Offline Reinforcement Learning with Domain-Unlabeled Data

Anonymous authors
Paper under double-blind review

Keywords: Offline Reinforcement Learning, Domain-Unlabeled Data, Positive-Unlabeled Learning

Summary

Offline reinforcement learning (RL) is vital in areas where active data collection is expensive or infeasible, such as robotics or healthcare. In the real world, offline datasets often involve multiple "domains" that share the same state and action spaces but have distinct dynamics, and only a small fraction of samples are clearly labeled as belonging to the target domain we are interested in. For example, in robotics, precise system identification may only have been performed for part of the deployments. To address this challenge, we consider Positive-Unlabeled Offline RL (PUORL), a novel offline RL setting in which we have a small amount of labeled target-domain data and a large amount of domain-unlabeled data from multiple domains, including the target domain. For PUORL, we propose a plug-and-play approach that leverages positive-unlabeled (PU) learning to train a domain classifier. The classifier then extracts target-domain samples from the domain-unlabeled data, augmenting the scarce target-domain data. Empirical results on a modified version of the D4RL benchmark demonstrate the effectiveness of our method: even when only 1%–3% of the dataset is domain-labeled, our approach accurately identifies target-domain samples and achieves high performance, even under substantial dynamics shift. Our plug-and-play algorithm seamlessly integrates PU learning with existing offline RL pipelines, enabling effective multi-domain data utilization in scenarios where comprehensive domain labeling is prohibitive.

Contribution(s)

1. We introduce Positive-Unlabeled Offline RL (PUORL), a novel offline RL setting with a small amount of data from a target domain and a large dataset containing data from multiple domains without domain labels. The goal is to learn a policy for the target domain.
   Context: Existing cross-domain offline RL methods (Liu et al., 2022; 2023; Wen et al., 2024) assume knowledge of the original domain of each transition, which is not accessible in our setting.
2. We propose a method that uses positive-unlabeled (PU) learning to filter the target-domain data from domain-unlabeled data.
   Context: Our approach uses PU learning (Li & Liu, 2003; Kiryo et al., 2017) to classify domain-unlabeled samples as "positive" (target) or "negative" (other). We then augment the labeled target-domain dataset with the domain-unlabeled samples predicted to be positive. This filtering can be integrated with value-based offline RL algorithms.
3. We empirically demonstrate that our PU-based method accurately filters domain-unlabeled data and achieves high performance in a modified version of D4RL.
   Context: We tested our approach on a modified D4RL benchmark (Fu et al., 2020), where only 1%–3% of samples contain domain labels, and the rest are domain-unlabeled, drawn from both the target and other domains with different dynamics. Even with this limited labeling, our method closely matches an oracle baseline (which has access to all target-domain data) and overall achieves higher average returns than the other baselines, even under substantial dynamics mismatch.

Offline Reinforcement Learning with Domain-Unlabeled Data

Anonymous authors
Paper under double-blind review

Abstract

Offline reinforcement learning (RL) is vital in areas where active data collection is expensive or infeasible, such as robotics or healthcare. In the real world, offline datasets often involve multiple "domains" that share the same state and action spaces but have distinct dynamics, and only a small fraction of samples are clearly labeled as belonging to the target domain we are interested in. For example, in robotics, precise system identification may only have been performed for part of the deployments. To address this challenge, we consider Positive-Unlabeled Offline RL (PUORL), a novel offline RL setting in which we have a small amount of labeled target-domain data and a large amount of domain-unlabeled data from multiple domains, including the target domain. For PUORL, we propose a plug-and-play approach that leverages positive-unlabeled (PU) learning to train a domain classifier. The classifier then extracts target-domain samples from the domain-unlabeled data, augmenting the scarce target-domain data. Empirical results on a modified version of the D4RL benchmark demonstrate the effectiveness of our method: even when only 1%–3% of the dataset is domain-labeled, our approach accurately identifies target-domain samples and achieves high performance, even under substantial dynamics shift. Our plug-and-play algorithm seamlessly integrates PU learning with existing offline RL pipelines, enabling effective multi-domain data utilization in scenarios where comprehensive domain labeling is prohibitive.

1 Introduction

Offline reinforcement learning (RL) (Levine et al., 2020) trains policies exclusively from pre-collected datasets without further environmental interaction. This paradigm has been applied to many real-world problems, including robotics (Kalashnikov et al., 2018; 2021) and healthcare (Guez et al., 2008; Killian et al., 2020), where live data collection is costly or infeasible. This paper examines an offline RL setting where the dataset is collected in multiple *domains*, environments that share the same state and action spaces but have different dynamics—with the goal of training a policy that performs well in a specific target domain. In practice, however, annotating domain labels is labor-intensive or impractical at scale, resulting in a small amount of domain-labeled target data alongside a large volume of domain-unlabeled samples drawn from various domains, including the target domain. One illustrative example arises in healthcare: if a specific disease significantly alters a patient's response to treatment, it effectively changes the transition dynamics. Only a small subset of patients are tested for disease with high cost of testing, leading to limited domain-labeled data and a predominance of domain-unlabeled samples (Claesen et al., 2015).

Since offline RL depends on large, diverse datasets (Kalashnikov et al., 2021; Padalkar et al., 2023), relying solely on the small domain-labeled subset may deteriorate policy performance. Consequently, there is a pressing need to incorporate domain-unlabeled data effectively. While recent studies have focused on enhancing target domain performance by utilizing data from a different domain (Liu et al., 2022; Wen et al., 2024; Xu et al., 2023b), these methods presuppose that clear domain labels are available for all samples, which does not hold in our setting.

To tackle this challenge, we propose a new offline RL setting called **P**ositive-**U**nlabeled **O**ffline **RL** (**PUORL**). In PUORL, we have two types of data: a small amount of target-domain (positive-domain) data and a large volume of domain-unlabeled data, a mixture of samples from the positive domain and other domains (negative domains). This setting is relevant in any setting where we aim to train agents based on a specific characteristic that significantly affects the dynamics. This includes cases where a particular disease influences medical outcomes, as noted above, and scenarios such as unique road conditions in autonomous driving or a standard actuator defect in robotics (Kiran et al., 2021; Padakandla, 2021; Shi et al., 2021).

