Improving Generalization in Offline Reinforcement Learning via Adversarial Data Splitting

Da Wang [1] **Lin Li** [1] **Wei Wei** [1] **Qixian Yu** [1] **Jianye Hao** [2] **Jiye Liang** [1]

Abstract

Offline Reinforcement Learning (RL) commonly suffers from the out-of-distribution (OOD) overestimation issue due to the distribution shift. Prior work gradually shifts their focus from suppressing OOD overestimation to avoiding overly conservative learning from suboptimal behavior policies to improve generalization. However, most approaches explicitly delimit boundaries for OOD actions based on the support in the dataset, which can potentially impede the data near these boundaries from acquiring realistic estimates. This paper investigates how to loosen the rigid demarcation of OOD boundaries, adaptively extracting knowledge from empirical data to implicitly improve the model's generalization to nearby unseen data. We introduce an adversarial data splitting (ADS) framework that enforces the model to generalize the distribution shifts simulated from the train/validation subsets splitting of the dataset. Specifically, ADS is modeled as a min-max optimization problem inspired by meta-learning and solved by iterating over the following two steps. First, we train the model on the train-subset to minimize its loss on the validation-subset. Then, we adversarially generate the "hardest" train/validation subsets with the maximum distribution shift, making the model incapable of generalization at that splitting. We derive a generalization error bound for theoretically understanding ADS and verify the effectiveness with extensive experiments. Code is available at `https://github.com/DkING-lv6/ADS`.

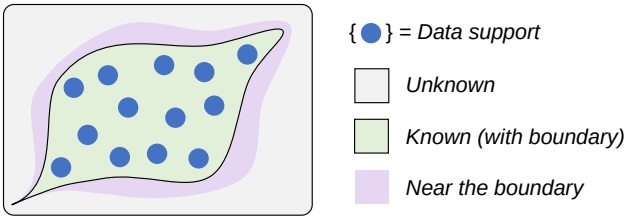

Figure 1: Abstract representation of our work. We aim to train the model capable of effectively generalizing the data near the boundary.

1. Introduction

Offline Reinforcement Learning (RL) (Levine et al., 2020) aims to learn a policy from a static dataset collected by unknown behavior policies. It saves resources and reduces risk by eliminating the need for environmental interaction during training, which has attracted significant interest from the research community. One of the fundamental challenges of offline RL is the distribution shift of state-action visitation frequency between the learned policy and the behavior policy. The inability to continuously correct the current policy's Q-value estimation for state-action pairs through exploration leads to a severe overestimation of out-of-distribution (OOD) actions (Fujimoto et al., 2019). This issue is further exacerbated through bootstrapping (Kumar et al., 2019). To tackle this issue, prior approaches deal with OOD actions based on conservative principles. The associated techniques can broadly be categorized as learning underestimated or conservative values (Kumar et al., 2020; Kostrikov et al., 2021; Ma et al., 2021; Wang et al., 2022), constraining policies (Fakoor et al., 2021; Fujimoto & Gu, 2021; Wu et al., 2022; Li et al., 2023), and uncertainty estimating (An et al., 2021; Bai et al., 2022; Wu et al., 2021).

Unfortunately, overly pessimistic conservatism hinders possible generalization and leads to the limited performance of the learned policy, especially when the behavior policy in the dataset used for constraint could be suboptimal. Existing research mitigates conservatism by incorporating policy regularization techniques, such as support constraint (Lyu et al., 2022; Mao et al., 2023) or nearest neighbor restriction (Ran et al., 2023). In practice, considering the continuous state-action space, the trained model (i.e., the neural

*Equal contribution [1]Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan, China. [2]College of Intelligence and Computing, Tianjin University, Tianjin, China. Correspondence to: Wei Wei <weiwei@sxu.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

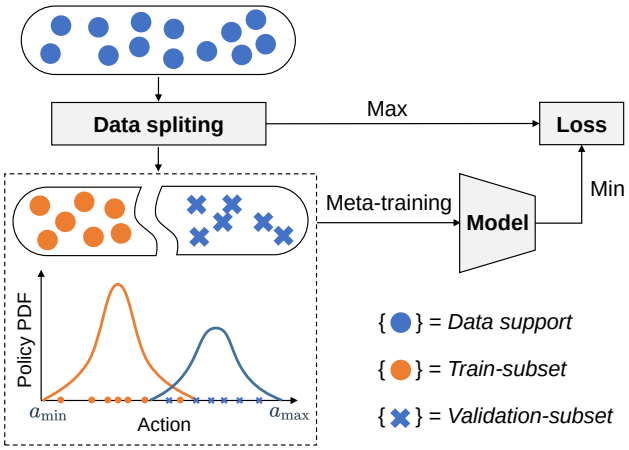

Figure 2: ADS framework for offline RL. We split the offline dataset into the train/validation subsets. The model trained on the train-subset will minimize the loss on the validation-subset, while the data splitting operation will maximize the generalization loss of the model on the validation-subset.

network) should be capable of extracting knowledge from the dataset and achieving generalization to nearby unknown states and actions. The above work, however, explicitly delimits boundaries for OOD actions, which can potentially impede the data near these boundaries from acquiring realistic estimates – actions that exceed the boundary are assigned pseudo Q-values, or they might be inaccurately estimated due to the inherent limitations in the model's generalization capabilities.

In this study, we explore methods to loosen the rigid demarcation of OOD boundaries and adaptively extract knowledge from empirical data, thus implicitly enhancing the model's generalization capabilities for data near the boundaries. The boundary is abstract, and the discussion surrounding it revolves around the extent of the distribution shift (Figure 1). Inspired by some meta-learning-based works (Qiao et al., 2020; Volpi et al., 2018; Gu et al., 2024), we innovatively introduce an adversarial data splitting (ADS) framework for offline RL. The fundamental idea is to simulate distribution shifts by splitting the dataset into train/validation subsets and then training the model capable of effectively generalizing across all possible train/validation splitting. This process enables successful generalization for scenarios encountered during testing, where distribution shifts are simulated in the splitting. However, training the model on all possible train/validation subsets is infeasible. In practice, we unify train/validation splitting and meta-learning into a min-max problem and implement ADS by iterating the two steps (Figure 2): (1) we utilize the adversarial idea to inversely generate the "hardest" train/validation subset with the maximum distribution shift; (2) we adopt a meta-learning approach to learn a generalizable model to minimize the distribution shift in the worst-case splitting.

Theoretical analysis and extensive experiments demonstrate the effectiveness of ADS in improving generalization. We apply ADS to related widely-used baselines, with the best performance surpassing some state-of-the-art offline algorithms.

In summary, our contribution is three-fold:

- We introduce an adversarial data splitting (ADS) framework for offline RL, which loosens the rigid demarcation of OOD boundaries and improves the model's generalization to nearby unseen data.
- We derive a generalization error bound based on a meta-learning framework for offline RL and analyze the effectiveness of ADS for improving generalization.
- We apply ADS to existing widely-used algorithms, significantly enhancing their performance and competitiveness. Moreover, we show that generalizing OOD boundary data with ADS performs better than assigning them pseudo Q-values.

2. Preliminaries

2.1. Markov Decision Processes (MDP)

We consider an infinite-horizon, discounted MDP denoted as $(\mathcal{S}, \mathcal{A}, P, r, \gamma, p_0)$, where $\mathcal{S}$ and $\mathcal{A}$ represent finite state and action spaces. $P(s'|s,a) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto [0,1]$ and $r(s,a) : \mathcal{S} \times \mathcal{A} \mapsto [0, R_{\max}]$ are the transition and reward function. $\gamma \in (0,1)$ is the discount factor and $p_0(s)$ is the distribution of the initial state (Sutton & Barto, 2018). The goal of RL is to find an optimal policy $\pi(\cdot|s)$ that maximizes the expected cumulative discounted reward $J(\pi) := \mathbb{E}_\pi[\Sigma_{t=0}^{\infty}\gamma^t r(s_t,a_t)]$, where the expectation is over trajectories sampled from $s_0 \sim \rho_0, a_t \sim \pi(\cdot|s_t)$, and $s_{t+1} \sim P(\cdot|s_t,a_t)$ for $t \geq 0$. The standard definition of the state value function is defined as $V^\pi(s) := \mathbb{E}_{a\sim\pi(\cdot|s)}[Q^\pi(s,a)]$, and its corresponding state-action value function, or Q-function as $Q^\pi(s,a) := \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t) \mid s_0 = s, a_0 = a\right]$. It is well known that the optimal policy $\pi^\star$ satisfies $\pi^\star(s) = \pi_{Q^\star}(s) := \arg\max_{a\in\mathcal{A}} Q^\star(s,a)$, and $Q^\star(s,a)$ obtained by the *Bellman equation* $Q^\star(s,a) = \mathcal{T}Q^\star(s,a)$, where $\mathcal{T} : \mathbb{R}^{\mathcal{S}\times\mathcal{A}} \to \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ is the *Bellman update operator*: $\forall f \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$,

$$(\mathcal{T}f)(s,a) := r(s,a) + \gamma\mathbb{E}_{s'\sim P(\cdot|s,a)}[V_f(s')], \quad (1)$$

where $V_f(s') := \max_{a'\in\mathcal{A}} f(s',a')$.

2.2. Offline RL and Bellman Error Minimization

In offline RL setting, the agent does not have direct access to the MDP. It learns a policy from a static dataset $\mathcal{D} = \{(s,a,r,s')\}$, which previously collected from unknown behavior policy $\mu(\cdot|s)$. In the training process, for

simplicity, we assume that (s, a) is generated $i.i.d.$ from the data distribution μ, and we would like to find $f \in \mathcal{F}$ ($\mathcal{F} \subset (\mathcal{S} \times \mathcal{A} \to [0, V_{\max}]), V_{\max} := R_{\max}/(1-\gamma)$) that approximates $Q^\star$ and outputs the greedy policy π_f. The objective is to minimize $V^\star(s) - V^{\pi_f}(s)$, however, this gap is highly nonsmooth in f, and a popular approach is to use a surrogate loss – the Bellman error.

Definition 2.1. (Bellman error). Under data distribution μ, we define the Bellman error of function $f \in \mathcal{F}$ as:

$$\mathcal{E}(f) := \|f - \mathcal{T}f\|_{2,\mu}^2. \tag{2}$$

Assume that μ is a distribution supported on the entire $\mathcal{S} \times \mathcal{A}$, then $\|f - \mathcal{T}f\|_{2,\mu}^2 = 0$ would guarantee that $f = Q^\star$. However, the knowledge of transition dynamics is unknown in the learning setting (recall Equation (1)), a natural choice is considering an empirical version of $\mathcal{E}(f)$ computed from samples (Chen & Jiang, 2019; Duan et al., 2021):

$$\mathcal{L}_{\mathcal{D}}(f) := \frac{1}{|\mathcal{D}|} \sum_{(s,a,r,s') \in \mathcal{D}} \left(f(s,a) - r - \gamma V_f(s')\right)^2. \tag{3}$$

That is, the Bellman error can be approximated as $\mathcal{E}(f) \approx \mathcal{L}_{\mathcal{D}}(f)$[1].

3. Our Method

In this section, we first model the generalization task of offline RL as a learning problem and propose an adversarial data splitting (ADS) framework to compromise the learning problem into a min-max problem (Section 3.1). Then, we discuss the optimization of the min-max problem and propose a practical implementation (Section 3.2). Finally, we theoretically derive a generalization error bound for understanding our method and then analyze it (Section 3.3).

