Adaptive Advantage-Guided Policy Regularization
for Offline Reinforcement Learning

Tenglong Liu [1] Yang Li [2] Yixing Lan [1] Hao Gao [1] Wei Pan [3] Xin Xu [1]

Abstract

In offline reinforcement learning, the challenge of out-of-distribution (OOD) is pronounced. To address this, existing methods often constrain the learned policy through policy regularization. However, these methods often suffer from the issue of unnecessary conservativeness, hampering policy improvement. This occurs due to the indiscriminate use of all actions from the behavior policy that generates the offline dataset as constraints. The problem becomes particularly noticeable when the quality of the dataset is suboptimal. Thus, we propose Adaptive Advantage-Guided Policy Regularization (A2PR), obtaining high-advantage actions from an augmented behavior policy combined with VAE to guide the learned policy. A2PR can select high-advantage actions that differ from those present in the dataset, while still effectively maintaining conservatism from OOD actions. This is achieved by harnessing the VAE capacity to generate samples matching the distribution of the data points. We theoretically prove that the improvement of the behavior policy is guaranteed. Besides, it effectively mitigates value overestimation with a bounded performance gap. Empirically, we conduct a series of experiments on the D4RL benchmark, where A2PR demonstrates state-of-the-art performance. Furthermore, experimental results on additional suboptimal mixed datasets reveal that A2PR exhibits superior performance. Code is available at https://github.com/ltlhuuu/A2PR.

1. Introduction

Reinforcement learning has made substantial breakthrough advancements over the past decades, such as chess games (Silver et al., 2016; Schrittwieser et al., 2020; Li et al., 2023), video games (Perolat et al., 2022; Vinyals et al., 2019), robotics (Hwangbo et al., 2019; Andrychowicz et al., 2020; Rajeswaran et al., 2017) and so on. Specifically, RL employs a trial-and-error approach, iteratively refining its performance through interactions with the environment in an online fashion (Sutton & Barto, 2018). However, the trial-and-error paradigm poses significant challenges to the seamless integration of RL into real-world applications such as autonomous driving, healthcare, and other tasks. The impracticality of trial-and-error in scenarios where active interaction with the environment is unfeasible renders each training run unrealistic. In recent times, offline RL has garnered considerable attention for its potential to exclusively learn from pre-collected datasets, eliminating the need for real-time interaction during training (Levine et al., 2020).

In offline RL, a key challenge is addressing the overestimation of Q-values caused by out-of-distribution (OOD) actions. Commonly, techniques rely on incorporating the dataset's behavior policy to tackle this challenge, constraining the learned policy through policy regularization methods (Levine et al., 2020). These regularization methods introduce an extra term to calculate divergence metrics between the learned policy and the behavior policy, employing widely-used metrics such as behavior clone (Fujimoto & Gu, 2021; Ran et al., 2023), Kullback-Leibler (KL) divergence (Jaques et al., 2019; Wu et al., 2019), fisher divergence (Kostrikov et al., 2021a), and Maximum Mean Discrepancy (MMD) (Kumar et al., 2019). To some extent, these methods alleviate overestimation from OOD actions (Levine et al., 2020). However, existing policy regularization methods are unnecessarily conservative (Hong et al., 2023b) since they force the learned policy to closely mimic actions of the behavior policy, even if those actions are suboptimal. Such unnecessary conservatism hampers policy performance, especially in datasets dominated by low-return trajectories with sparse high-return instances.

To address the issue of unnecessary conservatism (Hong et al., 2023b), we introduce an Adaptive Advantage-

[1]National University of Defense Technology, Changsha, China [2]Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, China [3]Delft University of Technology, The Netherlands. Correspondence to: Xin Xu <Xinxu@nudt.edu.cn>, Yixing Lan <lanyixing16@nudt.edu.cn>.

Guided Policy Regularization (A2PR) method for offline RL. This approach combines policy regularization with high-advantage actions and efficiently guides the learned policy toward improvement. In A2PR, a Variational Autoencoder (VAE) enhanced by an advantage function generates high-advantage actions by combining them with those from the dataset for policy regularization. This process is similar to directing the learned policy with sensible behavior from an augmented behavior policy. A2PR has the flexibility to select high-advantage actions differing from those in the dataset due to the augmented behavior policy. Additionally, it inherently exhibits a natural level of conservatism since the VAE generates samples from the same distribution as the data points. In contrast to prior methods that force the learned policy to closely mimic all data, A2PR promotes policy improvement while being guided by superior actions from the augmented behavior policy. It eases the constraint when encountering poorer data, alleviating the pessimistic issues associated with overly conservative constraints. Consequently, A2PR establishes a more effective and adaptive policy constraint.

A2PR can integrate into existing actor-critic offline RL algorithms. In our study, we implement a practical algorithm based on TD3 (Fujimoto et al., 2018), chosen for its straightforward and efficient implementation that yields remarkable performance. Subsequently, we perform a theoretical analysis investigating the performance improvement of the behavior policy. Our findings illustrate a reduction in the overestimation problem, substantiated by quantifying a bounded performance gap concerning the learned policy. The experimental results show that our proposed method attains state-of-the-art performance on the D4RL standard benchmark (Fu et al., 2020) for offline RL. Furthermore, we assess the method's performance on supplementary low-quality datasets, comprised of 99% random policy datasets. Our approach exhibits noteworthy performance improvements, particularly evident in the additional suboptimal or low-quality datasets.

2. Preliminaries

This section provides a concise introduction to the background and introduces some key notation. Offline RL, also referred to as batch RL or data-driven RL (Levine et al., 2020), constitutes a specialized category within RL. It operates within the framework of Markov decision processes (MDPs) denoted as $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$ (Sutton & Barto, 2018), where $\mathcal{S}$ represents the state space, $\mathcal{A}$ denotes the action space, $P(\cdot|s,a)$ characterizes the transition probability distribution function, γ is the discount factor, and $r(s,a)$ corresponds to the reward function for (s,a). Throughout, we consider $\forall (s,a) \in \mathcal{S} \times \mathcal{A}, \gamma \in (0,1], |r(s,a)| \leq R_{max}$, and $a \in [-A, A]$. The objective is to identify a policy π^*

that maximizes the expected return, commencing from any state $s \in S$: $\pi^* = \arg\max_\pi \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right]$.

Offline RL involves learning policies from a predetermined dataset $\mathcal{D}\{(s, a, r, s')\}$ gathered in advance through an unknown behavior policy π_β. This approach enhances sample efficiency by leveraging pre-collected data without requiring extensive direct interaction with the environment. Offline RL holds notable significance, especially in situations where interaction entails risks or incurs high costs. Consequently, the difference between the learned policy and the behavior policy often gives rise to OOD actions, leading to extrapolation error.

The Q-function $Q(s,a)$ signifies the expected discounted return starting from any state $s \in S$. The advantage function of the action a is defined as: $A(s,a) = Q(s,a) - V(s)$, where $V(s)$ represents the value function. For each policy π, there exists a corresponding Q-function obtained through the Bellman operator $\mathcal{T}$, defined as:

$$Q^\pi(s,a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, a_0 = a \right],$$
$$(\mathcal{T}Q^\pi)(s,a) = r(s,a) + \gamma \mathbb{E}_{\substack{s' \sim P(\cdot|s,a), \\ a' \sim \pi^*(\cdot|s')}} [Q(s', a')]. \tag{1}$$

Subsequently, the expected discounted reward $J(\pi)$ of policy π can be expressed as $J(\pi) = \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\pi(s)}[r(s)]$. The discounted state distribution of a policy π, also known as the occupancy measure d_π: $\mathcal{S} \to \mathbb{R}$, is defined as $d_\pi(s) = \sum_{t=0}^{\infty} \gamma^t p(s_t = s|\pi)$, where $p(s_t = s|\pi)$ represents the probability of state s_t being s under policy π.

The objective values utilized in Bellman backups for policy evaluation originate from actions sampled from the learned policy π. However, the Q-function is exclusively trained using actions sampled from the behavior policy π_β, responsible for generating the dataset $\mathcal{D}$. As π is optimized to maximize Q-values, a potential bias towards OOD actions may exist, resulting in inaccurately overestimated Q-values. We define δ_{error} as the overestimation error (Fujimoto et al., 2019), representing the disparity between an approximate estimate $\tilde{Q}^\pi$ and the true Q-value function Q^π: $\delta_{error} = \tilde{Q}^\pi(s,a) - Q^\pi(s,a)$.

3. Related Work

3.1. Offline RL with policy regularization.

Policy regularization is pivotal research in offline RL, addressing distribution shift challenges to mitigate OOD actions. TD3+BC (Fujimoto & Gu, 2021) enhances policy improvement by integrating a straightforward behavior cloning term, providing a clear estimate of the learned policy. Numerous divergence penalties compel the learned policy to

stay close to the behavior policy, including Maximum Mean Discrepancy (MMD) (Kumar et al., 2019), Fisher divergence (Kostrikov et al., 2021a), KL divergence (Jaques et al., 2019; Wu et al., 2019; Nair et al., 2020), and Wasserstein distance (Wu et al., 2019). BEAR (Kumar et al., 2019) employs MMD with a Gaussian kernel as divergence regularization for policy improvement but relies heavily on the approximate nature of low-sampled MMD. Advantage-weighted (Peng et al., 2019) regression utilizes supervised regression as learning subroutines to enhance the learned policy while concurrently enforcing an implicit KL-divergence constraint. BCQ (Fujimoto et al., 2019) and LAPO (Chen et al., 2022) implement the modeling of the behavior policy using Conditional VAE (Sohn et al., 2015), achieving proximity between the learned and behavior policies through an implicit constraint. SPOT (Wu et al., 2022) explicitly pre-trains a VAE to model the support set of the behavior policy, directly using behavior density to constrain the learned policy. OAP (Yang et al., 2023) utilizes complex RankNet pseudo-queries to select actions with higher Q-values for policy constraints from the current policy and dataset. PRDC (Ran et al., 2023) employs KD-tree (Bentley, 1975) to index potential actions corresponding to the current state from the entire dataset. However, it's worth noting that the use of KD-tree introduces a high computational complexity. Unlike PRDC, our methods use a simple and effective VAE to generate more potential high-advantage actions for policy regularization.

3.2. Data reweighting

In offline RL, ReDs (Singh et al., 2023) focuses on reweighting the data distribution solely for CQL (Kumar et al., 2020), aiming to achieve an approximate support constraint formulation. RB-CQL (Jiang et al., 2023) specializes in the context of CQL by incorporating a retrieval process that recalls past related experiences. AW (Hong et al., 2023a) reweights trajectories based on their returns, a process that necessitates the consideration of entire trajectories. DW (Hong et al., 2023b) emulates sampling from an alternate dataset through importance sampling, where the weighting function can be interpreted as the density ratio between the alternative dataset and the original one. OPER (Yue et al., 2023) adopts a priority function to prioritize a dataset for the enhancement of the learned policy. In the realm of offline imitating learning (IL) (Kim et al., 2021; Ma et al., 2022; Xu et al., 2022), the focus is on training an expert policy from a dataset that comprises a limited set of expert data along with a substantial amount of random data. These methods aim to train a model to learn a policy that closely aligns with the expert data, necessitating the separation of expert data from random data. In contrast, our methods solely require the advantage of selecting actions without the need for separable data.