For PUORL, we propose a general approach that uses *positive-unlabeled* (PU) learning (Li & Liu, 2003; Bekker & Davis, 2020; Sugiyama et al., 2022) to train a classifier to distinguish positive-domain data from other domains (Sec. 3.2). Using the trained classifier, we filter out negative-domain data from a large, domain-unlabeled dataset, thereby augmenting the small domain-labeled data with additional positive-domain samples. Then, we apply off-the-shelf offline RL algorithms to this augmented dataset. Our framework functions as a plug-and-play module compatible with any value-based offline RL method, allowing users to adopt their preferred offline RL algorithm for PUORL. Experiments on the modified version of the D4RL (Fu et al., 2020), where only 1%–3% of the data are domain-labeled, demonstrate that our method accurately identifies positive-domain data and effectively leverages the abundant domain-unlabeled dataset for offline RL (Sec. 4).

Related work. Cross-domain offline RL assumes fully domain-labeled datasets from two domains: a source domain with ample data and a target domain with fewer samples, where the goal is to effectively utilize the source domain data with different dynamics to improve the target domain performance (Liu et al., 2022; Xue et al., 2023; Xu et al., 2023b; Wen et al., 2024; Liu et al., 2023; Lyu et al., 2024). Some approaches fix the reward or filter transitions from a labeled source domain using discriminators (Liu et al., 2022; Wen et al., 2024), while others constrain policies to remain in regions aligned with target-domain data (Liu et al., 2023; Xue et al., 2023). Recently, Lyu et al. (2024) provided a benchmark for cross-domain offline RL. In contrast to most methods, which assume available domain labels for all samples, our work handles a large amount of domain-unlabeled data, which may include samples from both target and non-target domains. Please refer to App. A for the comprehensive related work.

Contributions. Our contributions are threefold: 1) we propose a new offline RL setting, PUORL, to handle the domain-unlabeled data, 2) we propose a method that leverages PU learning to train a precise domain classifier, augmenting the limited domain-labeled data, and 3) we demonstrate the effectiveness of our method on the modified version of the D4RL benchmark with dynamics shift, where only 1%–3% of the data are domain-labeled.

2 Preliminaries

Reinforcement learning (RL). RL (Sutton & Barto, 2018) is characterized by a Markov decision process (MDP) (Puterman, 2014), defined by 6-tuple: $\mathcal{M} := (\mathcal{S}, \mathcal{A}, P, p_0, R, \gamma)$. Here, $\mathcal{S}$ and $\mathcal{A}$ denote the continuous state and action spaces, respectively. $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ defines the transition density, $p_0 : \mathcal{S} \to [0, 1]$ denotes the initial state distribution, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ specifies the reward function, and $\gamma \in [0, 1)$ represents the discount factor. In RL, the primary objective is to learn a policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, maximizing the expected cumulative discounted reward $\mathbb{E}_{\pi,P}\left[\sum_{t=1}^{\infty} \gamma^t R(s_t, a_t)\right]$, where $\mathbb{E}_{\pi,P}[\cdot]$ denotes the expectation over the sequence of states and actions $(s_1, a_1, \dots)$ generated by the policy π and the transition density P.

In this paper, we assume that different domains correspond to distinct MDPs that differ only in their transition dynamics. For example, two domains, $\mathcal{M}_1$ and $\mathcal{M}_2$, have different transition dynamics (P_1 and P_2), with the other components being the same.

Offline RL. To address the limitations on direct agent-environment interactions, offline RL (Levine et al., 2020) employs a fixed dataset, $\mathcal{D} := \{(s_i, a_i, r_i, s'_i)\}_{i=1}^{N}$, collected by a behav-

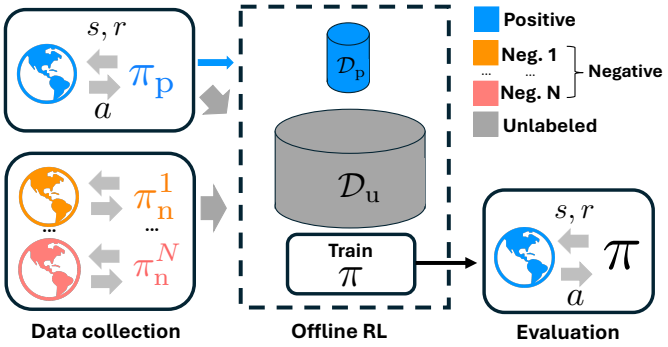

Figure 1: Diagram of Positive-Unlabeled Offline RL (PUORL). PUORL has a **positive domain** we target and **negative domains**, with different dynamics to the positive domain. We have two data types: **positive data** and **domain-unlabeled data**, which are mixtures of samples from the positive and negative domains. We train a policy to maximize the expected return in the positive domain.

ioral policy $\pi_\beta : \mathcal{S} \times \mathcal{A} \to [0, 1]$. Let $\mu_\beta(s, a)$ be the stationary distribution over the state-action pair induced by the behavioral policy π_β. The dataset $\mathcal{D}$ is assumed to be generated as follows: $(s_i, a_i) \sim \mu_\beta(s, a)$, $r_i = R(s_i, a_i)$, and $s'_i \sim P(\cdot|s_i, a_i)$.

Positive-unlabeled (PU) learning. PU learning is a method that trains a binary classifier using positive and unlabeled data (Li & Liu, 2003; Bekker & Davis, 2018; Sugiyama et al., 2022). Let $X \in \mathbb{R}^d$ and $Y \in \{-1, +1\}$ be the random variables of the input and label in a binary classification problem. We denote the data-generating joint density over (X, Y) by $p(x, y)$. Let $p_{\mathrm{p}}(x) := p(x|Y = +1)$ and $p_{\mathrm{n}}(x) := p(x|Y = -1)$ be the densities of x conditioned on the positive and negative labels respectively and $p(x) := \alpha_{\mathrm{p}} p_{\mathrm{p}}(x) + \alpha_{\mathrm{n}} p_{\mathrm{n}}(x)$ be the marginal density of the unlabeled data. $\alpha_{\mathrm{p}} := p(Y = +1)$ denotes the class prior probability (mixture proportion) for the positive label and $\alpha_{\mathrm{n}} := p(Y = -1) = 1 - \alpha_{\mathrm{p}}$ for the negative label. In PU learning, we assume that we have two types of data: Positively labeled data $\mathcal{X}_{\mathrm{p}} := \{x_i^{\mathrm{p}}\}_{i=1}^{n_{\mathrm{p}}} \overset{\text{i.i.d.}}{\sim} p_{\mathrm{p}}(x)$ and unlabeled data $\mathcal{X}_{\mathrm{u}} := \{x_i^{\mathrm{u}}\}_{i=1}^{n_{\mathrm{u}}} \overset{\text{i.i.d.}}{\sim} p(x)$. The task of PU learning is to train a binary classifier $f : X \to \{-1, +1\}$ from positive data $\mathcal{X}_{\mathrm{p}}$ and unlabeled data $\mathcal{X}_{\mathrm{u}}$. Generally, PU learning methods require information on the mixture proportion (α_p), and there are a bunch of mixture proportion estimation (MPE) methods (du Plessis & Sugiyama, 2014; Scott, 2015; du Plessis et al., 2017; Garg et al., 2021). Among the methods of PU learning, certain approaches, notably nnPU (Kiryo et al., 2017) and $(\mathrm{TED})^{\mathrm{n}}$ (Garg et al., 2021), demonstrate particular compatibility with neural networks.