3.1. Adversarial Data Splitting

Offline RL algorithms strive to address the challenge of distribution shift between the behavior policy μ and the learned policy π. Since the actions $a' \sim \pi(\cdot|s')$ utilized during the Bellman backup might lie outside the support of μ, the estimation of values on a', which are rarely corrected, could lead to extrapolation errors (Fujimoto et al., 2019). This situation could be further exacerbated by bootstrapping (Kumar et al., 2019). It has been observed that the focus of research has gradually shifted from suppressing OOD overestimation to avoiding overly conservative learning from suboptimal behavior policies to improve generalization. However, existing work explicitly delimits boundaries for OOD actions. It could impede the data near these boundaries from acquiring realistic estimates. Actions that exceed these boundaries are assigned pseudo Q-values or might be inaccurately estimated due to inherent limitations in the model's generalization capabilities.

Unlike previous work, we investigate how to loosen the rigid demarcation of the OOD boundaries and adaptively extract knowledge from empirical data to implicitly improve the model's generalization to nearby unseen data. Drawing inspiration from meta-learning, we simulate the distribution shift by splitting the offline dataset into train/validation ($\mathcal{D}_t/\mathcal{D}_v$) subsets that have distribution discrepancies. We then model the generalization task of offline RL as a learning problem that enables the model to generalize well over any train/validation subsets. Intuitively, if the model can effectively generalize across all the potential splittings, it is expected to outperform a model trained solely on the original dataset when encountering unseen yet relevant data. Building on this, we formulate our idea as a meta-learning-based bi-level optimization problem, as detailed below:

$$\begin{aligned} &\min_{w} \frac{1}{|\Lambda_\zeta|} \sum_{\mathcal{D}_v \in \Lambda_\zeta} \ell(\hat{\theta}(w); \mathcal{D}_v) + \mathcal{R}(w) \\ &\text{s.t. } \hat{\theta}(w) = \arg\min_{\theta} \ell(\theta; \mathcal{D}_t, w), \mathcal{D}_t = \mathcal{D} - \mathcal{D}_v, \end{aligned} \tag{4}$$

where $\ell(\hat{\theta}(w); \mathcal{D}_v) = \mathcal{L}_{\mathcal{D}_v}(f(\cdot, \hat{\theta}(w)))$ represents the generalization loss and $\ell(\theta; \mathcal{D}_t, w) = \mathcal{L}_{\mathcal{D}_t}(f(\cdot, \theta))$ represents the training loss. The $\mathcal{L}_{\mathcal{D}}(f)$ recall Equation (3) and we denote $f(\cdot, \theta)$ as the model parameterized by θ, introducing w as the initialization of θ (Li et al., 2018). Λ_ζ denotes the set of all possible validation-subsets of $\mathcal{D}$, and $\zeta = |\mathcal{D}_v|/|\mathcal{D}|$ denotes the relative size of validation-subset $\mathcal{D}_v$. $\mathcal{R}(w)$ is an additional regulation set to be the training loss on $\mathcal{D}_t$ (i.e., $\mathcal{R}(w) = \ell(w; \mathcal{D}_t)$).

In problem 4, we aim to update the initial parameters w to the final parameters learned on the task. Inspired by the meta-learning approach (Qiao et al., 2020; Volpi et al., 2018; Gu et al., 2024), we find a good meta parameter $\hat{\theta}(w)$ from the train-subset $\mathcal{D}_t$ by a nonconvex optimization function $\min_\theta \ell(\theta; \mathcal{D}_t, w)$ and then optimize the generalization loss on $\mathcal{D}_v$ based on $\hat{\theta}(w)$. Such an operation coupled with an additional constraint $\mathcal{R}(w)$ allows the model with w as the initialization to generalize well the distribution shift on any train/validation subset.

However, enumerating all possible train/validation splittings when solving problem 4 is intractable. Although the sample size of the pre-collected offline dataset is limited, it is still considerably large. Fortunately, it is easy to observe that models trained on the train-subset will perform better (resp. worse) on the validation-subset when the distribution shift between the train/validation subset is smaller (resp. larger), leading to a lower (resp. higher) generalization loss. Based on this observation, we naturally propose our ADS

[1]The original equation should be $\mathcal{E}(f) = \mathbb{E}_\mu \mathcal{L}_{\mathcal{D}}(f) - \mathbb{E}_\mu \operatorname{Var}_{s' \sim P(\cdot|s,a)}(V_f(s'))$, for the discussion of the additional variance terms is not the focus of this paper.

framework as an alternative:

$$\begin{aligned} &\min_{w} \max_{\mathcal{D}_v \in \Lambda_\zeta} \ell(\hat{\theta}(w); \mathcal{D}_v) + \mathcal{R}(w) \\ &\text{s.t. } \hat{\theta}(w) = \arg\min_{\theta} \ell(\theta; \mathcal{D}_t, w), \mathcal{D}_t = \mathcal{D} - \mathcal{D}_v. \end{aligned} \tag{5}$$

We replace problem 4 with the bi-level min-max problem 5. Finding the most challenging splitting with the maximum distribution shift is more straightforward than enumerating all possible train/validation splittings. Please note that the objective function in 5 serves as an upper bound of that in 4. In other words, we claim that a model demonstrating efficient performance for this worst-case splitting is likely to perform well for other splittings. Concretely, the problem 5 is solved by iterating over the following two steps. First, we train the model on the train-subset to minimize its loss on the validation-subset. Then, we adversarially generate the "hardest" splitting, designed to push the model to its limits of generalization. By doing so, we want to enhance the model's generalization capabilities to the greatest extent possible.

3.2. Optimization and Practical Implementation

We now discuss the optimization of problem 5 and then provide a practical implementation of our ADS framework. There has been extensive experience in solving bi-level min-max optimization problems. It commonly adopts gradient descent methods (Li et al., 2018) to solve the inner optimal $\hat{\theta}(w)$ approximately. The training procedure can be specifically written as:

$$\hat{\theta}(w) = w - \alpha \mathcal{G}_w^t, \tag{6}$$

where $\mathcal{G}_w^t = \nabla_\theta \ell(\theta; \mathcal{D}_t, w)$ and α is the step size. Applying Equation (6) to problem 5, we obtain:

$$\min_{w} \max_{\mathcal{D}_v \in \Lambda_\zeta} \ell(w - \alpha \mathcal{G}_w^t; \mathcal{D}_v) + \mathcal{R}(w). \tag{7}$$

Problem 7 can be solved by iteratively updating w and $\mathcal{D}_v$.

Updating w. Once fixing the $\mathcal{D}_v$, the update of w can be expressed as the common single-level optimization:

$$w = w - \eta \nabla_w (\ell(w - \alpha \mathcal{G}_w^t; \mathcal{D}_v) + \mathcal{R}(w)), \tag{8}$$

where η is the learning rate.

Updating $\mathcal{D}_v$. After fixing the parameter w, we aim to find the hardest splitting $\mathcal{D}_v$ for maximizing the generalization loss $\ell(w - \alpha \mathcal{G}_w^t; \mathcal{D}_v)$ in problem 7. This conveys our adversarial idea: find the validation-subset that remains inadequately generalized by the updated model trained on the train-subset. We further formalize the generalization loss by first-order Taylor expansion $\ell(w - \alpha \mathcal{G}_w^t; \mathcal{D}_v) \approx \ell(w; \mathcal{D}_v) - \alpha \langle \mathcal{G}_w^v, \mathcal{G}_w^t \rangle$, where $\mathcal{G}_w^v = \nabla_w \ell(w; \mathcal{D}_v)$ and $\langle \cdot, \cdot \rangle$ denotes the inner product. Then, we can rewright the maximization problem w.r.t. $\mathcal{D}_v$ as follows:

$$\begin{aligned} &\max_{\mathcal{D}_v, \mathbb{A}} \ell(w; \mathcal{D}_v) - \alpha \langle \nabla_w \ell(w; \mathcal{D}_v), \mathbb{A} \rangle \\ &\text{s.t. } \mathcal{D}_v \in \Lambda_\zeta, \mathbb{A} = \mathcal{G}_w^t. \end{aligned} \tag{9}$$

Since $\mathcal{D}_v$ and $\mathcal{D}_t$ are complementary ($\mathcal{D}_t = \mathcal{D} - \mathcal{D}_v$), solving problem 9 to find the optimal $\mathcal{D}_v$ is affected by $\mathcal{G}_w^t$. We therefore introduce an auxiliary variable $\mathbb{A}$ to denote $\mathcal{G}_w^t$ and update $\mathcal{D}_v$ and $\mathbb{A}$ alternately to solve this optimization problem. In this alternate iteration process, we first initialize $\mathbb{A}$ with the gradient of a randomly selected sample. Then, we compute the values of $\ell(w; \mathcal{D}) - \alpha \langle \nabla_w \ell(w; \mathcal{D}), \mathbb{A} \rangle$ for all the samples in $\mathcal{D}$ and select largest $\zeta|\mathcal{D}|$ samples to constitute the $\mathcal{D}_v$ ($\zeta = |\mathcal{D}_v|/|\mathcal{D}|$). Once obtain the $\mathcal{D}_v$ ($\mathcal{D}_t$ is then given), we update $\mathbb{A} = \frac{1}{|\mathcal{D}_t|} \sum \nabla_w \ell(w; \mathcal{D}_t)$. The process of updating $\mathcal{D}_v$ and $\mathbb{A}$ requires N iterations to convergence. In this way, we obtain the "hardest" splitting $\mathcal{D}_v$.

Overall, the optimization of the problem 5 is achieved by alternately using Equation (8) to update the model and solving problem 9 to find the "hardest" train/validation subsets splitting. Next, we provide the practical implementation of applying our ADS framework to offline RL.

Practical Implementation. We provide an illustration of actor-critic implementation with the ADS framework in Figure 3. Step 1 is a data preprocessing process performed only once at the beginning. Prior experience indicates that a crucial factor for the success of meta-learning-based methods is the requirement for the train/validation subsets to have either the same class or distributions that are closely similar to the task at hand. However, in the context of continuous offline RL data, there is an inherent absence of category information. We must strive to maintain as similar distributions as possible between the train and validation subsets. Existing work(Thompson, 2012; Arnab, 2017) highlights that stratified sampling can effectively maintain similar distributions, which is a more pragmatic approach than manual data preprocessing. The advantage of employing stratified sampling lies in its potential to make research more cost-effective and feasible by partitioning a large number of samples into smaller, homogenous groups. Concurrently, stratified sampling aids in preserving the diversity of the total population within the sample. Moreover, given that sampling is predicated on strata, these strata must be established before stratified sampling. Inspirations drawn from some works(Sun et al., 2020; Ran et al., 2023) indicate that direct distance calculations on continuous data are meaningful for discerning their similarity relationships. Therefore, we utilize a Gaussian Mixture Model (GMM) (Bishop, 2006) to cluster the state-action pairs into K strata. Through step 1, we obtain the initial $\mathcal{D}_t$ and $\mathcal{D}_v$ with hierarchical structures. Then, we use stratified sampling to form two mini-batches for training in step 2.

