Beyond OOD State Actions: Supported Cross-Domain Offline Reinforcement Learning

Jinxin Liu*, Ziqi Zhang*, Zhenyu Wei, Zifeng Zhuang, Yachen Kang, Sibo Gai, Donglin Wang†

School of Enginneering, Westlake University

Abstract

Offline reinforcement learning (RL) aims to learn a policy using only pre-collected and fixed data. Although avoiding the time-consuming online interactions in RL, it poses challenges for out-of-distribution (OOD) state actions and often suffers from data inefficiency for training. Despite many efforts being devoted to addressing OOD state actions, the latter (data inefficiency) receives little attention in offline RL. To address this, this paper proposes the cross-domain offline RL, which assumes offline data incorporate additional source-domain data from varying transition dynamics (environments), and expects it to contribute to the offline data efficiency. To do so, we identify a new challenge of OOD transition dynamics, beyond the common OOD state actions issue, when utilizing cross-domain offline data. Then, we propose our method BOSA, which employs two support-constrained objectives to address the above OOD issues. Through extensive experiments in the cross-domain offline RL setting, we demonstrate BOSA can greatly improve offline data efficiency: using only 10% of the target data, BOSA could achieve 74.4% of the SOTA offline RL performance that uses 100% of the target data. Additionally, we also show BOSA can be effortlessly plugged into model-based offline RL and noising data augmentation techniques (used for generating source-domain data), which naturally avoids the potential dynamics mismatch between target-domain data and newly generated source-domain data.

Introduction

Data-driven offline reinforcement learning (RL) holds the promise to learn a control policy from fixed and static dataset [Levine et al. 2020] while avoiding the costly and time-consuming online data acquisition in standard RL. Nevertheless, a notorious challenge in offline RL is the presence of extrapolation error, which tends to drive the learning policy towards OOD state actions and yields collapsed behaviors [Fujimoto, Meger, and Precup 2019]. Many recent works have been dedicated to tackling this challenge, leveraging policy regularization, value conservation, or supervised regression [Chen et al. 2021, Kostrikov, Nair, and Levine 2021, Fujimoto and Gu 2021]. Yet, such offline solutions tend to perform well when plenty of pre-collected training data is available and the testing environment keeps consistent with the data-collecting environment. However, collecting large-scale offline data for a domain-specific environment is still labor-intensive and expensive. Additionally, once we reduce the offline training data, the final performance drops quickly (Figure 1 *left* and Figure 2). Thus, although that advanced offline RL methods tend to be saturated with abundant data, the underlying offline data inefficiency makes them difficult to use for data-scarce scenarios.

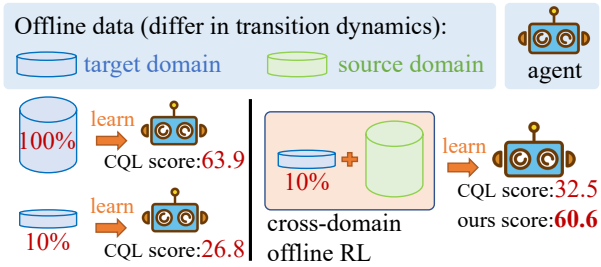

Figure 1: (*left*) The vanilla single-domain offline RL setting. (*right*) Our cross-domain offline RL, which learns a policy from offline cross-domain data, consists of *limited target-domain data* and plenty of source-domain data. Scores in this diagram are averaged over 9 D4RL Mujoco tasks, which serve as the target domain.

Motivated by the sample-efficient domain adaptation, this paper investigates the cross-domain offline RL. Specifically, we assume the agent can access a large amount of source-domain offline data from one or multiple separate source environments, and we are interested in adapting the learning policy with limited target-domain offline data (as shown in Figure 1 *right*). We also emphasize that such an additional source-domain data assumption can be naturally satisfied in practice. For example, it is common to have a large amount of data on driving behaviors in city roads (source domain), while only having a few samples in mountain environments (target domain) for autonomous driving tasks.

However, prior offline RL methods often work poorly in the cross-domain setting, particularly for source-domain data with large transition dynamics differences. As we show

*These authors contributed equally.

†Corresponding author: <wangdonglin@westlake.edu.cn>

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

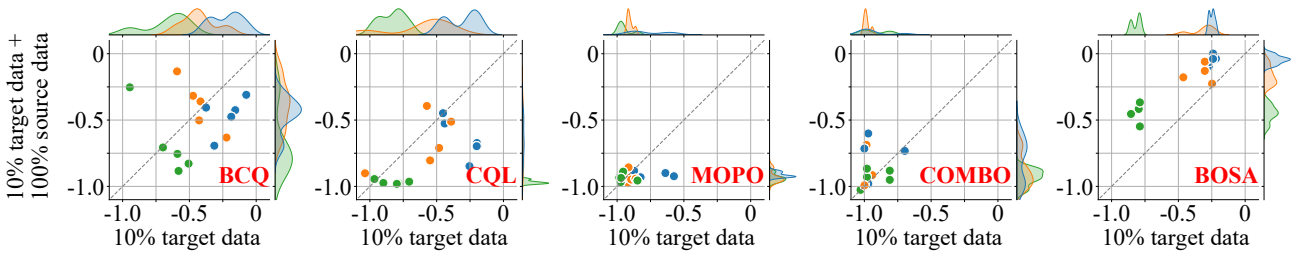

Figure 2: Performance difference between single-domain and cross-domain offline RL settings, where different colors represent different data qualities (blue: *-medium, green: *-medium-replay, orange: *-medium-expert) and multiple dots with the same color represent scores on multiple cross-domain tasks. We take D4RL [Fu et al. 2020] as the target domain, and take a modified D4RL (with transition dynamics shift) as the source domain. The x-axis represents the normalized performance improvement when only 10% of D4RL data (target) is available, and the y-axis represents the performance difference between learning with cross-domain offline data (10%target + 100%source) and learning with abundant target-domain offline data (100%target). We can observe that when we reduce the offline training data size, most offline RL methods suffer a clear drop in performance (*i.e.*, values on the x-axis is less than 0). Further, introducing additional source-domain data also does not bring any significant performance benefits (*i.e.*, below the dashed line), with the exception of our BOSA.

in Figure 2, simply combing the cross-domain data does not bring a positive performance improvement to that using only the limited target data. In fact, it can even result in a decline in performance for several tasks and offline baselines. In this paper, we attribute this transfer failure to the presence of OOD transition dynamics beyond those (OOD state actions) commonly encountered in single-domain offline RL setting. Intuitively, the presence of source-domain data can bias the agent's policy to visit transitions in source environments as long as they receive high values. Thus, cross-domain data easily lead the policy overfits to the source environment and hinders its transfer ability to the target domain.

