Policy-regularized Offline Multi-objective Reinforcement Learning

Qian Lin
Sun Yat-sen University
Guangzhou, China
linq67@mail2.sysu.edu.cn

Chao Yu
Sun Yat-sen University
Guangzhou, China
yuchao3@mail.sysu.edu.cn

Zongkai Liu
Sun Yat-sen University
Guangzhou, China
liuzk@mail2.sysu.edu.cn

Zifan Wu
Sun Yat-sen University
Guangzhou, China
wuzf5@mail2.sysu.edu.cn

ABSTRACT

In this paper, we aim to utilize only offline trajectory data to train a policy for multi-objective RL. We extend the offline policy-regularized method, a widely-adopted approach for single-objective offline RL problems, into the multi-objective setting in order to achieve the above goal. However, such methods face a new challenge in offline MORL settings, namely the *preference-inconsistent demonstration* problem. We propose two solutions to this problem: 1) filtering out preference-inconsistent demonstrations via approximating behavior preferences, and 2) adopting regularization techniques with high policy expressiveness. Moreover, we integrate the preference-conditioned scalarized update method into policy-regularized offline RL, in order to simultaneously learn a set of policies using a single policy network, thus reducing the computational costs induced by the training of a large number of individual policies for various preferences. Finally, we introduce *Regularization Weight Adaptation* to dynamically determine appropriate regularization weights for arbitrary target preferences during deployment. Empirical results on various multi-objective datasets demonstrate the capability of our approach in solving offline MORL problems. [1]

KEYWORDS

Offline RL; Multi-objective RL; Policy regularization

1 INTRODUCTION

In recent years, Reinforcement Learning (RL) [21] has achieved tremendous successes in solving various challenging problems including robotics [12, 13], games [25, 26] and recommendation systems [32]. In the standard RL setting, the primary goal is to optimize a policy that maximizes a cumulative scalar reward. However, there often exist multiple, potentially conflicting objectives in many real-world applications, which motivates the research of Multi-Objective Reinforcement Learning (MORL) [17]. In contrast to the characteristic of single-best solution in single-objective settings, the MORL paradigm accommodates multiple optimal policies catering to various preferences that indicate the trade-off among multiple objectives. Many MORL approaches have been proposed in the past few years [2, 17, 18]. However, most existing approaches focus on the online setting, and thus are inappropriate for many safety-critical real-world scenarios due to the potential risk from substantial online exploration. Offline MORL, which aims to leverage only offline datasets for the training of multi-objective policies, holds promise for solving this challenge but has received limited attention and research.

Current offline MORL algorithms either assume that the target preferences during deployment are static and pre-known [28], or require access to behavior policies [23] or their preference information [33], which can hinder the applicability of these algorithms in many offline MORL scenarios. The primary objective of this paper is to leverage trajectory data obtained from preference-unknown behavior policies to train a policy capable of achieving satisfactory performance for arbitrary target preferences during deployment. To achieve this goal, we extend the offline policy-regularized methods [10, 14, 27, 29], a kind of widely studied and effective offline RL approach, to the multi-objective linear-preference setting to address the offline MORL problem. First, we introduce the *preference-inconsistent demonstration* problem, a challenge encountered when directly applying offline policy-regularized methods to multi-objective datasets collected by behavior policies with diverse preferences. Specifically, when a policy makes decisions based on a preference ω' deviating from the target preference ω, trajectories generated by this policy can serve as arbitrarily inferior demonstrations under preference ω, due to conflicts in final decision-making objectives between ω and ω'. Consequently, these trajectories can misguide policy learning through the behavior cloning term used by offline policy-regularized methods. To address this problem, one intuitive approach is to directly filter out preference-inconsistent demonstrations from training datasets, which can be achieved by approximating the preferences of behavior policies. In addition, given that preference-inconsistent demonstrations can increase the difficulty of implicitly modeling behavior policies in offline

---

[1]The code for this paper can be found at https://github.com/qianlin04/PRMORL

policy-regularized methods, we mitigate this issue by adopting regularization techniques with high policy expressiveness.

Another challenge is that training an individual policy for each possible target preference can lead to considerable computational costs. To solve this problem, we integrate the preference-conditioned scalarized update method [2] into policy-regularized offline RL by employing the vector value estimate, scalarized value update and preference-conditioned actors and critics. This approach enables the simultaneous learning of a set of policies with a single policy network. Moreover, since the optimal regularization weight in offline MORL setting can vary across different preferences, *Regularization Weight Adaptation* is proposed to dynamically determine the appropriate regularization weights for each target preference during deployment. Specifically, we treat behavior cloning as an additional decision-making objective and the regularization weight as the corresponding preference weight, thus incorporating them into the original multi-objective setting during training. The appropriate regularization weights then can be dynamically adjusted using a limited number of collected trajectories during deployment.

We conduct several experiments on D4MORL [33], an offline MORL dataset designed for continuous control tasks. The results demonstrate that, compared to the state-of-the-art offline MORL method PEDA [33], our approach achieves competitive or even better performance without using the preference data of behavior policies. Additionally, since many current offline datasets are collected by behavior policies with a single and fixed preference, it is meaningful to explore the feasibility of utilizing these datasets for MORL. Therefore, we introduce the MOSB dataset, a multi-objective variant of D4RL [7] collected by single-preference behavior policies, which is different from D4MORL where behavior policies are explicitly trained for MORL. Our approach consistently exhibits outstanding performance on this dataset, highlighting its superiority in terms of generalizability across various offline MORL scenarios.

2 RELATED WORK

MORL   Existing work within the domain of MORL [17, 24] can be broadly categorized into two classes, namely single-policy and multi-policy methods. Single-policy methods aim to transform a multi-objective problem into a single-objective problem under one specific preference among multiple objectives, and then solve it to obtain a single optimal policy using standard RL approaches [1, 3]. Conversely, multi-policy methods learn a set of policies to approximate the Pareto front of optimal policies. To achieve this goal, one can explicitly maintain a finite set of policies by either applying a single-policy algorithm individually for each candidate preference [18, 20], or employing evolutionary algorithms to generate a population of diverse policies [11, 30]. However, these methods suffer from low scalability to high-dimensional preference space due to the limited number of learned policies. To tackle the above challenge, a feasible approach is to simultaneously learn a set of policies represented by a single network [2, 4, 31]. Conditioned Network (CN) method [2] utilizes a single preference-conditioned network to represent value functions over the entire preference space and applies the scalarized update method [18] to update the vector value function. Based on this method, several works [4, 31] are proposed to enhance performance while improving sample efficiency during online training.

Policy-regularized Offline RL   The main challenge of offline RL is the distribution shift between the state-action pairs in the dataset and those induced by the learned policy [16]. Policy regularization [10, 14, 27, 29] is proposed as a simple yet efficient approach to address this issue. This method typically incorporates a behavior cloning mechanism into the policy learning to regularize the learned policy to stay close to the behavior policy. BCQ [10] employs a conditional variational auto-encoder to model behavior policies and learns a perturbation model to generate bounded adjustments to the actions of the behavior policy. BEAR [14] regularizes the learned policy with the maximum mean discrepancy, which is estimated using multiple samples from the learned policy and pre-modeled behavior policy. BRAC [29] proposes a general regularization framework and comprehensively evaluates the existing techniques in recent methods. TD3+BC [8] adds maximum likelihood estimation (MLE) as a minimal change to the TD3 algorithm, which exhibits surprising effectiveness in addressing offline RL problems. Diffusion Q-learning [27] is a state-of-the-art offline policy regularization method that leverages the diffusion model as the learned policy while utilizing the training loss of the diffusion policy as a policy regularizer.

Offline MORL   There exist a limited number of works that focus on the offline MORL problem. PEDI [28] utilizes the dual gradient ascent and pessimism principle to find an optimal policy that achieves the highest utility under a fixed preference while keeping the vector value constrained within a target set. MO-SPIBB [23] adopts the Seldonian framework [22] to achieve safe policy improvement under some predefined preferences using offline datasets. These methods require prior knowledge of target preferences and cannot be adapted to new preferences during deployment. To learn a single policy that works for all preferences and solve the continuous control problem in MORL settings, PEDA [33] introduces D4MORL, a large-scale benchmark for offline MORL, and extends supervised offline RL methods [5, 6] to MORL by directly incorporating the preference information into the input of decision models.

