given $\mathcal{Y}$, we optimize a standard quantile loss that pairs each predicted quantile with all target particles to capture the distributional shape:

$$\mathcal{L}_t^{\text{quantile}} = \sum_{j=1}^{K_{\text{tgt}}} \sum_{i=1}^{K_{\text{onl}}} \ell_{\tau_i}^{\kappa}\Big(f_\theta(z_t, t, \tau_i, \boldsymbol{s}, \boldsymbol{a}),\, y_j\Big), \tag{9}$$

where $\{\tau_i\}_{i=1}^{K_{\text{onl}}} \subset (0,1)$ are quantile levels. However, this loss provides only *implicit* supervision per quantile level τ_i, and more critically, the bridge variable z_t for $t > 0$ is not naturally given by the network alone, it must be explicitly constructed from the target particles $\mathcal{Y}$. To this end, we compute the sample τ-quantile via order statistics:

$$\hat{y}_\tau := y_{(\lceil K_{\text{tgt}}\tau \rceil)}, \tag{10}$$

which is the minimizer of Eq. (9) with $\kappa = 0$. We provide a formal proof and further properties of sample quantiles in Appendix C.1. Using $\hat{y}_\tau$ together with the quantile level τ, we construct the bridge variable z_t for any $t \in [0,1]$; for $t = 0$, we simply set $z_0 = \tau$.

Anchor Loss. Beyond its role in constructing z_t, $\hat{y}_\tau$ also serves naturally as a supervision target, leading to our proposed *anchor loss*. For each quantile level τ_i, we impose a direct pointwise regression term via a robust Huber penalty:

$$\mathcal{L}_t^{\text{anchor}} = \sum_{i=1}^{K_{\text{onl}}} \mathcal{L}_\kappa^{\text{Huber}}\Big(f_\theta(z_t, t, \tau_i, \boldsymbol{s}, \boldsymbol{a}) - \hat{y}_{\tau_i}\Big). \tag{11}$$

The Huber loss provides robustness against individual mispredicted particles that might otherwise dominate the gradient. Combining the anchor loss with the standard quantile loss yields the total critic objective:

$$\mathcal{L}_t^{\text{total}} = \mathcal{L}_t^{\text{quantile}} + w \cdot \mathcal{L}_t^{\text{anchor}}, \tag{12}$$

where $w \geq 0$ controls the anchoring strength. We note two important properties of this objective. First, the sample quantile is a **biased** estimator (with bias $o(1/\sqrt{K_{\text{tgt}}})$; see Theorem C.4 in Appendix C.1), so the anchor loss should only be used as a light auxiliary regularizer. Second, despite this bias, we prove in Appendix C.2 that for any fixed training horizon there exists a sufficiently small $w > 0$ such that the combined objective attains a strictly lower finite-time error than the pure quantile loss Eq. (9) whenever the anchor bias is smaller than the remaining error (Theorem C.5, with no small-error linearization), and in Appendix C.3 that the anchored Bellman operator remains a γ-contraction whose fixed point deviates from the true one by only $O(wK_{\text{tgt}}^{-1/2})$, demonstrating the practical benefit of Eq. (11).

For critic updating, we adopt DropTop aggregation (See Appendix F) to mitigate value overestimation, and for gradient-based actor updates (e.g. SAC, TD3), backpropagation is performed through the full diffusion procedure.

4.3 Integral-consistent Discretization of DBC

The training objective above defines DBC in continuous bridge time, but policy optimization requires finite-step sampling from the critic. However, directly applying the bridge dynamics with a naive Euler discretization can introduce a systematic **endpoint bias**. To see this, substituting the UniDB solution into the controlled bridge dynamics yields the deterministic ODE

$$\frac{\mathrm{d}z_t}{\mathrm{d}t} = c(t)(z^{\text{end}} - z^{\text{start}}), \quad c(t) = \left(\beta_t + g_t^2 \frac{e^{-2\bar{\beta}_{t:1}}}{\bar{\sigma}_{t:1}^2}\right) \xi(t), \tag{13}$$

where z^{start} and z^{end} are the bridge endpoints. Since the continuous trajectory must satisfy $z_0 = z^{\text{start}}$ and $z_1 = z^{\text{end}}$, the coefficient satisfies $\int_0^1 c(t)\mathrm{d}t = 1$. Given a partition $0 = t_0 < t_1 < \cdots < t_M = 1$, a conventional Euler update gives

$$z_{t_{i+1}} = z_{t_i} + c(t_i)\Delta t_i (z^{\text{end}} - z^{\text{start}}), \quad \Delta t_i = t_{i+1} - t_i. \tag{14}$$

After M steps, the accumulated displacement is proportional to $\sum_{i=0}^{M-1} c(t_i)\Delta t_i$, which generally differs from the continuous integral $\int_0^1 c(t)\mathrm{d}t = 1$ for finite M. Consequently, the sampled trajectory

Table 2: Mean evaluation performance of different algorithms in the 5 MuJoCo benchmarks for 10^6 environment steps. All experiments are conducted with 5 random seeds, and the table reports the mean evaluation results across seeds.

Run	Ant-v5	HalfCheetah-v5	Hopper-v5	Humanoid-v5	Walker2d-v5
SAC	6170.8 ± 392.1	12738.1 ± 293.8	3648.7 ± 49.0	5170.7 ± 99.7	4912.7 ± 357.9
IQN	3449.0 ± 854.4	10973.9 ± 146.6	3282.0 ± 378.1	4771.8 ± 545.7	4728.4 ± 276.2
VD	3865.8 ± 957.5	4254.8 ± 502.1	1062.2 ± 19.0	4872.2 ± 233.4	2574.4 ± 275.2
VF	2634.4 ± 170.4	8215.9 ± 1441.3	3320.3 ± 26.6	5004.8 ± 58.5	2587.0 ± 1470.6
DSAC	2033.3 ± 625.5	12909.1 ± 195.5	3508.8 ± 22.3	3463.3 ± 232.0	3320.5 ± 1198.4
DBC(*)	**6481.9 ± 88.5**	**13172.7 ± 155.9**	**3754.9 ± 86.5**	**5877.2 ± 188.7**	**6162.8 ± 41.5**

may end at $z_{t_M} \neq z^{\text{end}}$, causing biased return samples for policy optimization. To remove this bias, we exploit the identity

$$c(t) = -\frac{\mathrm{d}\xi(t)}{\mathrm{d}t}, \quad \xi(0) = 1, \quad \xi(1) = 0. \tag{15}$$

Please refer to Appendix D for proof. Instead of approximating the interval contribution by $c(t_i)\Delta t_i$, we use the exact integral

$$\int_{t_i}^{t_{i+1}} c(t)\mathrm{d}t = \xi(t_i) - \xi(t_{i+1}). \tag{16}$$

This yields the integral-consistent update

$$z_{t_{i+1}} = z_{t_i} + \underbrace{\big(\xi(t_i) - \xi(t_{i+1})\big)}_{\tilde{c}(t_i)}(z^{\text{end}} - z^{\text{start}}). \tag{17}$$

Theorem 4.2 (Endpoint consistency of DBC discretization)**.** *For any partition $0 = t_0 < t_1 < \cdots < t_M = 1$, if $z_{t_0} = z^{start}$ and the trajectory is updated by Eq.* (17)*, then $z_{t_M} = z^{end}$ for any number of discretization steps M.*

Theorem 4.2 follows by telescoping the coefficients $\xi(t_i) - \xi(t_{i+1})$, and its full proof is provided in Appendix B.2. The resulting sampler preserves the bridge endpoint exactly while requiring only evaluations of $\xi(t)$, which is important for low-latency critic sampling with small M. Algorithm 2 summarizes the sampling procedure.

Policy training. After the critic is fully trained, we use the online network to generate K_{onl} return samples via Algorithm 2. These samples are averaged to obtain an expected return estimate q_{onl}, which is then provided to the actor for policy optimization.

Algorithm 2 Integral-consistent ODE Sampling for DBC

Input: online critic f_θ, state-action $(\boldsymbol{s}, \boldsymbol{a})$, quantiles $\{\tau_i\}_{i=1}^{K_{\text{onl}}}$ **Hyperparameters:** time grid $\{t_m\}_{m=0}^{M}$, interpolation $\xi(\cdot)$.

for $i = 1$ **to** K_{onl} (In Parallel) **do**
  $z \leftarrow z^{\text{start}} \leftarrow \tau_i$
  for $m = 0$ **to** $M - 1$ **do**
    $\hat{z}_{\tau_i, t_m} \leftarrow f_\theta(z, t_m, \tau_i, \boldsymbol{s}, \boldsymbol{a})$
    $z \leftarrow z + \tilde{c}(t_m)(\hat{z}_{\tau_i, t_m} - z^{\text{start}})$
  end for
  $z^{\text{end}}_{(i)} \leftarrow z$
end for

5 Experiments

In this section, we systematically evaluate the performance of DBC on multiple MuJoCo continuous-control benchmarks. We first compare DBC against several representative critic baselines to quantify its overall performance gains. We then assess DBC as a plug-and-play critic module by integrating it into different actor algorithms, thereby validating its generality across actor backbones. Finally, we conduct two ablation studies to analyze how key hyperparameters affect performance and training stability, including the time steps M and the anchor loss weight w.

5.1 Comparative Evaluation

We benchmark DBC against several critic baselines, including SAC (Haarnoja et al., 2018), DSAC (Duan et al., 2022), IQN (Dabney et al., 2018a), Value Diffusion (VD) (Hu et al., 2025), and Value Flows (VF) (Dong et al., 2025), all implemented within our codebase. Brief descriptions of these baselines are provided in Appendix F. Table 2 presents the main results. At each evaluation

Table 3: Mean evaluation performance of different algorithms in the 5 DMC tasks. All experiments are conducted with 5 random seeds, and the table reports the mean evaluation results across seeds.

Run	CartPole-v0	Humanoid-Run-v0	Humanoid-Stand-v0	Reacher-Easy-v0	Walker-Run-v0
DSAC	999.0 ± 0.4	1.6 ± 0.2	8.1 ± 0.7	363.8 ± 236.9	501.0 ± 21.6
IQN	999.1 ± 1.1	66.5 ± 56.3	710.7 ± 164.8	812.0 ± 142.4	671.9 ± 50.0
SAC	999.0 ± 0.9	77.4 ± 65.4	691.4 ± 145.2	871.3 ± 95.9	664.0 ± 29.1
DBC(*)	$\mathbf{999.3} \pm 0.2$	$\mathbf{138.9} \pm 23.6$	$\mathbf{854.1} \pm 37.3$	$\mathbf{972.6} \pm 8.0$	$\mathbf{675.1} \pm 57.7$

checkpoint, we roll out the policy for 10 episodes and average the returns to obtain a stable performance estimate. Detailed experimental settings are provided in Appendix G, and the complete breakdowns per-actor are reported in Appendix H. Details regarding the experimental runtime are provided in Appendix H.6.

As shown in Table 2, DBC achieves the best overall performance. DSAC, in contrast, suffers from a notable limitation: by explicitly parameterizing the return distribution with a single Gaussian, it lacks the representational capacity to capture rich, potentially multimodal return distributions, which consequently degrades its empirical performance. In addition, we evaluated the methods on the DM-Control suite, with results reported in Table 3. DBC again demonstrated the most favorable performance. For the relatively simple CartPole task, all critic methods achieved scores approaching the maximum. Learning curves of these baselines are demonstrated in Figure 3.

