Bootstrapped Bipartite Actor-Critic for Diffusion RL

Tianze Zhu[1†] **Yinuo Wang**[4†] **Letian Tao**[1†] **Tianyi Zhang**[1] **Likun Wang**[5]
Feihong Zhang[1] **Jingliang Duan**[1,2*] **Yao Lyu**[1*] **Shengbo Eben Li**[1,3*]

[1]School of Vehicle and Mobility, Tsinghua University
[2]School of Mechanical Engineering, University of Science and Technology Beijing
[3]College of Artificial Intelligence, Tsinghua University
[4]2030Lab, Yinwang Intelligent Technology Co., Ltd.
[5]National University of Singapore

[†]Equal contribution. [*]Corresponding authors.

Abstract

Diffusion policies have achieved strong performance in reinforcement learning (RL) due to the exceptional expressivity in capturing complex action distributions. However, existing approaches focus primarily on positive samples and lack explicit modeling of negative samples, thereby failing to fully exploit this representation capacity and ultimately limiting performance. To mitigate this problem, we propose BOBAC (BOotstrapped Bipartite Actor-Critic), a novel method for diffusion policy optimization via pairwise preference learning. Bridging the gap between distribution modeling and preference learning, we reformulate diffusion policy optimization within a preference-aware framework via the Bradley-Terry (BT) modeling, which enables the effective exploitation of negative samples through relative ranking. Since the original BT loss assigns the same weight to all pairs regardless of their varying importances, we further design a soft-gate mechanism to adaptively weight each pair according to its Q-gap. Experimental results on 6 MuJoCo tasks and 4 H1Bench tasks demonstrate that, compared with representative model-free and generative-policy RL baselines, BOBAC achieves state-of-the-art performance with consistent improvements in all continuous-control environments.

1 Introduction

Reinforcement learning (RL) has become a standard framework of sequential decision-making and control for autonomous driving and robotics [19, 35, 36, 21, 38, 40, 41, 43, 42]. Generative policies, especially diffusion policies, have emerged as powerful actors due to their ability to expressively model complex and multimodal action distributions [7, 20, 5, 1]. Unlike conventional unimodal Gaussian policies, diffusion policies can better capture settings where a single state admits multiple action modes, particularly in high-dimensional action spaces [16, 37]. However, how to fully exploit this modeling power during online RL optimization remains an open problem, since the critic-induced target distribution keeps shifting during training, making it difficult for the policy to capture the complex and non-stationary action-value landscape.

To mitigate this problem, existing approaches can be broadly grouped into two main categories. The first category consists of reparameterization-based methods, which optimize the policy by backpropagating signals of critic gradient through the denoising or generation process [33, 23, 44]. The second category encompasses Q-weighting methods, which improve diffusion policy optimization by Q-guided refinement [9, 25, 39, 28].

Despite their different forms, both categories mainly optimize diffusion policies with positive samples, while making limited use of negative samples as informative signals for policy improvement. This

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

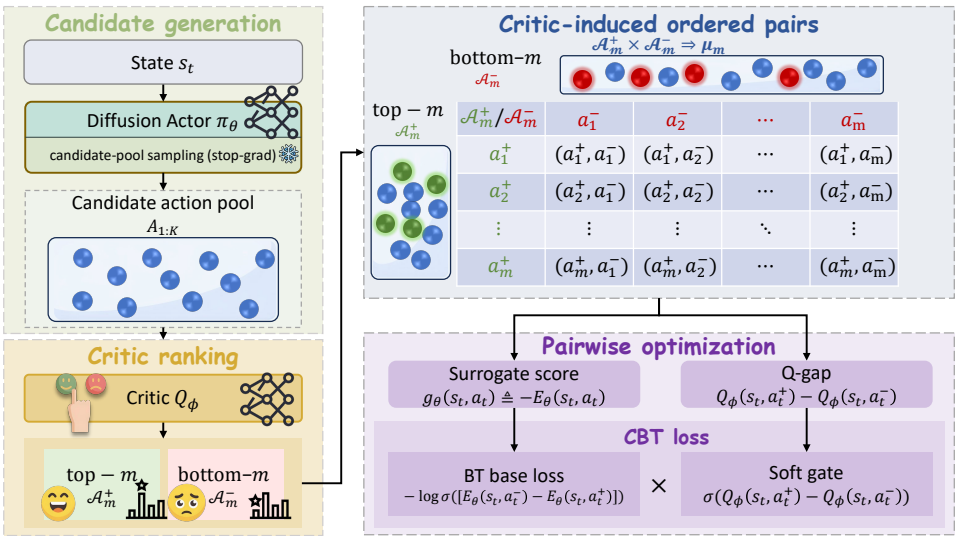


Figure 1: **The architectural overview of BOBAC.** BOBAC reformulates diffusion actor optimization as critic-induced pairwise preference learning. For each state, the actor samples a local candidate pool, the critic ranks it to construct top-m and bottom-m action subsets.

limitation is important because, for a given state, a diffusion actor can naturally generate multiple candidate actions that span both positive and negative samples with different Q-values. Existing objectives usually treat these actions independently, without explicitly modeling the relative preferences among them. As a result, the expressive power of diffusion policies is not fully activated during RL optimization, which limits their performance ceiling [11]. In contrast, jointly using positive and negative samples and explicitly modeling their relative preferences can provide richer supervision for diffusion policy optimization [8, 15, 2, 6, 27, 29].

Motivated by this observation, we propose **BO**otstrapped **B**ipartite **A**ctor-**C**ritic (BOBAC), a novel method for diffusion policy optimization in online RL. Specifically, for each state, we construct positive–negative action pairs according to their Q-values and leverage the Bradley–Terry (BT) loss with a surrogate objective for diffusion policies to learn pairwise preferences. Unlike existing updates that mainly rely on positive samples, the proposed BT loss guides diffusion policies through explicit relative preference learning among sampled actions, yielding an update signal that simultaneously favors better actions and suppresses worse ones. Moreover, different positive–negative pairs should not contribute equally, since their importance varies with the Q-gap between the two actions. To address this issue, we introduce a Q-gap-aware soft-gating mechanism. Using the Q-gap between positive and negative actions, this mechanism adaptively adjusts the weights of sampled comparisons, places greater emphasis on reliable pairwise supervision, and makes contrastive policy optimization more targeted and stable. Our main contributions are summarized as follows:

- We introduce the BT loss into diffusion policy optimization. By deriving a tractable surrogate score for diffusion policies, we make BT preference learning a practical training objective over positive and negative samples.
- We introduce a Q-gap-aware soft-gating mechanism. By adaptively reweighting sampled comparisons according to their Q-gap, it emphasizes more reliable pairwise supervision and improves the stability of policy optimization.
- We validate the proposed method on a 10-task benchmark spanning 6 MuJoCo tasks and 4 H1Bench tasks. BOBAC achieves the best performance across all evaluated tasks and outperforms 10 representative baselines, demonstrating the effectiveness and scalability of relative preference-based policy improvement for diffusion RL.

2 Related Works

2.1 Reparameterization-based Methods in Generative RL

These methods optimize diffusion policies by backpropagating critic signals through the denoising or generation process. By treating the reverse diffusion chain [14] or flow trajectory [22] as a differentiable computation graph, they directly optimize action quality through end-to-end value gradients. Representative methods such as the DACER series [33, 34, 44] and FlowRL [23] follow this paradigm. Although this line of work provides a direct mechanism for injecting reinforcement signals into expressive generative policies, it also introduces substantial optimization and computational challenges. More importantly, their updates are still mainly driven by moving sampled actions toward higher-value regions, rather than explicitly modeling which lower-value actions should be suppressed.

2.2 Q-weighted Methods in Generative RL

These methods optimize diffusion policies by incorporating critic-derived value information into the policy update, thereby biasing learning toward high-return regions. Representative works such as QVPO [9], DPMD [25], and SDAC [25] inject Q information into diffusion training through variational or exponential weighting schemes. DIPO [39] can also be viewed in this category, as it improves diffusion optimization through value-based target construction or dichotomous value-based supervision rather than explicit pairwise contrastive modeling. In addition, QSM [28] uses the Q-function to guide Langevin sampling toward higher-value actions. Although these methods differ in their specific instantiations, their updates remain fundamentally pointwise: they mainly improve sampled actions according to their individual values, without fully exploiting the within-state contrast among candidate actions, leading to less informative policy updates.

3 Preliminary

3.1 Reinforcement Learning

We consider a Markov decision process (MDP) defined by $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, $P(s' \mid s, a)$ is the transition dynamics, $r(s, a)$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor. A policy $\pi(a \mid s)$ specifies a distribution over actions conditioned on state s. The goal of reinforcement learning is to learn a policy that maximizes the expected discounted return.

In the off-policy actor-critic setting, transitions (s, a, r, s') are stored in a replay buffer $\mathcal{B}$, from which minibatches are sampled for training. The action-value function is defined as follows:

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}\Big[\sum_{t=0}^{\infty} \gamma^t r_t \,\Big|\, s_0 = s, a_0 = a\Big], \tag{1}$$

which measures the expected long-term return of taking action a at state s and following policy π thereafter. This value function is approximated by a parametric critic network $Q_\phi(s, a)$ with parameters ϕ. The actor is then optimized according to the critic's value signal.

3.2 Diffusion Model

We employ a conditional diffusion model, where x_0 denotes a clean sample and c its condition. The forward process perturbs x_0 into x_τ over T_{diff} steps, with

$$q(x_\tau \mid x_0) = \mathcal{N}(x_\tau; \sqrt{\bar{\alpha}_\tau} x_0, (1 - \bar{\alpha}_\tau) I), \tag{2}$$

which can be reparameterized using the standard Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$:

$$x_\tau = \sqrt{\bar{\alpha}_\tau} x_0 + \sqrt{1 - \bar{\alpha}_\tau}\epsilon, \tag{3}$$

where $\bar{\alpha}_\tau = \prod_{i=1}^{\tau} \alpha_i$ and $\alpha_\tau = 1 - \beta_\tau$ are determined by a predefined variance schedule $\{\beta_1, \ldots, \beta_{T_{\text{diff}}}\}$. At inference time, generation starts from $x_{T_{\text{diff}}} \sim \mathcal{N}(0, I)$ and iteratively denoises conditioned on c to produce x_0. During training, given a training dataset $\mathcal{D}$ of (c, x_0) pairs, the conditional diffusion model parameterized by θ is optimized using the simplified denoising objective:

$$\mathcal{L}_{\text{diff}}(\theta) = \mathbb{E}_{(c, x_0) \sim \mathcal{D},\, \tau,\, \epsilon}\left[\|\epsilon - \epsilon_\theta(x_\tau, c, \tau)\|_2^2\right], \tag{4}$$

where $\tau \sim \mathcal{U}(\{1, \ldots, T_{\text{diff}}\})$ and $\epsilon_\theta(x_\tau, c, \tau)$ predicts the injected noise.