4. Method

In this section, we present the A2PR algorithm. We commence by introducing elevating positive behavior learning in Section 4.1, aiming to procure more high-advantage actions with the support of the dataset. Subsequently, in Section 4.2, we delve into the adaptive advantage policy constraint, designed to reinforce policy improvement by prioritizing actions with higher advantages. The practical implementation details of the algorithm are outlined in Section 4.3. Finally, in Section 4.4, we provide theoretical analyses elucidating the performance improvement guarantee over the behavior policy and a bounded performance gap, addressing the overestimation issue.

4.1. Elevating Positive Behavior Learning

In offline RL, the sampled probability of actions remains fixed due to the unchanging nature of the pre-collected dataset. Notably, existing offline RL methods (Hong et al., 2023a) demonstrate that reweighting datasets based on trajectory return or episode advantage effectively regulates the implicit behavior policy, resulting in enhanced performance. This approach serves to provide a more favorable starting performance for the learned policy. Motivated by these findings, it is intuitive to consider that augmenting the probability of high-advantage actions can contribute to the improvement of the implicit behavior policy. Furthermore, we delve into a theoretical analysis guaranteeing the enhancement of the behavior policy.

Proposition 4.1. *Suppose that* $A^{\pi_\beta}(s,a)(\hat{\pi}_\beta(a|s) - \pi_\beta(a|s)) \geq 0$. *Then, we have*

$$J(\hat{\pi}_\beta) - J(\pi_\beta) \geq 0, \tag{2}$$

where $\hat{\pi}_\beta$ *is another behavior policy,* π_β *is the original behavior policy of the dataset. The proof is deferred to Appendix A.1.*

Using a VAE as the density estimator for the dataset (Kumar et al., 2019; Wu et al., 2022) proves to be a straightforward and effective method for learning the behavior policy. In this regard, we propose a specific approach to optimize the VAE output action in conjunction with the advantage function. In the realm of offline RL, where $A(s,a)$ represents the additional reward achievable by taking action a rather than the expected return, it serves as a quality indicator for actions. This information can be effectively incorporated with the reconstruction component of the VAE. Motivated by this insight, we introduce elevating positive behavior learning (EPBL) utilizing a VAE enhanced by the advantage function. The EPBL can be optimized jointly with the following evidence lower bound (ELBO):

$$\log p_\psi(a|s) \geq \mathbb{E}_{q_\varphi(z|a,s)} \left[f(A(s,a) > \epsilon_A) \log p_\psi(a|z,s) \right] - \text{KL} \left[q_\varphi(z|a,s) \parallel p(z|s) \right], \tag{3}$$

where $f(A(s,a) > \epsilon_A)$ represents $w_1 * \mathbb{1}(A(s,a) > \epsilon_A)$, w_1 is a hyperparameter. $A(s,a)$ represents the advantage of action a, $\log p_\psi(a|z,s)$ signifies the likelihood measure of the reconstructed action from the decoder, and $\text{KL}\left[q_\varphi(z|a,s) \parallel p(z|s)\right]$ represents the KL-divergence between the encoder output and the prior of z. ϵ_A is an advantage threshold. This threshold serves to constrain the quality of the reconstructed action. Restricting the reconstruction process to actions with a higher advantage from the dataset is achieved by introducing a simple constraint $\mathbb{1}(A(s,a) > \epsilon_A)$ before the reconstruction loss. This constraint ensures the reconstruction of actions with a higher advantage. As a result, the enhanced behavior policy $\hat{\pi}_\beta$ assigns a greater density ratio to high-advantage actions compared to the behavior policy π_β.

4.2. Adaptive Advantage Policy Constraint

A2PR faces the challenge of reconciling two conflicting objectives (Yang et al., 2023): policy improvement and policy constraint. An overly strict policy constraint within a suboptimal dataset may impede the policy's enhancement beyond the behavior policy. Conversely, a lax constraint might lead to distributional shift, causing the learned policy to fail in OOD actions. Achieving a balance between these aspects is imperative. By incorporating a policy constraint that prioritizes actions with a higher advantage, alongside policy improvement considerations, A2PR enables the learned policy to assimilate knowledge from the augmented behavior policy. These dual approaches of policy improvement and policy constraint can be expressed generically through the following equation:

$$\max_\phi \mathbb{E}_{s \sim \mathcal{D}}[Q_\theta(s, \pi_\phi(s))] \implies \text{Policy improvement,}$$
$$\text{s.t.} \quad \|\pi_\phi(s) - a\| < \epsilon_0 \implies \text{Policy constraint,}$$
$$(4)$$

where Q_θ represents the state-action value function, $\pi_\phi(\cdot)$ denotes the learned policy, and $\|\cdot\|$ stands for a norm. Selecting high-advantage actions from the augmented behavior policy is pivotal for policy regularization:

$$\tilde{a} = \underset{\dot{a} \in \{a, \hat{\pi}_\beta(s)\}}{\arg\max} A(s, \dot{a}), \quad (5)$$

where $\tilde{a}$ represents the high-advantage action, chosen between a and the generative high-advantage actions $\hat{\pi}_\beta(s)$.

To be more specific, if the actions adhere to the condition $A(s, \tilde{a}) \geq \epsilon_A$, the policy will be constrained in proximity to $\tilde{a}$. Conversely, when the condition $A(s, \tilde{a}) \geq \epsilon_A$ is not satisfied, the current policy should self-learn, indicating the necessity for a robust constraint. The advantageous action $\bar{a}$ is dynamically determined by the following equation:

$$\bar{a} = \begin{cases} \tilde{a}, & A(s, \tilde{a}) \geq \epsilon_A \\ \pi_\phi(s), & A(s, \tilde{a}) < \epsilon_A, \end{cases} \quad (6)$$

Once the advantageous action $\bar{a}$ is determined, the learned policy can adaptively achieve policy constraint, and the objective transforms to:

$$\mathcal{L}(\phi) = \mathbb{E}_{\substack{s,a \sim \mathcal{D}, \\ \bar{a} \in \{\tilde{a}, \pi_\phi(s)\}}} \left[-\lambda Q_\theta(s, \pi_\phi(s)) + (\pi_\phi(s) - \bar{a})^2 \right], \quad (7)$$

where λ represents a hyperparameter. A2PR steers the learned policy toward actions with high advantage, fostering improvement. Simultaneously, it dynamically balances the interplay between enhancing policy and imposing constraints. Consequently, A2PR safeguards the learned policy against the influence of suboptimal actions.

4.3. Practical Implementation

Our algorithm framework builds upon TD3+BC. The parameters $\theta_1, \theta_2, \phi, \psi$ pertain to two Q-networks, the policy network, and the value network, respectively. Additionally, $\theta_1', \theta_2', \phi'$ correspond to the parameters of the target Q-networks and the target policy network. To achieve a more balanced integration of Q-value and regularization, we formalize the Q-value within the policy loss as follows:
$$\mathcal{L}(\phi) = \mathbb{E}_{\substack{s,a \sim \mathcal{D}, \\ \bar{a} \in \{\tilde{a}, \pi_\phi(s)\}}} \left[-\lambda Q_\theta(s, \pi_\phi(s)) + (\pi_\phi(s) - \bar{a})^2 \right].$$
Here, $\lambda = \frac{\alpha N}{\sum_{s_i, a_i} Q(s_i, a_i)}$, where α is a hyperparameter and N represents the batch size (Fujimoto & Gu, 2021).

To derive the advantage function, our approach draws inspiration from IQL (Kostrikov et al., 2021b), which focuses on learning exclusively within the dataset's support to mitigate overestimation issues related to OOD actions. We have similarly customized the learning processes for both the Q-function and the Value-function. Initially, the parameter θ undergoes optimization by minimizing the following temporal difference (TD) error:

$$\mathcal{L}_Q(\theta_i) = \mathbb{E}_{(s,a,s') \sim \mathcal{D}, a' \sim h(s')}[(r(s,a) + \gamma \min_i Q_{\theta_i'}(s', a') - Q_{\theta_i}(s,a))^2], \quad (8)$$

where $Q_{\theta_i'}$ represents a target Q-value function, and $h(s') = \text{clip}(\pi_{\phi'}(s') + \epsilon_0, -A, A), \epsilon_0 \sim \text{clip}(\mathcal{N}(0, \hat{\sigma}^2), -c, c)^3, i \in 1, 2$. Here, c and $\hat{\sigma}$ denote two hyperparameters for exploration. A distinct value function is employed to approximate an expectile solely concerning the Q-function, leading to the ensuing loss:

$$\mathcal{L}_V(\varepsilon) = \mathbb{E}_{(s,a) \sim \mathcal{D}}[(Q_{\theta_i}(s,a) - V_\varepsilon(s))^2], \quad (9)$$

where V_ε represents the value function. This design ensures the avoidance of excessive conservatism. Subsequently, the equation for the advantage function is derived as:

$$A(s,a) = Q_{\theta_i}(s,a) - V_\varepsilon(s). \quad (10)$$

Algorithm 1 Adaptive Advantage-Guided Policy Regularization (A2PR)

Input: Replay buffer $\mathcal{D}$, hyper-parameters α, batch size N, target network update rate τ .
Initialize two Q networks with θ_1, θ_2, policy network with ϕ and value function network with ε, target Q and target policy network with $\theta_1' \leftarrow \theta_1, \theta_2' \leftarrow \theta_2, \phi' \leftarrow \phi$, VAE networks with $G_{\psi,\varphi} = \{E_\varphi, D_\psi\}$.
for $t = 1$ **to** T_1 **do**
    Sample mini-batch of transitions $(s, a, r, s') \sim \mathcal{D}$
    Advantage-guide VAE update:
        $\mu, \sigma = E_\varphi(s, a), \hat{a} = D_\psi(s, z), z \sim \mathcal{N}(\mu, \sigma)$
        Update it by minimizing Equation (3)
    Q-function and value-function update:
        Update Q-value by minimizing Equation (8)
        Update Value function by minimizing Equation (9)
    Adaptive Advantage Policy update:
        Update policy network by minimizing Equation (7)
    Update Target Networks:
        $\theta_i' \leftarrow \tau\theta + (1 - \tau)\theta_i', i = 1, 2$
end for

Consequently, the policy can take superior actions in states that go beyond the limitations of the dataset, alleviating undue pessimism associated with suboptimal or low-return behaviors arising from unnecessary policy constraints.