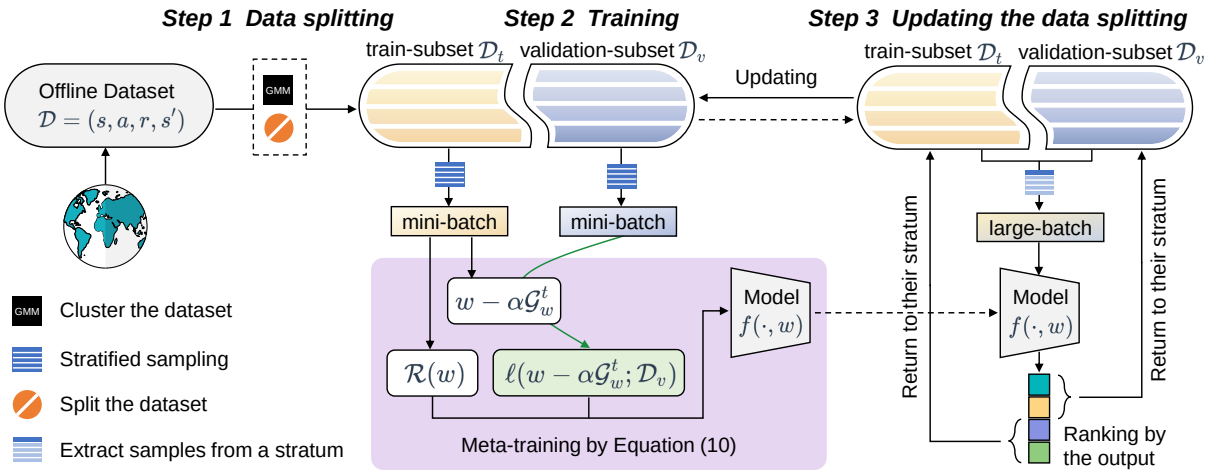

Figure 3: Illustration of actor-critic implementation with the ADS framework. Step 1: We cluster state-action pairs with GMM to form strata, then split the offline dataset into train/validation subsets. Step 2: We use stratified sampling to draw samples proportionally from each stratum of the train/validation subset to form two mini-batches, then train the model (both critic and actor) by meta-training. Step 3: We extract a large-batch from the same stratum of $\mathcal{D}_t$ and $\mathcal{D}_v$, rank these samples, and add the largest $\zeta|\mathcal{D}|$ to the corresponding stratum of $\mathcal{D}_v$ and the rest to that of $\mathcal{D}_t$. This step is repeated for all strata.

Steps 2 and 3 correspond to the alternating iterative optimization mentioned above for problem 5, which continuously performs along with the training process. The actor and critic networks we aim to train are collectively abstracted as the model $f(\cdot, w)$ in these two steps. It's worth noting that since RL cannot obtain the realistic values of samples in advance, it primarily calculates loss through the Bellman backup. The action a', which the backup depends on, is provided by the actor network. Therefore, in step 2, it is necessary to concurrently update the actor network when utilizing meta-learning methods to obtain optimal meta-parameters $\hat{\theta}$. Furthermore, although our method can implicitly handle OOD boundary data, we are still explicitly unable to determine which shifts are beyond our generalization capability and should be prevented from overestimation. For this reason, in our actual implementation, we make an optimistic adjustment by using the support as a boundary to determine whether a data point is OOD or not (in the case of CQL (Kumar et al., 2020)):

$$\ell = (1 - \lambda)\ell_{\text{ADS}} + \lambda \ell_{\text{CQL}}, \tag{10}$$

where $\ell_{\text{ADS}} = \ell(w; \mathcal{D})$ based on w which updated by Equation (8), $\lambda = \frac{\epsilon}{\epsilon_{\max} - \epsilon_{\min}}$, and $\epsilon = |Q_{a\sim\pi(\cdot|s)}(s, a) - Q_{a\sim\mu(\cdot|s)}(s, a)|$ is the value discrepancy between the learned policy π and the behavior policy μ, $\epsilon_{\max}$ and $\epsilon_{\min}$ are the maximum and minimum values of ϵ.

In addition, given the substantial size of the offline dataset, it would be inefficient to iterate over all the data in $\mathcal{D}$ when updating $\mathcal{D}_v$ in step 3. Therefore, we extract a large-batch from the same stratum in both $\mathcal{D}_t$ and $\mathcal{D}_v$, then rank and update them back into their corresponding strata. By repeating this operation across all strata, we approximate the optimization for problem 9. Appendix A provides the detailed algorithm.

3.3. Theoretical Analysis

To provide a deeper understanding of our method, we derive a generalization error bound for offline RL, using a broad meta-learning framework as our basis. We formulate the generalization problem for offline RL as minimizing the generalization error on an unseen target distribution $\mathcal{Q}$, where the model is trained on a source distribution $\mathcal{P}$. In this case, we learn the model on $\mathcal{P}$ (the samples in $\mathcal{D}$ are $i.i.d.$ sampled from $\mathcal{P}$) by using meta-learning techniques to find an $f \in \mathcal{F}_{\mathcal{D}_t} = \{f(\cdot, \hat{\theta}(w)) : \hat{\theta}(w) = \arg\min_\theta \ell(\theta; \mathcal{D}_t, w)\}$ to minimize the generalization loss on $\mathcal{D}_v$.

For simply, we denote $l(f) = (f(s, a) - r - \gamma V_f(s'))^2$ that omits $(s, a, r, s') \in \mathcal{D}$ and introduce a loss-related indicator $\Psi_l(f)$:

$$\Psi_l(f) = \begin{cases} 1 & \text{if } l(f) > \gamma, \\ 0 & \text{otherwise}, \end{cases} \tag{11}$$

where γ is a constant. We seek to establish a connection between the generalization error, $\mathcal{E}_{\mathcal{Q}}^{\Psi_l}(f) = \mathbb{E}_{\mathcal{Q}}[\mathbb{I}_{\Psi_l(f)=1}]$, and the empirical error, $\hat{\mathcal{E}}_{\mathcal{D}_v}^{\Psi_l}(f) = \frac{1}{|\mathcal{D}_v|}\sum \mathbb{I}_{\Psi_l(f)=1}$, thus helping to analyze the effectiveness of our ADS framework.

Theorem 3.1. *There exists an absolute constant $\gamma > 0$, if we assume*[2] $\mathbb{E}_{\mathcal{Q}}[\mathbb{I}_{\Psi_l(f)=1}] \geq \mathbb{E}_{\mathcal{P}}[\mathbb{I}_{\Psi_l(f)=1}]$, *for any* $\mathcal{D}_v \in$

[2]The assumption is realistic because the model trained on the $\mathcal{P}$ data should have a smaller classification loss on $\mathcal{P}$ than $\mathcal{Q}$.

Table 1: Average normalized scores over the final 10 evaluations and 5 seeds. We **bold** the highest mean.

Dataset	CQL	**CQL+ADS**	TD3+BC	**TD3+BC+ADS**	MCQ	**MCQ+ADS**
halfcheetah-medium	49.4 ± 0.2	**73.9** ± 2.8	48.2 ± 0.5	**49.0** ± 2.7	62.5 ± 3.1	**63.2** ± 2.7
hopper-medium	59.1 ± 4.1	**101.0** ± 6.0	60.8 ± 3.4	**73.7** ± 13.0	78.4 ± 4.3	**103.0** ± 1.5
walker2d-medium	83.6 ± 0.5	**91.3** ± 1.0	84.4 ± 2.1	**85.0** ± 1.1	91.0 ± 1.1	**94.5** ± 2.9
halfcheetah-medium-replay	47.0 ± 0.3	**49.6** ± 2.9	45.0 ± 0.5	**46.1** ± 2.9	56.2 ± 2.7	**59.4** ± 3.1
hopper-medium-replay	98.6 ± 1.5	**102.4** ± 1.2	67.3 ± 13.2	**100.3** ± 2.2	101.6 ± 1.0	**105.0** ± 0.9
walker2d-medium-replay	71.3 ± 17.9	**93.7** ± 1.1	83.4 ± 7.0	**91.3** ± 0.9	91.3 ± 1.8	**96.1** ± 0.6
halfcheetah-medium-expert	93.0 ± 2.2	**93.5** ± 4.0	90.7 ± 2.7	**96.6** ± 3.1	**80.1** ± 3.8	78.9 ± 0.5
hopper-medium-expert	111.4 ± 0.5	**113.3** ± 1.3	91.4 ± 11.3	**114.0** ± 1.9	87.8 ± 2.0	**105.8** ± 0.2
walker2d-medium-expert	109.8 ± 0.5	**112.1** ± 0.3	110.2 ± 0.3	**114.0** ± 1.1	**114.2** ± 0.9	108.3 ± 1.0
antmaze-umaze	82.6	**95.0**	73.0	**94.0**	0.0	**10.0**
antmaze-umaze-diverse	10.2	**82.0**	47.0	**90.0**	0.0	**30.0**
antmaze-medium-play	**59.0**	30.0	0.0	**10.0**	0.0	0.0
antmaze-medium-diverse	46.6	**64.0**	0.2	**10.0**	0.0	0.0
antmaze-large-play	**16.4**	10.0	0.0	0.0	0.0	0.0
antmaze-large-diverse	3.2	**12.0**	0.0	0.0	0.0	0.0

Λ_ζ *and* $\delta \in (0,1)$*, with probability at least* $1-2\delta$*, we have* $\forall f \in \mathcal{F}_{\mathcal{D}_t}$*,*

$$\mathcal{E}_{\mathcal{Q}}^{\Psi_l}(f) \leq \hat{\mathcal{E}}_{\mathcal{D}_v}^{\Psi_l}(f) - \inf_{f' \in \mathcal{F}_{\mathcal{D}_t}} \frac{1}{|\mathcal{D}_v|} \sum \mathbb{I}_{\Psi_l(f')=1} + \mathcal{C}, \quad (12)$$

where $\mathcal{C} = \mathcal{C}_{\mathfrak{R}} + \mathcal{C}_{gap}$ is a constant term in Theorem 3.1. Specifically, $\mathcal{C}_{\mathfrak{R}} = 2\sup_{\mathcal{D}'_v \in \Lambda_\zeta} \widehat{\mathfrak{R}}_{\mathcal{D}'_v}(\mathcal{F}^{\Psi_l}_{\mathcal{D}-\mathcal{D}'_v}) + 6\sqrt{\frac{\log\frac{2}{\delta}}{2\zeta|\mathcal{D}|}}$, in where $\mathcal{F}^{\Psi_l}_{\mathcal{D}_t} = \{\Psi_l \circ f : f \in \mathcal{F}_{\mathcal{D}_t}\}$ and $\widehat{\mathfrak{R}}_S(\mathcal{F})$ is the empirical Rademacher complexity[3] (Mohri et al., 2018) of $\mathcal{F}$, which is determined by the diversity of the hypothesis space and the number of training data, and is often informally regarded as a constant. $\mathcal{C}_{gap} = \sup_{f' \in \mathcal{F}_{\mathcal{D}_t}} \mathbb{E}_{\mathcal{Q}}[\mathbb{I}_{\Psi_l(f')=1}] + \inf_{f' \in \mathcal{F}_{\mathcal{D}_t}} [\mathcal{E}_{\mathcal{Q}}^{\Psi_l}(f') + \mathcal{E}_{\mathcal{P}}^{\Psi_l}(f')]$ reflects the distribution gap between $\mathcal{Q}$ and $\mathcal{D}_t$, that is, the models trained on $\mathcal{D}_t$ (i.e., $f' \in \mathcal{F}_{\mathcal{D}_t}$) would have a lower (resp. higher) loss on the $\mathcal{Q}$ data if the gap is smaller (resp. larger). However, we cannot directly estimate $\mathcal{C}_{gap}$ due to the lack of $\mathcal{Q}$ data and treat it as an objective constant.

Appendix B provides the proof of Theorem 3.1, and we now illustrate that our ADS model implicitly minimizes the rest terms in Equation (12) except for the constant $\mathcal{C}$. The first term $\hat{\mathcal{E}}_{\mathcal{D}_v}^{\Psi_l}(f)$ is the empirical error on $\mathcal{D}_v$, which is also the generalization loss $\ell(\hat{\theta}(w); \mathcal{D}_v)$ that our ADS model seeks to minimize. For the second term, minimizing $(-\inf_{f' \in \mathcal{F}_{\mathcal{D}_t}} \frac{1}{|\mathcal{D}_v|} \sum \mathbb{I}_{\Psi_l(f')=1})$ is equivalent to $\max_{\mathcal{D}_v \in \Lambda_\zeta} \inf_{f' \in \mathcal{F}_{\mathcal{D}_t}} \frac{1}{|\mathcal{D}_v|} \sum \mathbb{I}_{\Psi_l(f')=1})$, which is closely related to our min-max problem 5, in which we strive to find the hardest splitting $\mathcal{D}_v$ with maximum generalization loss and then minimize it. In summary, our ADS model is closely related to the minimization of the upper bound in Equation (12).