Compared to the above works, we provide an intuitive yet unexplored solution to offline MORL problems by integrating two well-established methods from the online MORL and offline RL domains, i.e., offline policy-regularized methods and the preference-conditioned scalarized update technique, respectively. The major advantage of our proposed method is that it can respond to arbitrary target preferences without access to behavior policies or their preference information, which is typically unavailable in real-world scenarios. The empirical results show that our approach demonstrates superior or competitive performance compared with PEDA on both datasets collected by multi-preference behavior policies and by single-preference behavior policies.

3 BACKGROUND

Multi-objective RL   The problem of multi-objective RL can be formulated as a multi-objective Markov decision process (MOMDP), which is represented by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \boldsymbol{r}, \gamma, \Omega, f_\Omega)$ with state space $\mathcal{S}$, action space $\mathcal{A}$, transition distribution $\mathcal{P}(s'|s, a)$, vector reward function $\boldsymbol{r}(s, a)$, discount factor $\gamma \in [0, 1]$, space of preferences Ω, and preference functions $f_{\boldsymbol{\omega}}(\boldsymbol{r})$. The vector reward function is $\boldsymbol{r}(s, a) = [r(s, a)_1, r(s, a)_2, ..., r(s, a)_n]^T$ where n is the

number of objectives and $r(s, a)_i$ is the reward of the ith objective. The preference function $f_{\boldsymbol{\omega}}(\boldsymbol{r})$ maps a reward vector $\boldsymbol{r}(s, a)$ to a scalar utility by using the given preference $\boldsymbol{\omega} \in \Omega$. The expected vector return G^{π} of a policy π is denoted as the discount sum of reward vectors, i.e., $G^{\pi} = \mathbb{E}_{\pi}[\sum_t \gamma^t \boldsymbol{r}(s_t, a_t)]$.

In multi-objective problems, there is typically no optimal policy that maximizes all objectives simultaneously due to the inner conflict among different goals. Instead, a set of non-dominated policies are needed in multi-objective problems. A policy π is Pareto optimal if it cannot be dominated by any other policy π', i.e., no other policy π' can achieve a better return in one objective without incurring a worse return than π in other objectives. In this work, we consider the common framework of an MOMDP with linear preferences, i.e., $f_{\boldsymbol{\omega}}(\boldsymbol{r}(s, a)) = \boldsymbol{\omega}^T \boldsymbol{r}(s, a)$ where $\boldsymbol{\omega}$ is a vector indicating the linear preference. Therefore, given a preference $\boldsymbol{\omega}$, we should train a policy π that maximizes the expected scalarized return $\boldsymbol{\omega}^T G^{\pi} = \mathbb{E}_{\pi}[\boldsymbol{\omega}^T \sum_t \gamma^t \boldsymbol{r}(s_t, a_t)]$. If all possible preferences in Ω are considered, a set of corresponding Pareto policies can be obtained to form an approximation of the Pareto front.

In the context of offline MORL, we assume no access to online interactions with the environment and offline dataset $\mathcal{D} = \{(s, a, s', \boldsymbol{r})\}$ is the only data available for training. The dataset $\mathcal{D}$ is generated by the behavior policies $\pi_{\beta}(\cdot|\boldsymbol{\omega})$ based on various preferences $\boldsymbol{\omega}$ of multiple objectives. Therefore, we define the behavior preference $\boldsymbol{\omega}_{\tau}$ as the preference of the policy that generates trajectory τ. Given that the preferences of behavior policies are either unknown or difficult to be explicitly represented in many real-world applications, in this paper, we address the offline MORL problems under the assumption that the real behavior preferences are inaccessible during the policy training process, and the goal is set to learn a policy from $\mathcal{D}$ that performs well for arbitrary target preference during deployment.

Preference-conditioned Scalarized Update To extend the single-objective methods to multi-objective settings, scalarized deep Q-learning [18] proposes to learn a vector-valued Q-value $\boldsymbol{Q} = [Q_1, Q_2, ..., Q_n]$, where Q_i is the Q value w.r.t. i^{th} objective. For a specific target preference $\boldsymbol{\omega}$, $\boldsymbol{Q}$ is updated by minimizing the loss:

$$L_{\boldsymbol{Q}} = \mathbb{E}_{(s,a,\boldsymbol{r},s') \sim D} \left[|\boldsymbol{y} - \boldsymbol{Q}(s, a)|^2 \right], \quad (1)$$

where $\boldsymbol{y} = \boldsymbol{r} + \gamma \boldsymbol{Q}(s', \arg\max_{a'}[\boldsymbol{\omega}^T \boldsymbol{Q}(s', a')])$. To respond to dynamically determined target preferences, Abels et al. [2] propose a vector-valued Q-network with outputs conditioned on the given preference, i.e., $\boldsymbol{Q}(s, a, \boldsymbol{\omega})$. This method can be extended to actor-critic framework by introducing the preference-conditioned vector critic $\boldsymbol{Q}(s, a, \boldsymbol{\omega})$ and actor $\pi(a|s, \boldsymbol{\omega})$. In this framework, for each possible target preference $\boldsymbol{\omega}$, the critic $\boldsymbol{Q}(s, a, \boldsymbol{\omega})$ is updated by minimizing the loss:

$$L_{\text{critic}} = \mathbb{E}_{(s,a,\boldsymbol{r},s') \sim D} \left[|\boldsymbol{y} - \boldsymbol{Q}(s, a, \boldsymbol{\omega})|^2 \right], \quad (2)$$

where $\boldsymbol{y} = \boldsymbol{r} + \gamma \mathbb{E}_{a' \sim \pi(\cdot|s, \boldsymbol{\omega})} \boldsymbol{Q}(s', a', \boldsymbol{\omega})$. As for the actor $\pi(a|s, \boldsymbol{\omega})$, the loss can be written as:

$$L_{\text{actor}} = -\mathbb{E}_{s \sim D} \left[\mathbb{E}_{a \sim \pi(\cdot|s, \boldsymbol{\omega})} \left[\boldsymbol{\omega}^T \boldsymbol{Q}(s, a, \boldsymbol{\omega}) \right] \right]. \quad (3)$$

Offline Policy-regularized Methods In offline RL, the errors in value estimation from out-of-distribution actions often lead to final poor performance. To address this issue, policy regularization

is applied to encourage the learned policy π to stay close to the behavior policy π_{β}, thus preventing out-of-distribution actions and avoiding performance degradation. In this paper, we focus on a common form of policy regularization, where a behavior cloning term is explicitly incorporated into the original actor loss to maximize the action-state value within the actor-critic framework:

$$L_{\text{actor}} = -\mathbb{E}_{(s,a) \sim D} \left[\mathbb{E}_{a' \sim \pi(\cdot|s)} \left[Q(s, a') \right] - \kappa L_{\text{bc}}(s, a) \right], \quad (4)$$

where $Q(s, a)$ is state-action value learned by critic, L_{bc} is a behavior cloning term and κ is a regularization weight to control the strength of the behavior cloning. The widely used behavior cloning terms in previous offline RL studies include: 1) **MSE regularization** [8] extends the TD3 [9] algorithm by adding a behavior cloning term in the form of mean square error (MSE) between the actions of the learned policy and the behavior policy in the dataset, i.e., $L_{\text{bc}}(s, a) = \mathbb{E}_{a' \sim \pi(\cdot|s)} \left[\|a - a'\|^2 \right]$; 2) **CVAE regularization** utilizes a policy class based on the conditional variational auto-encoder (CVAE) and employs the loss of evidence lower bound as the behavior cloning term, i.e. $L_{\text{bc}}(s, a) = \mathbb{E}_{z \sim q(\cdot|s,a)} \left[\log p(a|s, z) \right] - D_{\text{KL}}(q(z|s, a) \| \mathcal{N}(0, I))$, where q is an encoder and p is a decoder that is used for decision-making; and 3): **Diffusion regularization** [27] adopts the diffusion model as the learned policy and the diffusion reconstruction loss as the behavior cloning term, i.e., $\mathbb{E}_{i \sim \mathcal{U}, \epsilon \sim \mathcal{N}(0,I)} \left[\|\epsilon - \epsilon_{\theta}(\sqrt{\overline{\alpha}_i} a + \sqrt{1 - \overline{\alpha}_i} \epsilon, s, i) \|^2 \right]$, where i is the diffusion timestep, $\overline{\alpha}_i$ are pre-defined parameters and $\epsilon_{\theta}(\cdot)$ is a diffusion model serves as the learned policy.