4.4. Theoretical Analysis

We provide theoretical validation for the effectiveness of A2PR. Proposition 4.3 indicates that the high-advantage action, chosen by the maximum advantage function from the augmented behavior policy, allows for a superior behavior policy to guide the learned policy towards improvement. With adaptive advantage policy regularization, Theorem 4.4 illustrates that A2PR can alleviate the Q-value overestimation problem arising from OOD actions. Additionally, Theorem 4.5 highlights a performance gap between the optimal policy and the learned policy facilitated by A2PR.

Assumption 4.2. Supposed that $Q(s, a)$ and $P(s'|s, a)$ are Lispchitz continuous w.r.t a, then

$$||Q(s, a_1) - Q(s, a_2)|| \le L_Q||a_1 - a_2|| \quad (11)$$
$$||P(s'|s, a_1) - P(s'|s, a_2)|| \le L_P||a_1 - a_2|| \quad (12)$$

for all $(s, a_1), (s, a_2) \in \mathcal{S} \times \mathcal{A}$. L_Q and L_P represent the Lipschitz constants. Equation (11) is frequently employed in the theoretical analysis of RL (Saxena et al., 2023; Gouk et al., 2021). Equation (12) has received substantial attention in theoretical RL research (Dufour & Prieto-Rumeau, 2013).

Proposition 4.3 (Behavior Policy Improvement Guarantee). *Given the accurate state-action value function $Q(s, a)$, the high-advantage actions from the improved VAE and the*

dataset own accurate advantage $A(s, a)$. Then, we have

$$J(\tilde{\pi}_\beta) - J(\pi_\beta) \ge 0, \quad (13)$$

where denote the augmented behavior policy combined the dataset with the improved VAE as $\tilde{\pi}_\beta$, the original behavior policy of the pre-collected dataset as π_β.

The proof is provided in Appendix A.2. Proposition 4.3 implies that A2PR can acquire an augmented behavior policy to effectively constrain the learned policy, thereby ensuring a performance guarantee for the learned policy (Hong et al., 2023b).

Theorem 4.4. *With policy constraint, we have $||\pi_\phi(s) - \bar{a}|| \le \epsilon_0$. Then based on Equation (6), let $||\bar{a} - \pi_\beta(s)|| \le \epsilon_1$ due to high-advantage actions from the augmented behavior policy of A2PR. With Assumption 4.2, then we have*

$$||Q(s, \pi_\phi(s)) - Q(s, \pi_\beta(s))|| \le L_Q(\epsilon_0 + \epsilon_1), \quad (14)$$
for any $s \in \mathcal{S}$.

The proof is provided in Appendix A.3. For an accurate estimation of the true Q-function $Q^\pi(s', \pi(s'))$, an approximately correct estimate $\hat{Q}^\pi(s', \pi(s'))$ is required. With a sufficiently large number of samples, $\tilde{Q}^\pi(s', \pi(s'))$ will converge to $Q^\pi(s', \pi(s'))$, causing the overestimation error δ_{error} to approach zero (Fujimoto et al., 2019), $\delta_{error} = \tilde{Q}^\pi(s, a) - Q^\pi(s, a)$. Both $\tilde{Q}^\pi(s', \pi(s'))$ and $Q^\pi(s', \pi(s'))$ satisfy Assumption 4.2. Therefore, with Theorem 4.4, we have

$$||Q^\pi(s', \pi_\phi(s')) - \tilde{Q}^\pi(s', \pi_\beta(s'))|| \le 2L_Q(\epsilon_0 + \epsilon_1) + \delta_{error}, \quad (15)$$

The detailed proof is provided in Appendix A.3. In conclusion, A2PR demonstrates its effectiveness in mitigating the problem of overestimation in value estimation.

Theorem 4.5 (Performance Gap of A2PR). *Considering Equation (6), suppose $||\bar{a} - \tilde{\pi}_\beta(s)|| \le \tilde{\epsilon}_1$ and $\max_{s \in \mathcal{S}} |\pi^*(s) - \tilde{\pi}_\beta(s)| \le \tilde{\epsilon}_*$, conditions that can be satisfied by A2PR. Then we have*

$$|J(\pi^*) - J(\pi)| \le \frac{\mathcal{C}L_P R_{max}}{1 - \gamma}(\epsilon_0 + \tilde{\epsilon}_1 + \tilde{\epsilon}_*), \quad (16)$$

where $\mathcal{C}$ is a positive constant, and $\tilde{\epsilon}_1$ represents the extent of difference between the high-advantage actions and the actions from the original behavior policy.

The detailed proof is deferred to Appendix A.4. According to Theorem 4.5, the performance gap is influenced by $\tilde{\epsilon}_1$ and $\tilde{\epsilon}_*$. The high-advantage actions originate from both the improved VAE and the dataset. Considering VAE as a straightforward and effective method for learning the behavior policy (Wu et al., 2022; Zhou et al., 2021; Fujimoto et al., 2019), the high-advantage actions exhibit minimal deviation from the behavior policy.

In the standard approach without the advantage-guided method, the distance between the average action $\bar{a}$ and the behavior policy $\pi_\beta(s)$ is constrained by $||\bar{a} - \pi_\beta(s)|| \leq \epsilon_1$. However, in our advantage-guided approach, the modified behavior policy $\tilde{\pi}_\beta(s)$ results in a smaller distance, $||\bar{a} - \tilde{\pi}_\beta(s)|| \leq \tilde{\epsilon}_1$. The behavior policy $\pi_\beta(s)$ is typically used for generating datasets like the original medium-replay dataset. In contrast, our method enhances the behavior policy by selectively incorporating data that demonstrates a higher advantage. The actions produced by $\tilde{\pi}_\beta(s)$ not only exhibit a higher average advantage but also maintain a closer proximity to the high-advantage actions $\bar{a}$. Consequently, this leads to a reduced distance measure, where $\tilde{\epsilon}_1 \leq \epsilon_1$. This tighter bound implies that our advantage-guided method facilitates a more precise alignment with desirable actions, thereby enhancing the overall efficacy of the behavior policy.

Our method optimizes this training process by focusing primarily on high-advantage data from the datasets. This approach effectively filters out less beneficial initial data, while concurrently generating a refined behavior policy, denoted as $\tilde{\pi}_\beta(s)$. This improved behavior policy is more closely aligned with the optimal policy $\pi^*(s)$, resulting in a reduced error measure, where $\tilde{\epsilon}_* \leq \epsilon_*$. In the context of the original behavior policy $\pi_\beta(s)$, the performance bound is given by $|J(\pi^*) - J(\pi)| \leq \frac{\mathcal{C}L_P R_{max}}{1-\gamma}(\epsilon_0 + \epsilon_1 + \epsilon_*)$. With our advantage-guided approach, we refine this bound to $\epsilon_0 + \tilde{\epsilon}_1 + \tilde{\epsilon}_* \leq \epsilon_0 + \epsilon_1 + \epsilon_*$. This demonstrates that our method can effectively narrow the performance gap between the learned policy and the optimal policy. By selectively focusing on high-advantage data, our method not only enhances the quality of the behavior policy but also contributes to more efficient learning outcomes.

5. Experiments

In this section, we begin by detailing the experimental setup in Section 5.1. Following that, we present the primary results on the D4RL benchmark dataset in Section 5.2. Subsequently, A2PR undergoes evaluation on supplementary low-quality datasets to assess its generalization capabilities in Section 5.3. We then investigate its effectiveness in mitigating the overestimation issue in Section 5.4. Lastly, we conduct a comprehensive ablation study in Section 5.5.

5.1. Setup

Datasets We conduct our evaluations on two task domains from the D4RL benchmark (Fu et al., 2020): Gym and AntMaze. All datasets used are of the "v2" version. The Gym-MuJoCo locomotion tasks serve as widely recognized standard benchmarks for assessment, encompassing three diverse environments (halfcheetah, hopper, and walker2d). These environments feature a multitude of trajectories and

possess inherently smooth reward functions. The AntMaze tasks, on the other hand, present challenging scenarios with sparse rewards. These tasks require the agent to navigate through mazes, piecing together sub-optimal trajectories to reach specified goals. The AntMaze environment includes various maze layouts (umaze, medium, large), each presenting diverse challenges for an 8-DoF Ant robot.

Baselines We conduct a comparative analysis of our method against several robust baselines, incorporating three state-of-the-art algorithms: AW (Hong et al., 2023a), OAP (Yang et al., 2023), and PRDC (Ran et al., 2023). AW utilizes trajectory returns to reweight the dataset for policy improvement, relying solely on the return of entire trajectories without additional interaction data. OAP introduces different policy constraints by leveraging query preferences between pre-collected and learned policy. PRDC, on the other hand, constrains the policy by searching the dataset for the nearest state-action sample. Further details on the baseline algorithms are available in Appendix B.1.

5.2. Main results on benchmark

In this section, we present the results of A2PR and competing baselines on D4RL datasets, as summarized in Table 1. The baseline results are directly sourced from their respective papers. For our method, A2PR is trained for 1 million steps with 5 random seeds. The experimental outcomes showcase the superiority of our approach, outperforming other baselines and achieving state-of-the-art performance in 16 out of 18 tasks. Beyond the D4RL dataset performance in Table 1, we provide a more comprehensive evaluation of the algorithms introduced by (Tarasov et al., 2022), as depicted in Figure 2(a). The statistical robustness offered by the results in Figure 2(a) complements the findings in Table 1, robustly affirming the effectiveness of our method. Additional implementation details can be found in Appendix B.1.

5.3. Evaluation on additional low-quality datasets

Multiple target maze We investigate a maze task with continuous actions in a 2D space. The observation comprises the agent's location and velocities, and the action is represented as $a \in [-1, 1]$, indicating the linear force applied to the agent in the x and y directions. The environment features three destinations, each associated with a unique reward ($r = 4, 2, 1$), located at $(1, 1)$, $(6, 1)$, and $(1, 6)$ on the map, respectively. The goal in this task is to navigate from a predefined starting location to the position that offers the highest reward. In this section, we formulate an offline dataset $\mathcal{D} = (s_i, a_i, r_i, d_i)_{i=1}^M$ with $M = 100,000$. The dataset comprises trajectories targeting different positions. Specifically, the robot trajectory dataset is designed to reach positions $(1, 1)$, $(6, 1)$, and $(1, 6)$, accounting for 5%, 45%, and 50% of the dataset, respectively.