[3] $\widehat{\mathfrak{R}}_S(\mathcal{F}) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{f \in \mathcal{F}} \frac{1}{m} \sum_{i=1}^m \sigma_i f(x_i)\right]$, where S is a fixed data with m samples $(x_1, \ldots, x_m)$, $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_m)^\top$ consists of $i.i.d.$ random variables taking values in $\{-1, +1\}$.

Convergence. As our method ADS involves optimization of a bi-level problem, we theoretically show that ADS converges to the critical points under some mild conditions. To demonstrate its convergence fundamentally equates to establishing that the sequence concerning $\ell(w^{(k)}; \theta^{(k+1)})$ tends toward 0. In pursuit of this objective, we introduce the subsequent theorem:

Theorem 3.2. *Suppose the loss function* ℓ *is Lipschitz-smooth with constant L, and have* ρ*-bounded gradients with respect to train/validation data. Let the learning rate* η_k *satisfy* $\sum_{k=0}^{\infty} \eta_k = \infty, \sum_{k=0}^{\infty} \eta_k^2 < \infty$*, and* α_k*,* $1 \leq k \leq N$ *is a monotone descent sequence. Then,*

$$\lim_{k \to \infty} \mathbb{E}[\|\nabla \ell(w^{(k)}; \theta^{(k+1)})\|_2^2] = 0. \quad (13)$$

The proof is listed in Appendix C. The details and convergence of the iteration in problem 9 are also discussed.

4. Related Work

Improving generalization in offline RL is frequently explored in the context of balancing mild and conservative restrictions on OOD actions. Current experience indicates that avoiding overly pessimistic conservatism holds more promise than conservatively suppressing the overestimation of OOD actions. Mildly Conservative Q-Learning (MCQ) (Lyu et al., 2022) explores mild but enough conservatism by actively assigning OOD actions proper pseudo Q-values. Policy-guided Offline RL (POR) (Xu et al., 2022) inherits the training stability of imitation-style methods while still allowing logical OOD generalization. Policy Regularization with Dataset Constraint (PRDC) (Ran et al., 2023) allows the learned policy to choose optimal actions from all actions in the dataset, which is less conservative than the commonly used distribution and support constraints. Representation Distinction (RD) (Ma et al., 2023) proposes

Table 2: Average normalized scores over the final 10 evaluations and 5 seeds. We **bold** the highest mean and underline the second place.

Dataset	BC	IQL	POR	PRDC	STR	CQL+ADS	TD3+BC+ADS	MCQ+ADS
halfcheetah-medium	42.9	47.4	48.8	63.5	51.8	**73.9**	49.0	63.2
hopper-medium	56.1	65.7	78.6	100.3	101.3	101.0	73.7	**103.0**
walker2d-medium	76.6	81.1	81.1	85.2	85.9	91.3	85.0	**94.5**
halfcheetah-medium-replay	36.6	44.2	43.5	55.0	47.5	49.6	46.1	**59.4**
hopper-medium-replay	19.3	94.8	98.9	100.1	100.0	102.4	100.3	**105.0**
walker2d-medium-replay	24.8	77.3	76.6	92.0	85.7	93.7	91.3	**96.1**
halfcheetah-medium-expert	53.1	88.0	94.7	94.5	94.9	93.5	**96.6**	78.9
hopper-medium-expert	52.7	106.2	90.0	109.2	111.9	113.3	**114.0**	105.8
walker2d-medium-expert	102.5	108.3	109.1	111.2	110.2	112.1	**114.0**	108.3
Average Above	51.6	79.2	80.1	90.1	87.7	**92.3**	85.6	90.5

to timely suppress generalization (especially at the early learning stage) to tackle the problem of overgeneralization. Supported Trust Region optimization (STR) (Mao et al., 2023) performs trust region policy optimization within the behavior policy's support, benefiting from a less restrictive support constraint. In contrast, we aim to extract knowledge adaptively from empirical data, thereby implicitly improving the model's generalization to nearby unseen data. Our approach is distinct from the above research and remains largely unexplored. Additionally, using the loss function as a bridge, we can easily apply our approach to existing widely-used algorithms.

Although our approach employs meta-learning techniques, it falls into a different research area from the existing meta-RL (Kirsch et al., 2019; Lin et al., 2020) and offline meta-RL methods (Wang et al., 2023). The meta-RL aims to learn from multiple training tasks the ability to adapt efficiently to unseen test tasks. Our approach, ADS, still operates under the general settings of offline RL. ADS consolidates the meta-training process and adversarial data splitting into a cohesive framework, adaptively simulating distribution shifts from empirical data, thereby enhancing the robustness of the trained model.

5. Experiments

In this section, we conduct experiments to validate the effectiveness of ADS by answering the following questions:

(i) *How does ADS perform on the benchmarks by applying it to existing widely-used algorithms?*

(ii) *How to prove that generalizing OOD boundary data with ADS is better than assigning pseudo Q-values?*

(iii) *How do hyperparameters impact the performance of ADS? Is ADS time-consuming?*

(iv) *How does the adversarial idea impact the performance of ADS?*

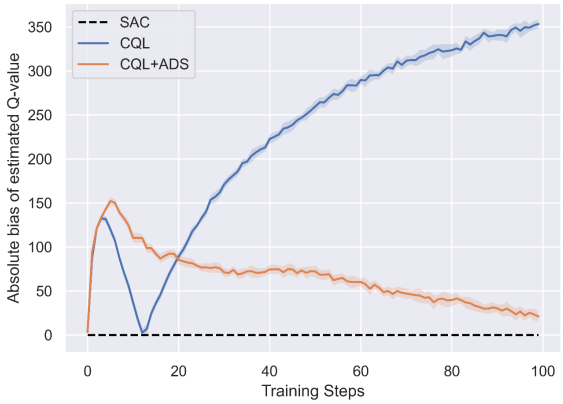

Figure 4: The absolute bias of the estimated Q-value. The baseline algorithm is SAC and the shaded area represents the standard deviation.

5.1. Main Results

First, we apply our ADS framework to existing widely-used algorithms, CQL (Kumar et al., 2020), TD3+BC (Fujimoto & Gu, 2021), and MCQ (Lyu et al., 2022), and conduct experiments on several D4RL (Fu et al., 2020) gym MuJoCo-v2 datasets. We procure the baseline results either by rerunning the official code or by directly extracting them from the original papers and report the result at 1M gradient step in Table 1. It is observable that the overall performance of both CQL, TD3+BC, and MCQ is enhanced when the ADS is applied. The CQL and MCQ are representative algorithms that assign pseudo Q-values. The ADS framework, utilizing the generalization capabilities of neural networks to deal with OOD actions near the boundary, exhibits more optimism than these two algorithms. The experimental outcomes indirectly convey that improving generalization is more beneficial than assigning pseudo Q-values. Furthermore, it's quite exhilarating to note that our algorithm demonstrates a lower standard deviation compared to the baseline in certain scenarios. This indicates that our ADS framework, by loosening rigid boundaries, helps to enhance the stability of the algorithm.

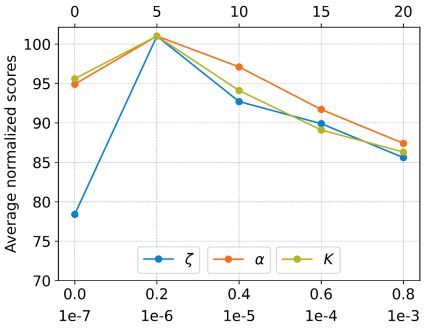

Figure 5: Parameters.

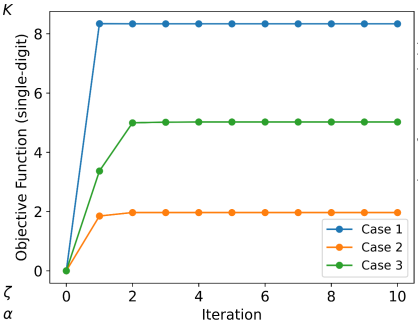

Figure 6: Convergence of the iteration.

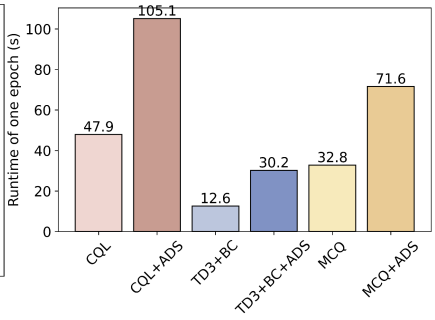

Figure 7: Run time.

It's known that the aforementioned value estimation-based baseline methods are not competent for sparse rewards tasks, such as AntMaze. We are interested in whether our framework can appropriately enhance the performance of these algorithms. Therefore, we also list the comparison results on AntMaze tasks in Table 1. It shows that, after applying the ADS framework, these baseline algorithms not only exhibit slight enhancements in tasks where they previously failed to make an impact but also demonstrate significantly better performance in tasks where they were already somewhat capable. This improvement can be attributed to the fact that when the baseline algorithms can approach the target point, ADS enhances the applicability of the policy in the vicinity of the target. It achieves this by bolstering the generalization capability, which subsequently leads to an increase in the success rate. Note that the primary objective behind designing the Antmaze task was to demonstrate the enhancement of the baseline's generalization ability through our proposed method. We do not claim that our framework is universally adept at handling tasks with sparse rewards.

Additionally, under the same experimental settings, we compare our method with behavior cloning (BC) and several model-free offline RL algorithms, including IQL (Kostrikov et al., 2021), POR (Xu et al., 2022), PRDC (Ran et al., 2023) and STR (Mao et al., 2023). As evidenced by Table 2, our representative algorithm delivers the best overall performance and achieves state-of-the-art results on several datasets. Among them, POR, PRDC, and STR are advanced methods that enhance generalization capabilities from different perspectives. In contrast, all three of our methods demonstrate competitiveness, both in terms of second-place performance and average scores. Our results present a relatively objective perspective. When dealing with non-expert datasets where the behavior policies are suboptimal, the optimistic approach MCQ+ADS proves more effective. However, as the proportion of expert data escalates, indicating policies that are nearing optimality, the more pessimistic methods of CQL+ADS and the behavior cloning-based TD3+BC+ADS demonstrate superior performance. More details and results are listed in Appendix D.

5.2. Value Estimation Error

Not content with the indirect reflection from the results in Table 1, we aim to investigate whether the ADS framework has succeeded in generalizing the OOD boundary data. A feasible solution is to compare the Q-value estimations for OOD actions. We use the Soft Actor-Critic (SAC) (Haarnoja et al., 2018) algorithm to interact online for 1M steps and gather this data. Then, we randomly extract 100 groups (each group containing 20 samples) from these online data. We conduct Q-value estimation tests during the training process of the CQL, CQL+ADS, and SAC algorithms, testing once every ten epochs. Among them, we run SAC on the MuJoCo halfcheetah-v2 environment and represent the realistic Q-value using the estimation of the test data. Moreover, we run CQL and CQL+ADS on the halfcheetah-medium dataset. In comparison, the 100 groups of samples collected through online interactions can be treated as OOD data to a certain extent. We calculated the absolute bias of the estimated Q-value for two algorithms (CQL, CQL+ADS) compared with the SAC and plotted the corresponding curves in Figure 4. These curves reflect that the gap between the estimations of CQL+ADS and the realistic values is gradually decreasing. In contrast, CQL, due to its over conservatism, is gradually moving away from better policies.