3 Method

This section introduces a novel offline RL problem setting for leveraging domain-unlabeled data. We then propose a simple algorithm using PU learning to address this problem.

3.1 Problem Formulation

We introduce **Positive-Unlabeled Offline RL (PUORL)** where the dataset is generated within multiple domains, with a small amount of data from one domain of our interest labeled and the rest provided as domain-unlabeled (Figure 1). In PUORL, we have a positive domain $\mathcal{M}_{\mathrm{p}} := (\mathcal{S}, \mathcal{A}, P_{\mathrm{p}}, \rho, R, \gamma)$, for which we aim to maximize the expected return and negative domains $\{\mathcal{M}_{\mathrm{n}}^k := (\mathcal{S}, \mathcal{A}, P_{\mathrm{n}}^k, \rho, R, \gamma)\}_{k=1}^N$, which share the same state and action spaces, initial state distribution, reward function and discount factor. For each domain, there exist fixed behavioral policies: π_{p} for positive domain and π_{n}^k for negative domains, and they induce the stationary distributions over the state-action pair denoted as $\mu_{\mathrm{p}}(s, a)$ and $\mu_{\mathrm{n}}^k(s, a)$ for all $k \in \{1, \ldots, N\}$. We define $\mu_{\mathrm{n}}(s, a) := \sum_{k=1}^N \eta_k \mu_{\mathrm{n}}^k(s, a)$, where $\eta_k \in [0, 1]$, $\sum_{k=1}^N \eta_k = 1$ is the domain-mixture proportion.

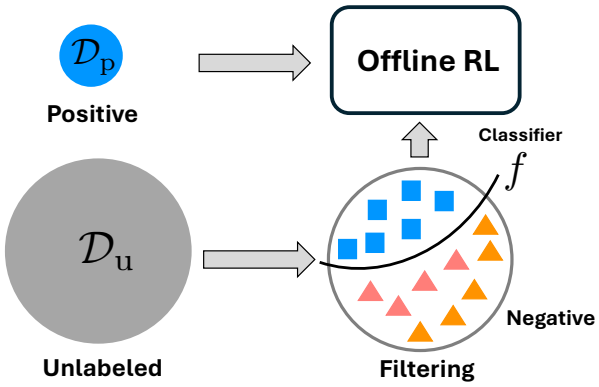

Figure 2: Diagram of our method. We first train a classifier f using **PU learning** to distinguish positive domain data from negative domain data. Then, we filter the positive domain data from domain-unlabeled data by applying classifier f to the domain-unlabeled dataset. Finally, we train a policy using off-the-shelf offline RL methods with the augmented dataset.

We are given two datasets:

- **Positive data**: explicitly labeled target-domain transitions, $\mathcal{D}_{\mathrm{p}} := \{(s_i, a_i, r_i, s'_i)\}_{i=1}^{n_{\mathrm{p}}}$. These transitions are i.i.d. samples from $\mu_{\mathrm{p}}(s, a)$, R, and P_{p}.
- **Domain-unlabeled data**: a mixture of positive and negative-domain transitions, $\mathcal{D}_{\mathrm{u}} := \{(s_i, a_i, r_i, s'_i)\}_{i=1}^{n_{\mathrm{u}}}$. These transitions are i.i.d. samples from $\mu_{\mathrm{u}}(s, a) := \alpha_{\mathrm{p}}\mu_{\mathrm{p}}(s, a) + \alpha_{\mathrm{n}}\mu_{\mathrm{n}}(s, a)$, R, and corresponding transition densites. We assume that $n_{\mathrm{u}} \gg n_{\mathrm{p}}$.

Henceforth, domain-unlabeled data will be referred to as *unlabeled data* when it is clear from the context. Although PUORL focuses on the difference in dynamics, we can generalize the problem set to encompass variations in the reward function. Refer to Appendix C for details. Here, the objective is to learn the optimal policy in the positive domain of our interest as

$$\pi^*(a|s) := \operatorname*{argmax}_{\pi} \mathbb{E}_{\pi, P_{\mathrm{p}}} \left[\sum_{t=1}^{\infty} \gamma^t R(s_t, a_t) \right]. \tag{1}$$

The most naive approach in this setup involves applying conventional offline RL methods on only a small amount of positive data $\mathcal{D}_{\mathrm{p}}$. However, using a small dataset increases the risk of encountering out-of-distribution state-action pairs due to the limited coverage of the dataset (Levine et al., 2020). Conversely, utilizing all available data $\mathcal{D}_{\mathrm{p}} \cup \mathcal{D}_{\mathrm{u}}$ to increase the dataset size can hinder the agent's performance due to the different dynamics (Liu et al., 2022).

3.2 Proposed Method

The key idea of our method is to filter positive-domain data from unlabeled data by training a domain classifier that leverages the differences in transition dynamics. Specifically, we propose a two-staged offline RL algorithm as in Figure 2.

Stage 1: Train a domain classifier by PU learning. We consider a binary classification problem where $\mathcal{S} \times \mathcal{A} \times \mathcal{S}'$ serves as the input space ($\mathcal{X}$ in Sec. 2). The label is defined as $Y = +1$ for the positive domain and $Y = -1$ for the negative domains. Since positive and negative domains differ in how they transition from (s, a) to s', the tuple (s, a, s') naturally captures these discrepancies, making it an effective signal for classification. Using positive data $\mathcal{D}_{\mathrm{p}}$ and unlabeled data $\mathcal{D}_{\mathrm{u}}$, we train a classifier $f : \mathcal{S} \times \mathcal{A} \times \mathcal{S}' \to \{+1, -1\}$ by PU learning (Kiryo et al., 2017; Sugiyama et al., 2022; Plessis et al., 2015). Because α_{p} is unknown in PUORL, we estimate it using mixture proportion estimation (MPE) (Garg et al., 2021; Sugiyama et al., 2022).