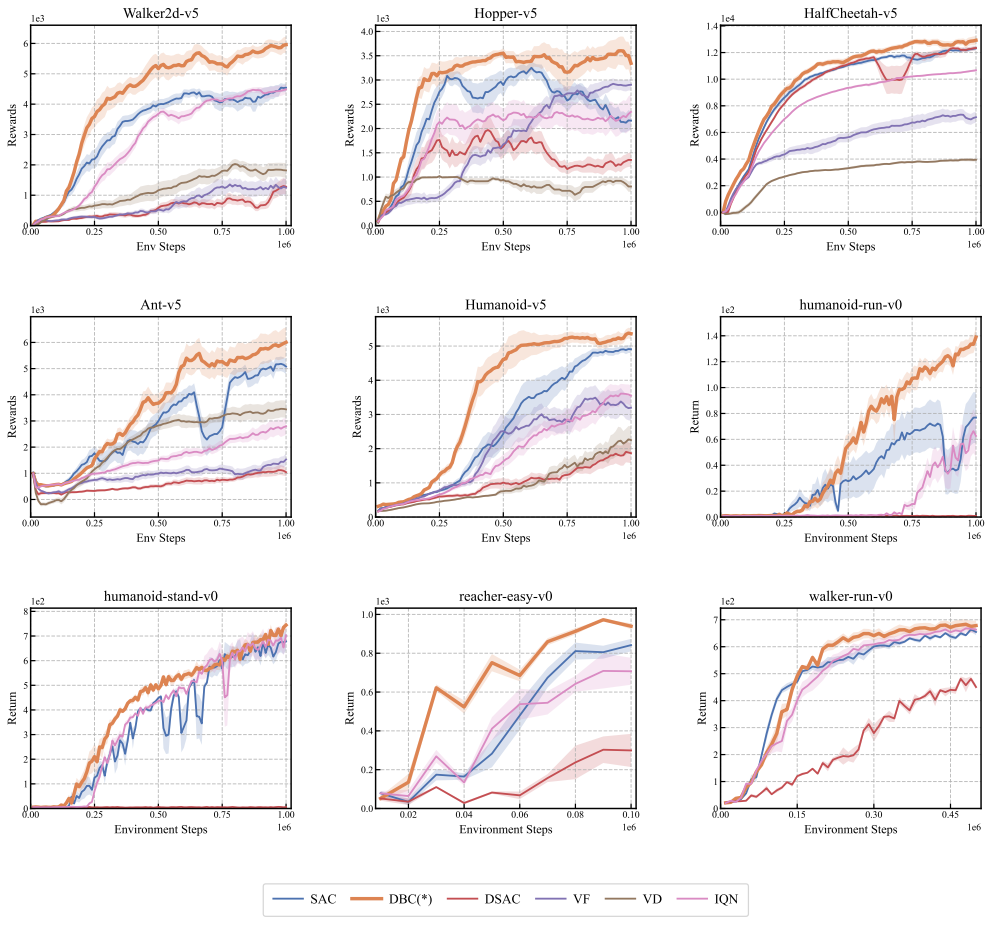


Figure 3: Learning curves of different algorithms on five MuJoCo benchmarks and four DM-Control tasks. The x-axis denotes the number of environment steps, and the y-axis denotes the episodic return. Curves are smoothed for improved visualization.

5.2 DBC AS A PLUG-AND-PLAY MODULE.

Table 4: DBC across different actor algorithms.

Actor	Critic	Humanoid-v5	Walker2d-v5
QVPO	CDQ	5037.9	5041.1
	DBC(*)	**5453.4**	**5502.6**
SAC	CDQ	5170.7	4912.7
	DBC(*)	**5877.2**	**6162.8**
TD3	CDQ	5098.2	5022.4
	DBC(*)	**5311.1**	**6373.5**

To validate the effectiveness of DBC as a plug-and-play critic module, we integrate it into three representative actor algorithms, SAC, TD3, and QVPO,and evaluate the resulting methods on two challenging continuous-control benchmarks, Humanoid-v5 and Walker2d-v5. As shown in Table 4, replacing CDQ with DBC consistently improves performance across all actor backbones. In particular, on Humanoid-v5, DBC yields a maximum improvement of 13.7% when paired with the SAC actor, while on Walker2d-v5, it achieves up to a 26.9% performance gain when integrated with TD3.

5.3 ABLATION

Time Steps. We ablate the number of time steps M to validate that our multi-step critic offers substantial gains beyond a naive integration of Diffusion Models and RL. As shown in Table 5, $M = 5$ outperforms $M = 1$ in all tasks, confirming that multi-step processing significantly enhances the ability to fit. However, increasing M to 10 introduces instability due to extended backpropagation chains. Therefore, we recommend $M = 5$ as the optimal trade-off between performance, cost, and stability.

Table 5: Ablation of time steps M.

Run	Ant-v5	Humanoid-v5	Walker2d-v5
M=1	5150.4	5524.7	5809.8
M=5	**6481.9**	**5877.2**	**6162.8**
M=10	5946.6	4848.7	6014.3

Table 6: Ablation of loss weight w.

Run	Ant-v5	Humanoid-v5	Walker2d-v5
w=0.0	4828.3	5807.3	5562.9
w=0.01	**6481.9**	**5877.2**	**6162.8**
w=0.02	3947.3	5743.4	6097.8
w=0.1	5716.2	5589.0	5974.6
w/o Eq.(9)	2517.1	3751.2	3859.3

Weight for Anchor Loss. Finally, we analyze the effect of the anchor loss introduced in Section 4.2. As shown in Table 6, the combined objective Eq. (12) achieves the best performance with a small anchor weight $w = 0.01$ across all tasks. In contrast, both using a larger weight (e.g., $w \geq 0.02$) and removing the anchor term entirely lead to clear performance degradation. These results are consistent with our theory: although the sample quantile is a biased target (Theorem C.4), a sufficiently small weight provably reduces the finite-time error (Theorem C.5) while shifting the value fixed point by at most $O(wK_{\text{tgt}}^{-1/2})$ (Theorem C.9), which supports using the anchor term as a light auxiliary regularizer. Based on these findings, we recommend $w = 0.01$ as the default choice.

6 CONCLUSION AND LIMITATION

In this work, we first theoretically prove the gaussian degradation problem in prior works of diffusion critic, and then propose DBC, a novel distributional RL method that leverages diffusion bridge models to accurately capture the return distribution. By learning the inverse CDF of the return distribution, DBC avoids the discrete quantile approximation and the Gaussian degradation that arises in vanilla diffusion critics under Bellman backups. Moreover, we derive an integral-consistent discretization to correct the bias induced by finite time steps, leading to more accurate and stable value estimation. As a plug-and-play component, DBC can be seamlessly integrated into existing RL algorithms, and empirical results on MuJoCo benchmarks and DMC benchmarks demonstrate consistent performance gains over prior distributional critics. These findings highlight the importance of expressive and reliable diffusion-based critics, and suggest that improving value estimation is a key direction for advancing diffusion-based RL. However, DBC incurs a higher computational cost than conventional methods due to the multi-step generation process and the need to output quantile values for multiple quantile levels per forward pass. The reduction of this overhead, such as distillation or improved network architecture design, will be considered in our future work.

AI USE STATEMENT

We used generative AI tools to assist with drafting parts of the manuscript and improving its wording and readability. The authors reviewed and revised the AI-assisted text and take responsibility for the final content of this paper, including its scientific claims and results.

REFERENCES

Bhavya Kumar Agrawalla, Michal Nauman, Khush Agrawal, and Aviral Kumar. floq: Training critics via flow-matching for scaling compute in value-based RL. In *The Fourteenth International Conference on Learning Representations*, 2026.

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022.

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International conference on machine learning*, pp. 449–458. PMLR, 2017.

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

Onur Celik, Zechu Li, Denis Blessing, Ge Li, Daniel Palanicek, Jan Peters, Georgia Chalvatzaki, and Gerhard Neumann. Dime: Diffusion-based maximum entropy reinforcement learning. *arXiv preprint arXiv:2502.02316*, 2025.

Deshu Chen, Yuchen Liu, Zhijian Zhou, Chao Qu, and Yuan Qi. Unleashing flow policies with distributional critics. *arXiv preprint arXiv:2509.23087*, 2025.

Tianrong Chen, Jiatao Gu, Laurent Dinh, Evangelos A Theodorou, Joshua Susskind, and Shuangfei Zhai. Generative Modeling with Phase Stochastic Bridges. *arXiv preprint arXiv:2310.07805*, 2023.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit quantile networks for distributional reinforcement learning. In *International conference on machine learning*, pp. 1096–1105. PMLR, 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*, volume 32, 2018b.

Shutong Ding, Ke Hu, Zhenhao Zhang, Kan Ren, Weinan Zhang, Jingyi Yu, Jingya Wang, and Ye Shi. Diffusion-based reinforcement learning via q-weighted variational policy optimization. *arXiv preprint arXiv:2405.16173*, 2024.

Shutong Ding, Ke Hu, Shan Zhong, Haoyang Luo, Weinan Zhang, Jingya Wang, Jun Wang, and Ye Shi. Genpo: Generative diffusion models meet on-policy reinforcement learning. *arXiv preprint arXiv:2505.18763*, 2025.

Perry Dong, Chongyi Zheng, Chelsea Finn, Dorsa Sadigh, and Benjamin Eysenbach. Value flows. *arXiv preprint arXiv:2510.07650*, 2025.

Jingliang Duan, Yang Guan, Shengbo Eben Li, Yangang Ren, Qi Sun, and Bo Cheng. Distributional soft actor-critic: Off-policy reinforcement learning for addressing value estimation errors. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Jingliang Duan, Wenxuan Wang, Liming Xiao, Jiaxin Gao, Shengbo Eben Li, Chang Liu, Ya-Qin Zhang, Bo Cheng, and Keqiang Li. Distributional soft actor-critic with three refinements. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Xiaoliang Hu, Fuyun Wang, Tong Zhang, and Zhen Cui. Value diffusion reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=thJ6aFoKrh.

Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

HJ Kappen. Stochastic optimal control theory. *ICML, Helsinki, Radbound University, Nijmegen, Netherlands*, 2008.

Arsenii Kuznetsov, Pavel Shvechikov, Alexander Grishin, and Dmitry Vetrov. Controlling overestimation bias with truncated mixture of continuous distributional quantile critics. In *International conference on machine learning*, pp. 5556–5566. PMLR, 2020.

Hojoon Lee, Youngdo Lee, Takuma Seno, Donghu Kim, Peter Stone, and Jaegul Choo. Hyperspherical normalization for scalable deep reinforcement learning. *arXiv preprint arXiv:2502.15280*, 2025.

Erich Leo Lehmann and George Casella. *Theory of point estimation*. Springer, 1998.

Bo Li, Kaitao Xue, Bin Liu, and Yu-Kun Lai. Bbdm: Image-to-image translation with brownian bridge diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern Recognition*, pp. 1952–1961, 2023.

Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A Theodorou, Weili Nie, and Anima Anandkumar. I2SB: Image-to-image schrödinger bridge. *arXiv preprint arXiv:2302.05872*, 2023.

Ziwei Luo, Fredrik K Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas B Schön. Image restoration with mean-reverting stochastic differential equations. *arXiv preprint arXiv:2301.11699*, 2023.

Haitong Ma, Tianyi Chen, Kai Wang, Na Li, and Bo Dai. Efficient online reinforcement learning for diffusion policy. *arXiv preprint arXiv:2502.00361*, 2025a.

Xiaoteng Ma, Junyao Chen, Li Xia, Jun Yang, Qianchuan Zhao, and Zhengyuan Zhou. Dsac: Distributional soft actor-critic for risk-sensitive reinforcement learning. *Journal of Artificial Intelligence Research*, 83, 2025b.