5.3. Sensitivity and Runtime Analysis

The effectiveness of our ADS framework is mainly influenced by several hyperparameters, including the ratio $\zeta = |\mathcal{D}_v|/|\mathcal{D}|$, the number of clusters K, and the step size α. For simplicity, we do not perform a combinatorial analysis of these parameters. We run MCQ+ADS on hopper-medium-v2 and plot the results in Figure 5. The larger the proportion (ζ) of $\mathcal{D}_v$, the worse the performance ($\zeta = 0$ indicates the score of MCQ). This is because we need to ensure sufficient training data to find good meta-parameters when updating w (Equation (8)). As for the effect of K, a larger K results in higher similarity among the samples obtained through stratified sampling, which contradicts the original intention of random sampling in RL training (which requires $i.i.d.$ sampling). It is worth stating that the process

Table 3: Average normalized scores over the final 10 evaluations and 5 seeds. We **bold** the highest mean.

Dataset	COL+ADS	COL+ADS w/o adversarial	TD3+BC+ADS	TD3+BC+ADS w/o adversarial	MCQ+ADS	MCQ+ADS w/o adversarial
halfcheetah-medium	**73.9** $\pm$ 2.8	68.6 $\pm$ 3.1	**49.0** $\pm$ 2.7	48.1 $\pm$ 2.8	**63.2** $\pm$ 2.7	60.1 $\pm$ 3.2
hopper-medium	**101.0** $\pm$ 6.0	91.3 $\pm$ 7.7	**73.7** $\pm$ 13.0	71.0 $\pm$ 14.6	**103.0** $\pm$ 1.5	92.4 $\pm$ 12.2
walker2d-medium	**91.3** $\pm$ 1.0	87.2 $\pm$ 1.5	**85.0** $\pm$ 1.1	81.9 $\pm$ 3.9	**94.5** $\pm$ 2.9	92.3 $\pm$ 3.6
halfcheetah-medium-replay	**49.6** $\pm$ 2.9	47.2 $\pm$ 3.1	**46.1** $\pm$ 2.9	45.5 $\pm$ 3.1	**59.4** $\pm$ 3.1	57.6 $\pm$ 3.3
hopper-medium-replay	**102.4** $\pm$ 1.2	101.0 $\pm$ 1.2	**100.3** $\pm$ 2.2	86.8 $\pm$ 15.5	**105.0** $\pm$ 0.9	102.0 $\pm$ 1.3
walker2d-medium-replay	**93.7** $\pm$ 1.1	87.9 $\pm$ 4.4	**91.3** $\pm$ 0.9	88.3 $\pm$ 2.9	**96.1** $\pm$ 0.6	93.6 $\pm$ 1.4
halfcheetah-medium-expert	**93.5** $\pm$ 4.0	91.5 $\pm$ 4.3	**96.6** $\pm$ 3.1	93.3 $\pm$ 5.7	**78.9** $\pm$ 0.5	76.6 $\pm$ 1.2
hopper-medium-expert	**113.3** $\pm$ 1.3	110.0 $\pm$ 1.1	**114.0** $\pm$ 1.9	98.5 $\pm$ 17.4	**105.8** $\pm$ 0.2	95.1 $\pm$ 11.4
walker2d-medium-expert	**112.1** $\pm$ 0.3	110.0 $\pm$ 0.7	**114.0** $\pm$ 1.1	110.0 $\pm$ 2.1	**108.3** $\pm$ 1.0	106.0 $\pm$ 1.6

of stratified sampling and the delineation of strata are intrinsically linked, and the delineation of strata is not executed under the condition when $K = 0$, which means that using normal sampling in both $\mathcal{D}_t$ and $\mathcal{D}_v$. We can see that the algorithm's performance deteriorates when $K = 0$. The results of α are relatively objective. A smaller α restricts the benefits of meta-learning and adversarial data splitting. A larger α affects gradient updates and may not be able to reduce training loss from an optimization perspective.

In finding the "hardest" $\mathcal{D}_v$, our method requires N iterations of alternating updates to $\mathbb{A}$ until convergence. We run MCQ+ADS on hopper-medium-v2 and randomly select three cases to show the convergence of this alternate iteration in Figure 6. The ordinate is the value of the objective function 9. It shows that the values converge after only a few iterations.

The time complexity is also a challenge in offline RL. While the process of the ADS framework might appear complex, it's important to note that the data preprocessing in step 1 is a one-time operation performed prior to the commencement of the task. The gradient computation and updates to $\mathcal{D}_v$ and $\mathcal{D}_t$ in steps 2 and 3 are comparatively straightforward. Consequently, the increase in computational cost relative to the baseline algorithm is minimal. We apply ADS to baselines and compare the run time of one epoch on the halfcheetah-medium-v2 dataset in Figure 7. It shows that although ADS has additional gradient calculation and data splitting processes, it runs at an acceptable speed. We highlight that the additional time cost allocats to extracting valuable information from the data, and the significance of this process outweighs the increase in the runtime.

5.4. Ablation Study

We design ablation experiments to elucidate the significance of the adversarial idea. We compare our methods against their variants (i.e., without adversarial). The adversarial idea works by allowing the model to generalize well even in the most challenging cases, thus ensuring its competence in a diverse range of cases. Consequently, the method devoid of adversarial component is designed to replace step 3 within the ADS framework by employing a stochastic splitting of the train/validation subset. We provide the experimental outcomes for CQL+ADS w/o adversarial, TD3+BC+ADS w/o adversarial, and MCQ+ADS w/o adversarial, as tabulated in Table 3. The results demonstrate that the algorithms' performance decreases after replacing the adversarial component and exhibits a high standard deviation in some cases, which underscores the indispensability of the adversarial idea.

6. Conclusion

We propose an adversarial data splitting (ADS) framework to improve the generalization in offline RL. ADS innovatively splits the offline dataset into train/validation subsets to simulate distribution shifts. It then employs adversarial principles to seek the "hardest" train/validation subsets with the maximum distribution shift. Crucially, the model is trained in a meta-learning manner to maintain robust generalization performance, even under the most challenging splitting scenarios. By solving the iterative optimization problem modeled by the above process, ADS loosens the rigid demarcation of the OOD boundaries and improves the model's generalization to nearby unseen data. We theoretically demonstrate that ADS can implicitly minimize the upper bound of generalization error for offline RL based on a meta-learning framework. Our extensive experiments show that combining ADS with existing widely-used algorithms can significantly enhance their performance. It also remains competitive in works related to improving generalization.

Our work focuses more on extracting knowledge from empirical data to promote the model, which has not been fully explored in previous studies. There remains room for improvement in ADS, such as extending its applicability to discrete control tasks and exploring alternative approaches to stratified sampling techniques. In addition, integrating ADS with the diffusion model is quite interesting. Deploying the diffusion model yields trajectories that closely mirror yet are slightly different from the dataset's trajectory distribution, and ADS facilitates a more valuable utilization of these generated data points. We hope that our work will stimulate more related research into the community.

Acknowledgements

This work is supported by the National Key Research and Development Program of China (2020AAA0106100), the National Natural Science Foundation of China (62276160), and the Natural Science Foundation of Shanxi Province, China (202203021211294). We extend our heartfelt gratitude to the anonymous reviewers for their invaluable and constructive comments, and we express our deep appreciation to Ting Guo and Yi Ma for their significant support.

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

An, G., Moon, S., Kim, J. H., and Song, H. O. Uncertainty-based offline reinforcement learning with diversified q-ensemble. In *Advances in neural information processing systems*, volume 34, pp. 7436–7447, 2021.

Arnab, R. *Survey sampling theory and applications*. Academic Press, 2017.

Bai, C., Wang, L., Yang, Z., Deng, Z., Garg, A., Liu, P., and Wang, Z. Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning. In *Proceedings of the Tenth International Conference on Learning Representations*, 2022.

Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, volume 19, 2006.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. A theory of learning from different domains. *Machine learning*, 79:151–175, 2010.

Bishop, C. M. Pattern recognition and machine learning. *Springer google schola*, 2:531–537, 2006.

Chen, J. and Jiang, N. Information-theoretic considerations in batch reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 1042–1051, 2019.

Duan, Y., Jin, C., and Li, Z. Risk bounds and rademacher complexity in batch reinforcement learning. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 2892–2902, 2021.

Fakoor, R., Mueller, J. W., Asadi, K., Chaudhari, P., and Smola, A. J. Continuous doubly constrained batch reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 11260–11273, 2021.

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

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

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

Gu, X., Sun, J., and Xu, Z. Adversarial data splitting for domain generalization. *Science China Information Sciences*, 67(5):1–15, 2024.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 1861–1870, 2018.

Kirsch, L., van Steenkiste, S., and Schmidhuber, J. Improving generalization in meta reinforcement learning using learned objectives. In *Proceedings of the Seventh International Conference on Learning Representations*, 2019.

Kostrikov, I., Fergus, R., Tompson, J., and Nachum, O. Offline reinforcement learning with fisher divergence critic regularization. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 5774–5783, 2021.

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

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

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

Li, D., Yang, Y., Song, Y., and Hospedales, T. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Li, J., Zhang, E., Yin, M., Bai, Q., Wang, Y., and Wang, W. Y. Offline reinforcement learning with closed-form policy improvement operators. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 20485–20528, 2023.

Lin, Z., Thomas, G., Yang, G., and Ma, T. Model-based adversarial meta-reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 10161–10173, 2020.

Lyu, J., Ma, X., Li, X., and Lu, Z. Mildly conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 1711–1724, 2022.

Ma, Y., Jayaraman, D., and Bastani, O. Conservative offline distributional reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 19235–19247, 2021.

Ma, Y., Tang, H., Li, D., and Meng, Z. Reining generalization in offline reinforcement learning via representation distinction. In *Advances in Neural Information Processing Systems*, volume 36, pp. 40773–40785, 2023.

Mairal, J. Stochastic majorization-minimization algorithms for large-scale optimization. In *Advances in Neural Information Processing Systems*, volume 26, pp. 2283–2291, 2013.

Mao, Y., Zhang, H., Chen, C., Xu, Y., and Ji, X. Supported trust region optimization for offline reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 23829–23851, 2023.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of machine learning*. 2018.

Qiao, F., Zhao, L., and Peng, X. Learning to learn single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12556–12565, 2020.

Ran, Y., Li, Y., Zhang, F., Zhang, Z., and Yu, Y. Policy regularization with dataset constraint for offline reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 28701–28717, 2023.

Sun, P., Zhou, W., and Li, H. Attentive experience replay. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, volume 34, pp. 5900–5907, 2020.

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

Thompson, S. K. *Sampling*, volume 755. John Wiley & Sons, 2012.

Volpi, R., Namkoong, H., Sener, O., Duchi, J. C., Murino, V., and Savarese, S. Generalizing to unseen domains via adversarial data augmentation. In *Advances in neural information processing systems*, volume 31, 2018.

Wang, J., Zhang, J., Jiang, H., Zhang, J., Wang, L., and Zhang, C. Offline meta reinforcement learning with in-distribution online adaptation. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Wang, Z., Hunt, J. J., and Zhou, M. Diffusion policies as an expressive policy class for offline reinforcement learning. In *Proceedings of the Tenth International Conference on Learning Representations*, 2022.

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

Wu, Y., Zhai, S., Srivastava, N., Susskind, J. M., Zhang, J., Salakhutdinov, R., and Goh, H. Uncertainty weighted actor-critic for offline reinforcement learning. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 11319–11328, 2021.