Stage 2: Data filtering and offline RL. We first filter the positive domain data from unlabeled data by applying classifier f to the unlabeled dataset to identify instances predicted as positive, denoted by $\mathcal{D}_{\mathrm{p}}^{f} := \{(s, a, r, s') \in \mathcal{D}_{\mathrm{u}} : f(s, a, s') = +1\}$, combining it with the positive data as $\tilde{\mathcal{D}}_{\mathrm{p}} := \mathcal{D}_{\mathrm{p}} \cup \mathcal{D}_{\mathrm{p}}^{f}$. Then, we train the policy using off-the-shelf offline RL methods with $\tilde{\mathcal{D}}_{\mathrm{p}}$.

The methodology details are outlined in Algo. 1.

Algorithm 1 Data filtering for the positive domain

Initialize classifier parameters ψ of classifier f
Initialize policy parameters θ and value function parameters ϕ
Initialize experience replay buffer $\mathcal{D}_{\mathrm{p}}$ and $\mathcal{D}_{\mathrm{u}}$
Specify epochs K_{PU}, K_{RL}
for iteration $k \in [0, \ldots, K_{\mathrm{PU}}]$ **do** ▷ PU learning routine
    Update ψ on $\mathcal{D}_{\mathrm{p}}$ and $\mathcal{D}_{\mathrm{u}}$ by PU learning with MPE
end for
$\tilde{\mathcal{D}}_{\mathrm{p}} \leftarrow \mathcal{D}_{\mathrm{p}} \cup \{(s, a, r, s') \in \mathcal{D}_{\mathrm{u}} : f_{\psi}(s, a, s') = +1\}$ ▷ Data filtering
for iteration $k \in [0, \ldots, K_{\mathrm{RL}}]$ **do** ▷ Offline RL routine
    Update θ and ϕ on $\tilde{\mathcal{D}}_{\mathrm{p}}$ by Offline RL method
end for
Output θ and ϕ

This algorithm exhibits considerable generality, accommodating a wide range of PU learning methodologies (Kiryo et al., 2017; Garg et al., 2021) and offline RL algorithm (Kumar et al., 2020; Kostrikov et al., 2022; Fujimoto & Gu, 2021; Fujimoto et al., 2023; Tarasov et al., 2023), allowing practitioners to choose the most suitable methods for their specific problem. An accurate classifier is necessary for the data filtering to work well. Conversely, less accurate classifiers result in the inclusion of negative-domain data in the filtered data $\mathcal{D}_{\mathrm{p}}^{f}$, potentially leading to a performance decline due to the different dynamics.

4 Experiment

We conduct experiments under various settings to investigate the following four questions: (i) Can the PU learning method accurately classify the domain from PU-formatted data? (ii) Can our method improve performance by augmenting positive data in various domain shift settings? (iii) How does the magnitude of the dynamics shift affect performance? (iv) How does the different quality of the negative-domain data affect the performance? We first explain the setup of our experiments and, subsequently, report the results.

4.1 Experimental Setup

Dataset. We utilized the modified version of D4RL benchmark (Fu et al., 2020) with dynamics shift, focusing on three control tasks: Halfcheetah, Hopper, and Walker2d. D4RL provides four different data qualities for each task: medium-expert (ME), medium-replay (MR), medium (M), and random (R). To examine the impact of dynamics shift on performance, we considered three types of dynamics shifts between positive and negative domains: **body mass shift**, **mixture shift**, and **entire body shift**. In all scenarios, we set the total number of samples to 1 million and maintained a 3:7 positive-to-negative ratio. We explored two labeled ratios: **0.03** and **0.01**, where only 30K and 10K samples were labeled positive, respectively. In the main text, we report the results with the labeled ratio of 0.01 and put the results with the labeled ratio of 0.03 in App. D.

We used the dataset provided by Liu et al. (2022) for the body mass shift and mixture shift. In body mass shift, the mass of specific body parts in the negative domain was modified. For the mixture shift, we mixed the data with body mass shift and data with joint noise with equal proportions to test whether our method can handle multiple negative domains. We prepared the entire body shift

Table 1: The results of the PU classifier in the body mass shift with labeled ratio = 0.01 and 0.03. For each setting, we reported the average and standard deviation of the test accuracy over 5 seeds.

Env	Ratio	ME/ME	ME/R	M/M	M/R
Hopper	0.01	99.54 ± 0.06	99.23 ± 0.08	99.77 ± 0.14	99.33 ± 0.07
	0.03	99.72 ± 0.06	99.89 ± 0.03	99.90 ± 0.03	99.32 ± 0.05
Halfcheetah	0.01	99.48 ± 0.04	99.45 ± 0.11	99.38 ± 0.18	99.33 ± 0.06
	0.03	99.63 ± 0.03	99.70 ± 0.10	99.66 ± 0.06	99.43 ± 0.07
Walker2d	0.01	99.00 ± 0.03	98.43 ± 0.04	98.36 ± 0.02	99.69 ± 0.10
	0.03	99.64 ± 0.02	99.49 ± 0.11	98.41 ± 0.06	99.39 ± 0.08

with Halfcheetah and Walker2d to test the performance with a large dynamics shift. Halfcheetah and Walker2d were paired as positive and negative domains in the entire body shift due to their entirely different body structures, yet they have the same state space of 17 dimensions.

To explore the effect of data quality on performance, we examined various combinations of data qualities, using abbreviations separated by a slash to denote pairs of positive and negative data with varying qualities, e.g., ME/ME, for medium-expert quality in both domains.

Offline RL algorithms and PU learning methods. We selected TD3+BC (Fujimoto & Gu, 2021) and IQL (Kostrikov et al., 2022) as our offline RL methods due to their widespread use and computational efficiency. We used the implementation of TD3+BC and IQL from JAX-CORL (Nishimori, 2024) and used the default hyperparameters for all experiments. The main results presented below pertain to TD3+BC. The results for IQL are reported in App. D.2. We trained the agent for 1 million steps and reported the average and 95% confidence interval of averaged evaluation results over 10 episodes and 10 different seeds for each setting.