To address this issue, we present BOSA (**B**eyond **O**OD **S**tate **A**ctions) for the cross-domain offline RL. Simply, BOSA aims to leverage cross-domain offline data (plenty of source data and limited target data) to improve the data efficiency for offline RL. The key idea behind BOSA is that we substantiate the inherent offline extrapolation error through OOD state actions and OOD transition dynamics, and try to filter out offline transitions that might cause state-actions shift and transition dynamics mismatch. Specifically, we propose a supported policy optimization for eliminating OOD state actions and a supported value optimization for addressing OOD transition dynamics. Additionally, to avoid exploiting overestimated Q-values for policy optimization over source-domain data, we introduce a conservation [Kumar et al. 2020] over the value optimization objective.

We conduct experiments with a variety of source domains that have transition dynamics mismatch and demonstrate that BOSA contributes to significant gains on learning from cross-domain offline data. Further, we show that BOSA can be plugged into more general cross-domain offline settings: model-based RL and (noising) data augmentation. Similarly, augmenting the offline data by a pseudo-transition model or random noise will also encounter OOD transitions that are not consistent with the target environment. Thus, we can naturally view the generated or noised data as a source domain, and then apply BOSA for handling dynamics mismatch.

The primary contributions of this work are as follows: 1) We identify an OOD transition dynamics issue in cross-domain offline RL and propose BOSA for handling it. 2) We show BOSA can greatly improve offline data efficiency and outperform prior state-of-the-art methods. 3) We show BOSA can be naturally plugged into model-based RL and (noising) data augmentation scenarios while eliminating the commonly overlooked OOD transition dynamics and thus facilitating positive transfer.

Related Work

Offline Reinforcement Learning. In standard (single domain) offline RL, the agent tries to learn a policy from static and fixed data that are pre-collected from a (target) environment [Levine et al. 2020, Fujimoto, Meger, and Precup 2019]. Yet, the overestimation of OOD state-action issues is often identified as a major issue. To solve this, most model-free offline methods typically augment existing off-policy algorithms with a penalty [Geist, Scherrer, and Pietquin 2019] measuring a divergence between the learning policy and the offline data (or the behavior policy). Recently, there have been many offline RL methods proposed to implement such a penalty, by introducing support constrains [Fujimoto, Meger, and Precup 2019, Ghasemipour, Schuurmans, and Gu 2021, Wu et al. 2022] or policy regularization [Kostrikov, Nair, and Levine 2021, Fujimoto and Gu 2021, Peng et al. 2019, Wu, Tucker, and Nachum 2019, Kumar et al. 2019a, Nachum et al. 2019]. Alternatively, some offline RL methods introduce the uncertainty estimation [Rezaeifar et al. 2021, An et al. 2021, Wu et al. 2021, Ma, Jayaraman, and Bastani 2021, Bai et al. 2022] or the conservation [Kumar et al. 2020, Lyu et al. 2022] over values to overcome the potential overestimation for OOD state actions. In the same spirit, model-based offline RL methods similarly employ the distribution (correcting) regularization [Hishinuma and Senda 2021, Zhang et al. 2022, Yang et al. 2022], uncertain estimation [Kidambi et al. 2020, Lu et al. 2021], and value conservation [Yu et al. 2021] to eliminate the OOD issues.

Cross-Domain Reinforcement Learning. To improve the sample efficiency, cross-domain RL introduces an additional source domain and regards the original task as the target domain. In this work, we assume source and target domains differ in their transition dynamics. This cross-dynamics setting has also helped improve the training efficiency for online RL [Wulfmeier, Posner, and Abbeel 2017, Eysenbach et al. 2020, 2021], reward-free RL [Liu et al. 2021], and reverse RL (imitation learning) [Fickinger et al. 2021, Viano et al. 2021, Franzmeyer, Torr, and Henriques 2022]. In the offline RL setting, our BOSA method bears a resemblance to that in DARA [Liu, Zhang, and Wang 2022], but a crucial algorithmic difference is that *DARA explicitly models both the source-domain and target-domain transitions* while *we only model target-domain transitions and do not model the source-domain transitions*, and also lead to improved performance empirically.

Background

Reinforcement Learning. We consider reinforcement learning (RL) in the Markov decision process (MDP) defined by a tuple $\mathcal{M} := (\mathcal{S}, \mathcal{A}, T, r, \gamma, p_0)$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ denotes the action space, $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}_+$ is the transition (dynamics) probability, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $\gamma \in [0, 1]$ is the discount factor, and p_0 is the distribution of initial states. The goal of RL is to find a policy $\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_+$ that maximizes the expected discounted cumulative return $J(\pi) = \mathbb{E}_{\tau\sim\pi(\tau)}\left[\sum_{t=0}^{T} \gamma^t r_t\right]$, where $\tau := (\mathbf{s}_0, \mathbf{a}_0, \mathbf{s}_1, \mathbf{a}_1 \cdots)$ denotes the rollout trajectory and $r_t := r(\mathbf{s}_t, \mathbf{a}_t)$ denotes the reward of transition $(\mathbf{s}_t, \mathbf{a}_t)$. Here we sightly abuse the notation $\pi(\tau)$ to denote the trajectory distribution induced by executing policy $\pi(\mathbf{a}|\mathbf{s})$ in $\mathcal{M}$, *i.e.*, $\mathbf{s}_0 \sim p_0(\mathbf{s}_0)$, $\mathbf{a}_t \sim \pi(\mathbf{a}_t|\mathbf{s}_t)$, and $\mathbf{s}_{t+1} \sim T(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$.

To optimize the above objective $J(\pi)$, off-policy RL methods introduce a Q-function defined by $Q^\pi(\mathbf{s}, \mathbf{a}) = \mathbb{E}_{\tau\sim\pi(\tau)}\left[\sum_{t=0}^{T} \gamma^t r_t|\mathbf{s}_0 = \mathbf{s}_t, \mathbf{a}_0 = \mathbf{a}\right]$. One property of such a Q-value function is that it satisfies the Bellman consistency criterion given by $\mathcal{T}^\pi Q^\pi(\mathbf{s}, \mathbf{a}) = Q^\pi(\mathbf{s}, \mathbf{a})\ \forall(\mathbf{s}, \mathbf{a})$, where $\mathcal{T}^\pi Q(\mathbf{s}, \mathbf{a}) := r(\mathbf{s}, \mathbf{a}) + \gamma\mathbb{E}_{\mathbf{s}'\sim T(\mathbf{s}'|\mathbf{s},\mathbf{a}),\mathbf{a}'\sim\pi(\mathbf{a}'|\mathbf{s}')}[Q(\mathbf{s}', \mathbf{a}')]$ is the Bellman operator. Given an experience replay buffer $\mathcal{B} := \{(\mathbf{s}, \mathbf{a}, r, \mathbf{s}')\}$ (that will be updated by executing the learning policy in the environment), standard approximate dynamic programming and actor-critic methods use this (Bellman consistency) criterion to iteratively learn a parametric Q-function Q_ϕ by minimizing,