Xu, H., Jiang, L., Li, J., and Zhan, X. A policy-guided imitation approach for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 4085–4098, 2022.

A. Algorithm

Our algorithm consists of the following two components. Algorithm 1 is the process of using Equation (8) to update the model. Algorithm 2 solves the problem 9 to find the "hardest" train/validation subsets splitting. These two algorithms correspond to the alternating iterative optimization for problem 5, which continuously performs along with the training process.

Algorithm 1 Training: actor-critic version in the case of CQL (Kumar et al., 2020)

Input: Offline dataset $\mathcal{D}$, initialization model $f(\cdot, w)$, the number of clusters K.
Output: Updated model $f(\cdot, w)$.
Use Gaussian mixture model to cluster $\mathcal{D}$ as K strata.
Stratified sample from the clustering results to obtain the train-subset $\mathcal{D}_t$ and the validation-subset $\mathcal{D}_v$.
for step $t = 1$ **to** T **do**
  Sample proportionally from each stratum of the train/validation subset to form two mini-batch data.
  Update model $f(\cdot, w)$ (including both critic and actor networks) by Equation (10) using SGD on the sampled data.
  if reach the data splitting update interval **then**
    Update $\mathcal{D}_t$ and $\mathcal{D}_v$ using Algorithm 2.
  end if
end for

Algorithm 2 Updating the data splitting

Input: Train-subset $\mathcal{D}_t$ and validation-subset $\mathcal{D}_v$, the updated model $f(\cdot, w)$, the number of clusters K, parameter ζ and the size of a big batch I.
for $k = 1$ **to** K **do**
  Extract $(1-\zeta)I$ samples from the k-th stratum of $\mathcal{D}_t$ and ζI samples from the k-th stratum of $\mathcal{D}_v$ to form a big batch B^k and meanwhile remove them from their stratum.
  Feed these samples to model $f(\cdot, w)$ and calculate the loss ℓ and its gradient $\mathcal{G}_w$.
  Randomly select a gradient to initialize $\mathbb{A}$.
  for $i = 1$ **to** N **do**
    Rank the samples in B^k with $\ell - \alpha \langle \nabla_w \ell, \mathbb{A} \rangle$ and select the largest ζI samples to constitute B_v^k.
    $B_t^k = B^k - B_v^k$
    Update $\mathbb{A}$ by $\frac{1}{|B_t^k|} \sum \nabla_w \ell(w; B_t^k)$
  end for
  Return the samples in B_t^k and B_v^k to their k-th stratum.
end for

B. Proof of Theorem 3.1

This section proves Theorem 3.1 in Section 3.3 of the paper. We first introduce the generalization bound based on empirical Rademacher complexity and the domain adaptation theory, then present two lemmas that will be used in the proof, and finally give the proof of Theorem 3.1.

B.1. Preliminary

The generalization bound based on empirical Rademacher complexity (Mohri et al., 2018):

Definition B.1. (Empirical Rademacher complexity) Let $\mathcal{H}$ be a family of functions mapping from X to $[0, 1]$ and S is a fixed data with m samples $(x_1, \ldots, x_m)$. Then, the empirical Rademacher complexity of $\mathcal{H}$ with respect to the sample S is defined as:

$$\widehat{\mathfrak{R}}_S(\mathcal{H}) = \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{h \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i h(x_i) \right], \tag{14}$$

where $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_m)^\top$ consists of $i.i.d.$ random variables taking values in $\{-1, +1\}$. The random variables σ_i are called Rademacher variables.

Theorem B.2. *Let $\mathcal{H}$ be a family of functions mapping from X to $[0,1]$. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$ over the draw of an i.i.d. sample S of size m, we have $\forall h \in \mathcal{H}$,*

$$\mathbb{E}[h(x)] \leq \frac{1}{m}\sum_{i=1}^{m} h(x_i) + 2\widehat{\mathfrak{R}}_S(\mathcal{H}) + 3\sqrt{\frac{\log\frac{2}{\delta}}{2m}}. \tag{15}$$

Domain adaptation theory (Ben-David et al., 2006; 2010):

Theorem B.3. *Let $\mathcal{P}$ be a source distribution and $\mathcal{Q}$ be a target distribution. For $\forall h \in \mathcal{H}$, we have,*

$$\mathcal{E}_{\mathcal{Q}}(h) \leq \mathcal{E}_{\mathcal{P}}(h) + \frac{1}{2}d_{\mathcal{H}}(\mathcal{P},\mathcal{Q}) + \lambda^{\star}, \tag{16}$$

where $\lambda^{\star} \geq \inf_{h' \in \mathcal{H}}\{\mathcal{E}_{\mathcal{P}}(h') + \mathcal{E}_{\mathcal{Q}}(h')\}$, $\mathcal{E}_{\mathcal{P}}(h) = \mathbb{E}_{(x,y)\sim\mathcal{P}}\left[\mathbb{I}_{\{(h(x))\neq y\}}\right]$ is the source error of $\{(x,y)\}$ i.i.d. sampled from $\mathcal{P}$ and

$$d_{\mathcal{H}}(\mathcal{P},\mathcal{Q}) = 2\sup_{h\in\mathcal{H}}|\mathbb{E}_{\mathcal{P}}[h=1] - \mathbb{E}_{\mathcal{Q}}[h=1]|$$

is the $\mathcal{H}$-divergence.

B.2. Lemmas

Lemma B.4. *Given $\mathcal{H} : X \mapsto \{-1,+1\}$ and $\mathcal{F}_{\mathcal{H}} = \{f_h : h \in \mathcal{H}\}$. Let $Z = X \times \{-1,+1\}$, for any $\delta \in (0,1)$, with probability at least $1-\delta$ over the draw of an i.i.d. sample S of size m, we have $\forall f_h \in \mathcal{F}_{\mathcal{H}}$,*

$$\mathbb{E}[f_h(z)] \leq \frac{1}{m}\sum_{i=1}^{m} f_h(z_i) + \widehat{\mathfrak{R}}_S(\mathcal{H}) + 3\sqrt{\frac{\log\frac{2}{\delta}}{2m}}. \tag{17}$$

Proof. Given $\mathcal{H} : X \mapsto \{-1,+1\}$ and $\mathcal{F}_{\mathcal{H}} = \{f_h : h \in \mathcal{H}\}$. Let $Z = X \times \{-1,+1\}$ ($y \in \{-1,+1\}$), we have the following transformation,

$$f_h(z) = f_h(x,y) = \mathbb{I}\left(h(x) \neq y\right).$$

Thus, the hypothesis space $\mathcal{H}$ with value range $\{-1,+1\}$ is transformed into the function space $\mathcal{F}_{\mathcal{H}} = \{f_h : h \in \mathcal{H}\}$ with value range $[0,1]$. From the Definition B.1 of empirical Rademacher complexity it follows:

$$\begin{aligned}
\widehat{\mathfrak{R}}_Z(\mathcal{F}_{\mathcal{H}}) &= \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{f_h\in\mathcal{F}_{\mathcal{H}}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i f_h(x_i,y_i)\right] \\
&= \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i\mathbb{I}\left(h(x_i)\neq y\right)\right] \\
&= \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i\frac{1-y_i h(x_i)}{2}\right] \\
&= \frac{1}{2}\mathbb{E}_{\boldsymbol{\sigma}}\left[\frac{1}{m}\sum_{i=1}^{m}\sigma_i + \sup_{h\in\mathcal{H}}\frac{1}{m}\sum_{i=1}^{m}(-y_i\sigma_i h(x_i))\right] \\
&= \frac{1}{2}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}}\frac{1}{m}\sum_{i=1}^{m}(-y_i\sigma_i h(x_i))\right] \\
&= \frac{1}{2}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}}\frac{1}{m}\sum_{i=1}^{m}(\sigma_i h(x_i))\right] \\
&= \frac{1}{2}\widehat{\mathfrak{R}}_S(\mathcal{H}).
\end{aligned} \tag{18}$$

Due to $y \in \{-1,+1\}$, $-y_i\sigma_i$ is equal to σ_i. Substituting Equation (18) for Theorem B.2, the Lemma B.4 holds. □

Lemma B.5. *For any $\mathcal{D}_v \in \Lambda_\zeta$, and $\mathcal{D}_t = \mathcal{D} - \mathcal{D}_v$, given $\delta \in (0,1)$, with probability at least $1-\delta$, we have $\forall f \in \mathcal{F}_{\mathcal{D}_t}$,*

$$\mathcal{E}_{\mathcal{P}}^{\Psi_l}(f) \leq \hat{\mathcal{E}}_{\mathcal{D}_v}^{\Psi_l}(f) + \widehat{\mathfrak{R}}_{\mathcal{D}_v}(\mathcal{F}_{\mathcal{D}_t}^{\Psi_l}) + 3\sqrt{\frac{\log\frac{2}{\delta}}{2m}}, \tag{19}$$

where $\mathcal{E}_{\mathcal{P}}^{\Psi_l}(f) = \mathbb{E}_{\mathcal{P}}[\mathbb{I}_{\Psi_l(f)=1}]$ is the generalization error on $\mathcal{P}$, $\hat{\mathcal{E}}_{\mathcal{D}_v}^{\Psi_l}(f) = \frac{1}{|\mathcal{D}_v|}\sum \mathbb{I}_{\Psi_l(f)=1}$ is the empirical error, $\mathcal{F}_{\mathcal{D}_t}^{\Psi_l} = \{\Psi_l \circ f : f \in \mathcal{F}_{\mathcal{D}_t}\}$ and $\Psi_l(f)$ is a loss-related indicator:

$$\Psi_l(f) = \begin{cases} 1 & \text{if } l(f) > \gamma, \\ 0 & \text{otherwise}, \end{cases} \tag{20}$$

where γ is a constant and $l(f)$ is the loss function. $l(f) > \gamma$ can be considered as a transformation to $h(x) \neq y$ in Lemma B.4.

Proof. From the definition of $\mathcal{F}_{\mathcal{D}_t}^{\Psi_l}$, there exits a $h_f \in \mathcal{F}_{\mathcal{D}_t}^{\Psi_l}$ that $h_f = \{\Psi_l \circ f\}$. Applying Lemma B.4, with probability at least $1-\delta$, we have $\forall f \in \mathcal{F}_{\mathcal{D}_t}$,

$$\begin{aligned} &\mathcal{E}_{\mathcal{P}}^{\Psi_l}(f) - \hat{\mathcal{E}}_{\mathcal{D}_v}^{\Psi_l}(f) \\ =&\mathbb{E}_{\mathcal{P}}[\mathbb{I}_{\Psi_l(f)=1}] - \frac{1}{|\mathcal{D}_v|}\sum \mathbb{I}_{\Psi_l(f)=1} \\ =&\mathbb{E}_{\mathcal{P}}[h_f] - \frac{1}{|\mathcal{D}_v|}\sum h_f \\ \leq&\widehat{\mathfrak{R}}_{\mathcal{D}_v}(\mathcal{F}_{\mathcal{D}_t}^{\Psi_l}) + 3\sqrt{\frac{\log\frac{2}{\delta}}{2m}}. \end{aligned} \tag{21}$$