3.3 Bradley–Terry Preference Modeling

For two candidate actions a^+ and a^- under the same state s, the Bradley–Terry (BT) model [3] defines the probability that a^+ is preferred over a^- as

$$P(a^+ \succ a^- \mid s) = \sigma\big(g_\psi(s, a^+) - g_\psi(s, a^-)\big), \tag{5}$$

where $\sigma(\cdot)$ is the sigmoid function, and $g_\psi(s, a)$ is a scalar score function parameterized by ψ, measuring the relative desirability of action a under state s.

Given a dataset of pairwise preferences $\mathcal{D}_{\text{pref}} = \{(s, a^+, a^-)\}$, the BT objective learns the model parameters by minimizing the negative log-likelihood:

$$\mathcal{L}_{\text{BT}}(\psi) = \mathbb{E}_{(s,a^+,a^-)\sim\mathcal{D}_{\text{pref}}}\left[-\log \sigma\big(g_\psi(s, a^+) - g_\psi(s, a^-)\big)\right]. \tag{6}$$

4 Methodology

In this section, we detail the design of BOBAC. Our method changes the actor update from a pointwise rule to a pairwise rule. Rather than only pulling the policy toward high-Q actions, we compare actions sampled under the same state and explicitly enlarge the gap between better and worse ones. In this way, the actor is encouraged to assign higher probability to positive samples while suppressing negative ones. We implement this idea with two parts: BT preference learning for diffusion policies and a Q-gap-aware soft gate. First, we build BT preference learning for diffusion policies by sampling a local action pool and using the critic Q_ϕ to form positive and negative pairs. Next, we build a tractable BT objective with a surrogate score for diffusion policies. We then introduce a Q-gap-aware soft gate and finally describe the training procedure for diffusion actors in online RL. Finally, we propose a practical algorithm for optimizing diffusion models.

4.1 BT Preference Learning for Diffusion Policies

4.1.1 Critic-guided Positive–Negative Pair Construction

For diffusion policies, a single state can admit several sampled actions with different values. We use this within-state difference to build a pairwise actor update: actions with higher values should become more likely, while worse actions from the same pool should become less likely. To realize this update, we first sample a candidate action pool from the current policy at state s_t:

$$A_{1:K} = (a_t^1, \ldots, a_t^K), \qquad a_t^k \sim \pi(\cdot \mid s_t),\ k = 1, \ldots, K. \tag{7}$$

This candidate pool limits subsequent comparisons to actions sampled under the same state and the same policy. We rank candidate actions using a learned value approximation $Q_\phi(s_t, a_t)$. Let ρ denote the permutation of $\{1, \ldots, K\}$ such that

$$Q_\phi(s_t, a_t^{\rho(1)}) \geq Q_\phi(s_t, a_t^{\rho(2)}) \geq \cdots \geq Q_\phi(s_t, a_t^{\rho(K)}). \tag{8}$$

Based on this ordering, we define the top-m and bottom-m subsets as

$$\mathcal{A}_m^+(s_t; A_{1:K}) = \{a_t^{\rho(1)}, \ldots, a_t^{\rho(m)}\}, \qquad \mathcal{A}_m^-(s_t; A_{1:K}) = \{a_t^{\rho(K-m+1)}, \ldots, a_t^{\rho(K)}\}, \tag{9}$$

and regard them as ordered action pairs drawn from a critic-induced joint distribution over $\mathcal{A} \times \mathcal{A}$. Given the sampled candidate pool, we define the empirical pair distribution as

$$\mu_m(a_t^+, a_t^- \mid s_t; A_{1:K}) = \frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} \mathbb{I}\Big((a_t^+, a_t^-) = (a_t^{\rho(i)}, a_t^{\rho(K-m+j)})\Big), \tag{10}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function. Concretely, the ordered pair (a_t^+, a_t^-) is sampled as

$$(a_t^+, a_t^-) \sim \mu_m(\cdot, \cdot \mid s_t; A_{1:K}). \tag{11}$$

4.1.2 Surrogate BT Objective for Diffusion Policies

After defining the empirical pair distribution μ_m in Section 4.1.1, we now build the BT objective for actor optimization. Following the generic BT formulation in Section 3.3, we instantiate the generic score function g_ψ with the diffusion-policy parameters θ, and denote the resulting method-specific score by $g_\theta(s_t, a_t)$. An ideal choice for this score is the policy conditional log-likelihood $\log \pi_\theta(a_t \mid s_t)$, since the BT model depends only on score differences. Under this ideal choice, Eq. (6) becomes

$$\mathcal{L}_{\mathrm{BT}}^{\mathrm{ideal}}(\theta) = \mathbb{E}_{(s_t, a_t^+, a_t^-) \sim \mathcal{D}_{\mathrm{pref}}} \left[-\log \sigma\big(\log \pi_\theta(a_t^+ \mid s_t) - \log \pi_\theta(a_t^- \mid s_t) \big) \right], \tag{12}$$

Minimizing this loss directly pushes the policy distribution toward the preferred actions. However, for diffusion policies, the exact conditional log-density $\log \pi_\theta(a_t \mid s_t)$ is generally intractable. Therefore, we adopt a diffusion-consistent surrogate score [26]. Under the variational formulation of diffusion models, the conditional log-likelihood admits the decomposition:

$$\log \pi_\theta(a_t \mid s_t) = \mathrm{ELBO}_\theta(s_t, a_t) + \Delta_{\mathrm{KL}}^\theta(s_t, a_t), \tag{13}$$

where ELBO denotes the evidence lower bound and $\Delta_{\mathrm{KL}}^\theta(s_t, a_t)$ denotes the variational gap, detailed in Appendix A.1, where we bound the effect of dropping this term. We then adopt $\mathrm{ELBO}_\theta(s_t, a_t)$ as a tractable surrogate to estimate the policy likelihood. Under the standard noise-prediction parameterization, the denoising objective

$$E_\theta(s_t, a_t) = \mathbb{E}_{\tau \sim \mathcal{U}(\{1, \ldots, T_{\mathrm{diff}}\}),\, \epsilon \sim \mathcal{N}(0, I)} \left[\|\epsilon - \epsilon_\theta(a_t^\tau, s_t, \tau)\|_2^2 \right] \tag{14}$$

matches the negative ELBO exactly, apart from a fixed constant [17], where a_t^τ denotes the noisy action at diffusion step τ obtained from the forward process applied to a_t, and $\epsilon_\theta(a_t^\tau, s_t, \tau)$ is the noise prediction network. Therefore, we substitute $\log \pi_\theta(a_t \mid s_t) \approx -E_\theta(s_t, a_t) + \text{Const}$ into our ideal BT formulation in Eq. (12), yielding a tractable base loss:

$$\ell_{\mathrm{BT}}^{\mathrm{base}}(s_t, a_t^+, a_t^-; \theta) = -\log \sigma\big[E_\theta(s_t, a_t^-) - E_\theta(s_t, a_t^+) \big]. \tag{15}$$

We justify this design in Appendix A.1. Thus, lower denoising error corresponds to a higher surrogate score, providing a tractable quantity for pairwise comparison.

4.2 Q-gap-aware Soft Gating

While optimizing $\ell_{\mathrm{BT}}^{\mathrm{base}}$ successfully enforces relative action preferences, it fundamentally relies only on binary ranking and fails to fully exploit the magnitude of the Q-value discrepancy. To address this limitation and inject denser supervisory signals from the critic, we introduce a Q-gap-aware pairwise soft gate:

$$\omega(s_t, a_t^+, a_t^-) = \sigma\big(Q_\phi(s_t, a_t^+) - Q_\phi(s_t, a_t^-) \big). \tag{16}$$

This gate adaptively weights each comparison according to the Q-gap. Intuitively, action pairs with larger Q-gaps represent more reliable relative-preference signals and thus contribute more strongly to the actor update.

Conditioned on the sampled candidate pool $A_{1:K}$ and the induced empirical pair distribution $\mu_m(\cdot, \cdot \mid s_t; A_{1:K})$, we define the per-state Contrastive Bradley–Terry (CBT) loss as

$$\ell_{\mathrm{CBT}}(s_t; \theta) = \mathbb{E}_{A_{1:K} \sim \pi(\cdot \mid s_t)^K} \left[\mathbb{E}_{(a_t^+, a_t^-) \sim \mu_m(\cdot, \cdot \mid s_t; A_{1:K})} \left[\underbrace{\omega(s_t, a_t^+, a_t^-)}_{\text{pairwise soft gate}} \underbrace{\ell_{\mathrm{BT}}^{\mathrm{base}}(s_t, a_t^+, a_t^-; \theta)}_{\text{BT base loss}} \right] \right], \tag{17}$$

where π denotes the current actor used only for candidate-pool sampling, and $\pi(\cdot \mid s_t)^K$ denotes the product distribution of K independent samples from $\pi(\cdot \mid s_t)$. When differentiating $\ell_{\mathrm{CBT}}(s_t; \theta)$ with respect to θ, the pair construction induced by $A_{1:K}$ is treated as fixed.

Remark 1. *Using a sampled candidate pool $A_{1:K} \sim \pi(\cdot \mid s_t)^K$, Eq. (17) can be estimated by*

$$\hat{\ell}_{\mathrm{CBT}}(s_t, A_{1:K}; \theta) = \left[\frac{1}{m^2} \sum_{a_t^+ \in \mathcal{A}_m^+(s_t; A_{1:K})} \sum_{a_t^- \in \mathcal{A}_m^-(s_t; A_{1:K})} \omega(s_t, a_t^+, a_t^-)\, \ell_{\mathrm{BT}}^{\mathrm{base}}(s_t, a_t^+, a_t^-; \theta) \right]. \tag{18}$$

This estimator can be viewed as a Monte Carlo estimate of Eq. (17), obtained by sampling a candidate pool $A_{1:K}$ from the current actor. It is unbiased with respect to the outer expectation over candidate pools. For a fixed sampled pool $A_{1:K}$, the inner expectation under $\mu_m(\cdot, \cdot \mid s_t; A_{1:K})$ is evaluated exactly by averaging over all m^2 ordered positive–negative pairs induced by critic ranking.