$$\mathcal{L}_\mathcal{B}(Q_\phi) := \mathbb{E}_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim\mathcal{B}}\left[Q_\phi(\mathbf{s}, \mathbf{a}) - \mathcal{T}^{\pi_\theta} Q_{\bar{\phi}}(\mathbf{s}, \mathbf{a})\right]^2, \quad (1)$$

and, following the deterministic policy gradient theorem, learn a parametric policy π_θ by maximizing:

$$\mathcal{J}_\mathcal{B}(\pi_\theta) := \mathbb{E}_{\mathbf{s}\sim\mathcal{B},\mathbf{a}\sim\pi_\theta(\mathbf{a}|\mathbf{s})}\left[Q_\phi(\mathbf{s}, \mathbf{a})\right], \quad (2)$$

where ϕ and θ are the parameters of the Q-function and the policy respectively, $\bar{\phi}$ is an EMA (exponential moving average) of ϕ: $\bar{\phi} \leftarrow \alpha\bar{\phi} + (1 - \alpha)\phi$, and α is the target network EMA parameter. For simplicity of notation, we drop the subscript t and use $\mathbf{s}'$ to denote the state at the next time step.

Offline Reinforcement Learning. In offline RL [Levine et al. 2020], we can not execute the learning policy in the environment to collect new online transitions, but rather have access to a fixed offline dataset $\mathcal{D} := \{(\mathbf{s}, \mathbf{a}, r, \mathbf{s}')\}$, collected by an unknown behavior policy (or by multiple behaviors) $\pi_\beta(\mathbf{a}|\mathbf{s})$ in the environment $\mathcal{M}$.

However, naively performing policy evaluation (Equation 1) and taking the expectation over fixed offline data $\mathcal{D}$ will inevitably require the Q-function to extrapolate to OOD state-action pairs. Iterating the offline policy improvement and policy evaluation *i.e.*, $\max \mathcal{J}_\mathcal{D}(\pi_\theta)$ and $\min \mathcal{L}_\mathcal{D}(Q_\phi)$, the potential extrapolation error will be further amplified, biasing the learned Q-value towards erroneously overestimated values and further biasing the learned policy towards unconfident actions. Unlike the online RL formulation, such an induced extrapolation error and the unconfident action will never be corrected due to the inability to collect new interaction data over the task environment.

Problem Formulation

Cross-Domain Offline RL

Problem statement. In our cross-domain offline RL setting, we assume the static offline data are collected from a set of environments/MDPs with varying transition dynamics (and different data-collecting behavior policies), rather than from a single fixed environment like the vanilla single-domain offline RL formulation in Levine et al. [2020].

Formally, considering a target offline RL task specified through $\mathcal{M}_{\text{target}}$, we define the mixed cross-domain offline data $\mathcal{D}_{\text{mix}} := \mathcal{D}_{\text{target}} \cup \mathcal{D}_{\text{source}}$, where $\mathcal{D}_{\text{target}}$ denotes the (limited) target data collected from the target MDP $\mathcal{M}_{\text{target}}$ and $\mathcal{D}_{\text{source}}$ denotes the source data collected from a set of source MDPs $\{\mathcal{M}^1_{\text{source}}, \cdots, \mathcal{M}^n_{\text{source}}\}$. We also assume that all of these MDPs share the same state space, action space, and reward function, while differing in their transition dynamics. The goal of cross-domain offline RL is to learn a policy that maximizes the expected return over the target environment $\mathcal{M}_{\text{target}}$ using the static cross-domain offline data $\mathcal{D}_{\text{mix}}$.

Compared to the vanilla single-domain offline RL, the cross-domain formulation naturally preserves the benefit of offline data transfer. Incorporating source-domain data can alleviate the challenge of offline RL data efficiency, which often requires a large number of target offline samples and demands substantial data collection efforts on target environment. Thus, we expect that cross-domain offline RL formulation can reduce the demanding requirements on target.

OOD Issues in Cross-Domain Offline RL

Before discussing the OOD issues in the cross-domain setting, we first review the extrapolation error and take a deeper look at how to eliminate it with support constraints.

OOD state actions. In typical single-domain offline RL (learning with offline data $\mathcal{D}$), performing the offline policy evaluation will suffer from the empirical extrapolation error $\mathbb{E}_{\mu_\pi(\mathbf{s})\pi(\mathbf{a}|\mathbf{s})}|\epsilon(\mathbf{s}, \mathbf{a})|$, where $\epsilon(\mathbf{s}, \mathbf{a}) = \mathcal{T}^\pi Q(\mathbf{s}, \mathbf{a}) - \hat{\mathcal{T}}^\pi Q(\mathbf{s}, \mathbf{a})$, and $\hat{\mathcal{T}}$ denotes the empirical Bellman operator implicitly defined by the offline transition dynamics $\hat{T}$ by randomly sampling transitions $(\mathbf{a}, \mathbf{s}, r, \mathbf{s}')$ from $\mathcal{D}$.

Thus, to evaluate a policy π exactly, we are required to ensure $\mathbb{E}_{\mu_\pi(\mathbf{s})\pi(\mathbf{a}|\mathbf{s})}|\epsilon(\mathbf{s},\mathbf{a})| = 0$ at relevant state-action transitions. For this purpose, BCQ [Fujimoto, Meger, and Precup 2019] (or BEAR [Kumar et al. 2019b]) identifies OOD state actions as the key source of the extrapolation error and thus proposes the following theorem.

Theorem 1. *Under deterministic environment dynamics, if we can ensure all possible state actions are contained in offline data $\mathcal{D}$, we can guarantee $T(\mathbf{s}'|\mathbf{s},\mathbf{a}) - \hat{T}(\mathbf{s}'|\mathbf{s},\mathbf{a}) = 0$ for all $\mathbf{s}' \in \mathcal{S}$ and $(\mathbf{s},\mathbf{a})$ such that $\mu_\pi(\mathbf{s})\pi(\mathbf{a}|\mathbf{s}) > 0$. Then, $\mathbb{E}_{\mu_\pi(\mathbf{s})\pi(\mathbf{a}|\mathbf{s})}|\epsilon(\mathbf{s},\mathbf{a})| = 0$ will be naturally satisfied.*

Based on such a theorem, BCQ suggests that one can eliminate the extrapolation error by instantiating offline RL over a support-constrained paradigm *which constraints the learned policy (actions) within the support set of the offline dataset* [Ghasemipour, Schuurmans, and Gu 2021].