□

Lemma B.6. *For any $\mathcal{D}_v \in \Lambda_\zeta$, and $\mathcal{D}_t = \mathcal{D} - \mathcal{D}_v$, let $g = \arg\inf_{f\in\mathcal{F}_{\mathcal{D}_t}} \mathcal{E}_{\mathcal{P}}^{\Psi_l}(f)$ and $\hat{g} = \arg\inf_{f\in\mathcal{F}_{\mathcal{D}_t}} \hat{\mathcal{E}}_{\mathcal{D}_v}^{\Psi_l}(f)$, given $\delta \in (0,1)$, with probability at least $1-\delta$, we have,*

$$\mathcal{E}_{\mathcal{P}}^{\Psi_l}(g) \geq \hat{\mathcal{E}}_{\mathcal{D}_v}^{\Psi_l}(\hat{g}) - \widehat{\mathfrak{R}}_{\mathcal{D}_v}(\mathcal{F}_{\mathcal{D}_t}^{\Psi_l}) - 3\sqrt{\frac{\log\frac{2}{\delta}}{2m}}. \tag{22}$$

Proof. From the definition of g and $\hat{g}$, we have $\hat{\mathcal{E}}_{\mathcal{D}_v}^{\Psi_l}(g) \geq \hat{\mathcal{E}}_{\mathcal{D}_v}^{\Psi_l}(\hat{g})$, thus

$$\begin{aligned} &\mathcal{E}_{\mathcal{P}}^{\Psi_l}(g) - \hat{\mathcal{E}}_{\mathcal{D}_v}^{\Psi_l}(\hat{g}) \\ =&\mathcal{E}_{\mathcal{P}}^{\Psi_l}(g) - \hat{\mathcal{E}}_{\mathcal{D}_v}^{\Psi_l}(g) + \hat{\mathcal{E}}_{\mathcal{D}_v}^{\Psi_l}(g) - \hat{\mathcal{E}}_{\mathcal{D}_v}^{\Psi_l}(\hat{g}) \\ \geq&\mathcal{E}_{\mathcal{P}}^{\Psi_l}(g) - \hat{\mathcal{E}}_{\mathcal{D}_v}^{\Psi_l}(g) \\ \geq&-\widehat{\mathfrak{R}}_{\mathcal{D}_v}(\mathcal{F}_{\mathcal{D}_t}^{\Psi_l}) - 3\sqrt{\frac{\log\frac{2}{\delta}}{2m}}. \end{aligned} \tag{23}$$

In the last inequality, we utilize the absolute value of the discrepancy of Lemma B.5:

$$|\mathcal{E}_{\mathcal{P}}^{\Psi_l}(f) - \hat{\mathcal{E}}_{\mathcal{D}_v}^{\Psi_l}(f)| \leq \widehat{\mathfrak{R}}_{\mathcal{D}_v}(\mathcal{F}_{\mathcal{D}_t}^{\Psi_l}) + 3\sqrt{\frac{\log\frac{2}{\delta}}{2m}}. \tag{24}$$

□

B.3. Proof of Theorem 3.1

Proof. We start with the $\mathcal{H}$-divergence between a source distribution $\mathcal{P}$ and a target distribution $\mathcal{Q}$, for any $f \in \mathcal{F}$, there exits a $h_f \in \mathcal{F}^{\Psi_l}$ that $h_f = \{\Psi_l \circ f\}$. From the definition of Ψ_l in Lemma B.5, if we assume $\mathbb{E}_{\mathcal{Q}}[\mathbb{I}_{\Psi_l(f)=1}] \geq \mathbb{E}_{\mathcal{P}}[\mathbb{I}_{\Psi_l(f)=1}]$, thus,

$$\begin{aligned} d_{\mathcal{F}^{\Psi_l}}(\mathcal{P}, \mathcal{Q}) &= 2 \sup_{h_f \in \mathcal{F}^{\Psi_l}} |\mathbb{E}_{\mathcal{P}}[h_f = 1] - \mathbb{E}_{\mathcal{Q}}[h_f = 1]| \\ &= 2 \sup_{f \in \mathcal{F}} |\mathbb{E}_{\mathcal{P}}[\mathbb{I}_{\Psi_l(f)=1}] - \mathbb{E}_{\mathcal{Q}}[\mathbb{I}_{\Psi_l(f)=1}]| \\ &= 2 \sup_{f \in \mathcal{F}} \{\mathbb{E}_{\mathcal{Q}}[\mathbb{I}_{\Psi_l(f)=1}] - \mathbb{E}_{\mathcal{P}}[\mathbb{I}_{\Psi_l(f)=1}]\} \\ &\leq 2 \sup_{f \in \mathcal{F}} \mathbb{E}_{\mathcal{Q}}[\mathbb{I}_{\Psi_l(f)=1}] - 2 \inf_{f \in \mathcal{F}} \mathbb{E}_{\mathcal{P}}[\mathbb{I}_{\Psi_l(f)=1}]. \end{aligned} \tag{25}$$

The assumption is realistic because the model trained on the $\mathcal{P}$ data should have a smaller classification loss on $\mathcal{P}$ than $\mathcal{Q}$. For any $\mathcal{D}_v \in \Lambda_\zeta$, and $\mathcal{D}_t = \mathcal{D} - \mathcal{D}_v$, we replace $\mathcal{F}$ by $\mathcal{F}_{\mathcal{D}_t}$, then

$$d_{\mathcal{F}^{\Psi_l}_{\mathcal{D}_t}}(\mathcal{P}, \mathcal{Q}) \leq 2 \sup_{f \in \mathcal{F}_{\mathcal{D}_t}} \mathbb{E}_{\mathcal{Q}}[\mathbb{I}_{\Psi_l(f)=1}] - 2 \inf_{f \in \mathcal{F}_{\mathcal{D}_t}} \mathbb{E}_{\mathcal{P}}[\mathbb{I}_{\Psi_l(f)=1}]. \tag{26}$$

Applying Theorem B.3, we have

$$\mathcal{E}^{\Psi_l}_{\mathcal{Q}}(f) \leq \mathcal{E}^{\Psi_l}_{\mathcal{P}}(f) + \sup_{f' \in \mathcal{F}_{\mathcal{D}_t}} \mathbb{E}_{\mathcal{Q}}[\mathbb{I}_{\Psi_l(f')=1}] - \inf_{f' \in \mathcal{F}_{\mathcal{D}_t}} \mathbb{E}_{\mathcal{P}}[\mathbb{I}_{\Psi_l(f')=1}] + \lambda^\star(\mathcal{D}_t), \tag{27}$$

where $\lambda^\star(\mathcal{D}_t) \geq \inf_{f' \in \mathcal{F}_{\mathcal{D}_t}} \{\mathcal{E}^{\Psi_l}_{\mathcal{P}}(f') + \mathcal{E}^{\Psi_l}_{\mathcal{Q}}(f')\}$. We let $\mathcal{C}_{gap} = \sup_{f' \in \mathcal{F}_{\mathcal{D}_t}} \mathbb{E}_{\mathcal{Q}}[\mathbb{I}_{\Psi_l(f')=1}] + \inf_{f' \in \mathcal{F}_{\mathcal{D}_t}} [\mathcal{E}^{\Psi_l}_{\mathcal{Q}}(f') + \mathcal{E}^{\Psi_l}_{\mathcal{P}}(f')]$, then

$$\mathcal{E}^{\Psi_l}_{\mathcal{Q}}(f) \leq \mathcal{E}^{\Psi_l}_{\mathcal{P}}(f) - \inf_{f' \in \mathcal{F}_{\mathcal{D}_t}} \mathbb{E}_{\mathcal{P}}[\mathbb{I}_{\Psi_l(f')=1}] + \mathcal{C}_{gap}. \tag{28}$$

In Equation (28), the first term of the right side can be replaced by (from Lemma B.5):

$$\mathcal{E}^{\Psi_l}_{\mathcal{P}}(f) \leq \hat{\mathcal{E}}^{\Psi_l}_{\mathcal{D}_v}(f) + \widehat{\mathfrak{R}}_{\mathcal{D}_v}(\mathcal{F}^{\Psi_l}_{\mathcal{D}_t}) + 3\sqrt{\frac{\log \frac{2}{\delta}}{2m}}. \tag{29}$$

Furthermore, the second term of the right side in Equation (28) can be replaced by (from Lemma B.6):

$$\inf_{f' \in \mathcal{F}_{\mathcal{D}_t}} \mathbb{E}_{\mathcal{P}}[\mathbb{I}_{\Psi_l(f')=1}] \geq \inf_{f' \in \mathcal{F}_{\mathcal{D}_t}} \frac{1}{|\mathcal{D}_v|} \sum \mathbb{I}_{\Psi_l(f')=1} - \widehat{\mathfrak{R}}_{\mathcal{D}_v}(\mathcal{F}^{\Psi_l}_{\mathcal{D}_t}) - 3\sqrt{\frac{\log \frac{2}{\delta}}{2m}}. \tag{30}$$

Combining Equation (28), Equation (29), Equation (30) and using the union bound, for any $\delta \in (0, 1)$, with probability at least $1 - 2\delta$, we have $\forall f \in \mathcal{F}_{\mathcal{D}_t}$,

$$\mathcal{E}^{\Psi_l}_{\mathcal{Q}}(f) \leq \hat{\mathcal{E}}^{\Psi_l}_{\mathcal{D}_v}(f) - \inf_{f' \in \mathcal{F}_{\mathcal{D}_t}} \frac{1}{|\mathcal{D}_v|} \sum \mathbb{I}_{\Psi_l(f')=1} + 2\widehat{\mathfrak{R}}_{\mathcal{D}_v}(\mathcal{F}^{\Psi_l}_{\mathcal{D}_t}) + 6\sqrt{\frac{\log \frac{2}{\delta}}{2m}} + \mathcal{C}_{gap}. \tag{31}$$

Since $|\mathcal{D}_v| = \zeta|\mathcal{D}|$, we let $\mathcal{C}_{\mathfrak{R}} = 2\sup_{\mathcal{D}'_v \in \Lambda_\zeta} \widehat{\mathfrak{R}}_{\mathcal{D}'_v}(\mathcal{F}^{\Psi_l}_{\mathcal{D}-\mathcal{D}'_v}) + 6\sqrt{\frac{\log \frac{2}{\delta}}{2\zeta|\mathcal{D}|}}$ and $\mathcal{C} = \mathcal{C}_{\mathfrak{R}} + \mathcal{C}_{gap}$, we have

$$\mathcal{E}^{\Psi_l}_{\mathcal{Q}}(f) \leq \hat{\mathcal{E}}^{\Psi_l}_{\mathcal{D}_v}(f) - \inf_{f' \in \mathcal{F}_{\mathcal{D}_t}} \frac{1}{|\mathcal{D}_v|} \sum \mathbb{I}_{\Psi_l(f')=1} + \mathcal{C}. \tag{32}$$

The proof of Theorem 3.1 is finished. □

C. Proof of Theorem 3.2

Proof. We first introduce the following Lemma:

Lemma C.1. *(Lemma A.5 in (Mairal, 2013)) Let $(a_n)_{n\leq 1}, (b_n)_{n\leq 1}$ be two non-negative real sequences such that the series $\sum_{i=1}^{\infty} a_n$ diverges, the series $\sum_{i=1}^{\infty} a_n b_n$ converges, and there exists $K > 0$ such that $|b_{n+1} - b_n| \leq K a_n$. Then the sequences $(b_n)_{n\leq 1}$ converges to 0.*

The objective function $\ell(w^{(k)}; \theta^{(k+1)})$ can be easily checked to be Lipschitz-smooth with constant L, and have ρ-bounded gradients with respect to training data. Therefore, we have:

$$
\begin{aligned}
&\ell(w^{(k+1)}; \theta^{(k+1)}) - \ell(w^{(k)}; \theta^{(k+1)}) \\
&\leq \langle \nabla \ell(w^{(k)}; \theta^{(k+1)}), w^{(k+1)} - w^{(k)} \rangle + \frac{L}{2}\|w^{(k+1)} - w^{(t)}\|_2^2 \\
&= \langle \nabla \ell(w^{(k)}; \theta^{(k+1)}), -\eta_k[\nabla \ell(w^{(k)}; \theta^{(k+1)}) + \psi^{(k)}] \rangle + \frac{L\eta_k^2}{2}\|\nabla \ell(w^{(k)}; \theta^{(k+1)}) + \psi^{(k)}\|_2^2 \\
&= -(\eta_k - \frac{L\eta_k^2}{2})\|\nabla \ell(w^{(k)}; \theta^{(k+1)})\|_2^2 + \frac{L\eta_k^2}{2}\|\psi^{(k)}\|_2^2 - (\eta_k - L\eta_k^2)\langle \nabla \ell(w^{(k)}; \theta^{(k+1)}), \psi^{(k)} \rangle,
\end{aligned}
$$

where $\psi^{(k)} = \nabla \ell(w^{(k)}; \theta^{(k+1)})|_{\mathcal{D}_t} - \nabla \ell(w^{(k)}; \theta^{(k+1)})$ is i.i.d. random variable with finite variance, since $\mathcal{D}_t$ are drawn i.i.d. with a finite number of samples. Furthermore, due to samples are drawn uniformly at random, we have $\mathbb{E}[\psi^{(k)}] = 0$ and $\mathbb{E}\|\psi^{(k)}\|_2^2 \leq \sigma^2$.