For PU learning, TED^{n} (Garg et al., 2021) was chosen owing to its effectiveness with neural networks (App. B.1) and used the official implementation provided by the authors. We trained the classifier for 100 epochs and reported the average and standard deviation of the test accuracy over 5 seeds. For more details, refer to App. B.

Baselines. To evaluate our method's efficacy, we established five baselines for comparison: *Only-Labeled-Positive (OLP)*, *Sharing-All*, *Dynamic-Aware Reward Augmentation (DARA)* (Liu et al., 2022), *Info-Gap Data Filtering (IGDF)* (Wen et al., 2024) and *Oracle*. The OLP baseline, utilizing only labeled positive data (only 1%–3% of the entire dataset), avoided dynamics shifts' issues at the expense of using a significantly reduced dataset size. This comparison assessed the benefit of augmenting data volume through our filtering method. The Sharing-All baseline employed positive and unlabeled data without preprocessing for offline RL, offering broader data coverage but posing the risk of performance degradation due to dynamics shifts. This comparison aimed to explore the impact of dynamics shifts and how our filtering technique can mitigate these effects. The Oracle baseline, training policy with positively labeled data, and all positive data within the unlabeled data provide the ideal performance our method strives to achieve.

In addition to those naive baselines, we also compared our method with cross-domain adaptation methods designed to improve performance in the target domain by leveraging source domain data with different dynamics. For these methods, we used the positive data as the target data and the unlabeled data as the source domain data. We chose two methods, DARA and IGDF, which apply to any offline RL algorithms and are, thereby, good candidates for comparison with our plug-and-play method. This comparison aimed to examine whether PUORL, where we have domain-unlabeled data alongside a limited amount of labeled target data, negatively impacts the performance of cross-domain adaptation methods. If such a decline occurs, it highlights the need for specialized methods, such as PU-based filtering, to handle this scenario effectively. For both algorithms, we re-implemented the algorithm in JAX (Bradbury et al., 2018) for parallelized training referring to the official implementations. For more details, refer to App. B.3.

Table 2: The average normalized score and 95% confidence interval calculated by the results from 10 different seeds in body mass shift (labeled ratio = 0.01) with TD3+BC. Of feasible methods (OLP, Sharing-All, DARA, IGDF, Ours), the best average is in **blue**. Separated by the double vertical line, we report Oracle as a reference.

Body mass shift							
Env	**Quality**	**OLP**	**Sharing-All**	**DARA**	**IGDF**	**Ours**	**Oracle**
Hopper	ME/ME	28.6 ± 7.1	45.7 ± 13.0	55.5 ± 11.9	50.4 ± 12.8	$\mathbf{98.3 \pm 5.9}$	98.2 ± 8.4
	ME/R	36.5 ± 7.5	73.9 ± 12.7	51.0 ± 9.1	40.3 ± 8.2	$\mathbf{100.8 \pm 6.4}$	98.2 ± 8.4
	M/M	37.9 ± 7.3	47.4 ± 3.4	$\mathbf{56.6 \pm 4.6}$	52.9 ± 2.4	48.3 ± 1.4	48.9 ± 2.8
	M/R	43.3 ± 4.6	45.8 ± 4.0	52.1 ± 4.8	50.5 ± 4.7	$\mathbf{52.1 \pm 2.9}$	48.9 ± 2.8
Halfcheetah	ME/ME	17.6 ± 3.1	$\mathbf{80.8 \pm 2.1}$	27.2 ± 3.1	21.3 ± 5.0	75.3 ± 10.2	86.9 ± 4.4
	ME/R	17.0 ± 2.7	72.5 ± 4.4	3.9 ± 2.7	7.4 ± 2.8	$\mathbf{80.4 \pm 8.7}$	86.9 ± 4.4
	M/M	32.0 ± 2.7	42.1 ± 1.3	41.3 ± 1.0	42.3 ± 0.9	$\mathbf{48.5 \pm 0.2}$	48.8 ± 0.3
	M/R	32.3 ± 3.0	37.8 ± 10.2	11.3 ± 5.3	8.6 ± 3.7	$\mathbf{48.9 \pm 0.2}$	48.8 ± 0.3
Walker2d	ME/ME	9.3 ± 4.4	88.5 ± 0.6	37.1 ± 14.8	59.6 ± 17.3	$\mathbf{108.2 \pm 0.4}$	108.5 ± 0.4
	ME/R	15.9 ± 5.8	78.0 ± 24.1	2.6 ± 1.8	4.5 ± 2.2	$\mathbf{108.1 \pm 0.8}$	108.5 ± 0.4
	M/M	16.4 ± 7.0	81.2 ± 0.8	37.0 ± 11.3	41.7 ± 7.6	$\mathbf{83.2 \pm 2.2}$	84.6 ± 0.6
	M/R	21.3 ± 7.9	80.0 ± 2.1	1.2 ± 1.1	0.9 ± 1.4	$\mathbf{84.0 \pm 0.3}$	84.6 ± 0.6

Table 3: The average normalized score and 95% confidence interval from 10 seeds in mixture shift (labeled ratio = 0.01) with TD3+BC. The format is the same as the table for body mass shift.