David McAllister, Songwei Ge, Brent Yi, Chung Min Kim, Ethan Weber, Hongsuk Choi, Haiwen Feng, and Angjoo Kanazawa. Flow matching policy gradients. *arXiv preprint arXiv:2507.21053*, 2025.

Daniel Palenicek, Florian Vogt, Joe Watson, Ingmar Posner, and Jan Peters. Xqc: Well-conditioned optimization accelerates deep reinforcement learning. *arXiv preprint arXiv:2509.25174*, 2025.

Michael Psenka, Alejandro Escontrela, Pieter Abbeel, and Yi Ma. Learning a diffusion model policy from rewards via q-score matching. *arXiv preprint arXiv:2312.11752*, 2023.

Allen Z Ren, Justin Lidard, Lars L Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. *arXiv preprint arXiv:2409.00588*, 2024.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

Hanting Wang, Tao Jin, Wang Lin, Shulei Wang, Hai Huang, Shengpeng Ji, and Zhou Zhao. Irbridge: Solving image restoration bridge with pre-trained generative diffusion models. *arXiv preprint arXiv:2505.24406*, 2025.

Yinuo Wang, Likun Wang, Yuxuan Jiang, Wenjun Zou, Tong Liu, Xujie Song, Wenxuan Wang, Liming Xiao, Jiang Wu, Jingliang Duan, et al. Diffusion actor-critic with entropy regulator. *Advances in Neural Information Processing Systems*, 37:54183–54204, 2024a.

Yuji Wang, Zehua Chen, Xiaoyu Chen, Yixiang Wei, Jun Zhu, and Jianfei Chen. Framebridge: Improving image-to-video generation with bridge models. *arXiv preprint arXiv:2410.15371*, 2024b.

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*, 2022.

Long Yang, Zhixiong Huang, Fenghao Lei, Yucun Zhong, Yiming Yang, Cong Fang, Shiting Wen, Binbin Zhou, and Zhouchen Lin. Policy representation via diffusion probability model for reinforcement learning. *arXiv preprint arXiv:2305.13122*, 2023.

Conghan Yue, Zhengwei Peng, Junlong Ma, Shiyan Du, Pengxu Wei, and Dongyu Zhang. Image Restoration Through Generalized Ornstein-Uhlenbeck Bridge. In *International Conference on Machine Learning*, 2024.

Jing Zhang, Linjiajie Fang, Kexin Shi, Wenjia Wang, and Bingyi Jing. Q-distribution guided q-learning for offline reinforcement learning: Uncertainty penalized q-value via consistency model. *Advances in Neural Information Processing Systems*, 37:54421–54462, 2024.

Shan Zhong, Shutong Ding, He Diao, Xiangyu Wang, Kah Chan Teh, and Bei Peng. Flow-critic: Bridging value estimation with flow matching in reinforcement learning. *arXiv preprint arXiv:2510.22686*, 2025.

Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising Diffusion Bridge Models. *arXiv preprint arXiv:2309.16948*, 2023.

Kaizhen Zhu, Mokai Pan, Yuexin Ma, Yanwei Fu, Jingyi Yu, Jingya Wang, and Ye Shi. UniDB: A Unified Diffusion Bridge Framework via Stochastic Optimal Control. In *Forty-second International Conference on Machine Learning*, 2025.

A NOTATION

Symbol	Meaning
MDP and distributional RL (Section 3.1)	
$\mathcal{S},\ \mathcal{A}$	state space and action space
$\mathcal{P},\ \mathcal{R},\ \gamma,\ \pi$	transition kernel, reward function, discount factor, policy
$\boldsymbol{s},\ \boldsymbol{a},\ r,\ \mathcal{D}$	state, action, per-step reward, replay buffer
$Z^\pi,\ Q^\pi$	random return, value function
$Z_\tau,\ \mathcal{T}^\pi$	τ-quantile of the return, $Z_\tau = F_Z^{-1}(\tau)$, and the distributional Bellman operator
$\mathbb{E},\ \mathbb{I},\ U(\cdot)$	expectation, indicator, uniform distribution
$O(\cdot),\ o(\cdot)$	asymptotic order symbols
Diffusion bridge and generator (Section 3.2)	
$t,\ \xi(t),\ z_t$	bridge time, endpoint interpolation coefficient, bridge state
$z^{\text{start}},\ z^{\text{end}},\ u_t,\ g_t,\ \omega$	bridge endpoints, control input, diffusion coefficient, terminal-cost weight of Eq. (4)
$t_i,\ \Delta t_i,\ c(t)$	discretization grid, step size, velocity coefficient
$\lambda^2,\ \sigma^2,\ \bar{\sigma}^2_{t_1:t_2}$	OU variance scale, variance of the degenerate Gaussian in Theorem 4.1, and its integrated form
$\beta(t),\ \beta_{\min},\ \beta_{\max}$	GOU drift coefficient and its schedule bounds
$M,\ N$	solver discretization steps, training iterations
Critic and objectives (Section 4)	
$f_\theta,\ \theta,\ \phi$	critic network, online and target parameters
$\tau_i,\ K_{\text{onl}},\ K_{\text{tgt}}$	quantile level, number of online / target particles
$\mathcal{Y},\ y_j,\ \hat{y}_\tau$	target particles and their sample τ-quantile
$\rho^\kappa_\tau,\ \ell^\kappa_\tau$	Huberized pinball loss, prediction-first loss form
$\mathcal{L}^{\text{Huber}}_\kappa$	Huber loss
$\mathcal{L}^{\text{quantile}},\ \mathcal{L}^{\text{anchor}},\ \mathcal{L}^{\text{total}}$	quantile, anchor and total critic losses
$w,\ H,\ d,\ \alpha$	anchor loss weight, number of heads, quantiles dropped per head, entropy temperature

B THEOREM PROOF

B.1 PROOF OF THEOREM 4.1

Theorem 4.1. (Gaussian Degradation of Diffusion Critics). *Vanilla Diffusion Critics f_θ finally degrades into a Gaussian distribution $\mathcal{N}(Q(s,a),\sigma^2)$ with the Bellman update:*

$$\mathcal{T}^\pi Z(s,a) = \mathcal{R}(s,a) + \gamma Z\left(s',a'\right), Z \sim f_\theta(s,a). \tag{6}$$

Proof. Due to the existence of approximation error ϵ_t in the tth Bellman expansion, the target distribution of diffusion is $\hat{Z}(s,a) = \mathcal{R}(s,a) + \epsilon_0 + \gamma(\hat{Z}(s',a') + \epsilon_1)$. Hence, we can derive that

$$\hat{Z}(s,a) = Z(s,a) + \sum_{t=0}^{\infty} \gamma^t \epsilon_t.$$

Then, for convenience let $Z(s,a) \overset{D}{=} Q(s,a) + \gamma^{-1}\epsilon_{-1}$ rewrite the $\hat{Z}(s,a) = Z(s,a) + \sum_{t=-1}^{\infty}\gamma^t\epsilon_t = Q(s,a) + \sum_{t=-1}^{\infty}\gamma^t\epsilon_t$. Since the approximation error can be assumed to be independent of each other with zero mean, we can apply the Central Limit Theorem: when $\gamma \to 1$, we have

$$\hat{Z}(s,a) \to \mathcal{N}(Q(s,a),\sigma^2).$$

In that case, with Bellman backup iteration, the vanilla diffusion critics will finally fall into a Gaussian distribution. □

Notably, the inherent problem here is that diffusion models are designed to model distributions rather than point-to-point mappings. In other words, the objective

$$\min_\theta \mathbb{E}_{\epsilon,t}\left[\left\|\epsilon-\epsilon_\theta\left(\sqrt{\bar\alpha_t}x+\sqrt{1-\bar\alpha_t}\boldsymbol{\epsilon},t\right)\right\|^2\right], \qquad x=\mathbb{E}(Z),$$

is not equivalent to

$$\min_\theta \mathbb{E}_{Z,\epsilon,t}\left[\left\|\epsilon-\epsilon_\theta\left(\sqrt{\bar\alpha_t}Z+\sqrt{1-\bar\alpha_t}\boldsymbol{\epsilon},t\right)\right\|^2\right], \qquad x=Z.$$

in the diffusion model. Therefore, the approximation error cannot be alleviated simply by collecting more samples (i.e., $\mathbb{E}(Z+\gamma\epsilon_t)=\mathbb{E}(Z)$).

In contrast, DBC establishes an explicit mapping from quantiles τ to the Q-value distribution Z_τ, rendering the two objectives equivalent. That corresponds to our quantile loss (9) and the anchor loss (11), which both drive the outputs of the diffusion bridge to converge to the corresponding Q-value quantiles Z_τ.

B.2 Proof of Eq. (15)

We formally establish the relationship between the time-dependent velocity coefficient $c(t)$ and the interpolation coefficient $\xi(t)$. Revisiting Eq. (5) from the main text:

$$\boldsymbol{u}_t^* = g_t\frac{e^{-2\bar\beta_{t:1}}}{\bar\sigma_{t:1}^2}(z^{\text{end}}-z_t), \quad z_t=\xi(t)z^{\text{start}}+(1-\xi(t))z^{\text{end}}. \tag{5}$$

where $\xi(t)$ is a smooth scalar function satisfying the boundary conditions $\xi(1)=0$ and $\xi(0)=1$.

To derive the ordinary differential equation (ODE) governing this process, we differentiate Eq. (5) w.r.t. time t:

$$\begin{aligned}\frac{dz_t}{dt} &= \frac{d}{dt}\left[\xi(t)z^{\text{start}}+z^{\text{end}}-\xi(t)z^{\text{end}}\right]\\ &= \dot\xi(t)z^{\text{start}}-\dot\xi(t)z^{\text{end}}\\ &= -\dot\xi(t)(z^{\text{end}}-z^{\text{start}}).\end{aligned} \tag{18}$$

B.3 Proof of Theorem 4.2

Theorem 4.2 (Endpoint consistency of DBC discretization). *For any partition $0=t_0<t_1<\cdots<t_M=1$, if $z_{t_0}=z^{start}$ and the trajectory is updated by Eq.* (17)*, then $z_{t_M}=z^{end}$ for any number of discretization steps M.*

Proof. Starting from $z_{t_0}=z^{\text{start}}$, recursively applying Eq. (17) gives

$$\begin{aligned}z_{t_M} &= z_{t_0}+\sum_{i=0}^{M-1}\big(\xi(t_i)-\xi(t_{i+1})\big)(z^{\text{end}}-z^{\text{start}})\\ &= z^{\text{start}}+\big(\xi(t_0)-\xi(t_M)\big)(z^{\text{end}}-z^{\text{start}})\\ &= z^{\text{start}}+(1-0)(z^{\text{end}}-z^{\text{start}})\\ &= z^{\text{end}},\end{aligned} \tag{19}$$

where the second equality follows from telescoping the coefficients, and the third equality uses $t_0=0$, $t_M=1$, $\xi(0)=1$, and $\xi(1)=0$. Therefore, the discretized trajectory reaches the prescribed endpoint for any partition and any number of steps M. □

Proposition B.1 (Endpoint error: IC vs. Euler)**.** *Let $\hat z_i=z^{\text{end}}+\epsilon_i$, $\tilde c_i=\xi(t_i)-\xi(t_{i+1})$, and $w_i=c(t_i)\Delta t_i$. Then*

$$z_M^{\text{IC}}-z^{\text{end}}=\sum_{i=0}^{M-1}\tilde c_i\epsilon_i, \qquad \tilde c_i\ge 0, \quad \sum_{i=0}^{M-1}\tilde c_i=1,$$

hence

$$\left|z_M^{\text{IC}}-z^{\text{end}}\right|\le\max_{0\le i<M}|\epsilon_i|.$$

For Euler,

$$z_M^{\mathrm{E}} - z^{\mathrm{end}} = (S_M - 1)\big(z^{\mathrm{end}} - z^{\mathrm{start}}\big) + \sum_{i=0}^{M-1} w_i \epsilon_i, \qquad S_M = \sum_{i=0}^{M-1} w_i.$$

C Sample Quantile

C.1 Properties of the Sample Quantile

This section presents a detailed analysis of the anchor proposed in the main text, focusing on its formal definition and asymptotic properties. Let X be a real-valued random variable with distribution function $F \in C^1$ and density $f = F'$, satisfying $\mathbb{E}[X^2] < \infty$. Let X_i be i.i.d. samples drawn from F, and denote the associated order statistics by

$$X_{(1)} \le X_{(2)} \le \cdots \le X_{(n)}.$$

The sample distribution function is defined as

$$\hat{F}_n(x) := \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(X_i \le x),$$

where $\mathbb{I}(\cdot)$ denotes the indicator function. For a fixed quantile level $\tau \in (0, 1)$, the (generalized) empirical τ-quantile is defined via the generalized inverse

$$\hat{F}_n^{-1}(\tau) := \inf\big\{x \in \mathbb{R} : \hat{F}_n(x) \ge \tau\big\}.$$

It follows immediately that

$$\hat{F}_n^{-1}(\tau) = X_{(\lceil n\tau \rceil)}.$$

For notational convenience, we write $\hat{X}_\tau := \hat{F}_n^{-1}(\tau)$.