Finally, the overall actor objective is

$$\mathcal{L}_{\mathrm{CBT}}(\theta) = \mathbb{E}_{s_t}\left[\ell_{\mathrm{CBT}}(s_t;\theta)\right]. \tag{19}$$

Remark 2. *Furthermore, under an idealized pair-sampling construction, the gradient induced by ℓ_{CBT} can be written as follows:*

$$-\nabla_\theta \ell_{\mathrm{CBT}}(s_t;\theta) = \mathbb{E}_{a_t\sim\pi(\cdot|s_t)}\left[\Psi(s_t,a_t)\nabla_\theta \log \pi_\theta(a_t \mid s_t)\right]. \tag{20}$$

Here, the weight $\Psi(s_t, a_t)$ is

$$\begin{aligned}\Psi(s_t,a_t) = \mathbb{E}_{b\sim\pi(\cdot|s_t)}\Big[&\sigma\big(g_\theta(s_t,b) - g_\theta(s_t,a_t)\big)\,\omega(s_t,a_t,b)\,\mathbb{I}\big[Q_\phi(s_t,a_t) > Q_\phi(s_t,b)\big] \\ &- \sigma\big(g_\theta(s_t,a_t) - g_\theta(s_t,b)\big)\,\omega(s_t,b,a_t)\,\mathbb{I}\big[Q_\phi(s_t,b) > Q_\phi(s_t,a_t)\big]\Big],\end{aligned} \tag{21}$$

where ω is the soft gate in Eq. (16) and all policy-dependent terms are evaluated at the current parameters. $\Psi(s_t, a_t)$ can be viewed as a generalized advantage term, as detailed in Appendix A.2 and A.3.

4.3 Training Procedure

To obtain a practical algorithm, we use a parameterized function approximation for the Q-function and the diffusion policy. In the critic component, we adopt the double Q-learning strategy [12] to alleviate overestimation bias. Specifically, we maintain two independent Q-function estimators, denoted as $Q_{\phi_1}(s_t, a_t)$ and $Q_{\phi_2}(s_t, a_t)$, which are trained to approximate the true action-value function. To enhance training stability, we introduce two corresponding target networks, $Q_{\bar{\phi}_1}(s_t, a_t)$ and $Q_{\bar{\phi}_2}(s_t, a_t)$, which are updated softly from the main networks following the technique in [32]. The Q-networks are optimized by minimizing the temporal difference error. For each network $Q_{\phi_i}(s_t, a_t)$, the objective $J_Q(\phi_i)$ is defined as follows:

$$J_Q(\phi_i) = \mathop{\mathbb{E}}_{\substack{(s_t,a_t,r_t,s_{t+1})\sim\mathcal{B}\\ a_{t+1}\sim\pi_\theta(\cdot|s_{t+1})}}\left[\left(\left(r(s_t,a_t) + \gamma \min_{j=1,2} Q_{\bar{\phi}_j}(s_{t+1},a_{t+1})\right) - Q_{\phi_i}(s_t,a_t)\right)^2\right], \tag{22}$$

where γ is discount factor, the target is derived from the minimum of the two target Q-values, $Q_{\bar{\phi}_1}(s_{t+1}, a_{t+1})$ and $Q_{\bar{\phi}_2}(s_{t+1}, a_{t+1})$, to prevent over-optimistic estimates. Furthermore, we adopt several stabilization details following DSACT [10] to further mitigate overestimation issues. The entire procedure loop is shown in Fig. 1, and the entire training pseudocode is shown in Algorithm 1.

5 Experiments

We conduct experiments on a 10-task benchmark consisting of 6 OpenAI Gym MuJoCo tasks [4] and 4 H1Bench tasks [31]. The MuJoCo tasks evaluate standard continuous-control and locomotion capabilities, while the H1Bench tasks introduce more challenging humanoid-control scenarios involving navigation, reaching, and whole-body coordination. With these results, we aim to answer three questions:

- How does BOBAC compare with previous online RL algorithms and existing generative-policy online RL algorithms?
- Do the performance gains specifically arise from structured contrastive preference learning, rather than from generic reweighting or arbitrary negative sampling?
- Is the method robust to different hyperparameter choices?

Algorithm 1 BOBAC: Bootstrapped Bipartite Actor-Critic

Input: Replay buffer $\mathcal{B}$, diffusion actor π_θ, twin critics Q_{ϕ_1}, Q_{ϕ_2}, target networks, candidate size K, subset size m

Initialize actor parameters θ, critic parameters ϕ_1, ϕ_2, and target networks

for each training iteration **do**

Sample a minibatch of B transitions $\{(s_t, a_t, r_t, s_{t+1})\}$ from $\mathcal{B}$

// Critic update

Update critics Q_{ϕ_1}, Q_{ϕ_2} by minimizing the Bellman loss in Eq. (22)

// Actor update

for each sampled state s_t in the minibatch **do**

Sample K candidate actions $A_{1:K}$ from $\pi_\theta(\cdot \mid s_t)$ according to Eq. (7)

Compute $\tilde{Q}(s_t, a_t) = \min\big(Q_{\phi_1}(s_t, a_t),\, Q_{\phi_2}(s_t, a_t)\big)$ for all $a_t \in A_{1:K}$

Rank $A_{1:K}$ by $\tilde{Q}(s_t, a_t)$ and construct $\mathcal{A}_m^+(s_t; A_{1:K})$ and $\mathcal{A}_m^-(s_t; A_{1:K})$ using Eq. (9)

Construct the empirical pair distribution $\mu_m(\cdot, \cdot \mid s_t; A_{1:K})$ according to Eq. (10)

Compute the base BT loss $\ell_{\mathrm{BT}}^{\mathrm{base}}$ using Eqs. (14) and (15)

Form the state-wise CBT loss $\ell_{\mathrm{CBT}}(s_t; \theta)$ according to Eq. (17)

Update actor θ by minimizing the CBT loss $\mathcal{L}_{\mathrm{CBT}}(\theta)$ in Eq. (19)

// Target update

Update target networks using delayed or soft target updates

return trained parameters θ, ϕ_1, ϕ_2

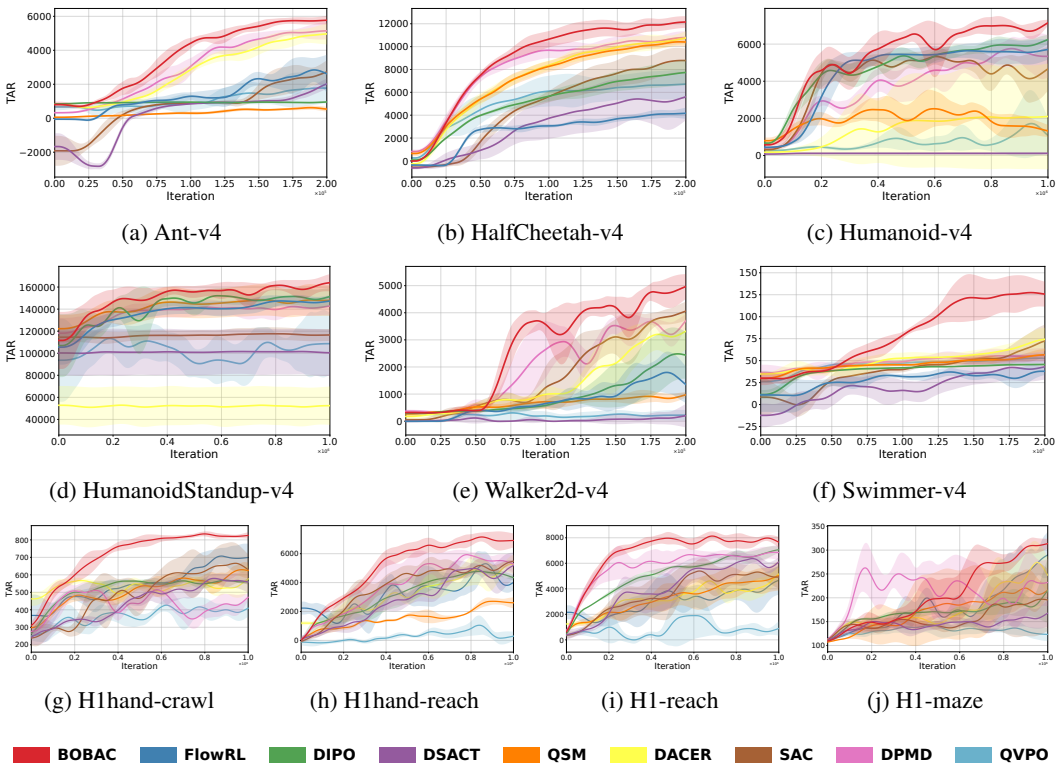
Table 1: Total Average Return (TAR) on 6 MuJoCo tasks and 4 H1Bench tasks. Results are reported as Mean ± Std over 5 seeds for MuJoCo tasks and H1Bench tasks. **Bold** and underline denote the best and second-best results, respectively. Higher is better.

		Humanoid-v4	Ant-v4	HalfCheetah-v4	Walker2d-v4	Swimmer-v4
Classic Model-Free RL	PPO	1158.3 ± 461.6	3004.8 ± 226.4	3157.1 ± 864.7	2609.9 ± 1211.8	125.2 ± 3.4
	TD3	5400.9 ± 138.9	3794.3 ± 800.7	10640.6 ± 578.7	1792.1 ± 1745.6	86.6 ± 36.3
	SAC	4388.6 ± 1121.1	2574.2 ± 561.2	8786.8 ± 1099.9	3966.0 ± 312.3	69.3 ± 17.3
	DSACT	114.3 ± 69.4	1849.6 ± 193.5	5421.2 ± 2320.2	154.6 ± 332.1	41.2 ± 14.0
Generative-Policy RL	QSM	1417.6 ± 282.0	584.7 ± 109.2	10426.1 ± 392.1	899.6 ± 126.9	56.2 ± 6.2
	DIPO	6090.2 ± 276.7	937.4 ± 27.4	7681.1 ± 1087.0	2493.5 ± 918.2	45.5 ± 1.1
	DACER	2076.5 ± 2812.3	4931.3 ± 544.6	10641.8 ± 480.2	3177.5 ± 721.3	73.0 ± 13.0
	FlowRL	5641.0 ± 602.9	2795.1 ± 761.0	4136.8 ± 362.4	1577.5 ± 701.4	37.9 ± 6.5
	QVPO	1462.7 ± 1376.1	1779.4 ± 946.3	6694.7 ± 1282.4	221.6 ± 195.8	52.8 ± 6.3
	DPMD	5339.7 ± 1242.9	5129.4 ± 499.1	10775.3 ± 558.7	3369.7 ± 374.7	56.0 ± 8.1
	BOBAC (ours)	**6905.3 ± 358.7**	**5745.7 ± 225.7**	**12128.9 ± 524.3**	**4857.8 ± 552.6**	**126.9 ± 15.2**
		HumanoidStandup-v4	H1-maze	H1-reach	H1hand-reach	H1hand-crawl
Classic Model-Free RL	PPO	114632.1 ± 14298.8	154.5 ± 25.8	3651.7 ± 162.5	2717.4 ± 219.8	324.0 ± 55.6
	TD3	120747.8 ± 17403.9	213.6 ± 71.4	5385.6 ± 39.9	5134.0 ± 401.8	696.7 ± 97.9
	SAC	116435.5 ± 5462.1	192.0 ± 65.6	5126.3 ± 670.4	5109.4 ± 164.7	649.2 ± 110.4
	DSACT	100589.9 ± 21175.7	162.0 ± 21.8	6026.2 ± 294.6	4648.3 ± 1027.6	561.8 ± 47.0
Generative-Policy RL	QSM	147862.5 ± 14062.9	216.6 ± 25.2	4784.2 ± 1058.9	2662.1 ± 99.7	615.0 ± 18.1
	DIPO	149635.6 ± 8245.9	194.4 ± 38.4	6865.1 ± 378.7	4521.4 ± 192.3	564.8 ± 7.2
	DACER	52415.7 ± 16996.3	254.2 ± 45.1	4772.3 ± 571.0	5244.3 ± 454.8	553.3 ± 70.6
	FlowRL	146440.3 ± 4975.4	282.9 ± 11.3	4492.4 ± 1082.6	3783.3 ± 405.9	695.6 ± 77.4
	QVPO	108635.1 ± 29424.9	125.3 ± 10.3	764.5 ± 468.0	214.8 ± 345.4	401.0 ± 34.9
	DPMD	141867.9 ± 7101.3	236.3 ± 55.2	6757.3 ± 452.0	5418.5 ± 417.4	447.3 ± 33.6
	BOBAC (ours)	**163074.9 ± 6743.7**	**311.8 ± 13.9**	**7800.1 ± 215.7**	**6894.8 ± 576.3**	**822.2 ± 12.3**