4 POLICY-REGULARIZED OFFLINE MORL

In this section, we first introduce the *preference-inconsistent demonstration* (PrefID) problem when applying offline policy-regularized methods to multi-objective datasets collected by behavior policies with diverse preferences, and present two approaches to mitigating this problem: 1) filtering out preference-inconsistent demonstrations via approximating behavior preferences and 2) adopting regularization techniques with high policy expressiveness. Then, a natural integration of policy-regularized offline RL and the preference-conditioned scalarized update method is proposed to simultaneously learn a set of policies using a single policy network. Finally, we introduce a regularization weight adaptation method to dynamically determine the appropriate regularization weights for each target preference during deployment.
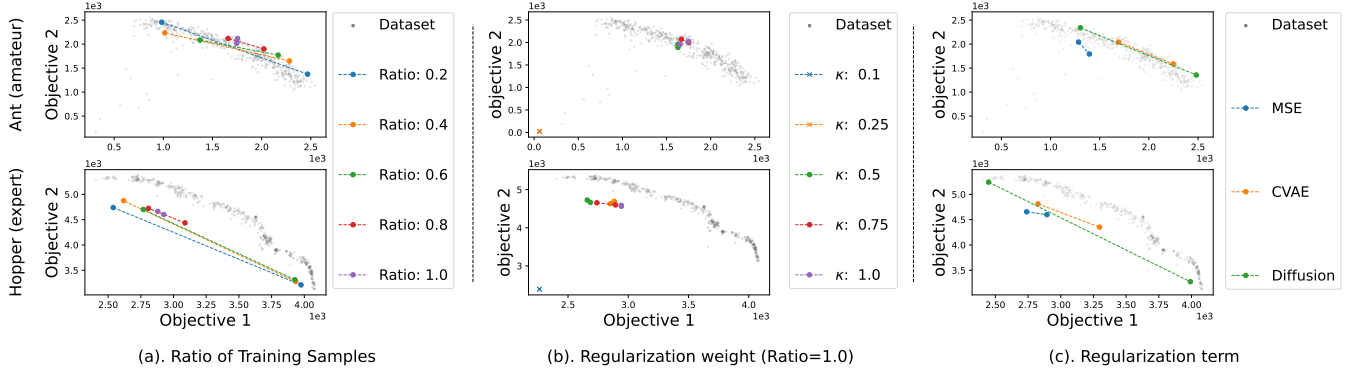
4.1 The PrefID Problem

When employing the offline policy-regularized method to train an individual policy for a specific target preference $\boldsymbol{\omega}$, we minimize the following loss:

$$L_{\text{actor}} = -\mathbb{E}_{(s,a) \sim D} \left[\mathbb{E}_{a' \sim \pi(\cdot|s)} \left[Q(s, a') \right] - \kappa L_{\text{bc}}(s, a) \right], \quad (5)$$

$$L_{\text{critic}} = \mathbb{E}_{(s,a,\boldsymbol{r},s') \sim D} \left[(y - Q(s, a'))^2 \right], \quad (6)$$

where $y = \boldsymbol{\omega}^T \boldsymbol{r} + \gamma \mathbb{E}_{a' \sim \pi(\cdot|s)} Q(s', a')$ and $Q(s, a')$ is the value function defined as the expectation of total scalar reward $\boldsymbol{\omega}^T \boldsymbol{r}$. Despite its effectiveness in solving traditional single-objective offline problems, this approach encounters a new challenge, i.e., the PrefID problem, when applied to multi-objective datasets collected by behavior policies with diverse preferences. Specifically, if a preference $\boldsymbol{\omega}'$ deviates from the target preference $\boldsymbol{\omega}$, the policy based on $\boldsymbol{\omega}'$

(a). Ratio of Training Samples

(b). Regularization weight (Ratio=1.0)

(c). Regularization term

Figure 1: The performance when applying policy-regularized methods on the Ant amateur and Hopper expert datasets within D4MORL. Two solid circles with the same color represent the expected vector return of policies trained under preferences ω_1 and ω_2 respectively (the farther these two circles extend to the high area of axes, the better performance they exhibit). Each cross signifies an aborted training due to the divergence of value estimates. The black dots represent trajectories of entire offline dataset, which reflect performance of behavior policies under various preferences.

can exhibit a conflict decision-making objective with the policy based on ω in decision-making. Therefore, trajectories generated from behavior preference ω' can become arbitrarily inferior demonstrations (i.e., preference-inconsistent demonstrations) during the policy training process under the target preference ω. These demonstrations can significantly misguide policy learning through the behavior cloning term used by offline policy-regularized methods and thus lead to performance degradation.

To study the above problem, we first introduce the sub-dataset $\mathcal{D}_{\omega}$, which is a subset of $\mathcal{D}$ that satisfies:

(1) $|\mathcal{D}_{\omega}|/|\mathcal{D}|$ = a certain ratio;
(2) $1 - \frac{\omega \cdot \omega_{\tau_i}}{||\omega|| \cdot ||\omega_{\tau_i}||} \leq 1 - \frac{\omega \cdot \omega_{\tau_j}}{||\omega|| \cdot ||\omega_{\tau_j}||}, \forall \tau_i \in \mathcal{D}_{\omega}, \tau_j \in \mathcal{D} \backslash \mathcal{D}_{\omega}$.

In other words, $\mathcal{D}_{\omega}$ contains a specific proportion of offline trajectories with behavior preferences having the smallest cosine distance to preferences ω within the entire offline dataset $\mathcal{D}$. Given that the divergence between two preferences can be effectively captured by their cosine distance under the linear preference setting, $\mathcal{D}_{\omega}$ with a smaller ratio can be considered to contain fewer preference-inconsistent demonstrations.

To further illustrate the PrefID problem and explore potential solutions, we conduct experiments on D4MORL by utilizing offline policy-regularized methods to train two policies individually, each maximizing the scalar utility under one of the two target preferences $\omega_1 = [1, 0]$ and $\omega_2 = [0, 1]$. Figure 1 (a) shows the performance of the policy trained under ω_i using TD3+BC [8] on sub-datasets $\mathcal{D}_{\omega_i}$ with various ratios $|\mathcal{D}_{\omega_i}|/|\mathcal{D}|$ for $i = 1, 2$. We observe that if the entire dataset is used for policy training (i.e., $|\mathcal{D}_{\omega_i}|/|\mathcal{D}|$=1.0), the policies learned under preferences ω_1 and ω_2 both exhibit inferior performance compared to behavior policies, while obtaining highly similar vector returns. Moreover, as preference-inconsistent samples are progressively excluded from the training dataset (i.e., $|\mathcal{D}_{\omega_i}|/|\mathcal{D}|$ decreases), the policy performance gradually improves, which demonstrates that the preference-inconsistent samples in the training dataset can hinder policy learning.

4.2 Mitigating the PrefID Problem

Based on the above results, one approach to addressing the PrefID problem is to filter out performance-inconsistent demonstrations from the training dataset based on the cosine distance between the behavior preference and the target preference. However, the real behavior preferences used for calculating cosine distance are inaccessible under our assumption. Inspired by the observation that the policy preferences and their associated expected vector return often exhibit strong correlation [4], we propose to approximate the behavior preference of trajectories by using the L1 normalization of their vector returns. Specifically, we assign each trajectory τ with an approximate behavior preference:

$$\hat{\omega}_{\tau} = R(\tau)/||R(\tau)||_1, \tag{7}$$

where $R(\tau) = \sum_t r_t$ is the total vector return of τ. Then we define the sub-dataset $\mathcal{D}(\omega)$ used in the policy training process under the target preference ω as:

$$\mathcal{D}(\omega) = \{\tau | \tau \in \mathcal{D}, 1 - \frac{\omega \cdot \hat{\omega}_{\tau}}{||\omega|| \cdot ||\hat{\omega}_{\tau}||} \leq 2\theta\}, \tag{8}$$

which indicates that the cosine distance between the approximate behavior preferences in $\mathcal{D}(\omega)$ and the target preference ω does not exceed the preset threshold 2θ. When θ is set to 1, i.e., $\mathcal{D}(\omega) = \mathcal{D}$, the entire offline dataset is employed for policy training under ω, while only the trajectories with an approximate behavior preference $\hat{\omega}_{\tau} = \omega$ are utilized when θ is set to 0. Although this approach is intuitive and simple, it can effectively mitigate the emergence of preference-inconsistent demonstrations in $\mathcal{D}(\omega)$. This is because trajectories generated by policies with inconsistent preferences typically exhibit noticeable differences in their vector returns, and thus such trajectories are scarcely included simultaneously in the same $\mathcal{D}(\omega)$ due to the similarity of $\hat{\omega}_{\tau}$ in $\mathcal{D}(\omega)$.