Table 1. The performance of A2PR and competing baselines on D4RL datasets (Gym, AntMaze). The results for A2PR correspond to the mean and standard errors of normalized D4RL scores over the final 10 evaluations and 5 random seeds.

| Task Name	TD3+BC	BCQ	BEAR	CQL	IQL	AW	OAP	PRDC	A2PR(ours)
halfcheetah-random	11.0	8.8	15.1	20.0	11.2	16.3	24.0 ± 1.6	26.9 ± 1.0	$\mathbf{31.77 \pm 0.63}$
hopper-random	8.5	7.1	14.2	8.3	7.9	7.9	8.8 ± 1.8	26.8 ± 9.3	$\mathbf{31.55 \pm 0.29}$
walker2d-random	1.6	6.5	**10.7**	8.3	5.9	4.8	5.1 ± 5.1	5.0 ± 1.2	5.0 ± 1.1
halfcheetah-medium	48.3	47.0	41.0	44.0	47.4	46.5	56.4 ± 4.3	63.5 ± 0.9	$\mathbf{68.61 \pm 0.37}$
hopper-medium	59.3	56.7	51.9	58.5	66.2	67.7	82.0 ± 6.6	100.3 ± 0.2	$\mathbf{100.79 \pm 0.32}$
walker2d-medium	83.7	72.6	80.9	72.5	78.3	81.3	85.6 ± 1.2	85.2 ± 0.4	$\mathbf{89.73 \pm 0.60}$
halfcheetah-medium-replay	44.6	40.4	29.7	45.5	44.2	44.7	53.4 ± 1.9	55.0 ± 1.1	$\mathbf{56.58 \pm 1.33}$
hopper-medium-replay	60.9	53.3	37.3	95.0	94.7	97.0	98.5 ± 2.5	100.1 ± 1.6	$\mathbf{101.54 \pm 0.90}$
walker2d-medium-replay	81.8	52.1	18.5	77.2	73.8	78.1	84.3 ± 2.7	92.0 ± 1.6	$\mathbf{94.42 \pm 1.54}$
halfcheetah-medium-expert	90.7	89.1	38.9	91.6	86.7	89.8	83.4 ± 5.3	94.5 ± 0.5	$\mathbf{98.25 \pm 3.20}$
hopper-medium-expert	98.0	81.8	17.7	105.4	91.5	104.6	85.9 ± 6.6	109.2 ± 4.0	$\mathbf{112.11 \pm 0.32}$
walker2d-medium-expert	110.1	109.5	95.4	108.8	109.6	109.4	111.1 ± 0.6	111.2 ± 0.6	$\mathbf{114.62 \pm 0.78}$
Gym Average	698.5	615.9	450.5	724.1	717.5	748.1	778.5	869.5	**944.27**
antmaze-umaze	91.3	0.0	73.0	84.8	88.2	77.3	90.4 ± 5.2	98.8 ± 1.0	$\mathbf{99.20 \pm 1.60}$
antmaze-umaze-diverse	54.6	61.0	61.0	43.3	66.7	36.0	75.0 ± 19.0	$\mathbf{90.0 \pm 6.8}$	84.80 ± 4.49
antmaze-medium-play	0.0	0.0	0.0	65.2	70.4	10.7	62.0 ± 10.0	82.8 ± 4.8	$\mathbf{85.60 \pm 9.75}$
antmaze-medium-diverse	0.0	0.0	8.0	54.0	74.6	6.0	54.5 ± 23.3	78.8 ± 6.9	$\mathbf{85.60 \pm 4.63}$
antmaze-large-play	0.0	6.7	0.0	18.8	43.5	1.3	0	54.8 ± 10.9	$\mathbf{71.20 \pm 5.74}$
antmaze-large-diverse	0.0	2.2	0.0	31.6	45.6	2.0	9.4 ± 8.4	50.0 ± 5.4	$\mathbf{52.80 \pm 9.77}$
Antmaze Average	145.9	69.9	142.0	297.7	389.0	133.3	291.3	455.2	**478.2**
Total Average	844.4	685.8	592.5	1021.8	1106.5	881.4	1069.8	1324.7	**1422.47**

To evaluate the effect of fixed policy constraints on low-quality trajectory datasets, we conducted a comparative study of A2PR and TD3+BC. Both methods underwent training for 500,000 steps to ensure sufficient convergence. Figure 1 illustrates all trajectories as well as those from the final 100,000 steps for both A2PR and TD3+BC. Notably, the policy derived from TD3+BC demonstrates limitations, as it remains confined to suboptimal performance levels and fails to converge towards the optimal target, as highlighted in Figure 1(c). In contrast, A2PR exhibits less susceptibility to the influence of suboptimal data, successfully generating trajectories that converge on the high-return target, as depicted in Figure 1(d). The unnecessary conservative policy constraint in TD3+BC compels the learned policy to incorporate all actions within a given state from the fixed dataset. This constraint, combined with the behavior policy, assigns greater density to lower-quality data. Consequently, the trajectories produced by TD3+BC are relatively homogenous and repetitive. In contrast, the trajectories generated by A2PR are diverse, containing broad high-return trajectories. This variation primarily stems from A2PR's capacity to generate a larger number of high-advantage actions. By leveraging an enhanced Variational Autoencoder (VAE), A2PR distinguishes these actions from those in the dataset more effectively. This enhancement significantly improves the behavior policy, which in turn, provides more accurate guidance for the learned policy towards optimal actions.

Mixed random policy lower-quality dataset In this section, we aim to validate the generalization capabilities of A2PR on lower-quality datasets that consist of a substantial proportion of low-quality demonstrations. To achieve this, we evaluate A2PR alongside two strong baselines, TD3+BC and IQL, on mixed policy datasets comprising a combination of random data and expert data. The results are presented in Figure 3(a). The mixed policy dataset comprises 100,000 state-action pairs, mirroring the size of each task in the D4RL dataset. Comprising 99% random policy data and 1% expert policy data, this dataset is designed to assess the algorithms' performance on mixed-quality data.

The results indicate A2PR's superior performance compared to TD3+BC and IQL on mixed policy datasets. A substantial performance gap exists between A2PR and TD3+BC, IQL, which exhibit notably poorer performance. A2PR demonstrates improved generalization, achieving remarkable normalized scores, particularly on the halfcheetah task, even in the presence of low-quality datasets. These findings underscore A2PR's ability to mitigate the over-constraint issue associated with poorer data.

5.4. Value estimation

Value overestimation poses a significant challenge in offline RL, and we assess the comparative performance of various methods in addressing this issue using the halfcheetah-
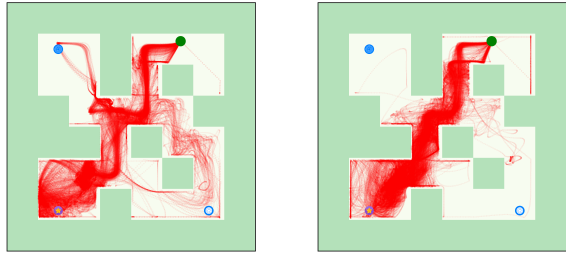
(a) TD3+BC: all trajectories (b) A2PR: all trajectories
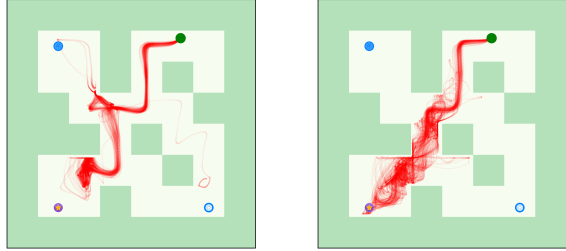
(c) TD3+BC: trajectories of final 100,000 step (d) A2PR: trajectories of final 100,000 step

Figure 1. All trajectories and the trajectories from the final 100, 000 steps of the trained policy for both A2PR and TD3+BC.
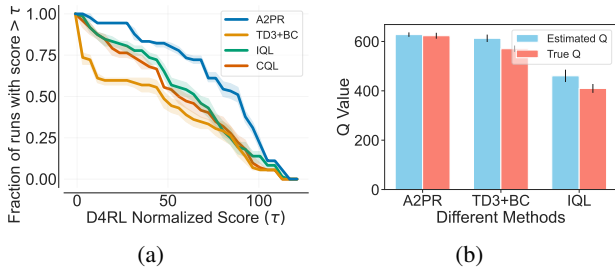

Figure 2. The performance profiles of reliable evaluation on D4RL based on 18 tasks and 5 random seeds for each task and the comparison between estimated Q-value and true Q-value of different methods.

medium-v2 dataset. True Q-values are determined through Monte-Carlo rollouts (Sutton & Barto, 2018). Over the 1M training steps, we randomly sample 10 states from the initial distribution, predict actions using the current policy, and interact with the environment for evaluation every 5k steps. To evaluate value estimation error, we conduct 10 final evaluations with 5 random seeds, estimating Q-values and comparing them with true Q-values across different methods. The results, depicted in Figure 2(b), highlight our method's ability to achieve higher true Q-values and lower value estimation error, indicating a smaller disparity between estimated and true Q-values compared to other methods. Therefore, our proposed adaptive policy regularization approach, grounded in behavior optimization, effectively mitigates the value overestimation problem.
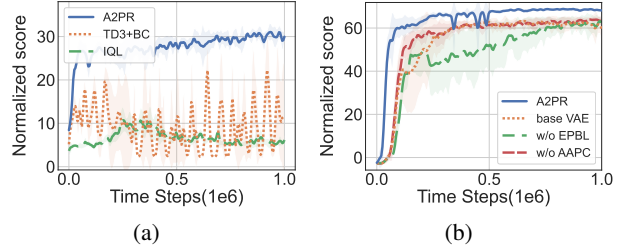

Figure 3. The performance of different methods in the mixed policy datasets and the comprehensive ablation study of A2PR on halfcheetah-medium-v2 with different components.

5.5. Ablation Study

In this section, we perform an ablation study to assess the contributions of the main components in our algorithm. We compare normalized scores on the halfcheetah-medium-v2, as depicted in Figure 3(b). Variants of A2PR include one without the elevated positive behavior learning (EPBL) component, denoted as w/o EPBL, and another incorporating a standard VAE instead of the improved version, referred to as base VAE. Additionally, a version of A2PR without adaptive advantage policy constraint is labeled as w/o AAPC. This analysis allows us to understand the individual impact of these components on the algorithm's performance.

Ablating the improved VAE leads to inferior outcomes, emphasizing the critical role of policy regularization with additional high-advantage actions from the augmented behavior policy. The convergence performance of the base VAE method experiences a slight decline, highlighting the importance of ensuring that generated actions have a higher likelihood of being high-advantage actions. Although w/o AAPC exhibits faster learning before 0.1M steps, the final performance also diminishes. This underscores the significance of selecting high-advantage actions for constraining the learned policy and guiding it toward effective policy improvement. Overall, these results underscore that A2PR achieves superior convergence performance with a swift learning pace and a high final score.

6. Discussion

A2PR aims to constrain the learned policy through high-advantage actions. Our initial idea aims to employ an enhanced behavior policy to restrict the learned policy, mitigating issues arising from unnecessary conservativeness towards inferior actions and preventing potential local optima or degraded policies. While using VAE appears intuitive for learning the behavior policy from the dataset, its effectiveness is hindered when the dataset itself contains more poor data. This is because VAE lacks a metric to distinguish good from bad state-action pairs during implicit variable learning, leading to an indiscriminate inclusion of all data. To address

this, we leverage the advantage function in RL as a metric to evaluate state-action pairs' quality. Motivated by this, we combine VAE with the advantage function, using the latter to guide VAE in learning from the offline dataset with higher advantages. In a given state, the improved VAE generates higher advantage actions, and we utilize these actions, alongside those from the dataset, for policy regularization. It mitigates the issue of unnecessary conservativeness by striking a suitable balance between policy improvement and policy constraint.