5.1 Experimental Setup

Baselines. The baselines encompass two categories of model-free RL algorithms. The first category consists of generative RL methods, including DACER [33], DPMD [25], QVPO [9], DIPO [39], QSM [28], and FlowRL [23]. The second category includes classic model-free online RL baselines, namely SAC [13], PPO [30], TD3 [12], and DSACT [10]. The experimental hyperparameters are provided in Appendix D.

Evaluation Setup. We implement the BOBAC algorithm in JAX and evaluate it on 6 OpenAI Gym MuJoCo v4 tasks and 4 H1Bench tasks. All experiments are conducted on a system equipped with an



(a) Ant-v4 (b) HalfCheetah-v4 (c) Humanoid-v4

(d) HumanoidStandup-v4 (e) Walker2d-v4 (f) Swimmer-v4

(g) H1hand-crawl (h) H1hand-reach (i) H1-reach (j) H1-maze

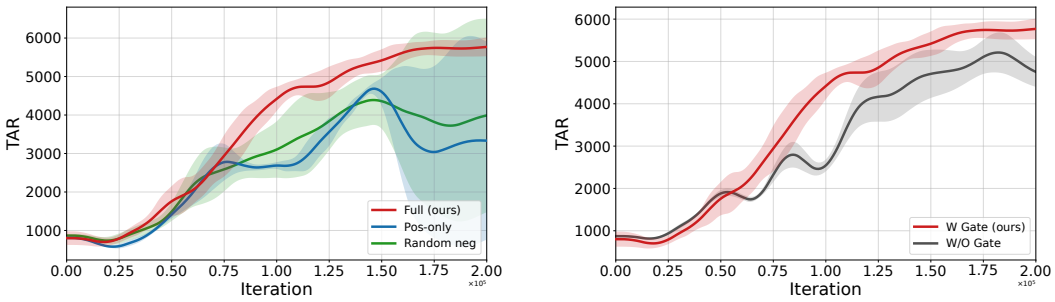
Figure 2: Learning curves on 6 MuJoCo tasks and 4 H1Bench tasks with 5 random seeds. Solid lines denote the mean Total Average Return (TAR), and shaded regions indicate 95% confidence interval. PPO and TD3 are omitted from curves for visual clarity but included in Table 1.

AMD Ryzen Threadripper 3960X 24-core processor and an NVIDIA GeForce RTX 4090 GPU. For the MuJoCo tasks, all environments except `Humanoid-v4` and `HumanoidStandup-v4` are trained for 200K iterations, corresponding to 1 million environmental interactions, while `Humanoid-v4` and `HumanoidStandup-v4` use five times as many interactions. For the H1Bench tasks, all methods are evaluated on `H1-maze`, `H1-reach`, `H1hand-reach`, and `H1hand-crawl`, with each task trained for 1 million environmental interactions.

5.2 Experimental Results

All the training curves are shown in Fig. 2, and the detailed numerical results are listed in Table 1. Overall, BOBAC achieves the best Total Average Return (TAR) across all 10 evaluated tasks, covering 6 standard MuJoCo tasks and 4 H1Bench humanoid-control tasks. These results demonstrate that contrastive preference-based actor optimization is effective across both conventional continuous-control benchmarks and more challenging high-dimensional humanoid-control scenarios.

On the MuJoCo suite, BOBAC achieves the best performance on all 6 tasks. Compared with SAC, BOBAC obtains large gains on representative challenging tasks, including `Humanoid-v4` (+57.3%), `Ant-v4` (+123.2%), `HalfCheetah-v4` (+38.0%), `Walker2d-v4` (+22.5%), and `HumanoidStandup-v4` (+40.1%). Moreover, BOBAC consistently surpasses the strongest generative-policy RL baseline on every task, with the largest gain of 73.8% on `Swimmer-v4`. These results indicate that pairwise preference learning provides a more effective actor update than both classic model-free RL and existing generative-policy RL methods. On the H1Bench suite, BOBAC achieves the best performance on all 4 tasks. Compared with the second-best method on each task, BOBAC improves performance by 10.2%, 13.6%, 27.2%, and 18.0% on `H1-maze`, `H1-reach`, `H1hand-reach`, and `H1hand-crawl`, respectively. This consistent advantage suggests that BOBAC scales well to complex humanoid-control tasks requiring whole-body coordination and high-dimensional action control.



(a) w/o negative guidance (b) w/o pairwise soft-gating

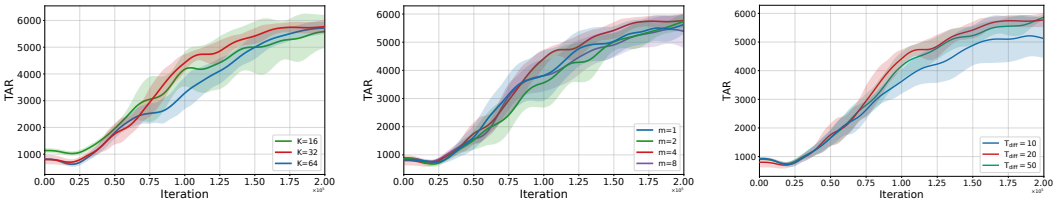
Figure 3: Ablation results on `Ant-v4` for negative guidance and pairwise soft-gating. Curves show the mean return over 5 random seeds. Shaded regions denote ± 1 standard deviation across runs at each training step (after smoothing), reflecting inter-run variability.

5.3 Ablation Study

In this section, we conduct ablation studies to examine the contributions of the key components in BOBAC and its sensitivity to major hyperparameters. Specifically, we study: 1) structured positive–negative preference modeling; 2) the pairwise soft gate; 3) the number of candidate actions K; 4) the top/bottom subset size m; and 5) the number of diffusion steps T_{diff}. The experiments are conducted on `Ant-v4`, and Appendix E.2 repeats the component ablations on the 61-dimensional `H1hand-reach` task with the same conclusions.

Positive–negative preference modeling. We compare BOBAC with two variants: **Pos-only**, which removes BT-style pairwise comparisons and updates the actor only with the high-ranked actions selected by the critic, using uniform weights, and **Random neg**, which replaces critic-guided negatives with randomly sampled ones. As shown in Fig. 3a, removing structured negative guidance substantially hurts both performance and stability. This suggests that the gains come not from one-sided reinforcement or arbitrary negative sampling, but from critic-guided pairwise preference learning.

Pairwise soft gate. We further evaluate the effect of the pairwise soft gate on performance. As shown in Fig. 3b, removing this component triggers a substantial drop in TAR and noticeably higher training instability. This indicates that not all pairwise comparisons are equally reliable, and that reliability-aware reweighting improves both optimization stability and final policy quality.



(a) sensitivity to K (b) sensitivity to m (c) sensitivity to T_{diff}

Figure 4: Hyperparameter sensitivity on `Ant-v4` for K, m, and T_{diff}. Curves show the mean TAR over 5 random seeds. Shaded regions denote ± 1 standard deviation across runs at each training step (after smoothing), reflecting inter-run variability. Our method remains reasonably stable around the default setting ($K = 32, m = 4, T_{\text{diff}} = 20$).

Hyperparameter sensitivity. Fig. 4 studies the effects of the candidate size K, subset size m, and diffusion steps T_{diff}. BOBAC remains reasonably stable around the default setting, with $K = 32$, $m = 4$, and $T_{\text{diff}} = 20$ providing a good trade-off between performance and computation. Too small K limits candidate diversity, while overly large m or T_{diff} brings no consistent improvement.

6 Conclusion

We present BOBAC, a contrastive actor optimization algorithm for online RL, which moves beyond conventional pointwise actor updates by formulating policy improvement as state-conditional pairwise preference learning. To this end, BOBAC constructs critic-guided positive–negative candidate actions from a local policy-sampled action pool and optimizes the actor with a unified CBT loss that combines BT-style preference modeling with Q-gap-aware pairwise soft-gating.

This formulation converts critic-induced rankings into structured pairwise supervision and yields an explicitly contrastive update through a diffusion-derived surrogate score based on denoising error, while the underlying objective remains broadly motivated by preference modeling under noisy critic supervision. Empirically, BOBAC achieves strong and consistent improvements over both standard model-free RL methods and recent generative-policy baselines across a 10-task benchmark spanning MuJoCo and H1Bench environments. Ablation studies indicate that the gains stem from the synergy between structured positive–negative preference modeling and the pairwise soft-gating mechanism, while compute-fairness analyses verify that the advantage is not merely an artifact of additional optimization-side computation.

Overall, our results suggest that contrastive actor optimization is a promising alternative to pointwise updates for online generative policy learning, especially in settings that require stable policy improvement over complex action distributions. A natural direction for future work is to extend this preference objective to other expressive policy classes beyond diffusion models and to explore more efficient preference construction strategies for improving sample efficiency and training stability in high-dimensional continuous-control RL.

Acknowledgments

This work was supported by the National Science and Technology Major Project (No. 2025ZD1606200), the Tsinghua–Efort Joint Research Center for EAI Computation and Perception, and SunRising AI Lab.