In addition to directly excluding performance-inconsistent demonstrations from training datasets, a parallel approach is to mitigate the adverse effect of preference-inconsistent demonstrations from behavior cloning by utilizing a smaller regularization weight. However, as depicted in Figure 1 (b), employing MSE regularization with a reduced regularization weight fails to improve policy performance, and even leads to the divergence of value estimates due to

---

Algorithm 1 Policy-regularized Offline MORL

1: **Input:** Offline dataset $\mathcal{D}$, hyperparameter θ and $\omega_{\text{bc}}^{\min}$, batch size B
2: Initialized policy π and critic $\hat{Q}$
3: **for** each interaction **do**
4:     Sample mini-batch $\mathcal{B} = \{(s_i, a_i, \boldsymbol{r}_i, s_i')\}_{i=1}^{B} \sim \mathcal{D}$
5:     Sample target preferences $\boldsymbol{\omega}_i$ from $P(\boldsymbol{\omega}|\tau_i)$ where τ_i is the trajectory containing $(s_i, a_i, \boldsymbol{r}_i, s_i')$
6:     Sample $\omega_{\text{bc},i}$ uniformly from $[\omega_{\text{bc}}^{\min}, 1]$
7:     Set $\hat{\boldsymbol{\omega}}_i = ((1 - \omega_{\text{bc},i}) \cdot \boldsymbol{\omega}, \omega_{\text{bc},i})$
8:     Update the policy by batch gradient descent on $\mathcal{B}$

$$L_{\text{actor}} = -\frac{1}{B} \sum_{i=1}^{B} \mathbb{E}_{a' \sim \pi(\cdot|s_i, \hat{\boldsymbol{\omega}}_i)} \left[\hat{\boldsymbol{\omega}}_i \hat{Q}(s_i, a', \hat{\boldsymbol{\omega}}_i) \right]$$

9:     Update the critc by batch gradient descent on $\mathcal{B}$

$$L_{\text{critic}} = \frac{1}{B} \sum_{i=1}^{B} (\boldsymbol{r}_i + \gamma \mathbb{E}_{a' \sim \pi(\cdot|s, \hat{\boldsymbol{\omega}}_i)} Q(s_i', a', \hat{\boldsymbol{\omega}}_i) - Q(s_i, a_i, \hat{\boldsymbol{\omega}}_i))^2$$

10: **end for**

---

distribution shift issues. Nevertheless, adopting regularization techniques with high policy expressiveness is an effective approach to mitigating the PrefID problem. Given that preference-inconsistent demonstrations generated from various preferences can exhibit entirely diverse behaviors, the behavior policy that is implicitly modeled by behavior cloning used in offline policy-regularized methods could demonstrate a high level of complexity, which can lead to inaccurate modeling of the behavior policy and subsequent policy degradation. This issue can be alleviated by employing regularization techniques with high policy expressiveness (e.g., CVAE and diffusion regularization) for policy learning, which has been proven effective in modeling the complex behavior policy [27]. Figure 1 (c) shows that using diffusion and CVAE regularization can indeed improve performance and the diffusion regularization method performs best among all these regularization techniques.

4.3 Responding to Arbitrary Target Preferences

To respond to arbitrary target preference during deployment, we can utilize offline policy-regularized methods to train an individual policy for each possible preference. However, this approach faces a significant challenge in terms of computational costs and storage requirements, as it requires the training of a large number of policies to cover all possible preferences. To address this challenge, we incorporate the preference-conditioned scalarized update method into policy-regularized offline RL, which enables simultaneous learning of a set of policies with a single policy network. The goal can be formulated as minimizing the loss:

$$
\begin{aligned}
L_{\text{actor}} = -\mathbb{E}_{\boldsymbol{\omega} \sim P(\Omega)} \Big[& \mathbb{E}_{(s,a) \sim D(\boldsymbol{\omega})} \\
& \big[\mathbb{E}_{a' \sim \pi(\cdot|s,\boldsymbol{\omega})} \big[\boldsymbol{\omega}^T Q(s, a', \boldsymbol{\omega}) \big] - \kappa L_{\text{bc}}(s,a) \big] \Big], \\
L_{\text{critic}} = \mathbb{E}_{\boldsymbol{\omega} \sim P(\Omega)} & \big[\mathbb{E}_{(s,a,\boldsymbol{r},s') \sim D(\boldsymbol{\omega})} \big[(\boldsymbol{y} - Q(s, a, \boldsymbol{\omega}))^2 \big] \big],
\end{aligned}
$$

$$(9)$$

where $\boldsymbol{y} = \boldsymbol{r} + \gamma \mathbb{E}_{a' \sim \pi(\cdot|s,\hat{\boldsymbol{\omega}})} Q(s', a', \boldsymbol{\omega})$, and $P(\Omega)$ is a prior distribution of $\boldsymbol{\omega}$. One challenge in the implementation of Eq. (9) is

computational complexity from reconstructing $\mathcal{D}(\boldsymbol{\omega})$ for each new $\boldsymbol{\omega}$ sampled from $P(\Omega)$ during training. Therefore, in our implementation, we adopt a sampling scheme that alters the sampling order of τ and $\boldsymbol{\omega}$, i.e., replace $\mathbb{E}_{\boldsymbol{\omega} \sim P(\boldsymbol{\omega})} \mathbb{E}_{\tau \sim D(\boldsymbol{\omega})}$ with $\mathbb{E}_{\tau \sim D} \mathbb{E}_{\boldsymbol{\omega} \sim P(\boldsymbol{\omega}|\tau)}$ where $P(\boldsymbol{\omega}|\tau) = U(\{\boldsymbol{\omega} \in \Omega | 1 - \frac{\boldsymbol{\omega} \cdot \hat{\boldsymbol{\omega}}_\tau}{||\boldsymbol{\omega}|| \cdot ||\hat{\boldsymbol{\omega}}_\tau||} \leq 2\theta\})$.

In Eq. (9), κ is the regularization weight that determines the trade-off between behavior cloning and policy improvement. An oversized κ can degenerate policy learning into mere behavior cloning, consequently impeding policy improvement, while an undersized κ can lead to the performance degradation due to severe distribution shift issues. In offline single-objective settings, an appropriate regularization weight can be determined through meticulous fine-tuning. However, the optimal regularization weight in offline MORL settings can vary across different preferences, due to the differing data coverage in the optimal trajectory space for various preferences. Therefore, the regularization weight should be dynamically adjusted to accommodate diverse preferences.