Limitation   However, there remain several limitations that require further refinement in future work. Sampling a single action using a VAE can sometimes lead to instability in learning, as the quality of a single sampled action is difficult to ensure. This variability can adversely impact the learning of the Q-function and, consequently, the policy learning process. Moving forward, selecting a more stable sampling method from the generative model will be crucial. Therefore, we need a more accurate indicator of the quality or the stability of the sampled actions. Additionally, the expressive capability of the behavior policy model is also important. A more expressive behavior policy model can more accurately identify high-advantage actions, thereby facilitating better policy improvement. Therefore, having a better behavior policy to guide the learned policy is crucial in the context of offline RL. In future research endeavors, we aim to explore advanced methodologies, such as the diffusion model, a more strong expressive generative model. This exploration is geared towards achieving even more effective policy improvement.

7. Conclusion

We introduce an innovative policy regularization approach, named Adaptive Advantage-Guided Policy Regularization (A2PR), designed for offline RL. To our knowledge, A2PR represents the first method to integrate VAE and the advantage function, providing a straightforward and efficient means to enhance the behavior policy. By leveraging the augmented behavior policy, A2PR effectively guides the learned policy to achieve policy improvement, mitigating the impact of suboptimal or out-of-distribution data. This approach introduces a region constraint, addressing the global constraint issues seen in prior policy regularization methods, which confined learned policies to actions within a specific state in the dataset. A2PR emerges as a promising solution within the realm of Offline RL, offering a robust and theoretically grounded strategy to counter unnecessary conservativeness and overestimation challenges. It attains state-of-the-art performance on the D4RL benchmark, showcasing its efficacy across diverse tasks and datasets. A2PR stands as a valuable contribution to the field, signaling potential advancements in Offline RL methodologies.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Agarwal, R., Schwarzer, M., Castro, P. S., Courville, A. C., and Bellemare, M. Deep reinforcement learning at the edge of the statistical precipice. *Advances in neural information processing systems*, 34:29304–29320, 2021.

Andrychowicz, O. M., Baker, B., Chociej, M., Jozefowicz, R., McGrew, B., Pachocki, J., Petron, A., Plappert, M., Powell, G., Ray, A., et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

Bentley, J. L. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18 (9):509–517, 1975.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Chen, X., Ghadirzadeh, A., Yu, T., Gao, Y., Wang, J., Li, W., Liang, B., Finn, C., and Zhang, C. Latent-variable advantage-weighted policy optimization for offline rl. *arXiv preprint arXiv:2203.08949*, 2022.

Dufour, F. and Prieto-Rumeau, T. Finite linear programming approximations of constrained discounted markov decision processes. *SIAM Journal on Control and Optimization*, 51(2):1298–1324, 2013.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Fujimoto, S. and Gu, S. S. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

Fujimoto, S., Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In *Interna-*

tional conference on machine learning, pp. 1587–1596. PMLR, 2018.

Fujimoto, S., Meger, D., and Precup, D. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pp. 2052–2062. PMLR, 2019.

Gouk, H., Frank, E., Pfahringer, B., and Cree, M. J. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110:393–416, 2021.

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

Hong, Z.-W., Agrawal, P., des Combes, R. T., and Laroche, R. Harnessing mixed offline reinforcement learning datasets via trajectory weighting. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=OhUAblg27z.

Hong, Z.-W., Kumar, A., Karnik, S., Bhandwaldar, A., Srivastava, A., Pajarinen, J., Laroche, R., Gupta, A., and Agrawal, P. Beyond uniform sampling: Offline reinforcement learning with imbalanced datasets. *arXiv preprint arXiv:2310.04413*, 2023b.

Hwangbo, J., Lee, J., Dosovitskiy, A., Bellicoso, D., Tsounis, V., Koltun, V., and Hutter, M. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019.

Jaques, N., Ghandeharioun, A., Shen, J. H., Ferguson, C., Lapedriza, A., Jones, N., Gu, S., and Picard, R. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*, 2019.

Jiang, L., Chen, S., Qiu, J., Xu, H., Chan, W. K., and Ding, Z. Offline reinforcement learning with imbalanced datasets. *arXiv preprint arXiv:2307.02752*, 2023.

Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pp. 267–274, 2002.

Kim, G.-H., Seo, S., Lee, J., Jeon, W., Hwang, H., Yang, H., and Kim, K.-E. Demodice: Offline imitation learning with supplementary imperfect demonstrations. In *International Conference on Learning Representations*, 2021.

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

Kostrikov, I., Fergus, R., Tompson, J., and Nachum, O. Offline reinforcement learning with fisher divergence critic regularization. In *International Conference on Machine Learning*, pp. 5774–5783. PMLR, 2021a.

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021b.

Kumar, A., Fu, J., Soh, M., Tucker, G., and Levine, S. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 1179–1191, 2020.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Li, Y., Xiong, K., Zhang, Y., Zhu, J., McAleer, S. M., Pan, W., Wang, J., Dai, Z., and Yang, Y. Jiangjun: Mastering xiangqi by tackling non-transitivity in two-player zero-sum games. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=MMsyqXIJuk.

Ma, Y., Shen, A., Jayaraman, D., and Bastani, O. Versatile offline imitation from observations and examples via regularized state-occupancy matching. In *International Conference on Machine Learning*, pp. 14639–14663. PMLR, 2022.

Nair, A., Gupta, A., Dalal, M., and Levine, S. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Peng, X. B., Kumar, A., Zhang, G., and Levine, S. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Perolat, J., De Vylder, B., Hennes, D., Tarassov, E., Strub, F., de Boer, V., Muller, P., Connor, J. T., Burch, N., Anthony, T., et al. Mastering the game of stratego with model-free multiagent reinforcement learning. *Science*, 378(6623): 990–996, 2022.

Rajeswaran, A., Kumar, V., Gupta, A., Vezzani, G., Schulman, J., Todorov, E., and Levine, S. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.

Ran, Y., Li, Y.-C., Zhang, F., Zhang, Z., and Yu, Y. Policy regularization with dataset constraint for offline reinforcement learning. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 28701–28717. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/ran23a.html.

Saxena, N., Khastagir, S., Shishir, N., and Bhatnagar, S. Off-policy average reward actor-critic with deterministic policy search. In *International Conference on Machine Learning*, pp. 30130–30203. PMLR, 2023.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839): 604–609, 2020.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

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

Singh, A., Kumar, A., Vuong, Q., Chebotar, Y., and Levine, S. Reds: Offline rl with heteroskedastic datasets via support constraints. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Sohn, K., Lee, H., and Yan, X. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

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

Tarasov, D., Nikulin, A., Akimov, D., Kurenkov, V., and Kolesnikov, S. Corl: Research-oriented deep offline reinforcement learning library. *arXiv preprint arXiv:2210.07105*, 2022.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J. P., Jaderberg, M., Vezhnevets, A. S., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky, Y., Molloy, J., Paine, T. L., Gulcehre, C., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T., Kavukcuoglu, K., Hassabis, D., Apps, C., and Silver, D. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 521:1–5, 2019.

Wu, J., Wu, H., Qiu, Z., Wang, J., and Long, M. Supported policy optimization for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 31278–31291, 2022.

Wu, Y., Tucker, G., and Nachum, O. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

Xiong, H., Xu, T., Zhao, L., Liang, Y., and Zhang, W. Deterministic policy gradient: Convergence analysis. In *Uncertainty in Artificial Intelligence*, pp. 2159–2169. PMLR, 2022.

Xu, H., Zhan, X., Yin, H., and Qin, H. Discriminator-weighted offline imitation learning from suboptimal demonstrations. In *International Conference on Machine Learning*, pp. 24725–24742. PMLR, 2022.

Yang, Q., Wang, S., Lin, M. G., Song, S., and Huang, G. Boosting offline reinforcement learning with action preference query. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 39509–39523. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/yang23o.html.

Yue, Y., Kang, B., Ma, X., Huang, G., Song, S., and Yan, S. Offline prioritized experience replay. *arXiv preprint arXiv:2306.05412*, 2023.

Zhou, W., Bajracharya, S., and Held, D. Plas: Latent action space for offline reinforcement learning. In *Conference on Robot Learning*, pp. 1719–1735. PMLR, 2021.

A. Theoretical Proofs

A.1. Proof of Proposition 4.1

We first start with a lemma considering the behavior policy improvement as follows:

Lemma A.1. *Given any two policies π_1 and π_2.*

$$J(\pi_1) - J(\pi_2) = \int_s d_{\pi_1}(s)(Q_{\pi_2}(s, \pi_1(s)) - V_{\pi_2}(s))\,\mathrm{d}s \tag{17}$$

$$= \int_s d_{\pi_1}(s) \int_a \pi_1(a|s) A^{\pi_2}(s, a)\,\mathrm{d}a\,\mathrm{d}s. \tag{18}$$

Proof. The deviation of Equation (17) in Lemma A.1 is related to (Yang et al., 2023; Kakade & Langford, 2002).

The deviation of Equation (18) in Lemma A.1 is related to (Yue et al., 2023). $\square$

Proposition A.2. *proposition1 Suppose that $A^{\pi_\beta}(s, a)(\hat{\pi}_\beta(a|s) - \pi_\beta(a|s)) \geq 0$. Then, we have*

$$J(\hat{\pi}_\beta) - J(\pi_\beta) \geq 0, \tag{19}$$

Proof. Based on Equation (3), it holds that $\forall s \in S$, $A^{\pi_\beta}(s, a)(\hat{\pi}_\beta(a|s) - \pi_\beta(a|s)) \geq 0$.

$$
\begin{aligned}
J(\hat{\pi}_\beta) - J(\pi_\beta) &= \int_s d_{\hat{\pi}_\beta}(s) \int_a \hat{\pi}_\beta(a|s) A^{\pi_\beta}(s, a)\,\mathrm{d}a\,\mathrm{d}s \\
&\geq \int_s d_{\hat{\pi}_\beta}(s) \int_a \pi_\beta(a|s) A^{\pi_\beta}(s, a)\,\mathrm{d}a\,\mathrm{d}s \\
&= 0.
\end{aligned}
\tag{20}
$$

Incorporating the advantage property $\int_a \pi_\beta(a|s) A^{\pi_\beta}(s, a)\,\mathrm{d}a = 0$, the above final derivation is as follows. So it follows that $J(\hat{\pi}_\beta) - J(\pi_\beta) \geq 0$. Theorem 4.4, suggests that the preference density estimator achieves policy improvement compared to behavior policy. $\square$

A.2. Proof of Proposition 4.3

Behavior Policy Improvement Guarantee.