References

[1] Anurag Ajay, Yilun Du, Abhi Gupta, Joshua B. Tenenbaum, Tommi S. Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision making? In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

[2] Erdem Biyik and Dorsa Sadigh. Batch active preference-based learning of reward functions. In *2nd Annual Conference on Robot Learning, CoRL 2018, Zürich, Switzerland, 29-31 October 2018, Proceedings*, Proceedings of Machine Learning Research, pages 519–528. PMLR, 2018.

[3] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[4] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[5] Huayu Chen, Cheng Lu, Chengyang Ying, Hang Su, and Jun Zhu. Offline reinforcement learning via high-fidelity generative behavior modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

[6] Xiaoyu Chen, Han Zhong, Zhuoran Yang, Zhaoran Wang, and Liwei Wang. Human-in-the-loop: Provably efficient preference-based reinforcement learning with general function approximation. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, Proceedings of Machine Learning Research, pages 3773–3793. PMLR, 2022.

[7] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *Int. J. Robotics Res.*, 44(10-11):1684–1704, 2025.

[8] Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4299–4307, 2017.

[9] 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. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[10] 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 Trans. Pattern Anal. Mach. Intell.*, 47(5):3935–3946, 2025.

[11] Kevin Frans, Seohong Park, Pieter Abbeel, and Sergey Levine. Diffusion guidance is a controllable policy improvement operator. *arXiv preprint arXiv:2505.23458*, 2025.

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

[13] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, Proceedings of Machine Learning Research, pages 1856–1865. PMLR, 2018.

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[15] Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in atari. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8022–8034, 2018.

[16] Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, Proceedings of Machine Learning Research, pages 9902–9915. PMLR, 2022.

[17] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

[18] Diederik P. Kingma and Ruiqi Gao. Understanding diffusion objectives as the ELBO with simple data augmentation. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[19] Shengbo Eben Li. *Reinforcement Learning for Sequential Decision and Optimal Control*. Springer Verlag, Singapore, 2023.

[20] Steven Li, Rickmer Krohn, Tao Chen, Anurag Ajay, Pulkit Agrawal, and Georgia Chalvatzaki. Learning multimodal behaviors from scratch with diffusion policy gradient. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[21] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[22] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

[23] Lei Lv, Yunfei Li, Yu Luo, Fuchun Sun, Tao Kong, Jiafeng Xu, and Xiao Ma. Flow-based policy for online reinforcement learning. *arXiv preprint arXiv:2506.12811*, 2025.

[24] Yao Lyu, Xiangteng Zhang, Shengbo Eben Li, Jingliang Duan, Letian Tao, Qing Xu, Lei He, and Keqiang Li. Conformal symplectic optimization for stable reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(6):11049–11063, 2025.

[25] Haitong Ma, Tianyi Chen, Kai Wang, Na Li, and Bo Dai. Efficient online reinforcement learning for diffusion policy. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research, 2025.

[26] 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.

[27] Katherine Metcalf, Miguel Sarabia, Natalie Mackraz, and Barry-John Theobald. Sample-efficient preference-based reinforcement learning with dynamics aware rewards. In Jie Tan, Marc Toussaint, and Kourosh Darvish, editors, *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, Proceedings of Machine Learning Research, pages 1484–1532. PMLR, 2023.

[28] Michael Psenka, Alejandro Escontrela, Pieter Abbeel, and Yi Ma. Learning a diffusion model policy from rewards via q-score matching. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 41163–41182, 2024.

[29] Aadirupa Saha, Aldo Pacchiano, and Jonathan Lee. Dueling RL: reinforcement learning with trajectory preferences. In Francisco J. R. Ruiz, Jennifer G. Dy, and Jan-Willem van de Meent, editors, *International Conference on Artificial Intelligence and Statistics, 25-27 April 2023, Palau de Congressos, Valencia, Spain*, Proceedings of Machine Learning Research, pages 6263–6289. PMLR, 2023.

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

[31] Carmelo Sferrazza, Dun-Ming Huang, Xingyu Lin, Youngwoon Lee, and Pieter Abbeel. Humanoidbench: Simulated humanoid benchmark for whole-body locomotion and manipulation. In Dana Kulic, Gentiane Venture, Kostas E. Bekris, and Enrique Coronado, editors, *Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024*, 2024.

[32] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

[33] Yinuo Wang, Likun Wang, Yuxuan Jiang, Wenjun Zou, Tong Liu, Xujie Song, Wenxuan Wang, Liming Xiao, Jiang Wu, Jingliang Duan, and Shengbo Li. Diffusion actor-critic with entropy regulator. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[34] Yinuo Wang, Likun Wang, Mining Tan, Wenjun Zou, Xujie Song, Wenxuan Wang, Tong Liu, Guojian Zhan, Tianze Zhu, Shiqi Liu, et al. Enhanced dacer algorithm with high diffusion efficiency. *arXiv preprint arXiv:2505.23426*, 2025.

[35] Yinuo Wang, Wenxuan Wang, Xujie Song, Tong Liu, Yuming Yin, Liangfa Chen, Likun Wang, Jingliang Duan, and Shengbo Eben Li. Ode-based smoothing neural network for reinforcement learning tasks. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025.

[36] 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, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

[37] Ziqi Wang, Jiashun Liu, and Ling Pan. Learning intractable multimodal policies with reparameterization and diversity regularization. *arXiv preprint arXiv:2511.01374*, 2025.

[38] Zuoshuai Wang, Yao Lyu, Congsheng Zhang, Zhe Xin, Jiaxin Gao, Beiyan Jiang, and Shengbo Eben Li. Feasible reinforcement learning for safety ensurance with stabilized training: Z. wang et al. *Automotive Innovation*, 9(2):340–353, 2026.

[39] 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.

[40] Di Zeng, Ling Zheng, Xiantong Yang, and Yinong Li. Reinforced imitative trajectory planning for urban automated driving: D. zeng et al. *Automotive Innovation*, pages 1–17, 2026.

[41] Feihong Zhang, Guojian Zhan, Bin Shuai, Tianyi Zhang, Jingliang Duan, and Shengbo Eben Li. Distributional soft actor-critic with harmonic gradient for safe and efficient autonomous driving in multi-lane scenarios. In *2025 IEEE Intelligent Vehicles Symposium (IV)*, pages 1964–1970. IEEE, 2025.

[42] Tianyi Zhang, Wenhan Cao, Chang Liu, Tao Zhang, Jiangtao Li, and Shengbo Eben Li. Robust state estimation for legged robots with dual beta kalman filter. *IEEE Robotics and Automation Letters*, 10(8):7955–7962, 2025.

[43] Tianyi Zhang, Likun Wang, Guojian Zhan, Feihong Zhang, Yang Guan, Yao Lyu, and Shengbo Eben Li. Langevin rollout optimization for modelic reinforcement learning. In *Forty-third International Conference on Machine Learning*.

[44] Tianze Zhu, Yinuo Wang, Wenjun Zou, Tianyi Zhang, Likun Wang, Letian Tao, Feihong Zhang, Yao Lyu, and Shengbo Eben Li. Real-time generative policy via langevin-guided flow matching for autonomous driving. *arXiv preprint arXiv:2603.02613*, 2026.

A Theoretical Proofs and Derivations

A.1 Surrogate Preference Score

Since BT modeling depends only on score differences, it suffices to justify the score g_θ at the pairwise level rather than at the level of exact action likelihoods. To make the approximation explicit, we first spell out the variational-gap term in Eq. (13) of the main text using standard diffusion notation. Here, $a^{1:T_{\text{diff}}} := (a^1, \dots, a^{T_{\text{diff}}})$ denotes the sequence of noisy actions, q denotes the forward diffusion distribution, and p_θ denotes the learned reverse-process distribution parameterized by θ.

Let the diffusion forward process be denoted by

$$q(a^{1:T_{\text{diff}}} \mid a, s) = \prod_{\tau=1}^{T_{\text{diff}}} q(a^\tau \mid a^{\tau-1}, s), \qquad a^0 \equiv a, \tag{23}$$

and the learned reverse process by

$$p_\theta(a^{1:T_{\text{diff}}} \mid a, s) = \frac{p_\theta(a, a^{1:T_{\text{diff}}} \mid s)}{\pi_\theta(a \mid s)}. \tag{24}$$

Then the variational gap in Eq. (13) can be written explicitly as

$$\Delta^\theta_{\text{KL}}(s, a) \triangleq D_{\text{KL}}\Big(q(a^{1:T_{\text{diff}}} \mid a, s) \,\|\, p_\theta(a^{1:T_{\text{diff}}} \mid a, s)\Big) \geq 0. \tag{25}$$

As discussed in the diffusion literature, the standard denoising objective is ELBO-aligned up to constants and weighting conventions [14, 17, 18]; see also Eqs. (14) and (15) in the main text. In particular, [26] adopts the same ELBO-surrogate viewpoint for diffusion policy optimization, replacing exact likelihood-based optimization with an ELBO-aligned surrogate objective. Therefore, for any ordered action pair (a^+, a^-), Eq. (13) implies

$$\log \pi_\theta(a^+ \mid s) - \log \pi_\theta(a^- \mid s) = \big[\text{ELBO}_\theta(s, a^+) - \text{ELBO}_\theta(s, a^-)\big] + \big[\Delta^\theta_{\text{KL}}(s, a^+) - \Delta^\theta_{\text{KL}}(s, a^-)\big]. \tag{26}$$

In the same spirit as [26], we use the diffusion score surrogate $g_\theta(s, a) \triangleq -E_\theta(s, a)$ and assume that the pairwise residual is bounded,

$$\big|\Delta^\theta_{\text{KL}}(s, a^+) - \Delta^\theta_{\text{KL}}(s, a^-)\big| \leq \varepsilon. \tag{27}$$

Since E_θ equals the negative ELBO up to an action-independent constant, Eq. (26) gives

$$\Big|\big[g_\theta(s, a^+) - g_\theta(s, a^-)\big] - \big[\log \pi_\theta(a^+ \mid s) - \log \pi_\theta(a^- \mid s)\big]\Big| \leq \varepsilon. \tag{28}$$

Because $-\log \sigma(\cdot)$ is 1-Lipschitz, the surrogate BT loss in Eq. (15) differs from the likelihood-based loss in Eq. (12) by at most ε for every pair. Moreover, whenever $|g_\theta(s, a^+) - g_\theta(s, a^-)| > \varepsilon$, the surrogate and the exact likelihood prefer the same action.

Hence, for any ordered action pair induced by the empirical joint distribution $\mu_m(\cdot, \cdot \mid s; A_{1:K})$ in the main text, or by its idealized continuous counterpart in the appendix, g_θ serves as a pairwise surrogate preference score whose BT loss stays within ε of the likelihood-based one.