Algorithm 2 Regularization Weight Adaption

1: **Input:** Target preference $\boldsymbol{\omega}$, number of update interaction N, number of trajectories collected within each iteration K
2: Initialized a truncated Gaussian distribution $\mathcal{N}(\mu, \sigma)$
3: **for** $i = 1, 2, .., N$ **do**
4:     Sample ω_{bc} from $\mathcal{N}(\mu, \sigma)$ and let $\hat{\boldsymbol{\omega}} = ((1 - \omega_{\text{bc}}) \cdot \boldsymbol{\omega}, \omega_{\text{bc}})$
5:     Collect K trajectories via $\pi(\cdot|\hat{\boldsymbol{\omega}})$ to form a mini-batch $\mathcal{B}$
6:     Update μ and σ by minimizing:

$$-\mathbb{E}_{\omega_{\text{bc}} \sim \mathcal{N}(\mu, \sigma)} \left[\mathbb{E}_{\tau \sim \mathcal{B}} \left[\boldsymbol{\omega} R(\tau) \right] \right].$$

7: **end for**
8: Return μ as final regularization weight ω_{bc}

To this end, we propose *Regularization Weight Adaptation* to achieve adaptive adjustments of the regularization weight for diverse preferences. First, we define a behavior cloning objective with a scalar value:

$$V_{\text{bc}}^{\boldsymbol{\omega}}(s) = -\mathbb{E}_{(s',a') \sim D(\boldsymbol{\omega})|s'=s} \left[\eta L_{\text{bc}}(s', a') \right], \tag{10}$$

where η is a scale hyperparameter. Then, we introduce its corresponding preference ω_{bc} which serves as an adaptable regularization weight. Next, we incorporate this behavior cloning objective and its corresponding preference into the original MORL framework by extending the value vector and preference vector to include the behavior cloning objective:

$$\hat{\boldsymbol{\omega}} \triangleq ((1 - \omega_{\text{bc}}) \cdot \boldsymbol{\omega}, \omega_{\text{bc}}), \tag{11}$$

$$\hat{Q}(s, a, \hat{\boldsymbol{\omega}}) \triangleq \left(Q(s, a, \hat{\boldsymbol{\omega}}), V_{\text{bc}}^{\hat{\boldsymbol{\omega}}}(s) \right). \tag{12}$$

Due to the incorporation of the behavior cloning objective, both the actor and the critic should be conditioned on an argumented preference $\hat{\boldsymbol{\omega}}$, i.e., $\pi(\cdot|s, \hat{\boldsymbol{\omega}})$, $Q(s, a, \hat{\boldsymbol{\omega}})$. Afterwards, the loss of actor and critic can be reformulated as follows:

$$
\begin{aligned}
L_{\text{actor}} = -\mathbb{E}_{\hat{\boldsymbol{\omega}} \sim P(\hat{\Omega})} \left[\mathbb{E}_{s \sim \mathcal{D}(\boldsymbol{\omega})} \left[\mathbb{E}_{a \sim \pi(\cdot|s, \hat{\boldsymbol{\omega}})} \left[\hat{\boldsymbol{\omega}}^T \hat{Q}(s, a, \hat{\boldsymbol{\omega}}) \right] \right] \right], \\
L_{\text{critic}} = \mathbb{E}_{\hat{\boldsymbol{\omega}} \sim P(\hat{\Omega})} \left[\mathbb{E}_{(s,a,\boldsymbol{r},s') \sim D(\boldsymbol{\omega})} \left[(\boldsymbol{y} - Q(s, a, \hat{\boldsymbol{\omega}}))^2 \right] \right].
\end{aligned}
$$

$$(13)$$

where $\boldsymbol{y} = \boldsymbol{r} + \gamma \mathbb{E}_{a' \sim \pi(\cdot|s, \hat{\boldsymbol{\omega}})} Q(s', a', \hat{\boldsymbol{\omega}})$. The argumented preference $\hat{\boldsymbol{\omega}}$ is obtained by combining $\boldsymbol{\omega}$ sampled from $P(\boldsymbol{\omega}|\tau)$ and ω_{bc}

sampled from $[\omega_{\mathrm{bc}}^{\min}, 1]$ where $\omega_{\mathrm{bc}}^{\min} > 0$ is a hyperparameter to circumvent instances of a zero value for ω_{bc}. The main discrepancy between Eq. (9) and Eq. (13) lies in the regularization weight, which is a preset hyperparameter κ in Eq. (9), but assumed unknown preference $\hat{\omega}$ in Eq. (13).

The remaining challenge lies in determining the regularization weight (i.e., behavior cloning preference) ω_{bc} for a given target preference ω to maximize the expected scalarized return ωG^{π}. Drawing inspiration from the policy adaptation method in [31], we employ a gradient approach to efficiently infer the regularization weight ω_{bc}. To respond to a given preference ω during deployment, we sample ω_{bc} from a truncated Gaussian distribution $\mathcal{N}(\mu, \sigma)$ with learnable parameters μ and σ, and then combine ω_{bc} and ω as inputs to the policy $\pi(\cdot|\hat{\omega})$ for decision-making. During policy deployment, we can iteratively optimize the parameters μ and σ through gradient decent using a limited number of trajectory samples collected by $\pi(\cdot|\hat{\omega})$ as follows:

$$\arg\max_{\mu,\sigma} \mathbb{E}_{\omega_{\mathrm{bc}} \sim \mathcal{N}(\mu,\sigma)} \left[\mathbb{E}_{\tau \sim \pi(\cdot|\hat{\omega})} \left[\omega^{T} R(\tau) \right] \right]. \qquad (14)$$

Despite the involvement of online interactions during the inference of ω_{bc}, the associated risks are constrained by two key factors: 1) the policy $\pi(\cdot|\hat{\omega})$ is well-trained to encompass a broad range of ω_{bc}; and 2) only a limited number of samples (e.g., 30 in our experiments) are sufficient to optimize the two scalar parameters μ and σ to achieve satisfactory performance. The pseudo codes of policy training and regularization weight adaptation are presented in Algorithm 1 and 2, respectively.

5 EXPERIMENTS

In this section, we evaluate our approach on D4MORL [33], an offline MORL dataset collected by behavior policies with diverse preferences, and a multi-objective variant of D4RL [7] generated by single-objective behavior policies (i.e., the MOSB dataset). Due to space limits, we present ablation experiments on the technique of filling out the preference-inconsistent samples and *Regularization Weight Adaptation* in Appendix E.1 and E.2, respectively.

Evaluation Protocols Given the fact that the true Pareto front is often unavailable in many problems, Hypervolume (**Hv**) and Sparsity (**Sp**) are commonly used metrics for evaluating the relative quality of the approximated Pareto front of different algorithms, without requiring knowledge of the true Pareto front [1, 4, 19, 30, 33]. Hv measures the volume enclosed by the returns of policies in the approximated Pareto set P, indicating the performance increase across all objectives, while Sp measures the average return distance of two adjacent solutions, offering insights into the density of policies in the approximated Pareto set. A larger Hv and a smaller Sp indicate a better performance, with Hv typically serving as the primary comparison metric and Sp as the secondary metric. This is because optimizing policies primarily based on the Sp metric can lead to a very narrow Pareto front and a low Hv. More details about metrics of Hv and Sp can be found in Appendix C. In all experiments, we evaluate the methods using 101 equidistant preference point pairs (i.e., [0.00, 1.00], [0.01, 0.99], ..., [1.00, 0.00]) and average the vector returns of 5 trajectories sampled from randomly initialized environments for each preference point. Hv and Sp metrics are calculated based on these averaged vector returns.

We perform 3 runs with various seeds for each algorithm and report the average. Except for Hv and Sp metrics, Appendix E.3 provides additional performance comparisons using the expected utility (EU) metric.

Comparative Algorithms We present the results of our proposed policy-regularized offline MORL method with MSE regularization, CVAE regularization and Diffusion regularization. Our primary comparative baseline is the state-of-the-art method **PEDA** [33], which extends the DT [5] and RvS [6] algorithms to MORL settings, referred as **MODT (P)** and **MORvS (P)**, respectively, by incorporating preference information and vector-valued returns into corresponding model architectures. In addition, we also provide the results of the following two methods: 1) preference-conditioned behavior cloning, denoted as **BC (P)**, where the policy receives not only the state but also its corresponding behavior preference as input and generates predicted actions; and 2) **MO-CQL**, where the Conservative Q-Learning (CQL) [15] is incorporated into the naive scalarized-updated actor-critic to cope with the distribution shift problem. Note that algorithms MODT (P), MORvS (P) and BC (P) require access to the preference information of behavior policies, whereas our approach and MO-CQL have no such an assumption. More implementation details of algorithms can be found in Appendix A and B.