Sample quantiles as a empirical minimizer of the quantile loss. Define the quantile loss $\rho_\tau : \mathbb{R} \to \mathbb{R}$ by

$$\rho_\tau(u) = u\big(\tau - \mathbb{I}(u < 0)\big), \qquad \forall u \in \mathbb{R}.$$

Consider the sample risk as a function of a scalar decision variable $\vartheta \in \mathbb{R}$:

$$L(\vartheta) := \sum_{i=1}^{n} \rho_\tau(X_i - \vartheta).$$

Then every minimizer of $L(\vartheta)$ is an sample τ-quantile.

Lemma C.1 (Sample quantile as a minimizer of the quantile loss)**.** *Let $\tau \in (0, 1)$ and define*

$$N_<(\vartheta) := \sum_{i=1}^{n} \mathbb{I}(X_i < \vartheta), \qquad N_\le(\vartheta) := \sum_{i=1}^{n} \mathbb{I}(X_i \le \vartheta).$$

Then the set of minimizers of $L(\vartheta)$ is

$$\arg\min_{\vartheta \in \mathbb{R}} L(\vartheta) = \{\vartheta \in \mathbb{R} :\ N_<(\vartheta) \le n\tau \le N_\le(\vartheta)\}.$$

In particular, $\hat{X}_\tau = X_{(\lceil n\tau \rceil)}$ is always a minimizer. Moreover, if $X_{(\lceil n\tau \rceil)} < X_{(\lceil n\tau \rceil + 1)}$ (i.e., there is no tie at the $\lceil n\tau \rceil$-th order statistic), then the minimizer is unique and equals $\hat{X}_\tau$.

Proof. For each i, define $\ell_i(\vartheta) = \rho_\tau(X_i - \vartheta)$. As a function of ϑ, ℓ_i is convex and piecewise linear. Its subgradient is

$$\partial \ell_i(\vartheta) = \begin{cases} \{-\tau\}, & X_i > \vartheta, \\ \{1 - \tau\}, & X_i < \vartheta, \\ [-\tau,\, 1 - \tau], & X_i = \vartheta, \end{cases}$$

where ∂ denotes the subdifferential (a set) in $\mathbb{R}$. By subdifferential calculus for sums of convex functions,

$$\partial L(\vartheta) = \sum_{i=1}^{n} \partial \ell_i(\vartheta).$$

Let $N_<(\vartheta)$ and $N_\le(\vartheta)$ be defined as in the statement. Then the indices with $X_i < \vartheta$ contribute $1-\tau$, those with $X_i > \vartheta$ contribute $-\tau$, and those with $X_i = \vartheta$ contribute an interval $[-\tau, 1-\tau]$. Collecting terms yields

$$\partial L(\vartheta) = \big[N_<(\vartheta) - n\tau,\ N_\le(\vartheta) - n\tau\big].$$

A point ϑ minimizes the convex function L if and only if $0 \in \partial L(\vartheta)$, which is equivalent to

$$N_<(\vartheta) \le n\tau \le N_\le(\vartheta).$$

This proves the characterization of $\arg\min L(\vartheta)$. Since $\hat{X}_\tau = \hat{F}_n^{-1}(\tau) = X_{(\lceil n\tau \rceil)}$ satisfies the above inequality, it is a minimizer. If additionally $X_{(\lceil n\tau \rceil)} < X_{(\lceil n\tau \rceil+1)}$, then the inequality pins down a single point, hence the minimizer is unique and equals $\hat{X}_\tau$. □

In addition, the sample quantile is guaranteed to converge to the true quantile as n approaches infinity. It is guaranteed by the following theorem.

Theorem C.2 (Consistency of the sample quantile)**.** *Let X be a real-valued random variable with cumulative distribution function F, and let*

$$X_\tau := F^{-1}(\tau) = \inf\{x \in \mathbb{R} : F(x) \ge \tau\}, \qquad \tau \in (0,1),$$

denote the population τ-quantile. Let $\hat{F}_n$ be the sample distribution function based on i.i.d. samples $\{X_i\}_{i=1}^n$, and define the sample τ-quantile by

$$\hat{X}_\tau := \hat{F}_n^{-1}(\tau).$$

Assume that X_τ is unique, i.e., F is strictly increasing in a neighborhood of x_τ, and that F is continuous at X_τ.Then

$$\hat{X}_\tau \xrightarrow{a.s.} X_\tau \quad \text{as } n \to \infty.$$

Proof. By the Glivenko-Cantelli theorem, the sample distribution function $\hat{F}_n$ converges uniformly to F almost surely. Since F is continuous and strictly increasing in a neighborhood of x_τ, the generalized inverse mapping F^{-1} is continuous at τ.The almost sure convergence $\hat{X}_\tau \to X_\tau$ then follows from the continuous mapping theorem. A detailed proof can be found in (Van der Vaart, 2000, p. 266). □

Asymptotic Normality of the Sample Quantile. We now establish the asymptotic distribution of the empirical quantile $\hat{X}_\tau$.

Theorem C.3 (Asymptotic normality of the sample quantile)**.** *Let $X_1, X_2, \ldots, X_n$ be i.i.d. random variables with cumulative distribution function F and probability density function f. Fix $\tau \in (0,1)$ and denote the population τ-quantile by $X_\tau = F^{-1}(\tau)$ and the empirical τ-quantile by $\hat{X}_\tau = \hat{F}_n^{-1}(\tau)$. If f is continuous at X_τ and $f(X_\tau) > 0$, then*

$$\sqrt{n}\,(\hat{X}_\tau - X_\tau) \xrightarrow{d} \mathcal{N}\left(0, \frac{\tau(1-\tau)}{f^2(X_\tau)}\right), \quad n \to \infty. \tag{20}$$

Proof. For any $t \in \mathbb{R}$, the event $\{\sqrt{n}(\hat{X}_\tau - X_\tau) \le t\}$ is equivalent to $\hat{F}_n(X_\tau + t/\sqrt{n}) \ge \tau$. Hence

$$P\Big(\sqrt{n}(\hat{X}_\tau - X_\tau) \le t\Big) = P\Big(\hat{F}_n(X_\tau + t/\sqrt{n}) \ge \tau\Big). \tag{21}$$

Introduce the decomposition

$$\sqrt{n}\big[\hat{F}_n(X_\tau + t/\sqrt{n}) - \tau\big] = A_n + B_n, \tag{22}$$

where

$$\begin{aligned} A_n &= \sqrt{n}\Big[\hat{F}_n(X_\tau + t/\sqrt{n}) - F(X_\tau + t/\sqrt{n})\Big], \\ B_n &= \sqrt{n}\Big[F(X_\tau + t/\sqrt{n}) - \tau\Big]. \end{aligned} \tag{23}$$

We analyze A_n and B_n separately.

For B_n, we expand F around X_τ and using $F(X_\tau) = \tau$ and obtain

$$B_n = f(X_\tau)t + o(1), \qquad n \to \infty. \tag{24}$$

For A_n, we define $\zeta_{n,i} = \mathbb{I}(X_i \le X_\tau + t/\sqrt{n})$ for $i = 1, \dots, n$. Then $\{\zeta_{n,i}\}$ are i.i.d. Bernoulli random variables with

$$\begin{aligned} \mathbb{E}[\zeta_{n,i}] &= F(X_\tau + t/\sqrt{n}), \\ \mathrm{Var}(\zeta_{n,i}) &= F(X_\tau + t/\sqrt{n})\big[1 - F(X_\tau + t/\sqrt{n})\big]. \end{aligned} \tag{25}$$

It is straightforward to verify that

$$\lim_{n\to\infty} \mathbb{E}[\zeta_{n,i}] = \tau, \qquad \lim_{n\to\infty} \mathrm{Var}(\zeta_{n,i}) = \tau(1-\tau). \tag{26}$$

Applying the Lindeberg-Feller central limit theorem yields

$$A_n = \sqrt{n}\Big(\frac{1}{n}\sum_{i=1}^{n} \zeta_{n,i} - F(X_\tau + t/\sqrt{n})\Big) \xrightarrow{d} \mathcal{N}\big(0, \tau(1-\tau)\big). \tag{27}$$

Now, combining the two terms and according to Slutsky's theorem, we have that

$$A_n + B_n \xrightarrow{d} \mathcal{N}\big(f(X_\tau)t,\ \tau(1-\tau)\big). \tag{28}$$

Consequently,

$$\begin{aligned} \lim_{n\to\infty} P\Big(\sqrt{n}(\hat{X}_\tau - X_\tau) \le t\Big) &= \lim_{n\to\infty} P(A_n + B_n \ge 0) \\ &= \Phi\left(\frac{f(X_\tau)t}{\sqrt{\tau(1-\tau)}}\right), \end{aligned} \tag{29}$$

where Φ denotes the standard normal distribution function. This shows that the limiting distribution of $\sqrt{n}(\hat{X}_\tau - X_\tau)$ is $\mathcal{N}\big(0, \tau(1-\tau)/f^2(X_\tau)\big)$, completing the proof. □

Bias of the Sample Quantile. Before presenting the formal result, we remark that no universally unbiased estimator exists for quantiles; we can only provide asymptotic statements. For a detailed discussion of this point, see (Lehmann & Casella, 1998, p. 84).

Theorem C.4 (Asymptotic bias of the sample quantile)**.** *Let $X_1, X_2, \dots, X_n$ be i.i.d. random variables with cumulative distribution function F and quantile function F^{-1}. Fix $\delta > 0$ and consider $\tau \in [\delta, 1-\delta]$. Assume that:*

(i) The density f satisfies $\inf_{\tau\in[\delta,1-\delta]} f(F^{-1}(\tau)) \ge c > 0$ for some constant c;

(ii) Q is differentiable on $[\delta, 1-\delta]$;

(iii) $\mathbb{E}[X^2] < \infty$.