OOD transition dynamics. We can observe that the above support-constrained derivation relies on the identification that if we can ensure $(\mathbf{s},\pi_\theta(\mathbf{s})) \in \mathcal{D}$ for all $\mathbf{s} \in \mathcal{D}$, then we can achieve zero extrapolation error. However, such an identification is only limited to the single-domain offline RL. The main reason is that in the cross-domain setting, it is easy to find a transition $(\mathbf{s},\mathbf{a},\mathbf{s}')$ such that $(\mathbf{s},\mathbf{a},\mathbf{s}') \in \mathcal{D}_{\text{mix}}$ and $T_{\text{target}}(\mathbf{s}'|\mathbf{s},\mathbf{a}) \neq \hat{T}_{\text{mix}}(\mathbf{s}'|\mathbf{s},\mathbf{a})$. Thus, we can not guarantee $T_{\text{target}}(\mathbf{s}'|\mathbf{s},\mathbf{a}) - \hat{T}_{\text{mix}}(\mathbf{s}'|\mathbf{s},\mathbf{a}) = 0$ for all $(\mathbf{s},\mathbf{a}) \in \mathcal{D}_{\text{mix}}$ and $\mathbf{s}' \in \mathcal{S}$ like Theorem 1. Even though we can restrict state actions $(\mathbf{s},\pi_\theta(\mathbf{s}))$ to lie in the support set of the offline data $\mathcal{D}_{\text{mix}}$, performing policy evaluation will still accumulate non-zero extrapolation errors due to the transition dynamics mismatch (between the target and source MDPs).

Lemma 2. *Under the cross-domain offline RL setting, constraining the policy within the support of cross-domain data $\mathcal{D}_{mix}$ can not guarantee zero extrapolation error for the target environment when performing offline policy evaluation.*

Thus, beyond the common OOD state actions issue identified in previous offline works, cross-domain offline RL also suffers from OOD transition dynamics (transition dynamics mismatch). In the next section, we will describe how our method, BOSA, addresses both of these issues jointly by introducing supported policy and value optimization.

Supported Policy and Value Optimization

In this section, we present BOSA (beyond OOD state actions). As aforementioned, cross-domain offline RL suffers from both OOD state actions and OOD dynamics issues. BOSA tackles these two issues jointly: considering the actor-critic framework, BOSA eliminates OOD state actions through a supported policy optimization and addresses OOD dynamics through a supported value optimization.

Supported Policy Optimization

Following the same spirit of BCQ [Fujimoto, Meger, and Precup 2019], we first introduce the supported policy optimization to address the OOD state actions issue. Note that the naive BCQ algorithm formulates the supported policy optimization over the Q-learning algorithm, utilizing the *optimal* Bellman operator. Here we rewrite it on top of the actor-critic methods, separating the policy improvement and the policy evaluation (value optimization). Specifically, considering a parametric Q-function Q_ϕ and a policy π_θ, we can perform offline *support-constrained* policy optimization by

$$\max_{\pi_\theta} \; \mathcal{J}_{\mathcal{D}_{\text{mix}}}(\pi_\theta) := \mathbb{E}_{\mathbf{s}\sim\mathcal{D}_{\text{mix}},\mathbf{a}\sim\pi_\theta(\mathbf{a}|\mathbf{s})}\left[Q_\phi(\mathbf{s},\mathbf{a})\right], \quad \text{s.t. } (\mathbf{s},\pi_\theta(\mathbf{s})) \in \mathcal{D}_{\text{mix}},\ \forall \mathbf{s} \in \mathcal{D}_{\text{mix}}, \tag{3}$$

where Equation 3 constraints the learned policy (actions) within the support set of the offline dataset, thus eliminating the OOD state actions issue as specified by Theorem 1.

Unfortunately, directly optimizing Equation 3 is often computationally expensive and requires a tabular representation for the state actions, which can quickly become impractical for large problems. Instead, we approximate it through an alternative objective by using a behavior policy $\hat{\pi}_{\beta_{\text{mix}}}$:

$$\max_{\pi_\theta} \; \mathcal{J}_{\mathcal{D}_{\text{mix}}}(\pi_\theta) := \mathbb{E}_{\mathbf{s}\sim\mathcal{D}_{\text{mix}},\mathbf{a}\sim\pi_\theta(\mathbf{a}|\mathbf{s})}\left[Q_\phi(\mathbf{s},\mathbf{a})\right], \quad \text{s.t. } \mathbb{E}_{\mathbf{s}\sim\mathcal{D}_{\text{mix}}}\left[\hat{\pi}_{\beta_{\text{mix}}}(\pi_\theta(\mathbf{s})|\mathbf{s})\right] > \epsilon_{\text{th}}, \tag{4}$$

where $\hat{\pi}_{\beta_{\text{mix}}} = \arg\max_{\beta_{\text{mix}}} \mathbb{E}_{(\mathbf{s},\mathbf{a})\sim\mathcal{D}_{\text{mix}}}\left[\log \hat{\pi}_{\beta_{\text{mix}}}(\mathbf{a}|\mathbf{s})\right]$ and ϵ_{th} denotes the threshold above which we retain the (state-action) support constrains. Note that compared to the common policy distribution matching regularization in prior offline methods (*e.g.*, restricting $\text{KL}(\pi_\theta(\mathbf{a}|\mathbf{s})||\pi_\beta(\mathbf{a}|\mathbf{s})) \leq \epsilon_{\text{th}}$), Equation 4 is essentially performing support constraints instead of distribution matching, which thus avoids the difficulty to trade off mode-covering and mode-seeking issues in distribution matching objective.

Supported Value Optimization

Next, we discuss how to tackle OOD transition dynamics. Similar to the above support-constrained policy optimization, the key idea is to constrain the value optimization (policy evaluation) over the transitions that do not expose the dynamics mismatch. As an example, one can directly use only the target-domain offline data $\mathcal{D}_{\text{target}}$ to perform value optimization by minimizing $\mathcal{L}_{\text{target}}(Q_\phi)$, where $\mathcal{L}_{\text{target}}(Q_\phi) := \mathbb{E}_{(\mathbf{s},\mathbf{a},r,\mathbf{s}')\sim\mathcal{D}_{\text{target}},\mathbf{a}'\sim\pi_\theta(\mathbf{a}'|\mathbf{s}')}\left[Q_\phi(\mathbf{s},\mathbf{a}) - r - Q_{\bar{\phi}}(\mathbf{s}',\mathbf{a}')\right]^2$. However, this naive method tends to suffer from low data efficiency and struggles with scarce (target-domain) offline data, especially in data-expensive offline RL tasks.