Proof. According to Equation (17) in Lemma A.1, it follows that

$$J(\tilde{\pi}_\beta) - J(\pi) = \int_s d_{\tilde{\pi}_\beta}(s)(Q(s, \tilde{\pi}_\beta(s)) - V(s))\,\mathrm{d}s \tag{21}$$

$$J(\pi_\beta) - J(\pi) = \int_s d_{\pi_\beta}(s)(Q(s, \pi_\beta(s)) - V(s))\,\mathrm{d}s. \tag{22}$$

Combining Equation (21) and Equation (22), we have

$$J(\tilde{\pi}_\beta) - J(\pi_\beta) = J(\tilde{\pi}_\beta) - J(\pi) + J(\pi) - J(\pi_\beta) \tag{23}$$

$$= (J(\tilde{\pi}_\beta) - J(\pi)) - (J(\pi_\beta) - J(\pi)) \tag{24}$$

$$= \int_s d_{\tilde{\pi}_\beta}(s)(Q(s, \tilde{\pi}_\beta(s)) - V(s))\,\mathrm{d}s - \int_s d_{\pi_\beta}(s)(Q(s, \pi_\beta(s)) - V(s))\,\mathrm{d}s \tag{25}$$

$$\overset{(i)}{\approx} \int_s d_{\pi_\beta}(s)(Q(s, \tilde{\pi}_\beta(s)) - Q(s, \pi_\beta(s)))\,\mathrm{d}s, \tag{26}$$

(i) represents $d_{\tilde{\pi}_\beta} \approx d_{\pi_\beta}$ because our method only updates policies for a finite set of states in the continuous state space at each iteration, the measure of these states in the entire state space is zero. More precisely, the probability of the measure of non-overlapping states between $\tilde{\pi}_\beta$ and π_β being zero is one. Hence, assuming that the original policy and the updated policy have approximately equal state visitation probabilities (Schulman et al., 2015).

Based on Equation (5), we have $A(s, \tilde{\pi}_\beta(s)) \geq A(s, \pi_\beta(s))$. With Equation (10), we have $Q(s, \tilde{\pi}_\beta(s)) \geq Q(s, \pi_\beta(s))$. Then,

$$
\begin{aligned}
J(\tilde{\pi}_\beta) - J(\pi_\beta) &\approx \int_s d_{\pi_\beta}(s)(Q(s, \tilde{\pi}_\beta(s)) - Q(s, \pi_\beta(s)))\, \mathrm{d}s \\
&\geq 0.
\end{aligned}
\tag{27}
$$

The proof of Proposition 4.3 is finished. □

A.3. Proof of Theorem 4.4

We first start with two lemmas as follows:

Lemma A.3 (Triangle inequality). *For any $x \in \mathbb{R}^n, y \in \mathbb{R}^n$,, it holds that,*

$$
||x + y|| \leq ||x|| + ||y||.
\tag{28}
$$

Lemma A.4. *With Assumption 4.2, it holds that,*

$$
\int_s |d_{\pi_\phi}(s) - d_{\pi_\beta}(s)|\, \mathrm{d}s \leq \mathcal{C} L_P \max_{s \in \mathcal{S}} ||\pi(s) - \pi_\beta(s)||,
\tag{29}
$$

where $\mathcal{C}$ is a positive constant.

Proof. The proof of Lemma A.4 can be found in the appendix of (Xiong et al., 2022).

Next, we will provide the proof of Theorem 4.4.

Based on Lemma A.3 and Equation (6), the left side of Equation (14) can be expanded as below

$$
\begin{aligned}
||Q(s, \pi_\phi(s)) - Q(s, \pi_\beta(s))|| &= ||Q(s, \pi_\phi(s)) - Q(s, \bar{a}) + Q(s, \bar{a}) + Q(s, \pi_\beta(s))|| \\
&\leq ||Q(s, \pi_\phi(s)) - Q(s, \bar{a})|| + ||Q(s, \bar{a}) + Q(s, \pi_\beta(s))|| \\
&\leq L_Q(||\pi_\phi(s) - \bar{a}|| + ||\bar{a} - \pi_\beta(s)||) \\
&\leq L_Q(\epsilon_0 + \epsilon_1).
\end{aligned}
\tag{30}
$$

The proof of Theorem 4.4 is finished. □

Proof. Next, we will demonstrate that A2PR effectively mitigates the value overestimation issue.

With the overestimation error (Fujimoto et al., 2019) and Assumption 4.2, then we have

$$
\delta_{error} = \tilde{Q}^\pi(s', \pi_\beta(s')) - Q^\pi(s', \pi_\beta(s')),
\tag{31}
$$

$$
||Q^\pi(s', \pi_\phi(s')) - Q^\pi(s', \pi_\beta(s'))|| \leq L_Q(\epsilon_0 + \epsilon_1),
\tag{32}
$$

$$
||\tilde{Q}^\pi(s', \pi_\phi(s')) - \tilde{Q}^\pi(s', \pi_\beta(s'))|| \leq L_Q(\epsilon_0 + \epsilon_1).
\tag{33}
$$

Combining Equation (31), Equation (32) and Equation (33), then with Lemma A.3 we have that

$$
\begin{aligned}
||\tilde{Q}^\pi(s', \pi_\phi(s')) - Q^\pi(s', \pi_\phi(s'))|| &= ||\tilde{Q}^\pi(s', \pi_\phi(s')) - \tilde{Q}^\pi(s', \pi_\beta(s')) + \tilde{Q}^\pi(s', \pi_\beta(s')) - Q^\pi(s', \pi_\phi(s'))|| \\
&= ||\tilde{Q}^\pi(s', \pi_\phi(s')) - \tilde{Q}^\pi(s', \pi_\beta(s')) + Q^\pi(s', \pi_\beta(s')) + \delta_{error} - Q^\pi(s', \pi_\phi(s'))|| \\
&\leq ||\tilde{Q}^\pi(s', \pi_\phi(s')) - \tilde{Q}^\pi(s', \pi_\beta(s'))|| + ||Q^\pi(s', \pi_\phi(s')) - Q^\pi(s', \pi_\beta(s'))|| + \delta_{error} \\
&\leq 2L_Q(\epsilon_0 + \epsilon_1) + \delta_{error}.
\end{aligned}
\tag{34}
$$

The proof of mitigating the value overestimation issue has been completed. □

A.4. Proof of Theorem 4.5

Proof. With Lemma A.3, it follows that

$$
\begin{aligned}
|J(\pi^*) - J(\pi)| &= |J(\pi^*) - J(\tilde{\pi}_\beta) + J(\tilde{\pi}_\beta) - J(\pi)| \\
&\le |J(\pi^*) - J(\tilde{\pi}_\beta)| + |J(\pi) - J(\tilde{\pi}_\beta)|.
\end{aligned}
\tag{35}
$$

Firstly, considering $|J(\pi) - J(\tilde{\pi}_\beta)|$ and Lemma A.4, we have

$$
\begin{aligned}
|J(\pi) - J(\tilde{\pi}_\beta)| &= |\frac{1}{1-\gamma}\mathbb{E}_{s \sim d_{\pi_\phi}}[r(s)] - \frac{1}{1-\gamma}\mathbb{E}_{s \sim d_{\tilde{\pi}_\beta}}[r(s)]| \\
&= \frac{1}{1-\gamma}|\int_s (d_{\pi_\phi}(s) - d_{\tilde{\pi}_\beta}(s))r(s)\,\mathrm{d}s| \\
&\le \frac{1}{1-\gamma}\int_s |d_{\pi_\phi}(s) - d_{\tilde{\pi}_\beta}(s)||r(s)|\,\mathrm{d}s \\
&\le \frac{R_{max}}{1-\gamma}\int_s |d_{\pi_\phi}(s) - d_{\tilde{\pi}_\beta}(s)|\,\mathrm{d}s \\
&\le \frac{\mathcal{C}L_P R_{max}}{1-\gamma}\max_{s \in S}||\pi(s) - \tilde{\pi}_\beta(s)|| \\
&= \frac{\mathcal{C}L_P R_{max}}{1-\gamma}\max_{s \in S}||\pi(s) - \bar{a} + \bar{a} - \tilde{\pi}_\beta(s)|| \\
&\le \frac{\mathcal{C}L_P R_{max}}{1-\gamma}(||\pi(s) - \bar{a}|| + ||\bar{a} - \tilde{\pi}_\beta(s)||) \\
&\le \frac{\mathcal{C}L_P R_{max}}{1-\gamma}(\epsilon_0 + \tilde{\epsilon}_1).
\end{aligned}
\tag{36}
$$

Then, considering $|J(\pi^*) - J(\tilde{\pi}_\beta)|$ and Lemma A.4, we get

$$
\begin{aligned}
|J(\pi^*) - J(\tilde{\pi}_\beta)| &= |\frac{1}{1-\gamma}\mathbb{E}_{s \sim d_{\pi_\phi^*}}[r(s)] - \frac{1}{1-\gamma}\mathbb{E}_{s \sim d_{\tilde{\pi}_\beta}}[r(s)]| \\
&= \frac{1}{1-\gamma}|\int_s (d_{\pi_\phi^*}(s) - d_{\tilde{\pi}_\beta}(s))r(s)\,\mathrm{d}s| \\
&\le \frac{1}{1-\gamma}\int_s |d_{\pi_\phi^*}(s) - d_{\tilde{\pi}_\beta}(s)||r(s)|\,\mathrm{d}s \\
&\le \frac{R_{max}}{1-\gamma}\int_s |d_{\pi_\phi^*}(s) - d_{\tilde{\pi}_\beta}(s)|\,\mathrm{d}s \\
&\le \frac{\mathcal{C}L_P R_{max}}{1-\gamma}\max_{s \in S}||\pi^*(s) - \tilde{\pi}_\beta(s)|| \\
&\le \frac{\mathcal{C}L_P R_{max}}{1-\gamma}\tilde{\epsilon}_*.
\end{aligned}
\tag{37}
$$

Finally, combining Equation (36) and Equation (37), we have that

$$
\begin{aligned}
|J(\pi^*) - J(\pi)| &= |J(\pi^*) - J(\tilde{\pi}_\beta) + J(\tilde{\pi}_\beta) - J(\pi)| \\
&\le |J(\pi^*) - J(\tilde{\pi}_\beta)| + |J(\tilde{\pi}_\beta) - J(\pi)| \\
&\le \frac{\mathcal{C}L_P R_{max}}{1-\gamma}(\epsilon_0 + \tilde{\epsilon}_1 + \tilde{\epsilon}_*).
\end{aligned}
\tag{38}
$$

The proof is finished. When not using the advantage-guided method, $||\bar{a} - \pi_\beta(s)|| \le \epsilon_1$. $\tilde{\pi}_\beta(s)$ produces actions that have a higher average advantage and a relatively smaller difference with actions $\bar{a}$ than the behavior policy $\pi_\beta(s)$, then $\tilde{\epsilon}_1 \le \epsilon_1$. Our method selects data with higher advantage through advantage-guided, which is equivalent to using a better behavior policy $\tilde{\pi}_\beta(s)$ for generating the data. Thus this better behavior policy $\tilde{\pi}_\beta(s)$ reduces the error with respect to the optimal policy $\pi^*(s)$ than the behavior policy $\pi_\beta(s)$, so $\tilde{\epsilon}_* \le \epsilon_*$. For $\pi_\beta(s)$, $|J(\pi^*) - J(\pi)| \le \frac{\mathcal{C}L_P R_{max}}{1-\gamma}(\epsilon_0 + \epsilon_1 + \epsilon_*)$. Then, $\epsilon_0 + \tilde{\epsilon}_1 + \tilde{\epsilon}_* \le \epsilon_0 + \epsilon_1 + \epsilon_*$. Thus, our advantage-guided method can reduce this performance gap.