Then the bias of the empirical quantile $\hat{F}_n^{-1}(\tau) = \hat{X}_\tau$ satisfies

$$\big|\mathbb{E}[\hat{F}_n^{-1}(\tau)] - F^{-1}(\tau)\big| = o\left(\frac{1}{\sqrt{n}}\right). \tag{30}$$

Proof. Write $U_i = F(X_i) \sim \mathrm{Uniform}(0,1)$ and let $U_{(1)} \le \cdots \le U_{(n)}$ be the order statistics. Set $k = \lceil n\tau \rceil$ and $\xi_n = U_{(k)} - \tau$. Then $\hat{F}_n^{-1}(\tau) = F^{-1}(U_{(k)}) = F^{-1}(\tau + \xi_n)$.

A first-order Taylor expansion yields

$$\hat{F}_n^{-1}(\tau) - F^{-1}(\tau) = (F^{-1})'(\tau)\xi_n + r(\xi_n), \tag{31}$$

where the remainder $r(v)$ satisfies the Peano form: for every $\eta > 0$ there exists $\delta_\eta > 0$ such that $|r(v)| \le \eta|v|$ whenever $|v| \le \delta_\eta$.

From properties of the Beta distribution,

$$\mathbb{E}[\xi_n] = \frac{k}{n+1} - \tau, \qquad \mathrm{Var}(\xi_n) = \frac{\tau(1-\tau)}{n} + O\left(\frac{1}{n^2}\right). \tag{32}$$

By Jensen's inequality,

$$\mathbb{E}\big[|\xi_n|\big] \le \sqrt{\mathbb{E}[\xi_n^2]} = \sqrt{\mathrm{Var}(\xi_n) + \big(\mathbb{E}[\xi_n]\big)^2} = O\left(\frac{1}{\sqrt{n}}\right). \tag{33}$$

To bound $\mathbb{E}[|r(\xi_n)|]$, we first estimate the second moment of $r(\xi_n)$. Using the elementary inequality $(a+b+c)^2 \le 3(a^2+b^2+c^2)$,

$$\begin{aligned} r(\xi_n)^2 &= \big(F^{-1}(\tau+\xi_n) - F^{-1}(\tau) - (F^{-1})'(\tau)\xi_n\big)^2 \\ &\le 3\Big(Q^2(\tau+\xi_n) + Q^2(\tau) + ((F^{-1})'(\tau))^2\xi_n^2\Big). \end{aligned} \tag{34}$$

Hence

$$\mathbb{E}\big[r(\xi_n)^2\big] \le 3\Big(\mathbb{E}\big[X_{(k)}^2\big] + Q^2(\tau) + ((F^{-1})'(\tau))^2\mathbb{E}\big[\xi_n^2\big]\Big), \tag{35}$$

where $X_{(k)} = F^{-1}(U_{(k)})$ is the k-th order statistic of the original sample. Because $\mathbb{E}[X^2] < \infty$, there exists a constant $A > 0$ such that $\mathbb{E}[r(\xi_n)^2] \le An$.

Next, the Dvoretzky-Kiefer-Wolfowitz inequality gives

$$P\big(|\xi_n| \ge t\big) \le 2\exp(-2nt^2), \qquad t > 0. \tag{36}$$

Split the expectation of $|r(\xi_n)|$ into two parts. For a given $\eta > 0$, take the corresponding δ_η:

$$\begin{aligned} \mathbb{E}\big[|r(\xi_n)|\big] &= \mathbb{E}\Big[|r(\xi_n)|\,\mathbb{I}\{|\xi_n| \le \delta_\eta\}\Big] + \mathbb{E}\Big[|r(\xi_n)|\,\mathbb{I}\{|\xi_n| > \delta_\eta\}\Big] \\ &\le \eta\,\mathbb{E}\big[|\xi_n|\big] + \sqrt{\mathbb{E}\big[r(\xi_n)^2\big]}\,\sqrt{P\big(|\xi_n| > \delta_\eta\big)} \\ &\le \eta\,O\Big(\frac{1}{\sqrt{n}}\Big) + \sqrt{An}\,\sqrt{2\exp(-2n\delta_\eta^2)} \\ &= \eta\,O\Big(\frac{1}{\sqrt{n}}\Big) + o\Big(\frac{1}{\sqrt{n}}\Big). \end{aligned} \tag{37}$$

Since $\eta > 0$ is arbitrary, we obtain

$$\mathbb{E}\big[|r(\xi_n)|\big] = o\Big(\frac{1}{\sqrt{n}}\Big). \tag{38}$$

Finally, combining Eq. (31), (33) and (38),

$$\begin{aligned} \big|\mathbb{E}[\hat{F}_n^{-1}(\tau)] - F^{-1}(\tau)\big| &\le |(F^{-1})'(\tau)|\,\big|\mathbb{E}[\xi_n]\big| + \mathbb{E}\big[|r(\xi_n)|\big] \\ &\le \frac{1}{c}\cdot\frac{1}{n+1} + o\Big(\frac{1}{\sqrt{n}}\Big) \\ &= o\Big(\frac{1}{\sqrt{n}}\Big), \end{aligned} \tag{39}$$

which completes the proof. □

Accordingly, letting $n = K_{\mathrm{tgt}}$, we show that, given K_{tgt} target samples, the resulting asymptotic bias is $o\big(1/\sqrt{K_{\mathrm{tgt}}}\big)$.

Remark. The assumption $\inf_{\tau\in[\delta,1-\delta]} f(F^{-1}(\tau)) \ge c > 0$ guarantees that $(F^{-1})'(\tau) = 1/f(F^{-1}(\tau))$ is bounded on $[\delta, 1-\delta]$. If one further assumes that Q' satisfies Lipschitz Condition, a sharper $O(1/n)$ bias bound can be derived. Restricting to the inner interval $[\delta, 1-\delta]$ covers the quantile levels that are most relevant in practice while providing the necessary regularity for the analysis.

C.2 Global Finite-Time Acceleration of the Anchor Loss

Recall from Section 4.2 that the DBC critic is trained on the combined objective Eq. (12), in which the anchor loss Eq. (11) supervises the critic with empirical quantiles of the target particles. Since DBC decomposes distribution learning into independent scalar estimation problems indexed by τ, we fix a single $\tau \in (0,1)$ and analyze the optimization dynamics induced by the anchor term. For analytical tractability we replace the huberized quantile loss with the standard pinball loss, which does not alter the qualitative conclusions.

We adopt the notation of Appendix C. Let X be a real-valued random variable with CDF $F \in C^1$ and density $f = F'$. Fix $\tau \in (0,1)$ and let $X_\tau = F^{-1}(\tau)$ denote the unique population τ-quantile. Let $\widehat{X}_\tau$ be the empirical τ-quantile based on n i.i.d. samples $\{X_i\}_{i=1}^n \sim F$, with bias

$\Delta_n = \mathbb{E}_{\mathcal{D}}[\widehat{X}_\tau] - X_\tau = o(1/\sqrt{n})$ (Theorem C.4). Define the pinball loss $\rho_\tau(u) = u(\tau - \mathbb{I}\{u < 0\})$. Consider a scalar parameter ϑ trained via gradient flow on the expected loss

$$\bar{\mathcal{L}}_w(\vartheta) = \mathbb{E}_{X\sim F}[\rho_\tau(X - \vartheta)] + \frac{w}{2}\,\mathbb{E}_{\mathcal{D}}[(\vartheta - \widehat{X}_\tau)^2], \qquad w \geq 0,$$

with training time $T \geq 0$. Let $e_w(T) = \vartheta_w(T) - X_\tau$ be the error at time T under weight w, and let $e_0(T)$ be the error when $w = 0$. All weights start from the same initial error $e_w(0) = e_{\text{init}}$.

Theorem C.5 (Global Finite-Time Acceleration of the Anchor Loss)**.** *Fix any finite training horizon $T > 0$. Assume:*

1. $e_0(T) \neq 0$*;*
2. $|\Delta_n| < |e_0(T)|$.

Then there exists $\overline{w} = \overline{w}_{T,n,e_{\text{init}}} > 0$ such that for every $w \in (0, \overline{w})$,

$$e_w(T)^2 < e_0(T)^2.$$

Proof. The gradient flow $\frac{d\vartheta}{dT} = -\frac{d\bar{\mathcal{L}}_w}{d\vartheta}$ yields the exact error dynamics

$$\frac{de_w}{dT} = -\big[F(X_\tau + e_w) - \tau\big] - w(e_w - \Delta_n). \tag{40}$$

Define $G(e) \triangleq F(X_\tau + e) - F(X_\tau) = F(X_\tau + e) - \tau$. Then $G(e) > 0$ for $e > 0$ and $G(e) < 0$ for $e < 0$, and (40) becomes

$$\frac{de_w}{dT} = -G(e_w) - w(e_w - \Delta_n). \tag{41}$$

Step 1: Baseline trajectory. Setting $w = 0$,

$$\frac{de_0}{dT} = -G(e_0). \tag{42}$$

Since $e\,G(e) > 0$ for $e \neq 0$,

$$\frac{d}{dT}\big(\tfrac{1}{2}e_0^2\big) = e_0\frac{de_0}{dT} = -e_0 G(e_0) < 0, \qquad e_0 \neq 0.$$

Hence $|e_0(T)|$ strictly decreases and $\operatorname{sgn}(e_0(T)) = \operatorname{sgn}(e_0(0))$ for all T. Consequently, for any $0 \leq s \leq T$,

$$\operatorname{sgn}(e_0(v)) = \operatorname{sgn}(e_0(T)), \qquad |e_0(v)| \geq |e_0(T)|. \tag{43}$$

Step 2: Sensitivity. Define

$$p(T) \triangleq \left.\frac{\partial e_w(T)}{\partial w}\right|_{w=0}. \tag{44}$$

Differentiating (41) with respect to w, interchanging derivatives, and setting $w = 0$ yields

$$\frac{dp}{dT} = -G'(e_0(T))\,p(T) - \big(e_0(T) - \Delta_n\big), \qquad p(0) = 0. \tag{45}$$

Since $G'(e) = f(X_\tau + e)$, (45) is a linear ODE. By the variation-of-constants formula,

$$p(T) = -\int_0^T \Phi(T, v)\,\big(e_0(v) - \Delta_n\big)\,ds, \tag{46}$$

where the kernel

$$\Phi(T, v) = \exp\Big(-\int_v^T f(X_\tau + e_0(u))\,du\Big) > 0. \tag{47}$$

Step 3: Sign of $p(T)$.

Case 1: $e_0(T) > 0$. By (43), $e_0(v) \geq e_0(T) > 0$. Assumption 2 gives $|\Delta_n| < e_0(T)$, so $\Delta_n < e_0(T) \leq e_0(v)$. Hence $e_0(v) - \Delta_n > 0$. Since $\Phi(T, v) > 0$, (46) implies $p(T) < 0$. Therefore $e_0(T)p(T) < 0$.

Case 2: $e_0(T) < 0$. By (43), $e_0(v) \leq e_0(T) < 0$. Assumption 2 gives $|\Delta_n| < -e_0(T)$, which implies $\Delta_n > e_0(T) \geq e_0(v)$. Hence $e_0(v) - \Delta_n < 0$, so (46) gives $p(T) > 0$. Again $e_0(T)p(T) < 0$.