To facilitate data-efficient value optimization, leveraging the source-domain data $\mathcal{D}_{\text{source}}$ is thus essential in the cross-domain setting. Following the same spirit of policy support constraints, we introduce the supported value optimization:

$$\min_{Q_\phi} \; \mathcal{L}_{\text{mix}}(Q_\phi) := \mathbb{E}_{(\mathbf{s},\mathbf{a},r,\mathbf{s}')\sim\mathcal{D}_{\text{mix}},\mathbf{a}'\sim\pi_\theta(\mathbf{a}'|\mathbf{s}')}\Big[\delta(Q_\phi)\cdot\mathbb{1}(\hat{T}_{\text{target}}(\mathbf{s}'|\mathbf{s},\mathbf{a}) > \epsilon'_{\text{th}})\Big] + \mathbb{E}_{(\mathbf{s},\mathbf{a})\sim\mathcal{D}_{\text{source}}}\left[Q_\phi(\mathbf{s},\mathbf{a})\right], \tag{5}$$

where $\delta(Q_\phi) := \big(Q_\phi(\mathbf{s},\mathbf{a}) - r - Q_{\bar{\phi}}(\mathbf{s}',\mathbf{a}')\big)^2$, $\hat{T}_{\text{target}}$ denotes the estimated target-domain transition dynamics, *i.e.*, $\hat{T}_{\text{target}} = \arg\max_{\hat{T}_{\text{target}}} \mathbb{E}_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim\mathcal{D}_{\text{target}}}\Big[\log \hat{T}_{\text{target}}(\mathbf{s}'|\mathbf{s},\mathbf{a})\Big]$, and $\mathbb{1}(\cdot)$ denotes the indicator function, which equals 1 if the argument is true, and 0 otherwise. Intuitively, the indicator function filters out OOD transitions that are likely

		single-domain setting (100% D4RL → 10% D4RL)				cross-domain setting (10% D4RL + source data)			
		BCQ	MOPO	CQL	SPOT	BCQ	CQL	SPOT	BOSA
mass	ha-m	40.7 → 37.6	42.3 → 3.2	44.4 → 35.4	58.4 → 45.4	35.1	32.2	50.3	**58.3** ± 2.5
	ha-mr	38.2 → 1.1	53.1 → −0.1	46.2 → 0.6	52.2 → 9.8	**40.1**	3.3	37.6	37.2 ± 0.7
	ha-me	64.7 → 37.3	63.5 → 4.2	62.4 → −3.3	86.9 → 46.2	26.4	12.9	33.8	**51.6** ± 0.1
	ho-m	54.5 → 37.1	28 → 4.1	58 → 43	86 → 62.5	25.7	44.9	**85.9**	82.4 ± 2.1
	ho-mr	33.1 → 9.3	67.5 → 1	48.6 → 9.6	100.2 → 13.7	28.7	1.4	15.5	**39.7** ± 0.1
	ho-me	110.9 → 58	23.7 → 1.6	98.7 → 59.7	99.3 → 69	75.4	53.6	75.5	**104.2** ± 0.5
	wa-m	53.1 → 32.8	17.8 → 7	79.2 → 42.9	86.4 → 65.4	50.9	80	22.5	**83** ± 2.9
	wa-mr	15 → 6.9	39 → 5.1	26.7 → 4.6	91.6 → 18.6	14.9	0.8	16	**21.4** ± 2
	wa-me	57.5 → 32.5	44.6 → 5.3	111 → 49.5	112 → 84	55.2	63.5	14.3	**86.5** ± 0.6
joints	ha-m	40.7 → 37.6	42.3 → 3.2	44.4 → 35.4	58.4 → 45.4	40	40.7	50.1	**56.2** ± 0.27
	ha-mr	38.2 → 1.1	53.1 → −0.1	46.2 → 0.6	52.2 → 9.8	39.4	2	41	**51.3** ± 1.1
	ha-me	64.7 → 37.3	63.5 → 4.2	62.4 → −3.3	86.9 → 46.2	**55.3**	7.7	38.1	52.8 ± 0.45
	ho-m	54.5 → 37.1	28 → 4.1	58 → 43	86 → 62.5	49	58	41.5	**78** ± 7.3
	ho-mr	33.1 → 9.3	67.5 → 1	48.6 → 9.6	100.2 → 13.7	23.8	2.6	23	**32.7** ± 1.3
	ho-me	110.9 → 58	23.7 → 1.6	98.7 → 59.7	99.3 → 69	96	73.4	52	**96.4** ± 0.5
	wa-m	53.1 → 32.8	17.8 → 7	79.2 → 42.9	86.4 → 65.4	44.9	73.2	38.8	**86.5** ± 5.6
	wa-mr	15 → 6.9	39 → 5.1	26.7 → 4.6	91.6 → 18.6	9.8	1.4	10.7	**38.2** ± 4.7
	wa-me	57.5 → 32.5	44.6 → 5.3	111 → 49.5	112 → 84	40.6	**109.9**	74.3	85.8 ± 0.3
Average† (%)		−48.3%	−88.7%	−61.5%	−47.1%				
Average‡ (%)						−50.1%	−59.4%	−50.9%	−**25.6**%

Table 1: Results on the single-domain and cross-domain offline RL. We take the baseline results (single-domain setting with 100% D4RL) from their original papers. We average our results over 5 seeds and for each seed, we compute the normalized average score using 10 episodes. In the cross-domain setting, the numbers to the left of the arrow (→) represent the scores trained on 100% D4RL data, and the numbers to the right of that represent the scores trained on only 10% D4RL data. In the left panel (single-domain setting), Average† represents the average performance change when the offline data is reduced (100%→10%). In the right panel (cross-domain setting), Average‡ represents the average performance difference between *the cross-domain results* and *the best results among baselines that are trained with 100% D4RL*. In each line, we bold the best score among baselines that are trained with 10% D4RL data, *i.e.*, including the single-domain 10% D4RL setting and the cross-domain setting. (ha: halfcheetah. ho: hopper. wa: walker2d. m: medium. mr: medium-replay. me: medium-expert.)

to yield dynamics mismatching[1]. Further, we also introduce a conservative regularization in Equation 5 that encourages learning conservative Q values for source-domain data, thus avoiding exploiting false and overestimated values when performing policy optimization in Equation 4.

Comparison to dynamics-aware reward modification. We also note that recent works propose to learn a dynamics-aware reward modification for the cross-domain (cross-dynamics) RL setting [Eysenbach et al. 2020], which *modifies the reward* by using two classifiers $q_{\text{sas}}(\cdot|\mathbf{s},\mathbf{a},\mathbf{s}')$ and $q_{\text{sa}}(\cdot|\mathbf{s},\mathbf{a})$ that distinguish between the source-domain and target-domain data, *i.e.*, $r_{\text{modified}}(\mathbf{s},\mathbf{a},\mathbf{s}') = r(\mathbf{s},\mathbf{a}) + \log \frac{q_{\text{sas}}(\text{target}|\mathbf{s},\mathbf{a},\mathbf{s}')}{q_{\text{sas}}(\text{source}|\mathbf{s},\mathbf{a},\mathbf{s}')} - \log \frac{q_{\text{sa}}(\text{target}|\mathbf{s},\mathbf{a})}{q_{\text{sa}}(\text{source}|\mathbf{s},\mathbf{a})}$. Compared to such a reward modification approach, our supported value optimization only learns a single transition model $\hat{T}_{\text{target}}$ that *merely fits the target-domain data, while does not explicitly fit the source-domain data*. As we will show in our experiment, our support optimization enjoys more stable training and achieves better performance especially when the source domain involves complex data distributions.

[1]In Figure 3, we also compare such a filter to a "soft" version.