Mixture shift							
Env	**Quality**	**OLP**	**Sharing-All**	**DARA**	**IGDF**	**Ours**	**Oracle**
Hopper	ME/ME	26.8 ± 6.2	73.0 ± 18.6	53.4 ± 8.8	42.4 ± 8.9	$\mathbf{92.6 \pm 9.7}$	96.4 ± 8.2
	ME/R	24.3 ± 7.0	84.9 ± 15.8	43.6 ± 7.6	42.5 ± 10.9	$\mathbf{97.0 \pm 7.5}$	96.4 ± 8.2
	M/M	40.6 ± 3.1	$\mathbf{56.8 \pm 7.9}$	55.4 ± 4.6	55.4 ± 5.8	46.9 ± 1.6	45.9 ± 1.5
	M/R	42.8 ± 2.2	43.7 ± 2.9	44.9 ± 4.6	$\mathbf{49.3 \pm 2.8}$	48.7 ± 1.5	45.9 ± 1.5
Halfcheetah	ME/ME	19.5 ± 5.2	78.6 ± 2.1	28.6 ± 3.6	29.9 ± 3.7	$\mathbf{82.4 \pm 6.8}$	81.3 ± 9.6
	ME/R	19.0 ± 3.1	$\mathbf{82.0 \pm 4.8}$	11.1 ± 4.0	9.3 ± 1.9	78.6 ± 8.5	81.3 ± 9.6
	M/M	35.8 ± 2.1	48.1 ± 1.3	39.8 ± 2.5	40.4 ± 2.9	$\mathbf{48.7 \pm 0.2}$	48.7 ± 0.2
	M/R	32.5 ± 2.0	$\mathbf{51.7 \pm 1.4}$	14.7 ± 3.2	16.8 ± 4.4	48.8 ± 0.3	48.7 ± 0.2
Walker2d	ME/ME	7.0 ± 3.1	104.4 ± 3.5	49.1 ± 20.2	46.0 ± 11.9	$\mathbf{107.6 \pm 2.0}$	108.5 ± 0.4
	ME/R	16.3 ± 6.3	107.2 ± 18.5	25.2 ± 4.9	37.6 ± 6.5	$\mathbf{108.7 \pm 0.3}$	108.5 ± 0.4
	M/M	17.3 ± 7.2	79.8 ± 1.6	55.1 ± 13.5	56.4 ± 12.4	$\mathbf{84.3 \pm 1.5}$	84.8 ± 1.4
	M/R	19.1 ± 7.3	78.7 ± 2.1	29.6 ± 11.8	41.6 ± 6.8	$\mathbf{83.0 \pm 3.5}$	84.8 ± 1.4

4.2 Results

We now present the experimental findings, organized around the four key questions posed in Section 4. Unless stated otherwise, all offline RL experiments use TD3+BC with a labeled ratio of 0.01. Full results for additional settings and labeled ratios are provided in Appendix D.

(i) PU classification performance. Table 1 reports the test accuracy of our PU classifier (based on TED$^{\text{n}}$; (Garg et al., 2021)) for Hopper, Halfcheetah, and Walker2d under body mass shift. The accuracy exceeds 98% in all cases, indicating that the classifier accurately distinguishes positive-domain data from unlabeled data. Similar performance appears under mixture shift and entire body shift, as detailed in Appendix D.3. These findings suggest that the data filtering employed by our method is highly reliable across various shift settings.

(ii) Policy performance with augmented positive data. Tables 2–4 summarize the performance of all methods under body mass shift, mixture shift, and entire body shift. In nearly all settings, our method achieves the highest or near-highest average normalized score among the feasible baselines (*OLP, Sharing-All, DARA, IGDF, Ours*), often approaching the performance of the *Oracle* (which has access to all positive samples). These results confirm that our method is effective even when only a tiny fraction of labeled positive samples are available.

Table 4: The average normalized score and 95% confidence interval from 10 seeds in entire body shift (labeled ratio = 0.01) with TD3+BC. The format is the same as the table for body mass shift.

Entire body shift							
Env	**Quality**	**OLP**	**Sharing-All**	**DARA**	**IGDF**	**Ours**	**Oracle**
Halfcheetah	ME/ME	18.4 ± 3.0	54.0 ± 4.8	14.6 ± 4.7	15.2 ± 5.0	$\mathbf{80.2 \pm 10.6}$	84.7 ± 4.9
	ME/R	21.3 ± 2.6	33.8 ± 10.2	9.9 ± 2.1	16.5 ± 4.0	$\mathbf{89.1 \pm 4.2}$	84.7 ± 4.9

(iii) Effect of dynamics shift magnitude. We examine performance across body mass shift, mixture shift, and entire body shift to analyze how outcomes change with increasing domain mismatch:

- **Robustness of our method.** Our method's performance remains consistently strong, showing minimal degradation under larger shifts (e.g., entire body shift in Table 4).
- **Sharing-All vs. large shift.** For smaller shifts (body mass or mixture shift), *Sharing-All* can occasionally yield competitive or high scores by exploiting the broader coverage. However, performance falls sharply as the shift increases (entire body shift).
- **Domain adaptation baselines (DARA, IGDF).** Although DARA (Liu et al., 2022) and IGDF (Wen et al., 2024) are designed to handle domain differences, both are worse than Sharing-All in most scenarios and degrade further with large shifts. A likely cause is their reliance on submodule training (e.g., domain classifiers or encoders) with very few labeled data, which can become unreliable when unlabeled data may also contain additional positive samples (App. D.1).

These patterns highlight that large domain shifts require careful data selection; our PU-based filtering remains effective, whereas both the naive Sharing-All and the domain adaptation baselines experience performance drops due to the dynamics shift.

(iv) Influence of negative-domain data quality. We analyze the influence of negative-domain data quality on the performance of our method and the baselines by comparing results with different negative-domain data quality. For example, compare ME/ME vs. ME/R or M/M vs.M/R with the same positive dataset quality. We observe:

- **Our method** remains robust regardless of negative-domain quality. The PU filtering consistently prevents the inclusion of harmful transitions, resulting in stable performance gains.
- **Sharing-All and domain adaptation baselines** degrade more significantly when the negative-domain quality is poor (e.g., R), suggesting that merging or adapting from such data can damage performance unless the shift and data mismatch is mild.

These findings indicate that negative-domain data quality is a key factor in the methods used to share unlabeled data. By contrast, PU-based filtering appears less sensitive to variations in the quality.

5 Conclusion and Future Work

This study introduced a novel offline RL setting, positive-unlabeled offline RL (PUORL), incorporating domain-unlabeled data. We then proposed a plug-and-play algorithmic framework for PUORL that uses PU learning to augment the positively labeled data with additional positive-domain samples from the unlabeled data. Experiments on the D4RL benchmark showed that our approach leverages large amounts of unlabeled data to train policies, achieving strong performance. Our method primarily focused on filtering positive data from unlabeled data and training a policy solely with the filtered samples, leaving efficient cross-domain sample sharing as a future direction. Since PU learning is a type of weakly supervised learning (WSL), we believe that extending this setting to other WSL problems could broaden offline RL's practical applications.

References

Joshua Achiam, Harrison Edwards, Dario Amodei, and P. Abbeel. Variational Option Discovery Algorithms. *arXiv preprint arXiv:1807.10299*, 2018.