Thus under Assumption 2,

$$e_0(T)p(T) < 0. \tag{48}$$

Step 4: Acceleration. Expand $e_w(T)^2$ to first order around $w = 0$:

$$e_w(T)^2 = e_0(T)^2 + 2\,e_0(T)p(T)\,w + o(w).$$

From (48),

$$\frac{d}{dw}\bigg|_{w=0} e_w(T)^2 = 2\,e_0(T)p(T) < 0.$$

By the definition of the derivative, there exists $\overline{w} > 0$ such that for all $w \in (0, \overline{w})$,

$$e_w(T)^2 < e_0(T)^2.$$

□

Remark. Theorem C.5 works directly with the exact nonlinear error dynamics Eq. (40) under the two stated assumptions. The only condition is that the anchor bias is strictly smaller than the remaining baseline error at time T, a natural regime during early-to-mid training. The threshold $\overline{w}$ may depend on T, n, and the specific initialization e_{init}.

C.3 Fixed-Point Deviation of the Anchored Bellman Operator

We adopt the notation of Appendix C and write $n = K_{\text{tgt}}$. Let μ, ν denote distributions over returns, with CDFs F_μ, F_ν and quantile functions F_μ^{-1}, F_ν^{-1}, and let

$$W_\infty(\mu, \nu) \triangleq \sup_{\tau \in (0,1)} \big|F_\mu^{-1}(\tau) - F_\nu^{-1}(\tau)\big|$$

be the sup-distance between quantile functions. For return-distribution functions Z_1, Z_2 we write $\overline{W}_\infty(Z_1, Z_2) \triangleq \sup_{(s,a)} W_\infty(Z_1(s,a), Z_2(s,a))$. Let $\hat{x}_{\mu,n}(\tau)$ be the empirical τ-quantile of n i.i.d. draws from μ (written $\widehat{X}_\tau$ in Appendix C.1, where the sample size is $n = K_{\text{tgt}}$), and set $\hat{x}_{\mu,n}(\tau) \triangleq \mathbb{E}_{\mathcal{D}}[\widehat{X}_\tau]$.

For a fixed quantile level τ, the anchored objective Eq. (12) has the first-order condition

$$h_\mu(x) \triangleq F_\mu(x) - \tau + w\big(x - \hat{x}_{\mu,n}(\tau)\big) = 0,$$

which admits a unique solution whenever $f_\mu = F_\mu' > 0$; we denote it by $F_{\mathcal{H}\mu}^{-1}(\tau)$. This defines the *anchor operator* $\mathcal{H}$, mapping a return distribution μ to the distribution $\mathcal{H}\mu$ whose τ-quantile solves

the anchored problem. With $\mathcal{T}^\pi$ the distributional Bellman operator of Eq. (2), we consider the composite operator $\mathcal{B}^\pi \triangleq \mathcal{H} \circ \mathcal{T}^\pi$ together with the two error quantities

$$\mathrm{dev}_{w,n} \triangleq \sup_\mu W_\infty(\mathcal{H}\mu, \mu), \qquad \eta_n \triangleq \sup_{\mu,\tau} \big|\hat{x}_{\mu,n}(\tau) - F_\mu^{-1}(\tau)\big|.$$

As in the remark following Theorem C.4, we assume throughout that the densities are uniformly bounded away from zero, $f_\mu \geq c > 0$.

Lemma C.6 (Nonexpansiveness of the anchor operator)**.** *For any two return distributions μ, ν,*

$$W_\infty(\mathcal{H}\mu, \mathcal{H}\nu) \leq W_\infty(\mu, \nu).$$

Proof. Let $d \triangleq W_\infty(\mu, \nu)$ and fix $\tau \in (0,1)$. Write $\vartheta_\mu \triangleq F_{\mathcal{H}\mu}^{-1}(\tau)$, $\vartheta_\nu \triangleq F_{\mathcal{H}\nu}^{-1}(\tau)$ and $k \triangleq \lceil n\tau \rceil$. Take the comonotone coupling $X_i^{(\mu)} = F_\mu^{-1}(U_i)$, $X_i^{(\nu)} = F_\nu^{-1}(U_i)$ with $U_1, \ldots, U_n \sim \mathrm{Unif}(0,1)$. Then $|X_i^{(\mu)} - X_i^{(\nu)}| \leq d$ almost surely, and since order statistics are 1-Lipschitz,

$$\big|\hat{x}_{\mu,n}(\tau) - \hat{x}_{\nu,n}(\tau)\big| = \big|\mathbb{E}[X_{(k)}^{(\mu)}] - \mathbb{E}[X_{(k)}^{(\nu)}]\big| \leq \mathbb{E}\big|X_{(k)}^{(\mu)} - X_{(k)}^{(\nu)}\big| \leq d.$$

Moreover $W_\infty(\mu, \nu) \leq d$ gives $F_\nu(x - d) \leq F_\mu(x) \leq F_\nu(x + d)$ for every x. Since $h_\mu(\vartheta_\mu) = 0$,

$$h_\nu(\vartheta_\mu + d) \geq F_\mu(\vartheta_\mu) - \tau + w\big(\vartheta_\mu + d - \hat{x}_{\nu,n}(\tau)\big) = w\big(d + \hat{x}_{\mu,n}(\tau) - \hat{x}_{\nu,n}(\tau)\big) \geq 0,$$

and symmetrically $h_\nu(\vartheta_\mu - d) \leq w\big(\hat{x}_{\mu,n}(\tau) - \hat{x}_{\nu,n}(\tau) - d\big) \leq 0$. Because h_ν is strictly increasing with $h_\nu(\vartheta_\nu) = 0$, we obtain $\vartheta_\mu - d \leq \vartheta_\nu \leq \vartheta_\mu + d$. Taking the supremum over τ yields $W_\infty(\mathcal{H}\mu, \mathcal{H}\nu) \leq d$. □

Theorem C.7 (The anchored Bellman operator is a contraction)**.** *$\mathcal{B}^\pi = \mathcal{H} \circ \mathcal{T}^\pi$ is a γ-contraction in $\overline{W}_\infty$, i.e.*

$$\overline{W}_\infty\big(\mathcal{B}^\pi Z_1, \mathcal{B}^\pi Z_2\big) \leq \gamma\, \overline{W}_\infty(Z_1, Z_2),$$

and therefore admits a unique fixed point $Z_{w,n}^{\pi,*}$.

Proof. By Lemma C.6 and the γ-contraction property of $\mathcal{T}^\pi$,

$$\overline{W}_\infty\big(\mathcal{H}\mathcal{T}^\pi Z_1, \mathcal{H}\mathcal{T}^\pi Z_2\big) \leq \overline{W}_\infty\big(\mathcal{T}^\pi Z_1, \mathcal{T}^\pi Z_2\big) \leq \gamma\, \overline{W}_\infty(Z_1, Z_2).$$

Existence and uniqueness of the fixed point then follow from Banach's fixed-point theorem on the complete metric space of return-distribution functions. The true quantile function $Z^{\pi,*}$ is the fixed point of $\mathcal{T}^\pi$, Eq. (2). □

Lemma C.8 (Single-step distortion)**.** *The anchor operator distorts a return distribution by at most*

$$\mathrm{dev}_{w,n} \leq \frac{w}{w + c}\, \eta_n.$$

Proof. Fix μ and τ, and write $\vartheta \triangleq F_{\mathcal{H}\mu}^{-1}(\tau)$, $\hat{a} \triangleq \hat{x}_{\mu,n}(\tau)$, $q \triangleq F_\mu^{-1}(\tau)$. From $h_\mu(\vartheta) = 0$ and $F_\mu(q) = \tau$ we get $F_\mu(\vartheta) - F_\mu(q) + w(\vartheta - a) = 0$. By the mean value theorem there exists ξ between ϑ and q with $F_\mu(\vartheta) - F_\mu(q) = f_\mu(\xi)\,(\vartheta - q)$, hence

$$\vartheta - q = \frac{w}{w + f_\mu(\xi)}\,(\hat{a} - q), \qquad \text{so} \qquad |\vartheta - q| \leq \frac{w}{w + c}\,|\hat{a} - q|$$

because $f_\mu(\xi) \geq c$. Taking the supremum over μ and τ gives the claim. □

Theorem C.9 (Fixed-point deviation of the anchored operator)**.** *Let $Z^{\pi,*}$ and $Z_{w,n}^{\pi,*}$ be the fixed points of $\mathcal{T}^\pi$ and $\mathcal{B}^\pi$, respectively. Then*

$$W_\infty\big(Z_{w,n}^{\pi,*}, Z^{\pi,*}\big) \leq \frac{w}{(1-\gamma)(w+c)} \cdot \frac{C}{\sqrt{n}} = O\big(wn^{-1/2}\big),$$

where C is the constant in the sample-quantile bound $\eta_n \leq C/\sqrt{n}$ implied by Theorem C.4.

Proof. By the triangle inequality,

$$W_\infty\big(Z_{w,n}^{\pi,*}, Z^{\pi,*}\big) \leq W_\infty\big(\mathcal{H}\mathcal{T}^\pi Z_{w,n}^{\pi,*},\, \mathcal{H}\mathcal{T}^\pi Z^{\pi,*}\big) + W_\infty\big(\mathcal{H}\mathcal{T}^\pi Z^{\pi,*},\, \mathcal{T}^\pi Z^{\pi,*}\big),$$

where the first term is at most $\gamma W_\infty(Z_{w,n}^{\pi,*}, Z^{\pi,*})$ by Theorem C.7 and the second at most $\mathrm{dev}_{w,n}$ by its definition. Hence

$$W_\infty\big(Z_{w,n}^{\pi,*}, Z^{\pi,*}\big) \leq \gamma W_\infty\big(Z_{w,n}^{\pi,*}, Z^{\pi,*}\big) + \mathrm{dev}_{w,n},$$

and rearranging while inserting Lemma C.8,

$$W_\infty\big(Z^{\pi,*}_{w,n}, Z^{\pi,*}\big) \le \frac{\mathrm{dev}_{w,n}}{1-\gamma} \le \frac{w}{(1-\gamma)(w+c)}\,\eta_n \le \frac{w}{(1-\gamma)(w+c)} \cdot \frac{C}{\sqrt{n}},$$

which is $O(wn^{-1/2})$. □

Remark. Theorems C.5 and C.9 address complementary questions about the biased anchor target: the first identifies the condition under which a small anchor weight reduces the finite-time error of the exact nonlinear dynamics, and the second shows that the same weight shifts the Bellman fixed point by at most $O(wn^{-1/2})$. The bias therefore does not compromise the convergence target of the critic: the deviation vanishes both as $n \to \infty$ and as $w \to 0$.

D Analysis of Discretization Error in Diffusion Bridge Critics

D.1 Schedules for β

The behavior of the GOU bridge is governed by the drift coefficient $\beta(t)$. In our experiments, we evaluated three distinct schedule types. For all schedules, we set the hyperparameters $\beta_{\min} = 0.1$ and $\beta_{\max} = 5.0$.

Constant Schedule: The drift rate remains invariant throughout the process.

$$\beta(t) = 1.0. \tag{49}$$

Linear Schedule: The drift rate increases linearly, enforcing stronger mean-reversion near the target $t = 1$.

$$\beta(t) = \beta_{\min} + (\beta_{\max} - \beta_{\min}) \cdot t. \tag{50}$$

Cosine Schedule: A smooth, non-linear schedule that transitions gently at the boundaries.

$$\beta(t) = \beta_{\min} + \frac{\beta_{\max} - \beta_{\min}}{2}\left(1 - \cos(\pi t)\right). \tag{51}$$

Table 7: Relative endpoint error (%) under different schedules and M.