Practical Implementation

We now describe our instantiation of BOSA for supported policy and value optimization. First, instead of directly maximizing the log-likelihood objective for estimating $\hat{\pi}_{\beta_{\text{mix}}}$ and $\hat{T}_{\text{target}}$, we opt to use the conditional variational auto-encoder (CVAE [Sohn, Lee, and Yan 2015]) for density estimation and likelihood inference (in Equations 4 and 5). Second, to tackle the constrained objective for supported policy optimization in Equation 4, we optimize it by using a Lagrangian relaxation. Third, for the filter operator $\mathbb{1}(\cdot)$ in supported value optimization (Equation 5), we learn an ensemble of target transition model $\hat{T}_{\text{target}}$ to maintain the model's uncertainty and take $\mathbb{1}(\cdot > \epsilon)$ over the minimum of ensemble models. Empirically, we find the performance can be improved by increasing the ensemble size, but improvement is saturated around 5. Thus, we learn 5 models in ensemble.

Experiments

The goal of our empirical evaluation is to answer the following questions: **1)** Can BOSA improve offline data efficiency by leveraging the additional source-domain data and achieve better performance than prior alternative methods? **2)** In some cases there exists no additional source data from other environments, can we retain the benefits of cross-

		Finetune	DARA*	MABE	BOSA
mass	ho-m	44.5	59.3	23.1	**80.5** $\pm$1.7
	ho-mr	27.5	34.1	20.4	**39.7** $\pm$0.1
	ho-me	85.9	99.7	38.9	**104.2** $\pm$0.5
	wa-m	72.3	81.7	56.7	**83** $\pm$2.9
	wa-mr	10.4	15.1	12.5	**21.4** $\pm$2
	wa-me	68.6	**93.3**	82.7	86.5 $\pm$0.6
joints	ho-m	52.5	58	57.7	**78** $\pm$7.3
	ho-mr	29.6	32	**35.4**	32.7 $\pm$1.3
	ho-me	107.3	**109**	104.8	96.4 $\pm$0.5
	wa-m	76.6	81.2	48.7	**86.5** $\pm$5.6
	wa-mr	13.5	16.4	1.6	**38.2** $\pm$4.7
	wa-me	104	**116.5**	82.6	85.8 $\pm$0.3

Table 2: Comparison on cross-domain tasks, where *DARA* denotes the best score* among offline baselines (BCQ, CQL, and MOPO) when using DARA-based reward modification.

domain data transfer by using BOSA? **3)** How do the different components of BOSA influence the performance?

In comparison, we consider the four most related offline RL methods: BCQ [Fujimoto, Meger, and Precup 2019], CQL [Kumar et al. 2020], MOPO [Yu et al. 2020], SPOT [Wu et al. 2022], where BCQ motivates us the support constraints, CQL motivates us the conservation over policy optimization, MOPO is a representative model-based approach which enjoys data efficiency, and SPOT is the current state-of-the-art baseline which also performs supported policy optimization in offline RL.

Offline cross-domain transfer. To answer the first question, we use the D4RL [Fu et al. 2020] offline data as the target domain and use the similar cross-domain dynamics modification utilized in DARA [Liu, Zhang, and Wang 2022] to collect source-domain data. Specifically, the source-domain data are collected by modifying the body **mass** or adding noise to **joints** of the agent and then following the same data-collection procedure as in D4RL. To study the data efficiency and make the cross-domain setting tractable, we only use 10% of the D4RL data in the target domain.

In Table 1, we provide the results of different methods using single-domain offline data or cross-domain data. We can see that in the single-domain setting, BCQ, CQL, MOPO, and SPOT both suffer a large performance drop when we reduce the training data size from 100% D4RL to 10% D4RL. Further, using additional source-domain data (*i.e.*, the cross-domain setting) also does not provide a clear performance improvement compared to that using only the target-domain data (10% D4RL). We can observe that in 6 out of the 18 tasks, cross-domain data even brings performance degradation for CQL. The main reason is that although cross-domain setting introduces additional source-domain data, it also raises the challenge of ODD transition dynamics. Aiming at improving the data efficiency and eliminating ODD transition dynamics, we can observe that our BOSA brings significant performance improvement (in 14/18 tasks) compared to baselines when using 10% D4RL data. In comparison to the best performance of baselines when using 100% D4RL data, BOSA receives the fewest performance degradation (-25.6%) among baselines when using 10% D4RL.

	pseudo-model *aug.*		noise *aug.*	
	SPOT	BOSA	SPOT	BOSA
ho-m	0.7 ± 0.6	**66.6** ± 1.6	29 ± 5	**76.6** ± 9.2
ho-mr	0.6 ± 0.3	**14.2** ± 0.2	12.8 ± 1.5	**16.5** ± 0.3
ho-me	0.6 ± 0.3	**70.7** ± 2	66.3 ± 1.9	**78.5** ± 1.7
wa-m	-2.1 ± 1.9	**76.7** ± 0.7	67.4 ± 10	**78.6** ± 3.3
wa-mr	1.4 ± 0.2	**20.1** ± 2.3	**14.8** ± 2.2	12.6 ± 1
wa-me	0.5 ± 1.3	**84.8** ± 0.4	82.6 ± 0.1	**84** ± 0.8

Table 3: When source-domain data is not available, we can use a (sub-optimal) pseudo-transition model or data augmentation to generate new transitions (source-domain data).

Then, we compare BOSA to cross-domain offline RL baselines: DARA [Liu, Zhang, and Wang 2022] and MABE [Cang et al. 2021], where DARA conducts the dynamics-aware reward modification and MABE learns a cross-domain behavior prior. Additionally, we also introduce a fine-tuning baseline (Finetune), which first trains a policy on the source-domain data and then finetunes it over the target-domain data (10% D4RL). We show the results in Table 2. We can see that BOSA is competitive with DARA (reward modification) in 9 out of 12 tasks and outperforms or matches Finetune and MABE on all 12 cross-domain tasks.

Model-based RL and (noising) data augmentation. For the second question, we answer it affirmatively. If we can not access additional source-domain data, we can learn a sub-optimal pseudo-transition model and use the learned model to generate new cross-domain transitions. Alternatively, we can also employ data augmentation (adding noise) to generate new cross-domain transitions. Then, we can directly treat the generated transitions as the source data. More importantly, here we do not require the learned pseudo-transition model to be optimal (in model-based setting) or to delicately balance the amplitude of noises (in data augmentation). Although the generated source-domain data might be OOD transitions, BOSA can filter out mismatched transitions, and preserve the target-relevant and beneficial transitions when performing supported policy and value optimization.

We provide the comparison results in Table 3. We can find that naively using a (sub-optimal) pseudo-transition model or employing data augmentation does not improve or even hurts their performance in most tasks. In contrast, BOSA can improve the cross-domain performance in 11 out of 12 tasks, thus facilitating effective cross-domain offline RL by automatically generating the source-domain data.