A.2 Advantage-Weighted Surrogate Gradient Interpretation of the CBT Loss under an Idealized Construction

We consider an idealized construction aligned with Remark 2 in the main text. For a fixed state s, define the idealized ordered-pair sampling distribution

$$\tilde{\mu}_Q(a^+, a^- \mid s) \propto \pi(a^+ \mid s)\, \pi(a^- \mid s)\, \mathbb{I}_{Q(s,a^+) > Q(s,a^-)}, \tag{29}$$

which retains only ordered action pairs satisfying $Q(s, a^+) > Q(s, a^-)$. This construction can be viewed as an idealized continuous counterpart of the empirical joint distribution $\mu_m(\cdot, \cdot \mid s; A_{1:K})$ used in the main text, rather than an exact term-by-term reformulation of the finite candidate-pool implementation. Its purpose is to reveal the advantage-weighted structure induced by the CBT update. Throughout this appendix, $\pi(\cdot|s)$ denotes the sampling distribution induced by the current actor.

When differentiating with respect to θ, the pair-sampling mechanism is treated as fixed. To keep the derivation exact, we first present the idealized likelihood-based form, where gradients propagate through the conditional log-likelihood $\log \pi_\theta(a \mid s)$. For diffusion policies, this likelihood score is replaced in practice by the surrogate score g_θ, whose pairwise differences match those of $\log \pi_\theta$ up to ε, as shown in Appendix A.1.

Theorem 1 (Surrogate Gradient Form under the Idealized Pair Distribution)**.** *For a fixed state s, let ordered action pairs (a^+, a^-) be sampled according to the idealized construction above. Define the state-conditional idealized CBT loss as*

$$\ell_{\mathrm{CBT}}^{\mathrm{ideal}}(s) = \iint_{Q(s,a^+)>Q(s,a^-)} \pi(a^+ \mid s)\,\pi(a^- \mid s)\left[-\log\sigma\left(\log\pi_\theta(a^+ \mid s) - \log\pi_\theta(a^- \mid s)\right)\right] \cdot w(a^+, a^-)\,da^+ da^-, \tag{30}$$

where

$$w(a^+, a^-) = \sigma\left(Q(s,a^+) - Q(s,a^-)\right). \tag{31}$$

Equivalently,

$$\ell_{\mathrm{CBT}}^{\mathrm{ideal}}(s) = \iint_{Q(a)>Q(b)} \pi(a \mid s)\pi(b \mid s)\left[-\log\sigma\big(\log\pi_\theta(a) - \log\pi_\theta(b)\big)\right] w(a,b)\,da\,db, \tag{32}$$

Then, evaluated at $\theta = \theta_{\mathrm{old}}$, the negative gradient of $\ell_{\mathrm{CBT}}^{\mathrm{ideal}}(s)$ can be written as

$$-\nabla_\theta \ell_{\mathrm{CBT}}^{\mathrm{ideal}}(s)\big|_{\theta_{\mathrm{old}}} = \mathbb{E}_{a\sim\pi(\cdot|s)}\left[\Psi(s,a)\nabla_\theta \log\pi_\theta(a \mid s)\right], \tag{33}$$

where the induced multiplier $\Psi(s,a)$ is defined by

$$\Psi(s,a) = \mathbb{E}_{b\sim\pi(\cdot|s)}\left[C(a,b)\mathbb{I}_{Q(a)>Q(b)} - C(b,a)\mathbb{I}_{Q(b)>Q(a)}\right], \tag{34}$$

with

$$C(x,y) = \sigma\left(-\left(g_{\mathrm{old}}(x) - g_{\mathrm{old}}(y)\right)\right) w(x,y) > 0, \qquad g_{\mathrm{old}}(a) \triangleq \log\pi_{\theta_{\mathrm{old}}}(a \mid s). \tag{35}$$

Therefore, the induced update admits a generalized advantage-weighted surrogate gradient interpretation: actions assigned higher critic values tend to receive larger positive weights, whereas lower-valued actions tend to receive negative weights. This aligns the update direction with the policy-improvement intuition of increasing the scores of better actions while suppressing those of worse actions.

Proof. For a fixed state s, let $\pi(a) \triangleq \pi(a \mid s)$, and suppress the explicit dependence on s in $\pi(a)$, $\log\pi_\theta(s,a)$, and $Q(s,a)$. Then

$$\ell_{\mathrm{CBT}}^{\mathrm{ideal}}(s) = \iint_{Q(a)>Q(b)} \pi(a)\pi(b)\left[-\log\sigma\big(\log\pi_\theta(a) - \log\pi_\theta(b)\big)\right] w(a,b)\,da\,db. \tag{36}$$

We differentiate with respect to θ and evaluate the result at $\theta = \theta_{\mathrm{old}}$. Using the identity

$$\nabla_x\big[-\log\sigma(x)\big] = -\sigma(-x), \tag{37}$$

we obtain

$$-\nabla_\theta \ell_{\mathrm{CBT}}^{\mathrm{ideal}}(s) = \int\int_{Q(a)>Q(b)} \pi(a)\pi(b)\sigma\left(-\left(g_{\mathrm{old}}(a) - g_{\mathrm{old}}(b)\right)\right) w(a,b)\nabla_\theta\left(\log\pi_\theta(a|s) - \log\pi_\theta(b|s)\right) da\,db, \tag{38}$$

where, for notational simplicity in this idealized likelihood-based derivation,

$$g_{\mathrm{old}}(a) \triangleq \log\pi_{\theta_{\mathrm{old}}}(a \mid s). \tag{39}$$

Define

$$C(a,b) = \sigma\big(-\left(g_{\mathrm{old}}(a) - g_{\mathrm{old}}(b)\right)\big)\, w(a,b) > 0. \tag{40}$$

Then Eq. (38) becomes

$$-\nabla_\theta \ell_{\mathrm{CBT}}^{\mathrm{ideal}}(s) = \iint_{Q(a)>Q(b)} \pi(a)\pi(b)\, C(a,b)\,\nabla_\theta\big(\log\pi_\theta(a) - \log\pi_\theta(b)\big)\, da\,db. \tag{41}$$

By linearity of differentiation, we can write

$$-\nabla_\theta \ell_{\mathrm{CBT}}^{\mathrm{ideal}}(s) = \iint_{Q(a)>Q(b)} \pi(a)\pi(b)\, C(a,b)\, \nabla_\theta \log \pi_\theta(a)\, da\, db - \iint_{Q(a)>Q(b)} \pi(a)\pi(b)\, C(a,b)\, \nabla_\theta \log \pi_\theta(b)\, da\, db. \tag{42}$$

For the second term, exchange the dummy variables a and b. The integration region $Q(a) > Q(b)$ then becomes $Q(b) > Q(a)$, yielding

$$\iint_{Q(a)>Q(b)} \pi(a)\pi(b)\, C(a,b)\, \nabla_\theta \log \pi_\theta(b)\, da\, db = \iint_{Q(b)>Q(a)} \pi(a)\pi(b)\, C(b,a)\, \nabla_\theta \log \pi_\theta(a)\, da\, db. \tag{43}$$

Substituting Eq. (43) into Eq. (42), we obtain

$$-\nabla_\theta \ell_{\mathrm{CBT}}^{\mathrm{ideal}}(s) = \iint_{Q(a)>Q(b)} \pi(a)\pi(b)\, C(a,b)\, \nabla_\theta \log \pi_\theta(a)\, da\, db - \iint_{Q(b)>Q(a)} \pi(a)\pi(b)\, C(b,a)\, \nabla_\theta \log \pi_\theta(a)\, da\, db. \tag{44}$$

Factoring out the common term $\nabla_\theta \log \pi_\theta(a)$ gives

$$-\nabla_\theta \ell_{\mathrm{CBT}}^{\mathrm{ideal}}(s) = \int_{\mathcal{A}} \pi(a) \nabla_\theta \log \pi_\theta(a) \left[\int_{Q(a)>Q(b)} \pi(b) C(a,b)\, db - \int_{Q(b)>Q(a)} \pi(b) C(b,a)\, db \right] da. \tag{45}$$

We therefore define

$$\Psi(s,a) = \mathbb{E}_{b\sim\pi(\cdot|s)} \left[C(a,b)\mathbb{I}_{Q(a)>Q(b)} - C(b,a)\mathbb{I}_{Q(b)>Q(a)} \right]. \tag{46}$$

Using this definition, Eq. (45) can be rewritten as

$$-\nabla_\theta \ell_{\mathrm{CBT}}^{\mathrm{ideal}}(s)\big|_{\theta_{\mathrm{old}}} = \mathbb{E}_{a\sim\pi(\cdot|s)} \left[\Psi(s,a) \nabla_\theta \log \pi_\theta(a \mid s) \right]. \tag{47}$$

Finally, taking expectation over states sampled from the replay buffer yields

$$-\nabla_\theta \mathcal{L}_{\mathrm{CBT}}(\theta)\big|_{\theta_{\mathrm{old}}} = \mathbb{E}_s \left[\mathbb{E}_{a\sim\pi(\cdot|s)} \left[\Psi(s,a) \nabla_\theta \log \pi_\theta(a \mid s) \right] \right]. \tag{48}$$

This shows that, under the idealized construction, the update induced by the CBT loss can be expressed as a $\Psi(s,a)$-weighted surrogate gradient, which admits a generalized advantage-weighted interpretation. □

A.3 Interpretation of the Induced Multiplier $\Psi(s,a)$ and Its Partial Monotonicity with Respect to the Critic Value

Under the idealized pair-sampling construction of Theorem 1, we further analyze the induced multiplier $\Psi(s,a)$ and study its partial monotonicity with respect to the critic value $Q(s,a)$. The proof is carried out directly in the continuous action space $\mathcal{A}$, which avoids technical ambiguities that may arise from mapping actions to a one-dimensional value space.

Assumption 1 (Regularity of the Critic Landscape)**.** *We assume that, for a fixed state s, the critic score function $b \mapsto Q(s,b)$ is continuously differentiable (C^1) with respect to the continuous action variable b. Moreover, on the relevant level sets of $Q(s,\cdot)$, the gradient is non-degenerate almost everywhere, i.e.,*

$$\|\nabla_b Q(s,b)\| > 0 \quad \textit{for almost every } b \textit{ on the relevant level sets.} \tag{49}$$

This assumption allows critical points on measure-zero sets (e.g., isolated extrema); its role is mainly to exclude flat level sets that would otherwise make the induced surface integral degenerate.