Regularization Weight Adaptation Phase Before evaluation, we conduct regularization weight adaptation to infer the appropriate regularization weight for each evaluated preference in our approach. In this phase, we employ gradient descent to update the parameters μ and σ of regularization weight distribution $\mathcal{N}(\mu, \sigma)$. Specifically, we conduct a total of 3 gradient descent iterations, with each iteration using 10 trajectories collected by $\pi(\cdot|\hat{\omega})$ from the environment. Finally, we utilize the updated μ as the final ω_{bc} for decision-making of policy $\pi(\cdot|\hat{\omega})$ during evaluation.

5.1 Experiments on D4MORL

Environments and Datasets D4MORL [33] is a dataset designed for offline MORL, which contains millions of trajectories collected by a set of multi-objective behavior policies in various multi-objective MuJoCo environments. D4MORL contains two types of datasets with different behavior policies: 1) **Amature Dataset**, a combination of expert demonstrations and suboptimal data generated via randomly perturbed expert policies; and 2) **Expert Dataset**, consisting entirely of trajectories collected by expert policies. In addition, the datasets we utilized for comparison consist of trajectories with behavior preferences uniformly sampled from a wide range of preference space. These datasets also serve as the primary dataset for experimentation in PEDA [33]. More details about D4MORL are provided in Appendix D.

Results and Analysis Table 1 provides the performance of different methods on the Amateur dataset and Expert dataset. First, we observe that three regularization techniques yield similar performance, but Diffusion regularization demonstrates an overall advantage compared to the others, which is consistent with the discussions about policy regularization techniques in Section 4.2. On amateur dataset, our approach consistently performs the best

Table 1: Results on D4MORL Amateur and Expert datasets. B indicates the performance of behavior policies from PEDA.

	Env	metric	B	BC (P)	MO-CQL	MODT (P) †	MORvS (P) †	Diffusion	CVAE	MSE
Amateur	Ant	HV (10^6)	5.61	$3.39_{\pm.81}$	$\mathbf{6.28}_{\pm.00}$	$5.63_{\pm.03}$	$5.81_{\pm.03}$	$\mathbf{6.15}_{\pm.01}$	$5.48_{\pm.01}$	$5.49_{\pm.02}$
		SP (10^4)	-	$2.44_{\pm2.27}$	$1.37_{\pm.02}$	$1.93_{\pm.27}$	$0.76_{\pm.08}$	$0.95_{\pm.03}$	$1.35_{\pm.38}$	$1.81_{\pm.19}$
	Swimmer	HV (10^4)	2.11	$2.79_{\pm.03}$	$\mathbf{2.94}_{\pm.03}$	$0.61_{\pm.01}$	$2.79_{\pm.05}$	$\mathbf{3.11}_{\pm.03}$	$\mathbf{3.14}_{\pm.01}$	$\mathbf{3.02}_{\pm.03}$
		SP (10^1)	-	$1.44_{\pm.09}$	$11.75_{\pm.78}$	$1.80_{\pm.08}$	$1.60_{\pm.40}$	$3.13_{\pm.42}$	$3.86_{\pm.62}$	$3.20_{\pm.62}$
	HalfCheetah	HV (10^6)	5.68	$5.42_{\pm.05}$	$5.59_{\pm.01}$	$5.63_{\pm.05}$	$5.74_{\pm.00}$	$5.72_{\pm.02}$	$5.74_{\pm.00}$	$\mathbf{5.75}_{\pm.01}$
		SP (10^4)	-	$1.21_{\pm.58}$	$4.11_{\pm.16}$	$0.46_{\pm.05}$	$0.34_{\pm.06}$	$0.53_{\pm.21}$	$0.51_{\pm.08}$	$0.57_{\pm.09}$
	Hopper	HV (10^7)	1.97	$1.60_{\pm.03}$	$\mathbf{1.84}_{\pm.00}$	$1.56_{\pm.19}$	$1.68_{\pm.07}$	$\mathbf{1.88}_{\pm.03}$	$1.75_{\pm.05}$	$1.83_{\pm.07}$
		SP (10^5)	-	$0.21_{\pm.03}$	$4.57_{\pm3.88}$	$2.13_{\pm2.15}$	$0.47_{\pm.16}$	$0.09_{\pm.03}$	$0.17_{\pm.03}$	$0.42_{\pm.15}$
	Walker2d	HV (10^6)	4.99	$3.67_{\pm.09}$	$4.15_{\pm.03}$	$3.17_{\pm.12}$	$\mathbf{4.93}_{\pm.05}$	$\mathbf{4.97}_{\pm.02}$	$4.94_{\pm.02}$	$\mathbf{4.99}_{\pm.07}$
		SP (10^4)	-	$18.67_{\pm16.26}$	$33.14_{\pm15.28}$	$46.88_{\pm11.87}$	$0.84_{\pm.12}$	$1.06_{\pm.17}$	$0.99_{\pm.15}$	$0.91_{\pm.27}$
Expert	Ant	HV (10^6)	6.32	$4.42_{\pm.57}$	$6.12_{\pm.01}$	$5.89_{\pm.03}$	$6.08_{\pm.04}$	$\mathbf{6.16}_{\pm.02}$	$5.72_{\pm.05}$	$5.75_{\pm.03}$
		SP (10^4)	-	$4.08_{\pm2.78}$	$1.55_{\pm.21}$	$2.33_{\pm.65}$	$1.15_{\pm.12}$	$0.71_{\pm.08}$	$2.04_{\pm1.19}$	$1.51_{\pm.13}$
	Swimmer	HV (10^4)	3.25	$\mathbf{3.18}_{\pm.01}$	$2.26_{\pm.04}$	$\mathbf{3.22}_{\pm.01}$	$\mathbf{3.22}_{\pm.01}$	$3.19_{\pm.01}$	$3.14_{\pm.03}$	$3.15_{\pm.02}$
		SP (10^1)	-	$4.00_{\pm.24}$	$0.02_{\pm.01}$	$2.96_{\pm.17}$	$4.38_{\pm.26}$	$5.44_{\pm1.25}$	$11.26_{\pm4.24}$	$10.60_{\pm4.41}$
	HalfCheetah	HV (10^6)	5.79	$\mathbf{5.60}_{\pm.03}$	$5.65_{\pm.02}$	$5.65_{\pm.00}$	$5.75_{\pm.00}$	$\mathbf{5.76}_{\pm.00}$	$5.75_{\pm.00}$	$\mathbf{5.76}_{\pm.00}$
		SP (10^4)	-	$0.49_{\pm.18}$	$1.05_{\pm.12}$	$0.77_{\pm.06}$	$0.68_{\pm.10}$	$0.34_{\pm.04}$	$0.64_{\pm.09}$	$0.58_{\pm.01}$
	Hopper	HV (10^7)	2.09	$0.88_{\pm.16}$	$1.71_{\pm.06}$	$1.79_{\pm.13}$	$\mathbf{1.97}_{\pm.01}$	$1.67_{\pm.02}$	$1.76_{\pm.13}$	$\mathbf{1.98}_{\pm.03}$
		SP (10^5)	-	$23.62_{\pm11.30}$	$3.47_{\pm.75}$	$1.21_{\pm.40}$	$2.05_{\pm1.75}$	$6.20_{\pm3.85}$	$2.28_{\pm1.37}$	$0.83_{\pm.16}$
	Walker2d	HV (10^6)	5.21	$2.12_{\pm.55}$	$2.55_{\pm.03}$	$5.00_{\pm.08}$	$\mathbf{5.05}_{\pm.07}$	$4.98_{\pm.06}$	$4.93_{\pm.11}$	$4.91_{\pm.04}$
		SP (10^4)	-	$34.77_{\pm23.21}$	$0.00_{\pm.00}$	$2.12_{\pm.60}$	$1.86_{\pm.51}$	$4.14_{\pm1.39}$	$3.67_{\pm1.71}$	$3.71_{\pm1.38}$