Ablation study. To answer the third question, we conduct a thorough ablation study on BOSA. We first investigate the sensitivity of BOSA on the amount of target-domain data. In Figure 3 (a, b), we present the cross-domain results across 5%, 10%, 30%, and 50% of target data. The results show that increasing the amount of target data generally improves the performance of both SPOT and BOSA, showing that data amount is critical to the offline performance. Consistently, BOSA achieves better performance than SPOT across a wide range of data amounts, showing that BOSA can contribute to

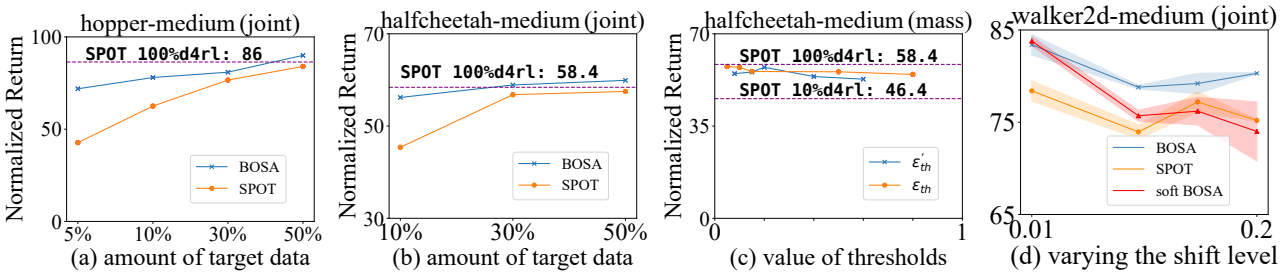

Figure 3: Sensitivity on (a, b) the amount of target-domain data and (c) the thresholds ϵ_{th} and ϵ'_{th}. (d) Varying the transition shift level. Across a range of amount of target data and thresholds, BOSA consistently achieves better results compared to baselines.

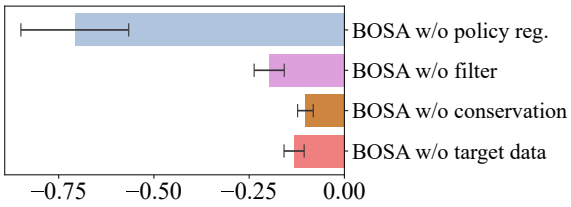

Figure 4: Performance changes when we ablate different components of our method. The x-axis denotes the average improvement of the ablated BOSA versus the full BOSA.

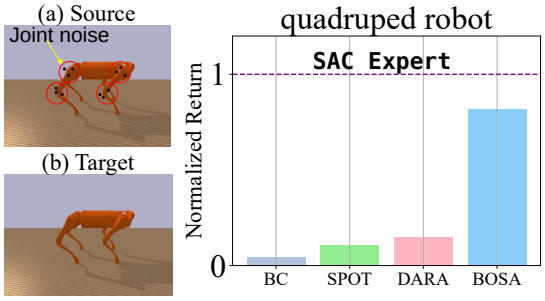

Figure 5: Cross-domain offline RL results on a simulated cross-domain quadruped locomotion task, where the y-axis denotes the normalized return recorded on the target domain.

target-domain sample efficiency by using cross-domain data.

Then, in Figure 3 (c), we study the hyper-parameter sensitivity on the thresholds ϵ_{th} and ϵ'_{th} in supported policy and value optimization respectively. We can see that BOSA is nearly robust when varying the ϵ_{th} and ϵ'_{th}, which consistently outperforms SPOT with 10% D4RL data.

Further, one may prefer to learn a "soft" filter to assign different weights over OOD samples, rather than the strict filter advocated in Equation 5. We thus implement such a "soft" BOSA here (by learning two transition models for weighting), and we provide the comparison when varying the transition shift level in Figure 3 (d). We can see that BOSA achieves more robust results compared to soft BOSA, especially as the distribution shift level increases.

To understand how the choice of different components of BOSA affects its performance, we continue conducting the following ablations[2]: **1)** *w/o policy reg.*: we remove the supported policy regularization in Equation 4. **2)** *w/o filter*: we remove the filter operator in value optimization (Equation 5). **3)** *w/o conservation*: we remove the conservation regularization (Equation 5) in value optimization. **4)** *w/ target data*: To eliminate the potential OOD transition dynamics, one can directly perform value optimization over the target-domain data. Thus, we replace the expectation of Equation 5 by target-domain data $\mathcal{D}_{\text{target}}$ and remove the corresponding filter operator. The results of our ablation studies are shown in Figure 4, where we present the percent difference (averaged over 9 tasks) in performance when removing the corresponding components of BOSA. As expected, removing any of the above components would result in performance degradation for cross-domain BOSA.

Quadrupted results. Aiming at improving the offline RL sample efficiency, we expect to deploy BOSA to more complex quadruped locomotion tasks. To do so, we have validated BOSA on two simulated quadruped locomotion environments (with transition dynamics mismatch, see details in appendix) and, following D4RL, collected 2×10^6 (source-domain) and 3×10^4 (target-domain) medium-expert offline transitions. Then, we conduct the standard offline RL training paradigm and test the performance in the target simulation environment. In Figure 5, we provide the comparison results of BC, SPOT, DARA, and BOSA. We can find that BOSA achieves significant performance gains in this cross-domain task, showing a greater potential that awaits future real-world quadruped robots.

Conclusion

In this paper, we formalize the cross-domain offline RL in an effort to improve offline data efficiency. Beyond the common OOD state actions issue,we identify a new challenge of OOD transition dynamics in the cross-domain offline setting and propose supported policy and value optimization. Empirically, we demonstrate in a variety of offline cross-domain tasks, BOSA can outperform existing cross-domain baselines and enjoys broad flexibility for cross-domain offline data transfer (being naturally plugged into model-based RL and (noising) data augmentation techniques).

[2]Due to page limitations, we leave the technical details and supplementary appendix to https://arxiv.org/pdf/2306.12755.pdf.

Ethical Statement

Limitations. Regarding the methodological assumption made in this paper, we assume source and target domains share the same state space, action space, and reward function. In more general real-world settings, such the state/action space and reward may be also different between the two domains. Thus, future works could further exploit source domain data with different state/action spaces, perhaps with different reward functions, to further improve the data efficiency of offline RL methods in low-data regimes. Another interesting direction is to use support policy/value constraints for online RL and safe RL [Schulman et al. 2015, Liu et al. 2022, Gu et al. 2022]. Similar to those trust region RL methods, we can use support constraints (instead of distribution matching objectives) to constrain the learning policy.

Social impacts. Typically, offline RL holds the promise of enabling RL agents to learn complex behaviors through fixed and static offline data. However, realizing this promise for data-limited tasks in the real world requires mechanisms to improve offline learning efficiency. This paper explicitly proposes cross-domain transfer and data augmentation for such data-limited offline scenes. We believe our work is an important step towards data-efficient offline RL while offering significant improvement, improving offline RL transferability, and providing a promising approach for real-world offline RL participation.