$\square$
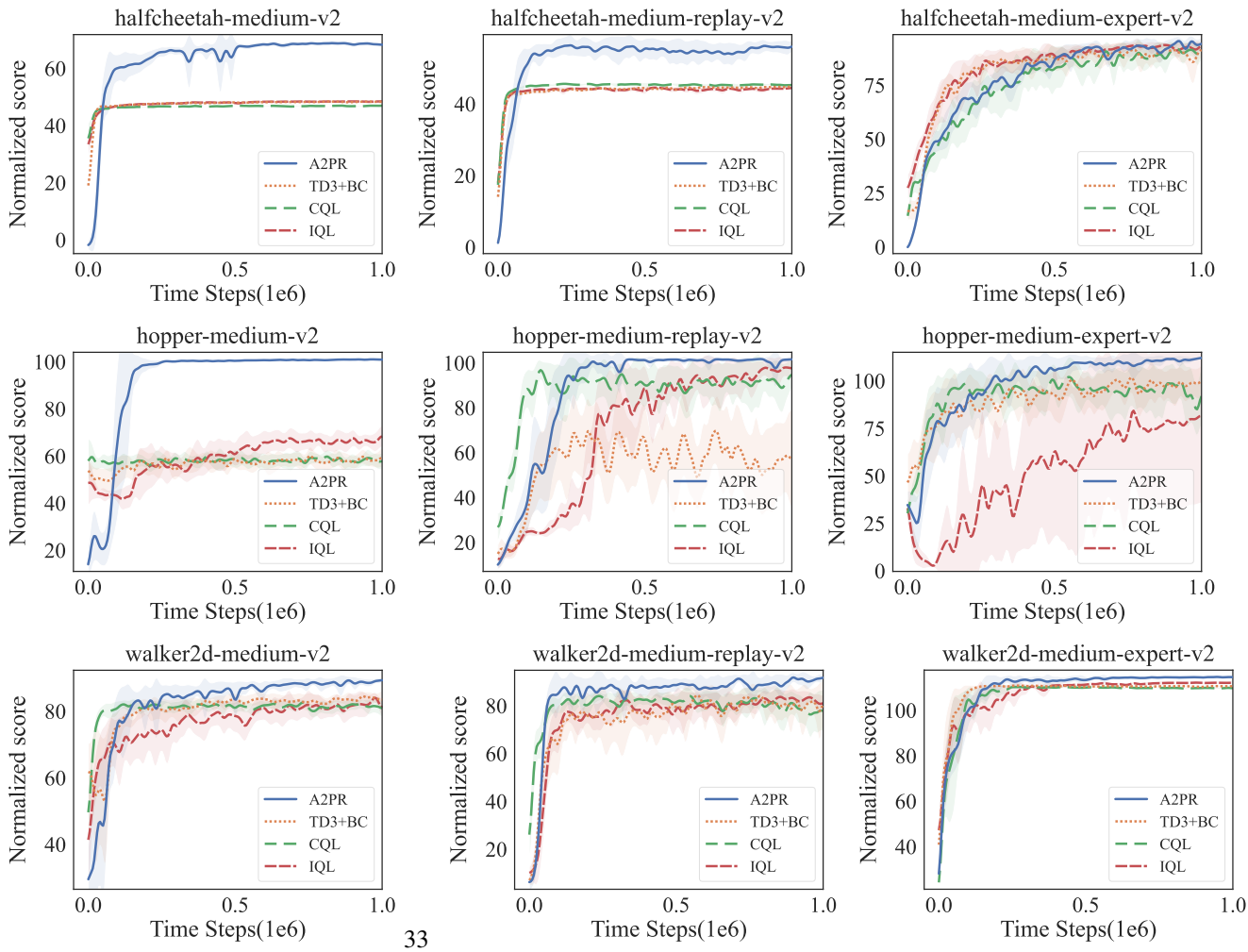
Figure 4. Results of performance comparisons conducted on nine original tasks in the D4RL dataset. The lines and shaded areas indicate the averages and standard deviations calculated over 5 random seeds, respectively.

B. More Results

B.1. Main results on benchmark

Baselines We compare our method with several strong baselines, including three state-of-the-art algorithms: AW (Hong et al., 2023a), OAP (Yang et al., 2023), and PRDC (Ran et al., 2023). Additionally, we consider policy regularization methods using behavior cloning, such as TD3+BC (Fujimoto & Gu, 2021); methods employing other divergences like BCQ (Fujimoto et al., 2019) and BEAR (Kumar et al., 2019) based on maximum mean discrepancy (MMD) and Gaussian kernel; Q-value constraint or critic penalty methods like CQL (Kumar et al., 2020), which lower-bounds the policy's true value with a conservative Q-value function; and implicit Q learning with expectile regression, avoiding queries to values of OOD actions as in IQL (Kostrikov et al., 2021b).

In addition to presenting the D4RL dataset performance in Table 1, we provide a more thorough evaluation of the algorithms implemented by (Tarasov et al., 2022), depicted in Figure 2(a). The training curves of A2PR are compared with TD3+BC, CQL, and IQL, and the results are illustrated in Figure 4. Leveraging metrics from a reputable source (Agarwal et al., 2021) enhances result confidence by addressing statistical uncertainty across multiple runs. Improved outcomes are indicated by higher mean, median, and IQM scores, along with a lower optimality gap, as illustrated in Figure 5. The results robustly confirm the superiority of our method.
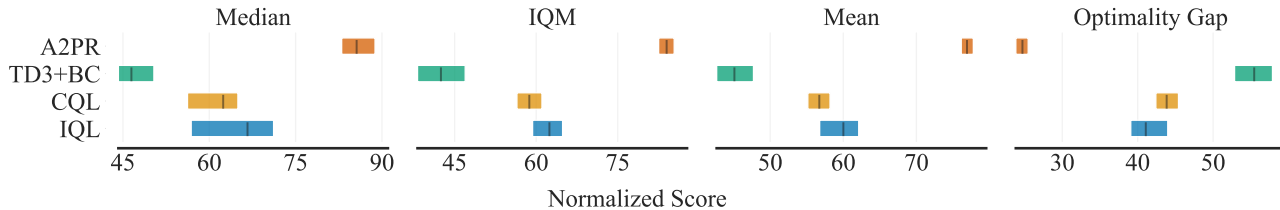
Figure 5. Reliable evaluation for statistical uncertainty on D4RL with 95% CIs based on 18 tasks and 5 random seeds for each task.

Table 2. The influence of different advantage thresholds ϵ_A on performance of D4RL datasets. The results for A2PR correspond to the mean and standard errors of normalized D4RL scores over 5 random seeds.

Task Name	$\epsilon_A = -0.5$	$\epsilon_A = -0.1$	$\epsilon_A = 0$	$\epsilon_A = 0.1$	$\epsilon_A = 0.5$
Halfcheetah-medium-v2	67.45 ± 0.674	67.03 ± 2.56	68.61 ± 0.37	$\mathbf{69.66 \pm 0.52}$	65.71 ± 6.27
Hopper-medium-v2	100.1 ± 0.88	99.88 ± 0.07	$\mathbf{100.79 \pm 0.32}$	100.1 ± 2.21	95.76 ± 8.27
Walker2d-medium-v2	76.79 ± 7.72	82.41 ± 3.77	89.73 ± 0.60	88.9 ± 0.62	$\mathbf{90.07 \pm 3.30}$
Halfcheetah-medium-replay-v2	42.80 ± 0.89	52.12 ± 1.71	$\mathbf{56.58 \pm 1.33}$	52.58 ± 3.46	52.72 ± 0.75
Hopper-medium-replay-v2	$\mathbf{101.7 \pm 0.75}$	100.9 ± 0.5	101.54 ± 0.90	99.55 ± 1.86	101.2 ± 0.37
Walker2d-medium-replay-v2	95.92 ± 1.13	90.99 ± 7.56	94.42 ± 1.54	88.22 ± 2.83	$\mathbf{96.31 \pm 1.96}$
Halfcheetah-medium-expert-v2	87.06 ± 5.91	97.57 ± 2.30	$\mathbf{98.25 \pm 3.20}$	93.29 ± 4.38	93.5 ± 6.06
Hopper-medium-expert-v2	112.1 ± 0.28	107.54 ± 2.26	$\mathbf{112.11 \pm 0.32}$	105.35 ± 4.38	96.44 ± 4.58
Walker2d-medium-expert-v2	110.2 ± 2.55	112.42 ± 0.87	$\mathbf{114.62 \pm 0.78}$	105 ± 6.27	112.4 ± 1.25

B.2. Sensitivity on the different advantage thresholds ϵ_A

This section explores the impact of varying advantage thresholds, denoted as ϵ_A, on the performance of our policy. We conduct experiments using the A2PR algorithm across three datasets: Hopper, HalfCheetah, and Walker2d (specifically -medium-replay-v2, -medium-v2, -medium-expert-v2). For each dataset, the model was trained for 1 million steps across five different seeds, as shown in Table 2. We assessed the performance of the policy with advantage thresholds set at $\epsilon_A \in \{-0.5, -0.1, 0, 0.1, 0.5\}$. The results indicate that the optimal performance is achieved when $\epsilon_A = 0$. This setting allows for a balanced approach to action selection, effectively avoiding the pitfalls of high thresholds that may exclude potentially beneficial actions due to their sparse occurrence, as well as low thresholds that might include more suboptimal actions. Essentially, a zero threshold maintains a healthy balance, enabling the selection of actions that contribute positively to learning outcomes without compromising the robustness of the algorithm. Furthermore, the consistent performance across varying thresholds suggests that our algorithm is robust to changes in ϵ_A. This adaptability underscores the utility of A2PR in diverse settings, making it a reliable choice for applications requiring a stable learning process.