† Due to slight differences in evaluation protocols, the presented results of PEDA exhibit minor discrepancies compared to the original paper.
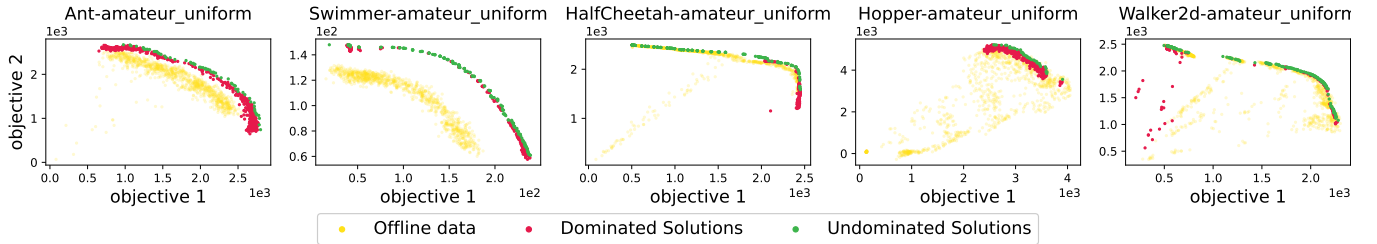
Figure 2: Approximate Pareto fronts learned by our approach with Diffusion regularization on the Amateur datasets.

for the primary metric Hv, especially in the Ant, Simmer and Hopper environments, where our approach achieves significant improvement compared to the state-of-the-art baseline PEDA. This is because, in contrast to the supervised offline models utilized in PEDA, which strongly rely on high-quality demonstrations for decision-making, our approach utilizes explicit policy improvement through the guidance of value estimates rather than solely following the expert demonstrations, making it more robust to suboptimal demonstrations. In the Expert dataset, where the data are entirely collected by expert behavior policies, our approach still exhibits competitive performance compared to PEDA. Finally, we observe that our approach performs slightly worse than PEDA in the secondary metric, Sp. A reasonable explanation is that, given the fixed number of sampling preferences, the improvement in Hv, signifying the expansion of the Pareto front, can inevitably lead to increased distances between solutions, consequently leading to an increase in Sp. The approximate Pareto fronts are partially shown in Figure 2 and fully displayed in Appendix E.5, which demonstrate the capability of our approach in learning a broad and dense approximate Pareto front.

5.2 Experiments on MOSB

Environments and Datasets   In many real-world scenarios, the offline data are collected by single-objective policies or policies with a fixed preference over multiple objectives. To explore the feasibility of utilizing these datasets for MORL, we introduce a new offline multi-objective dataset, namely MOSB, which is built upon the D4RL [7] Mujoco dataset. The main differences between the MOSB dataset and the D4MORL dataset include: 1) behavior preferences, where the behavior preferences in the MOSB dataset are identical and inaccessible, whereas those in D4MORL are sampled from a wide range of preference spaces; 2) behavior policy types, where MOSB is collected from diverse sources, including the partially-trained policy (medium), the replay buffer (medium-replay), and a combination of the partially-trained policy and expert policy (medium-expert); and 3) data quantity, where MOSB contains fewer transition data compared to D4MORL ($0.2 \sim 2$M vs. 25M). More details about the MOSB dataset are provided in Appendix D.

Results and Analysis   The results of different algorithms on the MOSB dataset are presented in Table 2. Our approaches equipped with different regularization techniques consistently outperform other methods. Compared to the vector return of the offline data

Table 2: Results on MOSB dataset. D indicate the Hv and Sp of the vector return front of offline data.

	Env	Metric	D	BC(P)[†]	MORvS(p)[†]	MO-CQL	Diffusion	CVAE	MSE
medium-replay	HalfCheetah	Hv (10^7)	1.21	$1.56_{\pm.07}$	$1.71_{\pm.02}$	$1.59_{\pm.07}$	$\mathbf{3.25}_{\pm.04}$	$3.08_{\pm.10}$	$2.93_{\pm.07}$
		Sp (10^5)	2.74	$4.52_{\pm.46}$	$1.18_{\pm.36}$	$9.23_{\pm1.69}$	$3.32_{\pm.75}$	$3.55_{\pm1.69}$	$3.75_{\pm1.17}$
	Hopper	Hv (10^7)	1.22	$0.64_{\pm.15}$	$0.53_{\pm.03}$	$\mathbf{1.37}_{\pm.07}$	$1.46_{\pm.01}$	$1.06_{\pm.08}$	$\mathbf{1.27}_{\pm.06}$
		Sp (10^5)	14.87	$7.36_{\pm4.89}$	$13.01_{\pm4.98}$	$0.59_{\pm.55}$	$0.48_{\pm.36}$	$2.76_{\pm1.89}$	$0.93_{\pm.38}$
	Walker2d	Hv (10^7)	1.02	$0.42_{\pm.27}$	$0.96_{\pm.03}$	$0.37_{\pm.06}$	$1.46_{\pm.06}$	$0.95_{\pm.04}$	$1.17_{\pm.10}$
		Sp (10^4)	9.71	$32.37_{\pm3.84}$	$234.56_{\pm74.90}$	$190.24_{\pm1.04}$	$2.24_{\pm1.26}$	$0.08_{\pm.00}$	$1.04_{\pm.63}$
medium	HalfCheetah	Hv (10^7)	1.32	$1.17_{\pm.07}$	$1.17_{\pm.08}$	$1.19_{\pm.08}$	$\mathbf{3.56}_{\pm.11}$	$3.18_{\pm.37}$	$\mathbf{3.39}_{\pm.03}$
		Sp (10^5)	2.29	$0.29_{\pm.22}$	$0.08_{\pm.05}$	$58.81_{\pm1.80}$	$2.08_{\pm.77}$	$2.19_{\pm.79}$	$1.00_{\pm.11}$
	Hopper	Hv (10^7)	1.19	$0.43_{\pm.04}$	$0.52_{\pm.04}$	$\mathbf{1.28}_{\pm.02}$	$1.37_{\pm.02}$	$1.42_{\pm.03}$	$1.25_{\pm.03}$
		Sp (10^3)	0.00	$6.87_{\pm2.72}$	$14.33_{\pm2.27}$	$29.22_{\pm1.96}$	$0.20_{\pm.05}$	$2.97_{\pm1.33}$	$0.78_{\pm1.10}$
	Walker2d	Hv (10^7)	0.95	$0.64_{\pm.41}$	$0.97_{\pm.02}$	$0.80_{\pm.00}$	$1.27_{\pm.03}$	$\mathbf{1.66}_{\pm.05}$	$1.17_{\pm.02}$
		Sp (10^4)	18.82	$3.87_{\pm2.29}$	$13.65_{\pm2.68}$	$1.06_{\pm.95}$	$0.48_{\pm.07}$	$0.49_{\pm.03}$	$0.29_{\pm.07}$
medium-expert	HalfCheetah	Hv (10^7)	2.70	$1.09_{\pm.13}$	$2.83_{\pm.22}$	$2.88_{\pm.16}$	$3.39_{\pm.15}$	$2.00_{\pm.66}$	$\mathbf{3.83}_{\pm.29}$
		Sp (10^6)	1.73	$0.26_{\pm.22}$	$1.73_{\pm1.31}$	$0.05_{\pm.02}$	$2.02_{\pm.88}$	$1.08_{\pm.98}$	$0.75_{\pm.26}$
	Hopper	Hv (10^7)	$\mathbf{1.48}$	$0.93_{\pm.45}$	$1.54_{\pm.01}$	$1.54_{\pm.01}$	$\mathbf{1.56}_{\pm.01}$	$1.52_{\pm.01}$	$1.48_{\pm.01}$
		Sp (10^2)	15.97	$9.70_{\pm1.37}$	$3.87_{\pm1.89}$	$8.04_{\pm1.50}$	$2.05_{\pm.67}$	$1.88_{\pm1.15}$	$0.72_{\pm.44}$
	Walker2d	Hv (10^7)	1.15	$1.02_{\pm.06}$	$\mathbf{1.29}_{\pm.00}$	$1.08_{\pm.02}$	$1.46_{\pm.01}$	$0.93_{\pm.81}$	$1.34_{\pm.00}$
		Sp (10^5)	9.76	$0.44_{\pm.16}$	$3.74_{\pm1.53}$	$1.27_{\pm1.26}$	$0.01_{\pm.00}$	$0.01_{\pm.01}$	$0.00_{\pm.00}$

[†] Since these algorithms require real behavior preferences, which are unavailable in MOSB, we use the approximate preferences in Eq. (7) as substitutes for policy training.
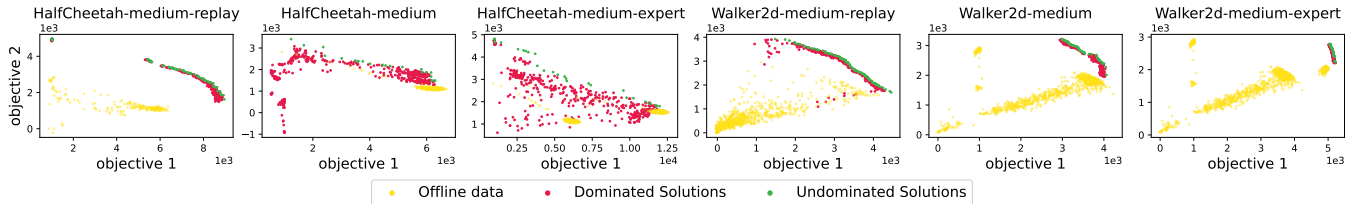
Figure 3: Vector returns of trajectories in MOSB dataset and approximate Pareto fronts learned by Diffusion regularization.