Assumption 2 (Non-Degenerate Policy Support)**.** *We assume that the sampling distribution $\pi(\cdot \mid s)$ does not collapse to a Dirac delta distribution. Specifically, for any target action a, the set*

$$\{b \in \mathcal{A} : Q(s,b) \neq Q(s,a),\ \pi(b \mid s) > 0\} \tag{50}$$

has positive measure.

Proposition 1 (Strict Partial Monotonicity of Ψ with Respect to the Critic Value)**.** *Under Assumption 1 and Assumption 2, and under the idealized pair-sampling construction of Theorem 1, fix a state s and a target action a, and define*

$$\begin{aligned} Q_a &\triangleq Q(s,a), & g_a &\triangleq \log \pi(a \mid s), \\ Q_b &\triangleq Q(s,b), & g_b &\triangleq \log \pi(b \mid s). \end{aligned} \tag{51}$$

Treating Q_a and g_a as locally independent scalar variables, the induced multiplier $\Psi(s,a)$ is strictly partially monotone with respect to Q_a, i.e.,

$$\frac{\partial \Psi(s,a)}{\partial Q_a} > 0. \tag{52}$$

Proof. By Theorem 1, under the idealized pair-sampling construction, the induced multiplier for action a can be written as

$$\Psi(s,a) = \int_{\mathcal{A}} \pi(b \mid s) \Big[C(a,b)\mathbb{I}_{Q_a > Q_b} - C(b,a)\mathbb{I}_{Q_b > Q_a} \Big]\, db, \tag{53}$$

where the pairwise soft-gating term is defined as

$$C(x,y) = \sigma\big(-(g_x - g_y)\big)\, \sigma(Q_x - Q_y). \tag{54}$$

To analyze the local response of $\Psi(s,a)$ to the critic value Q_a, we differentiate Eq. (53) with respect to Q_a while holding g_a fixed. Formally, using the distributional derivative of the indicator function

$$\frac{\partial}{\partial x}\mathbb{I}_{x>y} = \delta(x-y), \tag{55}$$

the derivative of $\Psi(s,a)$ can be decomposed into two parts: a boundary term and an interior term.

Boundary term. Differentiating the indicator functions produces a Dirac delta term $\delta(Q_a - Q_b)$. By the coarea formula, the corresponding volume integral can be rewritten as a surface integral over the level set $\{\, b : Q_b = Q_a \,\}$:

$$\int_{\mathcal{A}} \pi(b \mid s) \Big[C(a,b) + C(b,a) \Big] \delta(Q_a - Q_b)\, db = \int_{\{b:Q_b=Q_a\}} \frac{\pi(b \mid s)\big(C(a,b) + C(b,a)\big)}{\|\nabla_b Q(s,b)\|}\, dS(b). \tag{56}$$

Since $C(\cdot,\cdot) > 0$, $\pi(b \mid s) \geq 0$, and Assumption 1 guarantees that $\|\nabla_b Q(s,b)\| > 0$ almost everywhere on the relevant level set, the surface integral in Eq. (56) is nonnegative.

Interior term. We next differentiate the continuous part $C(\cdot,\cdot)$ with respect to Q_a over the corresponding active regions:

$$\int_{\mathcal{A}} \pi(b \mid s) \left[\frac{\partial C(a,b)}{\partial Q_a}\mathbb{I}_{Q_a > Q_b} - \frac{\partial C(b,a)}{\partial Q_a}\mathbb{I}_{Q_b > Q_a} \right] db. \tag{57}$$

For the first term, on the region $Q_a > Q_b$,

$$\frac{\partial C(a,b)}{\partial Q_a} = \sigma\big(-(g_a - g_b)\big)\, \sigma'(Q_a - Q_b) > 0. \tag{58}$$

For the second term, on the region $Q_b > Q_a$, applying the chain rule to $\sigma(Q_b - Q_a)$ yields

$$\frac{\partial C(b,a)}{\partial Q_a} = \sigma\big(-(g_b - g_a)\big)\, \sigma'(Q_b - Q_a) \cdot (-1) < 0. \tag{59}$$

Hence, because this second contribution appears with a minus sign in Eq. (57), it is also positive:

$$-\frac{\partial C(b,a)}{\partial Q_a} > 0. \tag{60}$$

□

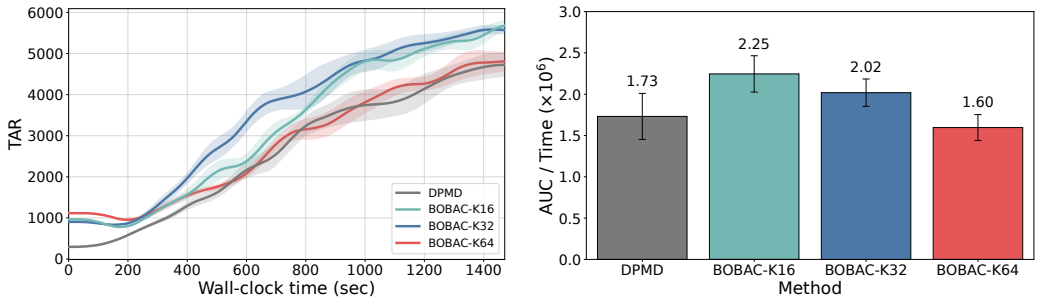
Remark 3. *Combining the two parts, the boundary contribution is nonnegative, while the interior contribution is strictly positive on its active regions. Moreover, by Assumption 2, the set of b satisfying $Q_b \neq Q_a$ and $\pi(b \mid s) > 0$ has positive measure, so the positive interior contribution cannot vanish identically. Therefore,*

$$\frac{\partial \Psi(s,a)}{\partial Q_a} > 0. \tag{61}$$

Thus, when g_a is held locally fixed, $\Psi(s,a)$ is monotone increasing in Q_a. In other words, assigning a higher critic value to action a leads to a larger induced gradient weight, so that $\Psi(s,a)$ plays a role analogous to a generalized advantage signal.

B Additional Compute-Fairness Analysis

To examine whether the performance gain of BOBAC mainly comes from additional optimization-side computation, we provide a compute-fairness analysis on `Ant-v4`. All methods are compared under the same environment interaction budget, while wall-clock time is measured explicitly to account for the extra computational cost introduced by candidate comparison. We compare DPMD with BOBAC using different candidate set sizes $K \in \{16, 32, 64\}$.



(a) Performance measured against wall-clock time. (b) Time-normalized Area Under the Curve (AUC).

Figure 5: Additional compute-fairness results on `Ant-v4`. **Left:** TAR versus wall-clock time under the same interaction budget. **Right:** Time-normalized AUC of the return trajectory. Shaded regions and error bars represent standard error and standard deviation across random seeds, respectively.

Wall-clock time comparison. Figure 5a compares methods using wall-clock time instead of environment steps. Under this metric, both BOBAC-K16 and BOBAC-K32 remain clearly stronger than DPMD throughout most of the training process, indicating that the advantage of BOBAC is not merely an artifact of step-based evaluation. Among the tested variants, BOBAC-K16 exhibits the most favorable early-to-mid training efficiency, while BOBAC-K32 remains competitive and attains slightly higher final return under the standard fixed-interaction protocol. In contrast, BOBAC-K64 progresses more slowly under wall-clock accounting, suggesting that excessively large candidate sets introduce non-negligible optimization overhead.

Efficiency summary via AUC per unit time. Figure 5b summarizes training efficiency using the **Area Under the Curve (AUC)** of the return trajectory normalized by wall-clock time. BOBAC-K16 achieves the highest AUC per wall-clock second, followed by BOBAC-K32, while BOBAC-K64 falls below DPMD on this metric. This reveals a clear compute–performance trade-off as K increases: moderate candidate sizes already capture most of the performance benefit, whereas overly large candidate sets substantially reduce time efficiency.

Quantitative trade-off. Table 2 quantifies the same trend. All BOBAC variants outperform DPMD in final return under the same interaction budget, confirming that the performance gain is not specific to wall-clock visualization. More importantly, the improvement remains meaningful after accounting for training time. BOBAC-K16 improves the final return by 21.0% with only 8.2% additional wall-clock time, and also achieves the highest AUC/time among all compared methods. BOBAC-K32 attains the best final return, improving over DPMD by 21.6%, but incurs a larger time overhead of 17.5%. In contrast, BOBAC-K64 yields a smaller return gain (16.3%) while introducing

Table 2: Compute-performance trade-off on `Ant-v4`. Results are aggregated across seeds and reported as mean ± standard deviation. Gain and overhead are computed relative to the mean of DPMD.

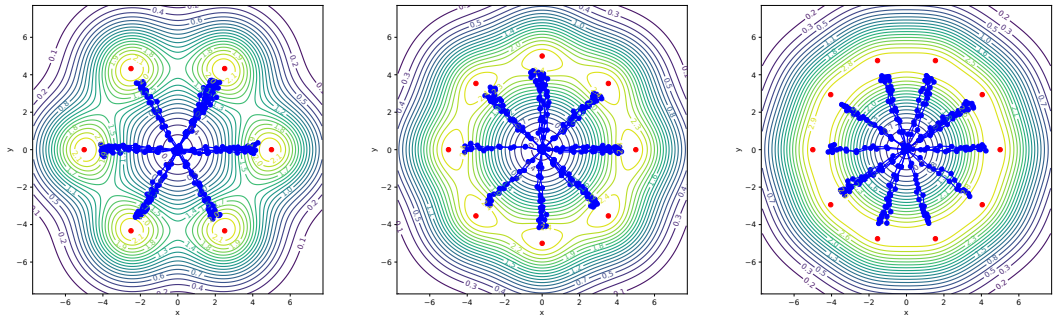
Method	K	TAR ↑	Gain over DPMD (%) ↑	AUC ($\times 10^9$) ↑
DPMD	–	4,715.5 ± 704.9	–	2.56 ± 0.42
BOBAC-K16	16	5,707.2 ± 358.7	21.0	3.59 ± 0.36
BOBAC-K32	32	5,732.3 ± 353.8	21.6	3.51 ± 0.35
BOBAC-K64	64	5,482.2 ± 576.1	16.3	3.29 ± 0.34

Method	K	Total Time (s) ↓	Time Overhead (%) ↓	AUC / Time ($\times 10^6$) ↑
DPMD	–	1,477.9 ± 6.8	–	1.73 ± 0.28
BOBAC-K16	16	1,599.7 ± 7.4	8.2	2.25 ± 0.22
BOBAC-K32	32	1,736.8 ± 39.5	17.5	2.02 ± 0.17
BOBAC-K64	64	2,062.5 ± 31.4	39.6	1.60 ± 0.16

a substantially larger time overhead (39.6%), resulting in the weakest efficiency among the BOBAC variants.