B.3. Mean advantage based on the same state

A2PR aims to utilize high-advantage actions to adaptively constrain the learned policy. To examine whether A2PR has learned actions with higher advantages from the low-quality dataset, A2PR, TD3+BC, and IQL are evaluated on the same 1,000 states randomly sampled from the halfcheetah-medium task dataset with 5 random seeds. Comparisons among the mean advantage curves of the actions from different methods are shown in Figure 6(a). The results demonstrate that our method selects actions with higher advantages based on the same states compared to TD3+BC and IQL. Moreover, the mean advantage from our method is positive in all 1000 states with 5 random seeds. These findings provide additional confirmation that A2PR has successfully acquired advantageous actions, even when exposed to a low-quality dataset.

B.4. A2PR implementation based on SAC framework

We implement the A2PR algorithm on the Soft Actor-Critic (SAC) (Haarnoja et al., 2018) framework, Which is compared to a version based on the TD3 (Fujimoto et al., 2018) framework. Our experimental evaluation spans several datasets:

halfcheetah-random-v2, halfcheetah-medium-v2, halfcheetah-medium-expert-v2, and halfcheetah-medium-replay-v2. The comparative analysis indicates that while the performance of the two approaches was broadly similar, the TD3 variant consistently outperformed the SAC variant, as shown in Figure 6(b).
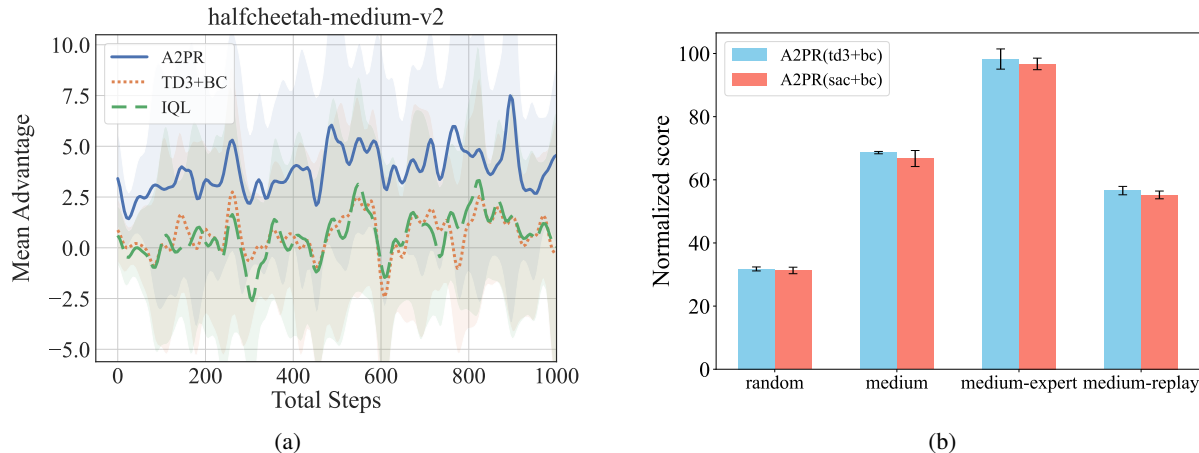

(a)

(b)

Figure 6. (a) The comparisons between the mean advantage curves of the actions of different methods on halfcheetah-medium-v2. (b) The performance comparison of A2PR on the TD3 variant and the SAC variant.

B.5. Training time

Managing time complexity presents a significant challenge in offline RL. We conduct experiments by running our methods and the baselines on the same dataset and machine for 1M steps. The re-training of TD3+BC, AWAC, IQL, and CQL was performed on the halfcheetah-medium-v2 dataset, utilizing implementations from https://github.com/tinkoff-ai/CORL (Tarasov et al., 2022). The re-training of PRDC was implemented using its official code from the original paper. The results in Table 3 indicate that our method performs faster than other baselines on the halfcheetah-medium-v2 dataset, particularly when compared to CQL and PRDC, which require KD-tree for retrieval. Thanks to the efficiency of VAE's powerful generation, our method demonstrates notable speed.

Table 3. The training time of the different methods

Methods	TD3+BC	AWAC	IQL	PRDC	CQL	A2PR
Train time	2h18m	3h40m	5h23m	6h49m	9h2m	3h59m

B.6. The generalization of A2PR on noisy datasets

We conducted experiments using the Multiple Target Maze and Mixed Random Policy Low-Quality datasets, illustrated in Figures 1 and 3(a) in Section 5.3. These datasets differ significantly from those in the D4RL benchmark. Our results indicate that the A2PR algorithm outperforms established baselines, demonstrating noteworthy generalization capabilities. To further assess the robustness and generalization performance of A2PR, we introduced Gaussian noise $\mathcal{N}(0, 1)$ to the state inputs during the evaluation phase. We conduct experiments using the A2PR algorithm across three datasets: Hopper, HalfCheetah, and Walker2d (specifically -medium-v2, -medium-replay-v2, -medium-expert-v2). For each dataset, the model was trained for 1 million steps across five different seeds. The results are shown in Figure 7. This variant, A2PR(noisy), was compared against a similarly modified version of the TD3+BC algorithm, labeled TD3+BC(noisy), where noise was also added to the state inputs. The findings reveal that A2PR(noisy) not only consistently outperforms TD3+BC(noisy) but also maintains superior performance over the original TD3+BC across most tasks, despite a slight decrease in performance compared to its noise-free version. Notably, A2PR's performance on noisy states frequently surpasses that of TD3+BC on noise-free states, further underscoring A2PR's enhanced ability to generalize well under conditions of state perturbation.
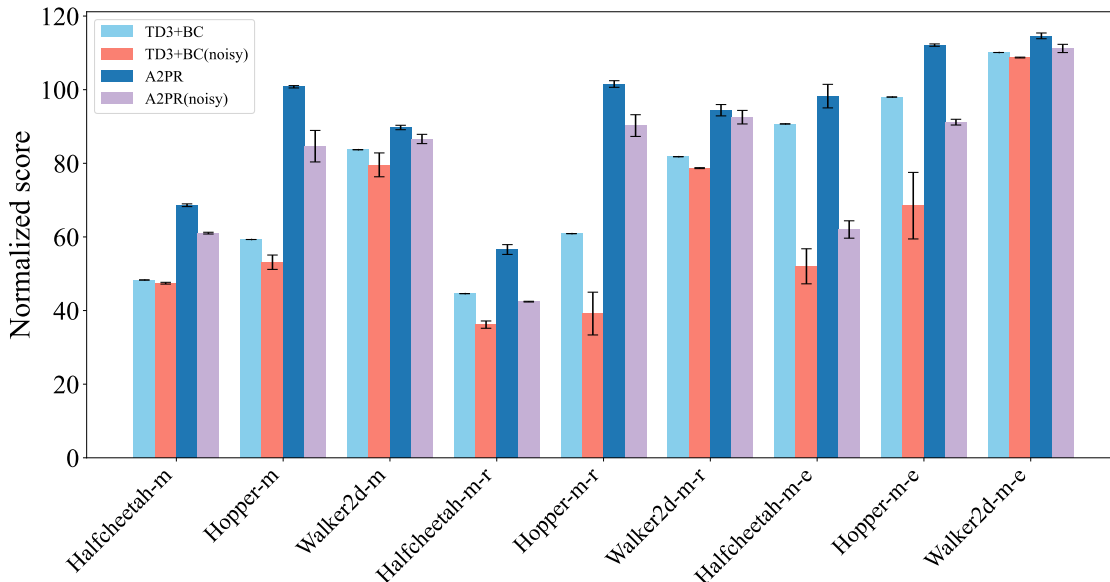
Figure 7. The performance comparison of A2PR and TD3+BC on noisy datasets.

C. Implement details

A2PR was implemented using PyTorch based on the TD3+BC implementation. The Elevating Positive Behavior Learning (EPBL) and Adaptive Advantage Policy Constraint (AAPC) components were implemented by ourselves. The hyperparameters of the algorithm are detailed in Table 4 and Table 5. To fully demonstrate the performance of A2PR and ensure fairness in comparison with the latest state-of-the-art research, the policy update equation introduces a new hyperparameter w_2 and retains the α values from the PRDC (Ran et al., 2023) algorithm, which means retaining the same λ values.

$$\mathcal{L}(\phi) = \mathbb{E}_{\substack{s,a\sim\mathcal{D}, \\ \bar{a}\in\{\tilde{a},\pi_\phi(s)\}}} \left[-\lambda Q_\theta(s, \pi_\phi(s)) + w_2(\pi_\phi(s) - \bar{a})^2 \right].$$

C.1. Hardware

We use the following hardware:

1. NVIDIA RTX 3090

2. 12th Gen Intel(R) Core(TM) i7-12900K

C.2. Software

We use the following software versions:

1. Python 3.9.19

2. D4RL 1.1 (Fu et al., 2020)

3. Mujoco 3.1.5 (Todorov et al., 2012)

4. Gym 0.23.1 (Brockman et al., 2016)

5. Mujoco-py 2.1.2.14

6. Pytorch 1.13.1 + cu11.7 (Paszke et al., 2019)

The v2 version of D4RL benchmark datasets is utilized in Gym locomotion and AntMaze tasks.

Table 4. Hyperparameter Table

	Hyper-parameters	Value
TD3	Number of iterations	1e6
	Target update rate τ	5e-3
	Policy noise	0.2
	Policy noise clipping	(-0.5,0.5)
	Policy update frequency	2
	Discount γ for Mujoco	0.99, 0.995
	Discount γ for Antmaze	0.995
	Actor learning rate	3e-4
	Critic learning rate for Mujoco	3e-4
	Critic learning rate for Antmaze	1e-4
Network	Q-Critic hidden dim	256
	Q-Critic hidden layers	3
	Q-Critic Activation function	ReLU
	V-Critic hidden dim	256
	V-Critic hidden layers	3
	V-Critic Activation function	ReLU
	Actor hidden dim	256
	Actor hidden layers	2
	Actor Activation function	ReLU
	Mini-batch size	256
	Optimizer	Adam (Kingma & Ba, 2014)
A2PR	Normalized state	True
	α for Mujoco	40.0, 2.5
	α for Antmaze	{2.5, 7.5, 20.0}
	ϵ_A	0

Table 5. Hyperparameter values for different tasks

Task name	w_1	w_2	γ	α
Halfcheetah-random-v2	1.0	1.0	0.99	40.0
Halfcheetah-medium-v2	1.0	1.0	0.99	40.0
Halfcheetah-medium-replay-v2	1.5	0.8	0.995	40.0
Halfcheetah-medium-expert-v2	1.0	15.0	0.99	40.0
Hopper-random-v2	1.5	1.5	0.995	2.5
Hopper-medium-v2	1.0	0.4	0.995	2.5
Hopper-medium-replay-v2	1.5	0.5	0.99	2.5
Hopper-medium-expert-v2	1.0	4.0	0.99	2.5
Walker2d-random-v2	1.0	1.0	0.995	2.5
Walker2d-medium-v2	1.5	1.0	0.99	2.5
Walker2d-medium-replay-v2	1.5	1.5	0.99	2.5
Walker2d-medium-expert-v2	1.0	0.8	0.99	2.5