Acknowledgments

We sincerely thank the anonymous reviewers for their insightful suggestions. This work was supported by the National Science and Technology Innovation 2030 - Major Project (Grant No. 2022ZD0208800), and NSFC General Program (Grant No. 62176215).

References

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

Bai, C.; Wang, L.; Yang, Z.; Deng, Z.; Garg, A.; Liu, P.; and Wang, Z. 2022. Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning. *arXiv preprint arXiv:2202.11566*.

Cang, C.; Rajeswaran, A.; Abbeel, P.; and Laskin, M. 2021. Behavioral priors and dynamics models: Improving performance and domain transfer in offline rl. *arXiv preprint arXiv:2106.09119*.

Chen, L.; Lu, K.; Rajeswaran, A.; Lee, K.; Grover, A.; Laskin, M.; Abbeel, P.; Srinivas, A.; and Mordatch, I. 2021. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34: 15084–15097.

Eysenbach, B.; Asawa, S.; Chaudhari, S.; Levine, S.; and Salakhutdinov, R. 2020. Off-dynamics reinforcement learning: Training for transfer with domain classifiers. *arXiv preprint arXiv:2006.13916*.

Eysenbach, B.; Khazatsky, A.; Levine, S.; and Salakhutdinov, R. 2021. Mismatched No More: Joint Model-Policy Optimization for Model-Based RL. *arXiv preprint arXiv:2110.02758*.

Fickinger, A.; Cohen, S.; Russell, S.; and Amos, B. 2021. Cross-Domain Imitation Learning via Optimal Transport. *arXiv preprint arXiv:2110.03684*.

Franzmeyer, T.; Torr, P. H.; and Henriques, J. F. 2022. Learn what matters: cross-domain imitation learning with task-relevant embeddings. *arXiv preprint arXiv:2209.12093*.

Fu, J.; Kumar, A.; Nachum, O.; Tucker, G.; and Levine, S. 2020. D4RL: Datasets for Deep Data-Driven Reinforcement Learning. *CoRR*, abs/2004.07219.

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

Fujimoto, S.; Meger, D.; and Precup, D. 2019. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, 2052–2062. PMLR.

Geist, M.; Scherrer, B.; and Pietquin, O. 2019. A theory of regularized markov decision processes. In *International Conference on Machine Learning*, 2160–2169. PMLR.

Ghasemipour, S. K. S.; Schuurmans, D.; and Gu, S. S. 2021. Emaq: Expected-max q-learning operator for simple yet effective offline and online rl. In *International Conference on Machine Learning*, 3682–3691. PMLR.

Gu, S.; Yang, L.; Du, Y.; Chen, G.; Walter, F.; Wang, J.; Yang, Y.; and Knoll, A. 2022. A Review of Safe Reinforcement Learning: Methods, Theory and Applications. *arXiv preprint arXiv:2205.10330*.

Hishinuma, T.; and Senda, K. 2021. Weighted model estimation for offline model-based reinforcement learning. *Advances in Neural Information Processing Systems*, 34: 17789–17800.

Kidambi, R.; Rajeswaran, A.; Netrapalli, P.; and Joachims, T. 2020. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33: 21810–21823.

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

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

Kumar, A.; Fu, J.; Tucker, G.; and Levine, S. 2019b. Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction. *CoRR*, abs/1906.00949.

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

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

Liu, J.; Shen, H.; Wang, D.; Kang, Y.; and Tian, Q. 2021. Unsupervised Domain Adaptation with Dynamics-Aware Rewards in Reinforcement Learning. *Advances in Neural Information Processing Systems*, 34: 28784–28797.

Liu, J.; Zhang, H.; and Wang, D. 2022. DARA: Dynamics-Aware Reward Augmentation in Offline Reinforcement Learning. *arXiv preprint arXiv:2203.06662*.

Liu, Z.; Cen, Z.; Isenbaev, V.; Liu, W.; Wu, S.; Li, B.; and Zhao, D. 2022. Constrained variational policy optimization for safe reinforcement learning. In *International Conference on Machine Learning*, 13644–13668. PMLR.

Lu, C.; Ball, P. J.; Parker-Holder, J.; Osborne, M. A.; and Roberts, S. J. 2021. Revisiting Design Choices in Model-Based Offline Reinforcement Learning. *arXiv preprint arXiv:2110.04135*.

Lyu, J.; Ma, X.; Li, X.; and Lu, Z. 2022. Mildly conservative Q-learning for offline reinforcement learning. *arXiv preprint arXiv:2206.04745*.

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

Nachum, O.; Dai, B.; Kostrikov, I.; Chow, Y.; Li, L.; and Schuurmans, D. 2019. Algaedice: Policy gradient from arbitrary experience. *arXiv preprint arXiv:1912.02074*.

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

Rezaeifar, S.; Dadashi, R.; Vieillard, N.; Hussenot, L.; Bachem, O.; Pietquin, O.; and Geist, M. 2021. Offline reinforcement learning as anti-exploration. *arXiv preprint arXiv:2106.06431*.

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

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

Viano, L.; Huang, Y.-T.; Kamalaruban, P.; Weller, A.; and Cevher, V. 2021. Robust inverse reinforcement learning under transition dynamics mismatch. *Advances in Neural Information Processing Systems*, 34: 25917–25931.

Wu, J.; Wu, H.; Qiu, Z.; Wang, J.; and Long, M. 2022. Supported Policy Optimization for Offline Reinforcement Learning. *arXiv preprint arXiv:2202.06239*.

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

Wu, Y.; Zhai, S.; Srivastava, N.; Susskind, J.; Zhang, J.; Salakhutdinov, R.; and Goh, H. 2021. Uncertainty weighted actor-critic for offline reinforcement learning. *arXiv preprint arXiv:2105.08140*.

Wulfmeier, M.; Posner, I.; and Abbeel, P. 2017. Mutual alignment transfer learning. In *Conference on Robot Learning*, 281–290. PMLR.

Yang, S.; Feng, Y.; Zhang, S.; and Zhou, M. 2022. Regularizing a model-based policy stationary distribution to stabilize offline reinforcement learning. In *International Conference on Machine Learning*, 24980–25006. PMLR.

Yu, T.; Kumar, A.; Rafailov, R.; Rajeswaran, A.; Levine, S.; and Finn, C. 2021. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34: 28954–28967.

Yu, T.; Thomas, G.; Yu, L.; Ermon, S.; Zou, J. Y.; Levine, S.; Finn, C.; and Ma, T. 2020. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33: 14129–14142.

Zhang, W.; Xu, H.; Niu, H.; Cheng, P.; Li, M.; Zhang, H.; Zhou, G.; and Zhan, X. 2022. Discriminator-Guided Model-Based Offline Imitation Learning. *arXiv preprint arXiv:2207.00244*.