C Multimodal Experiments

We evaluate the trained policy in the `Multi-goal` environment by initializing the agent at the origin and sampling 100 trajectories. We conduct experiments across configurations ranging from 6 to 10 symmetrically arranged goal points, with results reported for 6, 8, and 10 goals. As illustrated in Fig. 6, BOBAC consistently reaches all target locations with approximately uniform coverage across all configurations. These experimental results underscore the superior exploratory capability of BOBAC, enabling it to more effectively capture diverse, mode-separated policies in multimodal environments.



(a) 6 goals (b) 8 goals (c) 10 goals

Figure 6: **Multi-goal Task.** Trajectories generated by policies learned using our proposed BOBAC are shown, with the x-axis and y-axis representing 2D positions (states). The agent is initialized at the origin, and the goals are marked as red dots. The level curves indicate the reward, and reaching within 1 of the endpoint signifies task completion. Results are shown for 6, 8, and 10 goal configurations.

D Detailed Hyperparameters

This appendix summarizes the hyperparameters used in our experiments. We first report key settings of generative-policy baselines, followed by the default configuration of BOBAC. Each baseline follows the configuration of its own paper or released code, while the number of transitions collected per iteration (5 for off-policy methods) and the total interaction budget are unified across all methods. BOBAC uses the learning-rate setting of DPMD, whose diffusion-policy backbone it builds on. DSACT uses its published learning rate under this unified protocol, and with its native schedule

of 20 transitions per iteration under the same 5M-interaction budget, it reaches 3997.6 ± 126.3 on `Humanoid-v4`.

Table 3: Key hyperparameters of generative-policy baselines. Shared settings are described in the text, and remaining method-specific settings follow official or released default configurations.

Parameter	**DACER**	**DPMD**	**QVPO**	**DIPO**	**QSM**	**FlowRL**
Actor architecture	3×256	3×256	3×256	3×256	3×256	3×256
Critic architecture	3×256	3×256	3×256	3×256	3×256	3×256
Actor learning rate	1×10^{-4}	linear: $3\times10^{-4} \rightarrow 3\times10^{-5}$	1×10^{-4}	1×10^{-4}	1×10^{-4}	3×10^{-4}
Critic learning rate	1×10^{-4}	3×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}	3×10^{-4}
Diffusion / flow sampling steps	20	20	20	20	20	1
Action gradient steps	–	–	–	20	–	–

Table 4 reports the default configuration of BOBAC, including training settings, diffusion-actor parameters, and network architectures. Unless otherwise specified, the same BOBAC configuration is used across all MuJoCo and H1Bench tasks. For the Q-gap-aware soft gate, we normalize the gate over the m^2 positive–negative pairs for each state before computing the weighted BT loss.

Table 4: Hyperparameter settings of BOBAC.

Hyperparameter	**Value**	**Hyperparameter**	**Value**
Training			
Critic learning rate	3×10^{-4}	Actor learning rate	linear: $3 \times 10^{-4} \rightarrow 3 \times 10^{-5}$
Optimizer	RAD [24]	Discount factor γ	0.99
Target smoothing τ	5×10^{-3}	Batch size	256
Replay buffer capacity	1×10^{6}	Buffer warm-up size	3×10^{4} steps
Reward scale	0.2	Policy update interval	2 steps
Actor			
Diffusion time steps T_{diff}	20	Candidate actions per state K	32
Diffusion noise schedule	Cosine	Top/bottom subset size m	4
Action bound	$[-1, 1]$		
Architecture			
Actor hidden layers	3	Critic hidden layers	3
Actor hidden width	256	Critic hidden width	256
Actor/Critic activation	Mish	Time embedding (diffusion)	sinusoidal, $\dim = 16$

E Additional Results

E.1 Comparison with SDAC

DPMD and SDAC are introduced in the same work [25], and BOBAC builds on the same diffusion-policy backbone as DPMD. Table 5 compares the three methods under the protocol of Table 1 over 5 seeds.

Table 5: TAR of DPMD, SDAC, and BOBAC under the protocol of Table 1. Results are reported as Mean ± Std over 5 seeds. **Bold** denotes the best result.

Task	DPMD	SDAC	BOBAC (ours)
`Ant-v4`	5129.4 ± 499.1	962.6 ± 136.4	**5745.7 ± 225.7**
`HalfCheetah-v4`	10775.3 ± 558.7	11960.8 ± 263.0	**12128.9 ± 524.3**
`Humanoid-v4`	5339.7 ± 1242.9	4426.2 ± 641.8	**6905.3 ± 358.7**
`HumanoidStandup-v4`	141867.9 ± 7101.3	155244.2 ± 926.9	**163074.9 ± 6743.7**
`Swimmer-v4`	56.0 ± 8.1	74.6 ± 11.6	**126.9 ± 15.2**
`Walker2d-v4`	3369.7 ± 374.7	2657.0 ± 742.6	**4857.8 ± 552.6**
`H1-reach`	6757.3 ± 452.0	3594.4 ± 396.4	**7800.1 ± 215.7**
`H1-maze`	236.3 ± 55.2	131.7 ± 9.0	**311.8 ± 13.9**
`H1hand-reach`	5418.5 ± 417.4	2186.2 ± 552.1	**6894.8 ± 576.3**
`H1hand-crawl`	447.3 ± 33.6	759.3 ± 20.3	**822.2 ± 12.3**

BOBAC achieves the highest mean TAR among the three methods on all ten tasks. Since BOBAC and DPMD share the diffusion-policy backbone, these gains are not attributable to the backbone alone.

E.2 Component Ablations on `H1hand-reach`

We repeat the component ablations of Section 5 on `H1hand-reach`, whose action space has 61 dimensions, over 5 seeds.

Table 6: Component ablations on `H1hand-reach`. Results are reported as Mean ± Std over 5 seeds.

Variant	TAR	Change
BOBAC (full)	**6894.8 ± 576.3**	–
W/o gate	6030.0 ± 148.9	-12.5%
Pos-only	5373.3 ± 63.2	-22.1%
Random neg	4948.3 ± 34.3	-28.2%

The ordering matches that on `Ant-v4`. Replacing critic-guided negatives with random ones causes the largest drop, and keeping only positives also falls well short of the full method, so the gain comes from contrasting actions selected by the critic. Removing the Q-gap-aware gate lowers the return by 12.5%.

E.3 Sparse Rewards and Stochastic Action Execution

Pair construction and gating depend only on Q-gaps between actions at the same state. An additive critic offset $c(s)$ shared by all candidates of a state therefore leaves the ranking, the soft gate, and the CBT loss unchanged. To test BOBAC when critic learning is harder, we build a variant of `H1-maze` that removes all per-step shaping terms and keeps only checkpoint-completion rewards. We also add Gaussian noise with standard deviation 0.1 to each normalized action dimension before execution, during both training and evaluation. Early nAUC is the normalized area under the return curve over the first 30% of training. All results use 5 seeds.

Table 7: `H1-maze` with sparse rewards and stochastic action execution. Results are reported as Mean $\pm$ Std over 5 seeds.

Method	Final TAR	Early nAUC (first 30%)
DPMD	146.2 ± 54.2	35.0 ± 26.2
BOBAC, w/o gate	134.8 ± 52.8	31.4 ± 20.8
BOBAC (ours)	$\mathbf{204.7 \pm 33.8}$	$\mathbf{72.1 \pm 20.3}$

BOBAC achieves the best final return and the fastest early learning, with an Early nAUC 2.1 times that of DPMD. Under sparse rewards and stochastic action execution, removing the gate causes a 34.1% drop in final TAR, compared with a 15.0% drop on `Ant-v4`.

F Limitations

BOBAC requires sampling and ranking multiple candidate actions per state, which leads to some additional optimization cost compared with simpler pointwise actor updates. In the current implementation, using a moderately large candidate size K is helpful for exposing actions with clearer value contrast and for forming informative pairwise comparisons. Further improving the efficiency of candidate construction, so that similar gains can be achieved with a smaller K, is a meaningful direction for future work.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The main claims in the abstract and introduction accurately reflect the scope and contributions of the paper. The paper clearly states that BOBAC introduces BT-style pairwise preference learning for diffusion policy optimization, a Q-gap-aware soft-gating mechanism, and empirical validation on a 10-task benchmark consisting of 6 MuJoCo tasks and 4 H1Bench tasks.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: The paper includes a dedicated limitations discussion in Appendix F, where we discuss the additional optimization cost introduced by sampling and ranking multiple candidate actions per state and identify more efficient candidate construction as a future direction.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes].

   Justification: The detailed assumptions (Assumptions 1, 2), complete proofs for Theorem 1 and Proposition 1, and related derivations are provided in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: We describe the evaluation benchmarks, baseline setups, evaluation protocol, and implementation details in Section 5, while the algorithmic procedure is provided in Algorithm 1 and detailed hyperparameter settings are documented in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: We provide an anonymized implementation of the core algorithm in a ZIP file submitted as supplementary material. All reported experiments are based on publicly available benchmark environments, including OpenAI Gym MuJoCo and H1Bench, and no private datasets are used.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: All relevant training details, including hyperparameters and optimizers, are reported in Section 5 and Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

   Justification: The main results are reported over 5 random seeds with mean and standard deviation in the tables. The training curves report mean performance with shaded uncertainty regions, and the ablation and sensitivity studies also report variability across runs.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: The paper reports the compute resources used for experiments in Section 5, including an AMD Ryzen Threadripper 3960X 24-core CPU and an NVIDIA GeForce RTX 4090 GPU. It also specifies the training interaction budgets for different environments, and reports the per-run wall-clock time and compute-performance trade-off in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The research conforms to the NeurIPS Code of Ethics. The work is conducted on standard simulated control benchmarks, does not involve human subjects or private data, and the submission preserves anonymity.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A].

    Justification: This work is a foundational RL algorithm study evaluated only on simulated continuous-control benchmarks. It does not involve human subjects, private data, real-world deployment, or application-specific decision making, and therefore has no direct path to immediate societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: The paper does not release high-risk assets such as pretrained language models, image generators, scraped datasets, or other resources with a high risk of misuse. Therefore, no special release safeguards are required.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes].

Justification: The paper uses standard benchmark environments and existing baseline implementations, and properly credits the original sources. The corresponding licenses and terms of use are documented and respected in the supplementary material.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes].

Justification: We introduce the BOBAC algorithm codebase, which is well documented and provided as an anonymized supplementary ZIP file. The codebase will be released openly upon publication.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A].

Justification: The paper does not involve crowdsourcing, user studies, human annotation, or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The paper does not involve human subjects, so IRB approval or an equivalent review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: LLMs were used only for language polishing, grammar correction, and formatting assistance during paper writing. They were not used as an important, original, or non-standard component of the core methodology, experimental design, implementation, data analysis, or scientific conclusions.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.