Johannes Ackermann, Takayuki Osa, and Masashi Sugiyama. Offline reinforcement learning from datasets with structured non-stationarity. *arXiv preprint arXiv:2405.14114*, 2024.

Jessa Bekker and Jesse Davis. Estimating the class prior in positive and unlabeled data through decision tree induction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8.

Jessa Bekker and Jesse Davis. Learning from positive and unlabeled data: a survey. *Machine Learning*, 109(4):719–760, April 2020. ISSN 1573-0565. DOI: 10.1007/s10994-020-05877-5.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/jax-ml/jax.

Marc Claesen, Frank De Smet, Pieter Gillard, Chantal Mathieu, and Bart De Moor. Building classifiers to predict the start of glucose-lowering pharmacotherapy using belgian health expenditure data. *arXiv preprint arXiv:1504.07389*, 2015.

Ignasi Clavera, Anusha Nagabandi, Ronald S Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt: Meta-learning for model-based control. *arXiv preprint arXiv:1803.11347*, 3:3, 2018.

Ron Dorfman, Idan Shenfeld, and Aviv Tamar. Offline Meta Reinforcement Learning – Identifiability Challenges and Effective Data Collection Strategies. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 4607–4618. Curran Associates, Inc., 2021.

Finale Doshi-Velez and George Konidaris. Hidden parameter markov decision processes: a semiparametric regression approach for discovering latent task parametrizations. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pp. 1432–1440. AAAI Press, 2016. ISBN 9781577357704.

M. C. du Plessis and M. Sugiyama. Class prior estimation from positive and unlabeled data. *IEICE Transactions on Information and Systems*, E97-D(5):1358–1362, 2014.

M. C. du Plessis, G. Niu, and M. Sugiyama. Class-prior estimation for learning from positive and unlabeled data. *Machine Learning*, 106(4):463–492, 2017.

Benjamin Eysenbach, Shreyas Chaudhari, Swapnil Asawa, Sergey Levine, and Ruslan Salakhutdinov. Off-Dynamics Reinforcement Learning: Training for Transfer with Domain Classifiers. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: Datasets for Deep Data-Driven Reinforcement Learning. *arXiv preprint arXiv:2004.07219*, 2020.

Scott Fujimoto and Shixiang (Shane) Gu. A Minimalist Approach to Offline Reinforcement Learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 20132–20145. Curran Associates, Inc., 2021.

Scott Fujimoto, Wei-Di Chang, Edward J. Smith, Shixiang Shane Gu, Doina Precup, and David Meger. For SALE: State-Action Representation Learning for Deep Reinforcement Learning. 2023.

Saurabh Garg, Yifan Wu, Alexander J Smola, Sivaraman Balakrishnan, and Zachary Lipton. Mixture Proportion Estimation and PU Learning:A Modern Approach. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 8532–8544. Curran Associates, Inc., 2021.

Arthur Guez, Robert D. Vincent, Massimo Avoli, and Joelle Pineau. Adaptive Treatment of Epilepsy via Batch-mode Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*, 2008.

Assaf Hallak, Dotan Di Castro, and Shie Mannor. Contextual Markov Decision Processes. *arXiv preprint arXiv:1502.02259*, 2015.

Donald Hejna, Lerrel Pinto, and Pieter Abbeel. Hierarchically decoupled imitation for morphological transfer. In *International Conference on Machine Learning*, pp. 4159–4171. PMLR, 2020.

Jan Humplik, Alexandre Galashov, Leonard Hasenclever, Pedro A. Ortega, Yee Whye Teh, and Nicolas Heess. Meta reinforcement learning as task inference. *arXiv preprint arXiv:1905.06424*, 2019.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and Sergey Levine. Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation. In Aude Billard, Anca Dragan, Jan Peters, and Jun Morimoto (eds.), *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pp. 651–673. PMLR, 29–31 Oct 2018.

Dmitry Kalashnikov, Jacob Varley, Yevgen Chebotar, Benjamin Swanson, Rico Jonschkowski, Chelsea Finn, Sergey Levine, and Karol Hausman. MT-Opt: Continuous Multi-Task Robotic Reinforcement Learning at Scale. *arXiv preprint arXiv:2104.08212*, 2021.

Taylor W. Killian, Haoran Zhang, Jayakumar Subramanian, Mehdi Fatemi, and Marzyeh Ghassemi. An Empirical Study of Representation Learning for Reinforcement Learning in Healthcare. 2020.

Kuno Kim, Yihong Gu, Jiaming Song, Shengjia Zhao, and Stefano Ermon. Domain adaptive imitation learning. In *International Conference on Machine Learning*, pp. 5286–5295. PMLR, 2020.

B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE transactions on intelligent transportation systems*, 23(6):4909–4926, 2021.

Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A Survey of Zero-shot Generalisation in Deep Reinforcement Learning. *Journal of Artificial Intelligence Research*, 76: 201–264, January 2023. ISSN 1076-9757. DOI: 10.1613/jair.1.14174.

Ryuichi Kiryo, Gang Niu, Marthinus C du Plessis, and Masashi Sugiyama. Positive-Unlabeled Learning with Non-Negative Risk Estimator. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline Reinforcement Learning with Implicit Q-Learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-Learning for Offline Reinforcement Learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1179–1191. Curran Associates, Inc., 2020.

Sergey Levine, Aviral Kumar, G. Tucker, and Justin Fu. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. *arXiv preprint arXiv:2005.01643*, 2020.

Jiachen Li, Quan Vuong, Shuang Liu, Minghua Liu, Kamil Ciosek, Henrik Christensen, and Hao Su. Multi-task Batch Reinforcement Learning with Metric Learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6197–6210. Curran Associates, Inc., 2020.

Xiaoli Li and Bing Liu. Learning to classify texts using positive and unlabeled data. In *IJCAI*, volume 3, pp. 587–592. Citeseer, 2003.

Jinxin Liu, Hongyin Zhang, and Donglin Wang. DARA: Dynamics-Aware Reward Augmentation in Offline Reinforcement Learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Jinxin Liu, Ziqi Zhang, Zhenyu Wei, Zifeng Zhuang, Yachen Kang, Sibo Gai, and Donglin Wang. Beyond OOD State Actions: Supported Cross-Domain Offline Reinforcement Learning. *arXiv preprint arXiv:2306.12755*, 2023.