The above inequality implies that $\sum_{k=1}^{\infty} \eta_k \mathbb{E}[\|\nabla \ell(w^{(k)}; \theta^{(k+1)})\|] < \infty$, since $\sum_{k=0}^{\infty} \eta_k = \infty$, we need to substantiate $\lim_{k\to\infty} \mathbb{E}[\|\nabla \ell(w^{(k)}; \theta^{(k+1)})\|_2^2] = 0$. By Lemma C.1, it only needs to prove:

$$
\left| \mathbb{E}[\|\nabla \ell(w^{(k+1)}; \theta^{(k+2)})\|_2^2] - \mathbb{E}[\|\nabla \ell(w^{(k)}; \theta^{(k+1)})\|_2^2] \right| \leq C\eta_k,
$$

for some constant C. Based on the inequality $|(\|a\| + \|b\|)(\|a\| - \|b\|)| \leq \|a + b\|\|a - b\|$, we have

$$
\begin{aligned}
&\left| \mathbb{E}\left[\|\nabla \ell(w^{(k+1)}; \theta^{(k+2)})\|_2^2\right] - \mathbb{E}\left[\|\nabla \ell(w^{(k)}; \theta^{(k+1)})\|_2^2\right] \right| \\
&= \left| \mathbb{E}\left[(\|\nabla \ell(w^{(k+1)}; \theta^{(k+2)})\|_2 + \|\nabla \ell(w^{(k)}; \theta^{(k+1)})\|_2)(\|\nabla \ell(w^{(k+1)}; \theta^{(k+2)})\|_2 - \|\nabla \ell(w^{(k)}; \theta^{(k+1)})\|_2)\right] \right| \\
&\leq \mathbb{E}\left[\left| \|\nabla \ell(w^{(k+1)}; \theta^{(k+1)})\|_2 + \|\nabla \ell(w^{(k)}; \theta^{(k)})\|_2 \right| \left| \|\nabla \ell(w^{(k+1)}; \theta^{(k+2)})\|_2 - \|\nabla \ell(w^{(k)}; \theta^{(k+1)})\|_2 \right| \right] \\
&\leq \mathbb{E}\left[\|\nabla \ell(w^{(k+1)}; \theta^{(k+2)}) + \nabla \ell(w^{(k)}; \theta^{(k+1)})\|_2 \|\nabla \ell(w^{(k+1)}; \theta^{(k+2)}) - \nabla \ell(w^{(k)}; \theta^{(k+1)})\|_2 \right] \\
&\leq \mathbb{E}\left[(\|\nabla \ell(w^{(k+1)}; \theta^{(k+2)})\|_2 + \|\nabla \ell(w^{(k)}; \theta^{(k+1)})\|_2) \|\nabla \ell(w^{(k+1)}; \theta^{(k+2)}) - \nabla \ell(w^{(k)}; \theta^{(k+1)})\|_2 \right] \\
&\leq 2L\rho \mathbb{E}\left[(w^{(k+1)}; \theta^{(k+2)}) - (w^{(k)}; \theta^{(k+1)}) \right] \\
&\leq 2L\rho\eta_k\alpha_k \mathbb{E}\left[\|(\nabla \ell(w^{(k)}; \theta^{(k+1)}) + \psi^{(k)}, \nabla \ell(\theta^{(k+1)}) + \xi^{(k+1)})\|_2 \right] \\
&\leq 2L\rho\eta_k\alpha_k \mathbb{E}\left[\sqrt{\|\nabla \ell(w^{(k)}; \theta^{(k+1)}) + \psi^{(k)}\|_2^2} + \sqrt{\|\nabla \ell(\theta^{(k+1)}) + \xi^{(k+1)}\|_2^2} \right] \\
&\leq 2L\rho\eta_k\alpha_k \sqrt{\mathbb{E}\left[\|\nabla \ell(w^{(k)}; \theta^{(k+1)}) + \psi^{(k)}\|_2^2\right] + \mathbb{E}\left[\|\nabla \ell(\theta^{(k+1)}) + \xi^{(k+1)}\|_2^2\right]} \\
&\leq 2L\rho\eta_k\alpha_k \sqrt{\mathbb{E}\left[\|\nabla \ell(w^{(k)}; \theta^{(k+1)})\|_2^2\right] + \mathbb{E}\left[\|\psi^{(k)}\|_2^2\right] + \mathbb{E}\left[\|\xi^{(k+1)}\|_2^2\right] + \mathbb{E}\left[\|\nabla \ell(\theta^{(k+1)})\|_2^2\right]} \\
&\leq 2L\rho\eta_k\alpha_k \sqrt{2\sigma^2 + 2\rho^2} \\
&\leq 2\sqrt{2(\sigma^2 + \rho^2)} L\rho\alpha_1\eta_k,
\end{aligned}
$$

where $\xi^{(k+1)} = \nabla \ell(\theta^{(k+1)})|_{\mathcal{D}_v} - \nabla \ell(\theta^{(k+1)})$. There has $C = 2\sqrt{2(\sigma^2 + \rho^2)} L\rho\alpha_1$ that we can achieve $\lim_{k\to\infty} \mathbb{E}[\|\nabla \ell(w^{(k)}; \theta^{(k+1)})\|_2^2] = 0$ according to the Lemma C.1.

The proof is finished. □

In addition, regarding problem 9, it fundamentally presents an alternating optimization problem (alternatively updating $\mathcal{D}_v$ and $\mathbb{A}$). For this problem, we only need to discuss the convergence of updating $\mathbb{A}$ (since $\mathbb{A}$ is computed through $\mathcal{D}_t$, implying that the update of $\mathcal{D}_t$ has converged, hence $\mathcal{D}_v = \mathcal{D} - \mathcal{D}_t$ also converges). Experimental analysis of the convergence situation for this optimization problem is provided in the original paper, as seen in Figure 6. The detailed alternating update process is as follows:

Initialization. We first initialize $\mathbb{A}$ with the gradient of a sample randomly selected from $\mathcal{D}$. After initialization, we alternately update $\mathcal{D}_v$ and $\mathbb{A}$ as follows.

Updating $\mathcal{D}_v$. Given $\mathbb{A}$, $\mathcal{D}_v$ is updated by solving

$$\begin{aligned}&\max_{\mathcal{D}_v} \ell(w;\mathcal{D}_v) - \alpha\left\langle \nabla_w \ell(w;\mathcal{D}_v), \mathbb{A}\right\rangle\\ &\text{s.t. } \mathcal{D}_v \in \mathcal{D}, |\mathcal{D}_v|/|\mathcal{D}| = \zeta.\end{aligned}$$

The above equation indicates that the optimal $\mathcal{D}_v$ consists of $\zeta|\mathcal{D}|$ samples that have the largest values of $\ell(w;\mathcal{D}) - \alpha\left\langle \nabla_w \ell(w;\mathcal{D}), \mathbb{A}\right\rangle$. Therefore in practice, when fixing $\mathbb{A}$, we can compute and rank the values $\ell(w;\mathcal{D}) - \alpha\left\langle \nabla_w \ell(w;\mathcal{D}), \mathbb{A}\right\rangle$ for all samples in $\zeta|\mathcal{D}|$ and select the largest $\zeta|\mathcal{D}|$ samples to constitute the $\mathcal{D}_v$.

Updating $\mathbb{A}$. Given $\mathcal{D}_v$ ($\mathcal{D}_t = \mathcal{D} - \mathcal{D}_v$ is then given). We update $\mathbb{A}$ by

$$\begin{aligned}\mathbb{A} = \mathcal{G}_w^t &= \nabla_\theta \ell(\theta;\mathcal{D}_t, w)\\ &= \frac{1}{|\mathcal{D}_t|}\sum \nabla_\theta \ell(\theta;\mathcal{D}_t)|_{\theta=w}\\ &= \frac{1}{|\mathcal{D}_t|}\sum \nabla_w \ell(w;\mathcal{D}_t),\end{aligned}$$

where the second equation utilizes the fact that w is the initialization of θ.

The experimental results illustrated in Figure 6 demonstrate that the problem 9 can easily achieve convergence, typically within just 2-3 iterations in a 10-iteration setting. This rapid convergence is fundamentally attributed to the data transfer that occurs between $\mathcal{D}_t$ and $\mathcal{D}_v$. As a consequence of this transfer, A becomes progressively smaller after each iteration compared to the previous one. Given the constant sample sizes of $\mathcal{D}_t$ and $\mathcal{D}_v$, a stable point of convergence is consistently reached after a handful of iterations. For the theoretical analysis of the convergence, we take it as our future work.

D. Experimental Details and Additional Results

In this section, we provide the experimental details of our paper.

D.1. D4RL Experiments

Data collection The datasets in D4RL are generated using the following methodology:

- Medium: 1M samples are derived from a policy trained to achieve approximately one-third of the expert's performance.
- Medium-replay: The replay buffer of a policy trained up to the performance level of the medium agent.
- Medium-expert: A balanced mix of medium and expert data, with a 50-50 split.

For all datasets, we utilize the v2 version.

Implementation details All of the baseline algorithms including CQL, TD3+BC, and MCQ come from the code library [https://github.com/yihaosun1124/OfflineRL-Kit].

D.2. Additional Results

We also provide the results of comparing our ADS framework with different algorithms on AntMaze tasks. We must emphasize that the baseline algorithm we employ is intrinsically unable to cope with the antmaze task. Consequently, our equipped algorithm, in comparison to other algorithms, is still not in a position of advantage. Nevertheless, it continues to demonstrate competitive performance on the antmaze-umaze and antmaze-umaze-diverse tasks.

Table 4: Average normalized scores over the final 10 evaluations and 5 seeds. We **bold** the highest mean and underline the second place.

Task Name	BC	IQL	POR	PRDC	STR	**CQL+ADS**	**TD3+BC+ADS**	**MCQ+ADS**
antmaze-umaze	66.8	89.6	90.6	**98.8**	93.6	95.0	94.0	10.0
antmaze-umaze-diverse	56.8	66.7	71.3	**90.0**	77.4	82.0	**90.0**	30.0
antmaze-medium-play	0.0	76.4	**84.6**	82.8	82.6	30.0	10.0	0.0
antmaze-medium-diverse	0.0	72.8	79.2	78.8	**87.0**	64.0	10.0	0.0
antmaze-large-play	0.0	42.0	**58.0**	54.8	42.8	10.0	0.0	0.0
antmaze-large-diverse	0.0	45.6	**73.4**	50.0	46.8	12.0	0.0	0.0