(i.e., D in Table 2), our approach shows significant improvements in both Hv and Sp metrics. These improvements indicate that our approach has learned a broadly expanded approximate Pareto front with a dense policy set, which is consistent with the results in Figure 3. In contrast, BC (P) and PEDA rely on accurate behavior preference data, leading to their poorer performance on MOSB.

We present vector returns of trajectories in MOSB and the approximate Pareto front learned from our approach on HalfCheetah and Walker2d in Figure 3, and a comprehensive display of other environments is provided in Appendix E.5. Due to the fixed preferences of behavior policies, the vector returns distribution of trajectories in the offline dataset is extremely unbalanced. There are a greater number of trajectories with high returns on objective 1, while fewer trajectories exhibit high returns on objective 2. However, our approach can still achieve a broad approximate Pareto front with a dense policy set, which demonstrates the feasibility of learning multi-objective policies from the dataset collected by single-preference behavior policies. Moreover, our obtained approximate Pareto front significantly expands outward compared to the front composed of vector returns in the dataset, which indicates that our approach can achieve substantial policy improvement rather than simply imitate behavior policies.

6 CONCLUSION

In this paper, we make a first attempt to employ offline policy-regularized methods to tackle the offline MORL problems. We provide two intuitive yet effective solutions to mitigating the PrefID problem encountered in such problems. In addition, we introduce the preference-conditioned scalarized update method to policy-regularized offline RL, in order to simultaneously learn a set of policies using a single policy network. Moreover, a novel method *Regularization Weight Adaptation* is proposed to dynamically determine the appropriate regularization weights for each target preference during deployment. Empirically, the experiments conducted on various multi-objective datasets demonstrate that, compared with the state-of-the-art offline MORL algorithms, our approach can achieve competitive or superior performance without requiring preference information of behavior policies.

REFERENCES

[1] Abbas Abdolmaleki, Sandy Huang, Leonard Hasenclever, Michael Neunert, Francis Song, Martina Zambelli, Murilo Martins, Nicolas Heess, Raia Hadsell, and Martin Riedmiller. 2020. A distributional view on multi-objective policy optimization. In *International conference on machine learning*. PMLR, 11–22.

[2] Axel Abels, Diederik Roijers, Tom Lenaerts, Ann Nowé, and Denis Steckelmacher. 2019. Dynamic weights in multi-objective deep reinforcement learning. In *International conference on machine learning*. PMLR, 11–20.

[3] Mridul Agarwal, Vaneet Aggarwal, and Tian Lan. 2022. Multi-objective reinforcement learning with non-linear scalarization. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*. 9–17.

[4] Toygun Basaklar, Suat Gumussoy, and Umit Y Ogras. 2022. Pd-morl: Preference-driven multi-objective reinforcement learning algorithm. *arXiv preprint arXiv:2208.07914* (2022).

[5] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems* 34 (2021), 15084–15097.

[6] Scott Emmons, Benjamin Eysenbach, Ilya Kostrikov, and Sergey Levine. 2021. Rvs: What is essential for offline rl via supervised learning? *arXiv preprint arXiv:2112.10751* (2021).

[7] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. 2020. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219* (2020).

[8] Scott Fujimoto and Shixiang Shane Gu. 2021. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems* 34 (2021), 20132–20145.

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

[10] Scott Fujimoto, David Meger, and Doina Precup. 2019. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*. PMLR, 2052–2062.

[11] Hisashi Handa. 2009. Solving multi-objective reinforcement learning problems by EDA-RL-acquisition of various strategies. In *2009 ninth international conference on intelligent systems design and applications*. IEEE, 426–431.

[12] Josiah P Hanna, Siddharth Desai, Haresh Karnan, Garrett Warnell, and Peter Stone. 2021. Grounded action transformation for sim-to-real reinforcement learning. *Machine Learning* 110, 9 (2021), 2469–2499.

[13] Elia Kaufmann, Leonard Bauersfeld, Antonio Loquercio, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. 2023. Champion-level drone racing using deep reinforcement learning. *Nature* 620, 7976 (2023), 982–987.

[14] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. 2019. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems* 32 (2019).

[15] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems* 33 (2020), 1179–1191.

[16] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643* (2020).

[17] Chunming Liu, Xin Xu, and Dewen Hu. 2014. Multiobjective reinforcement learning: A comprehensive overview. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45, 3 (2014), 385–398.

[18] Hossam Mossalam, Yannis M Assael, Diederik M Roijers, and Shimon Whiteson. 2016. Multi-objective deep reinforcement learning. *arXiv preprint arXiv:1610.02707* (2016).

[19] Mathieu Reymond, Eugenio Bargiacchi, and Ann Nowé. 2022. Pareto Conditioned Networks. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*. 1110–1118.

[20] Diederik M Roijers, Shimon Whiteson, and Frans A Oliehoek. 2014. Linear support for multi-objective coordination graphs. In *AAMAS'14: PROCEEDINGS OF THE 2014 INTERNATIONAL CONFERENCE ON AUTONOMOUS AGENTS & MULTIAGENT SYSTEMS*, Vol. 2. IFAAMAS/ACM, 1297–1304.

[21] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement Learning: An Introduction*. MIT press.

[22] Philip S Thomas, Bruno Castro da Silva, Andrew G Barto, Stephen Giguere, Yuriy Brun, and Emma Brunskill. 2019. Preventing undesirable behavior of intelligent machines. *Science* 366, 6468 (2019), 999–1004.

[23] Philip S Thomas, Joelle Pineau, Romain Laroche, et al. 2021. Multi-objective spibb: Seldonian offline policy improvement with safety constraints in finite mdps. *Advances in Neural Information Processing Systems* 34 (2021), 2004–2017.

[24] Peter Vamplew, Richard Dazeley, Adam Berry, Rustam Issabekov, and Evan Dekker. 2011. Empirical evaluation methods for multiobjective reinforcement learning algorithms. *Machine learning* 84 (2011), 51–80.

[25] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575, 7782 (2019), 350–354.

[26] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291* (2023).

[27] Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. 2022. Diffusion policies as an expressive policy class for offline reinforcement learning. *arXiv preprint arXiv:2208.06193* (2022).

[28] Runzhe Wu, Yufeng Zhang, Zhuoran Yang, and Zhaoran Wang. 2021. Offline constrained multi-objective reinforcement learning via pessimistic dual value iteration. *Advances in Neural Information Processing Systems* 34 (2021), 25439–25451.

[29] Yifan Wu, George Tucker, and Ofir Nachum. 2019. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361* (2019).

[30] Jie Xu, Yunsheng Tian, Pingchuan Ma, Daniela Rus, Shinjiro Sueda, and Wojciech Matusik. 2020. Prediction-guided multi-objective reinforcement learning for continuous robot control. In *International conference on machine learning*. PMLR, 10607–10616.

[31] Runzhe Yang, Xingyuan Sun, and Karthik Narasimhan. 2019. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. *Advances in neural information processing systems* 32 (2019).

[32] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. 2018. DRN: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 World Wide Web Conference*. 167–176.

[33] Baiting Zhu, Meihua Dang, and Aditya Grover. 2023. Scaling pareto-efficient decision making via offline multi-objective rl. *arXiv preprint arXiv:2305.00567* (2023).