Steps (M)	**Euler Discretization Error (%)**		
	Constant	Linear	Cosine
1	14.91	21.44	21.44
2	9.48	6.93	18.75
5	4.29	5.41	6.85
10	2.23	3.07	3.42
20	1.13	1.62	1.71
50	0.46	0.67	0.68
100	0.23	0.34	0.34
1000	0.02	0.03	0.03

D.2 Quantitative Analysis of Endpoint Bias

Standard Euler discretization approximates the continuous integral $\int c(t)dt$ with a discrete sum $\sum c(t_i)\Delta t_i$. Due to the non-linearity of $c(t)$, particularly under dynamic drift schedules, this approximation introduces a systematic discrepancy that prevents the trajectory from reaching the intended boundary. We term this phenomenon **Endpoint Bias** and quantify it using the relative error $\mathcal{E} = |1 - \sum_{i=0}^{M-1} c(t_i)\Delta t_i| \times 100\%$.

Table 7 presents the errors calculated in various inference steps (M). The results reveal that standard Euler discretization incurs substantial bias, exceeding 21% in low-latency regimes ($M \le 5$), which can lead to unstable generative semantics and inaccurate reward estimation in online reinforcement learning. In particular, while the Euler method requires a large number of steps ($M \ge 1000$) to suppress bias to negligible levels, our proposed **Integral-Consistent Discretization** mathematically eliminates this error, maintaining a consistent $\mathcal{E} = 0$ regardless of the number of steps or the complexity of the drift schedule.

E FULL ALGORITHM

Here, the ***generator*** corresponds to a specific instantiation of Eq. (4) under a predefined scheduling scheme. The full algorithm is shown in Algorithm 3.

Algorithm 3 DBC: Training and Inference

Inputs: replay buffer $\mathcal{D}$; online parameters θ; target parameters ϕ
Hyperparameters: heads H; discount γ; temperature α; target atoms K_{tgt}; online atoms K_{onl}; solver steps M; TQC drop d; training iterations N; target EMA coefficient τ_{tgt}; Generator
Notation: $m \in \{0,1\}$ is the bootstrap mask ($m = 1$ for non-terminal transitions); $\mathsf{sac} = \alpha \log \pi(\boldsymbol{a}' \mid \boldsymbol{s}')$

Part I: Training (performed for N iterations)
for $k = 1$ **to** N **do**
  Sample $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}', m) \sim \mathcal{D}$; sample $\boldsymbol{a}' \sim \pi(\cdot \mid \boldsymbol{s}')$ and set $\mathsf{sac} \leftarrow \alpha \log \pi(\boldsymbol{a}' \mid \boldsymbol{s}')$
  (A) Target atoms (no gradient).
  Sample $\tau' \sim U([0,1])^{K_{\text{tgt}}}$ and set $z'^{\text{start}} \leftarrow \tau'$
  $\{z^{(h)}_{\text{target}}\}_{h=1}^{H} \leftarrow \text{Generator}(z'^{\text{start}}, \boldsymbol{s}', \boldsymbol{a}', \tau', \phi)$ by Eq. (17)
  Stack atoms $Z \leftarrow \text{Concat}(z^{(1)}_{\text{target}}, \ldots, z^{(H)}_{\text{target}})$
  Apply truncation $Z_{\text{tqc}} \leftarrow \text{DropTop}(Z, dH)$
  Build target particles $\mathcal{Y} \leftarrow \{r + m\gamma\,(z - \mathsf{sac})\}_{z \in Z_{\text{tqc}}}$ by Eq. (8)
  (B) Online samples and empirical quantiles.
  Sample $\tau \sim U([0,1])^{K_{\text{onl}}}$ and set $z^{\text{start}} \leftarrow \tau$
  $\hat{y}_\tau \leftarrow \text{SampleQuantile}(\mathcal{Y}, \tau)$ by Eq. (10)
  (C) Online predictions at $t = 0$ and $t = t_{\text{mid}}$.
  Sample $t_{\text{mid}} \sim \text{Unif}(0,1)$
  $z_{t_{\text{mid}}} \leftarrow \xi(t_{\text{mid}})\, z^{\text{start}} + (1 - \xi(t_{\text{mid}}))\, \hat{y}_\tau$
  for $h = 1$ **to** H **do**
    $\hat{z}^{(h)}_{\tau,0} \leftarrow f_{\theta,h}(z^{\text{start}}, \boldsymbol{s}, \boldsymbol{a}, \tau, 0)$
    $\hat{z}^{(h)}_{\tau,t_{\text{mid}}} \leftarrow f_{\theta,h}(z_{t_{\text{mid}}}, \boldsymbol{s}, \boldsymbol{a}, \tau, t_{\text{mid}})$
    $\mathcal{L}^{(h)} \leftarrow \sum_{t \in \{0,\, t_{\text{mid}}\}} \mathcal{L}^{\text{total}}_t\left(\hat{z}^{(h)}_{\tau,t}\right)$ by Eq. (12)
  end for
  (D) Loss and updates.
  $\mathcal{L}_{\text{total}} \leftarrow \frac{1}{H} \sum_{h=1}^{H} \mathcal{L}^{(h)}$
  Update θ by descending $\nabla_\theta \mathcal{L}_{\text{total}}$
  Soft-update target parameters $\phi \leftarrow \tau_{\text{tgt}}\theta + (1 - \tau_{\text{tgt}})\phi$
end for

Part II: Inference (Q-value evaluation)
Input: a state-action pair $(\boldsymbol{s}, \boldsymbol{a})$
Sample $\tau \sim U([0,1])^{K_{\text{onl}}}$ and set $z^{\text{start}} \leftarrow \tau$
$\{z^{(h)}\}_{h=1}^{H} \leftarrow \text{Generator}(z^{\text{start}}, \boldsymbol{s}, \boldsymbol{a}, \tau, \theta)$ by Algorithm 2
Stack atoms $Z \leftarrow \text{Concat}(z^{(1)}, \ldots, z^{(H)})$
Output $Q(\boldsymbol{s}, \boldsymbol{a}) \leftarrow \text{Mean}(Z)$

F BASELINE METHODS AND IMPLEMENTATION DETAILS

Here we provide additional details on the baseline methods reported in Table 2 of the main paper. We summarize their distributional assumptions and generative mechanisms, and clarify the implementation and comparison protocols adopted in our experiments.

TD3 Fujimoto et al. (2018). CDQ (Clipped Double Q-Learning), originally introduced in TD3, models the critic as a scalar-valued function and mitigates overestimation by taking the minimum over multiple Q-value estimates. It does not explicitly maintain a return distribution and thus corresponds to a non-distributional critic with an implicit point-mass assumption. The official torch implementation for TD3 is available at:

`https://github.com/sfujim/TD3`.

SAC Haarnoja et al. (2018). SAC (Soft Actor-Critic) is an off-policy actor-critic algorithm that learns a stochastic policy under the maximum-entropy reinforcement learning framework. By explicitly encouraging policy entropy during training, SAC improves exploration and robustness while maintaining strong sample efficiency. In practice, SAC maintains a pair of Q-functions and uses a clipped double-Q target to mitigate overestimation. It also supports automatic temperature adjustment, which adapts the strength of the entropy regularization to match a target entropy. We implement SAC in our unified codebase and use it as the default actor backbone in our experiments. The official tensorflow implementation for SAC is available at:

`https://github.com/haarnoja/sac`.

QVPO Ding et al. (2024). QVPO (Q-weighted Variational Policy Optimization) is an online diffusion-policy reinforcement learning algorithm that parameterizes the actor as a conditional diffusion model, thereby capturing multimodal action distributions. To bridge diffusion-model training with online policy optimization, QVPO introduces a Q-weighted variational objective that re-weights the diffusion training signal using value estimates and further employs a weight transformation to handle general settings where raw Q-values may not be strictly positive. In addition, QVPO incorporates a tailored entropy regularization to encourage exploration and proposes an efficient behavior/action selection strategy to reduce the variance of diffusion-policy rollouts during online interaction. We follow the official implementation and integrate it into our unified framework:

`https://github.com/wadx2019/qvpo`.

DSAC Duan et al. (2022). DSAC models the return distribution as a Gaussian distribution, parameterized by a learned mean and variance. Under this assumption, the entire return distribution can be represented compactly, and target sampling reduces to sampling from a Gaussian. We base our implementation on the official DSAC codebase:

`https://github.com/Jingliang-Duan/DSAC-v2`.

Specifically, we integrate the `DSAC-v1` implementation into our unified code framework and use it as the DSAC baseline.

IQN Dabney et al. (2018a). IQN explicitly models the continuous quantile function of the return distribution. Quantile levels are embedded using cosine features, which improves the representation of high-frequency distributional structures. We implement IQN following the original paper. When combined with a SAC actor, IQN can be viewed as a particular instantiation of the DSAC Ma et al. (2025b) framework, where DSAC here denotes the general distributional SAC formulation rather than the Gaussian-based DSAC baseline discussed earlier.

VF Dong et al. (2025). Value Flows (VF) combine Flow Matching with distributional critics and primarily target offline reinforcement learning. The original implementation is written in JAX and available at:

`https://github.com/chongyi-zheng/value-flows`.

We re-implement the method in PyTorch based on the official code and extract only the critic component for comparison, decoupling it from the original actor design.

VD Hu et al. (2025). VDRL (Value Diffusion Reinforcement Learning) models the return distribution using diffusion models and is designed for online reinforcement learning. At the time of submission, no official implementation is publicly available. We implement VDRL from scratch based on the descriptions in the original paper and integrate it into our unified framework as a baseline.

Implementation and comparison protocol. To ensure a fair and consistent comparison across baselines, we adopt the following unified design principles:

- **Implementation source.** For methods with official implementations (e.g., DSAC, VF), we base our implementations on the official code. For methods originally implemented in JAX (e.g., VF), we provide corresponding PyTorch re-implementations. All baselines are integrated into the same unified code framework.
- **Actor-critic decoupling.** For all baselines, we decouple the critic from the actor and evaluate only the critic component under a unified training and evaluation pipeline.

Table 8: Comparison of baseline methods by distributional assumption and generative mechanism.

Method	Distribution Assumption	Representation	Generative Mechanism	Aggregation
CDQ	Point mass	Scalar $Q(s,a)$	Deterministic evaluation	Min
DSAC	Gaussian	(μ, σ)	Gaussian sampling	Mean
IQN	Quantile	$F^{-1}(\tau)$	Quantile sampling	Mean/Min
VF	Flow	Velocity field	Flow-based sampling	Mean/Min
VDRL	Diffusion	noise predictor	Reverse diffusion	Mean/Min
DBC (Ours)	Diffusion Bridge	End Point $\hat{z}(\tau, t)$	ODE bridge sampling	DropTop

- **Target construction.** For entropy-regularized methods, target return samples are constructed as
$$r + \gamma\big(Z^{\mathrm{tgt}}(s', a') - \alpha \log \pi(a' \mid s')\big).$$
For non-entropy-based methods, target return samples are constructed as
$$r + \gamma Z^{\mathrm{tgt}}(s', a').$$
- **Distributional aggregation.** For critics with multiple heads or multiple return distributions, we consider the following aggregation schemes:
  1. Mean aggregation:
  $$\frac{1}{H}\sum_{h=1}^{H} \mathbb{E}[Z_h].$$
  2. Min aggregation:
  $$\mathbb{E}\big[\min_h Z_h\big].$$
  3. Drop-Top aggregation:
  $$\frac{1}{HK - d \cdot H} \sum_{l=1}^{HK - d\cdot H} Z_{(l)},$$
  where $\{Z_{(l)}\}_{l=1}^{HK}$ are the order statistics of all quantile estimates from H heads (each with K outputs), we use $H = 2$ in DBC.

These design choices ensure that all baselines are evaluated under consistent training signals and target semantics, isolating the effects of different distributional assumptions and generative mechanisms. We provide a concise summary of these methods in Table 8 to facilitate a clear comparison and highlight the key differences among them.

G Detailed Experimental Settings

In this section, we briefly describe the hyperparameters used for each baseline in our experiments. Since our codebase adopts a fully decoupled design for the actor and the critic, we present their hyperparameters separately. Specifically, we report the settings for the actor algorithms, QVPO, SAC, and TD3, as well as for the critic algorithms, DSAC, IQN, DBC, CDQ, VF, and VD. A detailed summary of the actor hyperparameters is provided in Table 9.

Table 9: Actor-side hyperparameters for QVPO, SAC, and TD3. Here, "–" indicates that the corresponding actor does not involve this hyperparameter.

Parameter	**QVPO**	**SAC**	**TD3**
Base Settings			
Discount factor γ	0.99	0.99	0.99
Target update rate τ	0.005	0.005	0.005
Actor Network Architecture			
Hidden units	256	256	256
Time embedding dim	32	–	–
Hidden layers	3	3	3
Activation layer	Mish	ReLU	ReLU
Diffusion Parameterization			
Number of diffusion steps T	20	–	–
Beta schedule	Cosine	–	–
Predict noise ϵ	True	–	–
Noise ratio	1.0	–	–
Optimization (Actor Only)			
Actor learning rate	1×10^{-4}	3×10^{-4}	3×10^{-4}
Gradient clipping	2.0	1.0	1.0
Sampling / Policy Evaluation			
Train samples per state	64	–	–
Eval samples per state	32	–	–
Regularization / Entropy			
Entropy coefficient α	0.02	1	–
Minimum α	0.002	–	–
Update Frequency			
Policy Delay	1	1	2

Due to the nature of distributional reinforcement learning, the sampling and aggregation schemes of distributional critics are an important part of the experimental configuration and thus are reported separately in Table 11. In this table, we only specify the aggregation mode for combining the outputs of heads; by default, the final critic output is obtained by averaging over samples (i.e. $\frac{1}{n}\sum_{i=1}^{n} z_i$). DSAC is excluded from Table 11 because it models the return distribution as a Gaussian and directly predicts the return mean (and variance), rather than relying on multi-sample estimation and aggregation as in other distributional methods.

Table 10: Common critic hyperparameters across methods.

Parameter	**DSAC-V1**	**IQN**	**DBC**	**CDQ**	**ValueFlows**	**VD**
Network Structure	MLP	IQN	MLP	MLP	MLP	MLP
Hidden width	256	512	512	512	512	256
Parameter Counts	0.14M	0.58M	1.14M	1.08M	1.08M	0.29M
Activation	GELU	ReLU	ReLU	ReLU	ReLU	Mish
Number of heads	1	2	2	2	2	2
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam
Learning rate	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}
Adam ϵ	10^{-5}	10^{-5}	10^{-5}	10^{-5}	10^{-5}	10^{-8}
Gradient clip (max norm)	1.0	1.0	1.0	1.0	1.0	1.0
Discount γ	0.99	0.99	0.99	0.99	0.99	0.99
Target Polyak τ	0.005	0.005	0.005	0.005	0.005	0.005

Table 11: Comparison of online/target sampling counts and aggregation strategies. IQN aggregates via minimum over sampled quantiles;

Method	Aggregation	Online Samples	Target Samples	Embedding Dimension
VF	Mean	1	1	1
VD	Min	5	10	16
IQN	Min	32	32	64
DBC	Drop-top	64	128	32

Table 12 presents the configurations for DBC.

Table 12: Drop ratios for DBC across environments. DBC uses environment-specific truncation ratios under a higher-resolution target distribution.

	HalfCheetah-v5	Ant-v5	Walker2d-v5	Humanoid-v5	Hopper-v5
Drop / Total	$0/128$	$12/128$	$14/128$	$12/128$	$32/128$

H Full Experiments

Experimental Environment. All experiments were conducted on a Linux server running Ubuntu 22.04.5 LTS with kernel version 5.15.0. The machine is equipped with two Intel Xeon Platinum 8480+ CPUs, providing 224 logical cores in total and 2 TiB of system memory. For acceleration, we use 8 NVIDIA H20 GPUs, each with 96 GB of HBM memory, driven by NVIDIA driver 580.95.05. The software stack consists of Python 3.12.12 and PyTorch 2.9.1 compiled with CUDA 12.8, with CUDA runtime support verified during execution. Unless otherwise stated, all reported results are obtained under this hardware and software configuration.

Throughout this section, unless stated otherwise, we adopt the following default hyperparameters: $M = 5$, $w = 0.01$, and $\beta(t) = 1$.

H.1 Detailed Results for All Baselines

We present the complete version of Table 2 in Table 13, in which all corresponding actor-critic combinations are fully annotated.

Table 13: Experiments Performance for all baselines on MuJoCo benchmarks

Run	Ant-v5	HalfCheetah-v5	Hopper-v5	Humanoid-v5	Walker2d-v5
QVPO+CDQ	6418.0 ± 61.8	12043.4 ± 438.8	3603.5 ± 46.2	5037.9 ± 29.6	5041.1 ± 264.2
SAC+CDQ	6170.8 ± 392.1	12738.1 ± 293.8	3648.7 ± 49.0	5170.7 ± 99.7	4912.7 ± 357.9
SAC+DSAC	2033.3 ± 625.5	12909.1 ± 195.5	3508.8 ± 22.3	3463.3 ± 232.0	3320.5 ± 1198.4
SAC+IQN	3449.0 ± 854.4	10973.9 ± 146.6	3282.0 ± 378.1	4771.8 ± 545.7	4728.4 ± 276.2
SAC+VD	3865.8 ± 957.5	4254.8 ± 502.1	1062.2 ± 19.0	4872.2 ± 233.4	2574.4 ± 275.2
SAC+VF	2634.4 ± 170.4	8215.9 ± 1441.3	3320.3 ± 26.6	5004.8 ± 58.5	2587.0 ± 1470.6
TD3+CDQ	4329.4 ± 1609.0	11725.7 ± 611.2	3614.2 ± 40.0	5098.2 ± 32.0	5022.4 ± 468.0
SAC+DBC(*)	6481.9 ± 88.5	13172.7 ± 155.9	**3754.9 ± 86.5**	**5877.2 ± 188.7**	6162.8 ± 41.5
QVPO+DBC(*)	**6593.9 ± 73.3**	13129.4 ± 265.0	3711.0 ± 121.3	5453.4 ± 8.5	5502.6 ± 181.4
TD3+DBC(*)	5359.1 ± 1212.4	**13801.6 ± 336.1**	3700.4 ± 70.2	5311.1 ± 207.3	**6373.5 ± 253.2**

H.2 Ablation for Different Schedules

We observe that the constant and linear schedules exhibit comparable performance, whereas the cosine schedule results in substantially higher variance. Therefore, for the case of $M = 5$, we recommend using either the constant or the linear schedule.

Table 14: Ablation for different schedules

Schedule	Ant-v5	HalfCheetah-v5	Hopper-v5	Humanoid-v5	Walker2d-v5
Constant	6481.9 ± 88.5	**13172.7 ± 155.9**	**3754.9 ± 86.5**	**5877.2 ± 188.7**	6162.8 ± 41.5
Cosine	5358.3 ± 1628.9	12538.9 ± 266.1	**3748.6 ± 144.4**	5623.8 ± 368.7	**6326.6 ± 751.9**
Linear	**6544.5 ± 113.9**	13004.6 ± 174.7	3728.6 ± 50.9	5594.4 ± 305.7	6116.7 ± 517.3

H.3 Bridge Type: GOU versus Flow Matching

Within our framework the bridge is determined by the interpolation coefficient $\xi(t)$ in Eq. (5). Flow Matching corresponds to the zero-drift limit $\beta(t) \to 0$, for which $\xi(t) = 1 - t$, i.e. the linear interpolation between the two endpoints; the GOU bridge instead uses $\beta(t) > 0$, which makes $\xi(t)$ non-linear in t and adds a mean-reverting pull towards the data endpoint. The two generators therefore differ only in how $\xi(t)$ is chosen.

Table 15: GOUB versus Flow Matching, i.e. the two choices of $\xi(t)$ in Eq. (5): $\beta(t) > 0$ (GOUB) and the zero-drift limit $\xi(t) = 1 - t$ (FM).

Generator	Hopper-v5	Walker2d-v5
GOUB	**3870.0**	**6238.7**
FM	3767.1	5779.8

We re-ran the comparison under the same hyperparameters as the main experiments, changing only the generator. The results are reported in Table 15.

H.4 Ablation for Different Weights

We further investigate the effect of different values of the weight w across a wider range of tasks. Empirical results show that a small weight w (e.g., 0.01) consistently leads to performance improvements. However, as w increases, the performance may begin to deteriorate. For reference, when training is performed solely with the Anchor Loss, the resulting performance is generally poor. This indicates that the Anchor Loss should not be used in isolation, but rather as a weak auxiliary loss to facilitate convergence. Based on these observations, we recommend setting $w = 0.01$.

Table 16: Ablation of Anchor Loss Weight w

	Ant-v5	HalfCheetah-v5	Hopper-v5	Humanoid-v5	Walker2d-v5
$w = 0.0$	4828.3 ± 1439.0	12990.1 ± 719.7	3646.3 ± 84.9	5807.3 ± 175.8	5562.9 ± 423.2
$w = 0.01$	**6481.9 ± 88.5**	**13172.7 ± 155.9**	**3754.9 ± 86.5**	**5877.2 ± 188.7**	**6162.8 ± 41.5**
$w = 0.02$	3947.3 ± 2151.8	13165.8 ± 301.2	3702.6 ± 86.5	5743.4 ± 301.1	6097.8 ± 375.1
$w = 0.1$	5716.2 ± 741.3	12775.5 ± 127.8	3591.1 ± 79.2	5589.0 ± 50.3	5974.6 ± 603.2
No Eq. (9)	2517.1 ± 2464.7	12393.1 ± 314.9	3244.0 ± 497.5	3751.2 ± 321.1	3859.3 ± 1828.7

Table 17: Ablation of Flow Steps M

	Ant-v5	HalfCheetah-v5	Hopper-v5	Humanoid-v5	Walker2d-v5
M=1	5150.4 ± 2226.3	12821.5 ± 26.4	**3748.1 ± 46.8**	5524.7 ± 132.7	5809.8 ± 308.6
M=5	**6481.9 ± 88.5**	**13172.7 ± 155.9**	**3754.9 ± 86.5**	**5877.2 ± 188.7**	**6162.8 ± 41.5**
M=10	5946.6 ± 379.7	13066.6 ± 203.0	3607.9 ± 30.5	4848.7 ± 88.1	6014.3 ± 561.6

Table 18: Training time (in hours) required by different critics to complete 1M environment steps. All experiments use SAC as the actor backbone and are evaluated on MuJoCo v5 benchmarks.

Critic	**Ant-v5**	**HalfCheetah-v5**	**Hopper-v5**	**Walker2d-v5**	**Humanoid-v5**
CDQ	3.0	3.0	3.0	3.0	3.5
IQN	4.0	4.0	4.0	4.0	3.8
DBC	7.0	7.0	7.0	7.0	8.0

H.5 Ablation for Different Time Steps

We further examine the impact of the number of time steps (M) across a wider range of tasks. Consistent with the results in the main text, we find that (M = 5) yields the best overall performance. Results are shown in Table 17.

H.6 Training Time Across Benchmarks

As shown in Table 18, although DBC employs a multi-step reward generation process in the critic, the resulting training-time overhead remains manageable. Compared to other baselines, DBC incurs only an approximately $2\times$ increase in training time.