Jiafei Lyu, Kang Xu, Jiacheng Xu, Mengbei Yan, Jingwen Yang, Zongzhang Zhang, Chenjia Bai, Zongqing Lu, and Xiu Li. Odrl: A benchmark for off-dynamics reinforcement learning. *arXiv preprint arXiv:2410.20750*, 2024.

Bhairav Mehta, Manfred Diaz, Florian Golemo, Christopher J Pal, and Liam Paull. Active domain randomization. In *Conference on Robot Learning*, pp. 1162–1176. PMLR, 2020.

Soichiro Nishimori. Jax-corl: Clean sigle-file implementations of offline rl algorithms in jax. 2024. URL https://github.com/nissymori/JAX-CORL.

Sindhu Padakandla. A survey of reinforcement learning algorithms for dynamically varying environments. *ACM Computing Surveys (CSUR)*, 54(6):1–25, 2021.

Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

Christian F. Perez, Felipe Petroski Such, and Theofanis Karaletsos. Generalized Hidden Parameter MDPs: Transferable Model-Based RL in a Handful of Trials. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020*, pp. 5403–5411. AAAI Press, 2020. DOI: 10.1609/AAAI.V34I04.5989.

Marthinus Du Plessis, Gang Niu, and Masashi Sugiyama. Convex Formulation for Learning from Positive and Unlabeled Data. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1386–1394, Lille, France, 07–09 Jul 2015. PMLR.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, August 2014. ISBN 978-1-118-62587-3. Google-Books-ID: VvBjBAAAQBAJ.

Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient Off-Policy Meta-Reinforcement Learning via Probabilistic Context Variables. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5331–5340. PMLR, 09–15 Jun 2019.

Clayton Scott. A Rate of Convergence for Mixture Proportion Estimation, with Application to Learning from Noisy Labels. In Guy Lebanon and S. V. N. Vishwanathan (eds.), *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of

Proceedings of Machine Learning Research, pp. 838–846, San Diego, California, USA, 09–12 May 2015. PMLR.

Tianyu Shi, Dong Chen, Kaian Chen, and Zhaojian Li. Offline reinforcement learning for autonomous driving with safety and exploration enhancement. *arXiv preprint arXiv:2110.07067*, 2021.

Reda Bahi Slaoui, William R Clements, Jakob N Foerster, and Sébastien Toth. Robust visual domain randomization for reinforcement learning. *arXiv preprint arXiv:1910.10537*, 2019.

Shagun Sodhani, Amy Zhang, and Joelle Pineau. Multi-Task Reinforcement Learning with Context-based Representations. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9767–9779. PMLR, 18–24 Jul 2021.

Masashi Sugiyama, Han Bao, Takashi Ishida, Nan Lu, and Tomoya Sakai. *Machine learning from weak supervision: An empirical risk minimization approach*. MIT Press, 2022.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning, second edition: An Introduction*. MIT Press, November 2018. ISBN 978-0-262-35270-3. Google-Books-ID: uWV0DwAAQBAJ.

Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36:11592–11620, 2023.

Qiang Wang, Robert McCarthy, David Cordova Bulens, Kevin McGuinness, Noel E. O'Connor, Francisco Roldan Sanchez, Nico Gürtler, Felix Widmaier, and Stephen J. Redmond. Improving Behavioural Cloning with Positive Unlabeled Learning. In *7th Annual Conference on Robot Learning*, 2023.

Xiaoyu Wen, Chenjia Bai, Kang Xu, Xudong Yu, Yang Zhang, Xuelong Li, and Zhen Wang. Contrastive representation for data filtering in cross-domain offline reinforcement learning. *arXiv preprint arXiv:2405.06192*, 2024.

Jinwei Xing, Takashi Nagata, Kexin Chen, Xinyun Zou, Emre Neftci, and Jeffrey L Krichmar. Domain adaptation in reinforcement learning via latent unified state representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10452–10459, 2021.

Danfei Xu and Misha Denil. Positive-Unlabeled Reward Learning. In Jens Kober, Fabio Ramos, and Claire Tomlin (eds.), *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pp. 205–219. PMLR, 16–18 Nov 2021.

Kang Xu, Chenjia Bai, Xiaoteng Ma, Dong Wang, Bin Zhao, Zhen Wang, Xuelong Li, and Wei Li. Cross-domain policy adaptation via value-guided data filtering. *Advances in Neural Information Processing Systems*, 36:73395–73421, 2023a.

Kang Xu, Chenjia Bai, Xiaoteng Ma, Dong Wang, Bin Zhao, Zhen Wang, Xuelong Li, and Wei Li. Cross-Domain Policy Adaptation via Value-Guided Data Filtering, October 2023b. arXiv:2305.17625 [cs].

Zhenghai Xue, Qingpeng Cai, Shuchang Liu, Dong Zheng, Peng Jiang, Kun Gai, and Bo An. State Regularized Policy Optimization on Data with Dynamics Shift. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Kai Yan, Alexander G. Schwing, and Yu-Xiong Wang. A Simple Solution for Offline Imitation from Observations and Examples with Possibly Incomplete Trajectories, 2023.

Minjong Yoo, Sangwoo Cho, and Honguk Woo. Skills Regularized Task Decomposition for Multi-task Offline Reinforcement Learning. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

Tianhe Yu, Aviral Kumar, Yevgen Chebotar, Karol Hausman, Chelsea Finn, and Sergey Levine. How to Leverage Unlabeled Data in Offline Reinforcement Learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 25611–25635. PMLR, 17–23 Jul 2022.

Amy Zhang, Shagun Sodhani, Khimya Khetarpal, and Joelle Pineau. Multi-Task Reinforcement Learning as a Hidden-Parameter Block MDP. *arXiv preprint arXiv:2007.07206*, 2020.

Luisa Zintgraf, Sebastian Schulze, Cong Lu, Leo Feng, Maximilian Igl, Kyriacos Shiarlis, Yarin Gal, Katja Hofmann, and Shimon Whiteson. VariBAD: Variational Bayes-Adaptive Deep RL via Meta-Learning. *Journal of Machine Learning Research*, 22(289):1–39, 2021.

Konrad Zolna, Alexander Novikov, Ksenia Konyushkova, Caglar Gulcehre, Ziyun Wang, Yusuf Aytar, Misha Denil, Nando de Freitas, and Scott E. Reed. Offline Learning from Demonstrations and Unlabeled Experience. *arXiv preprint arXiv:2011.13885*, 2020.

Supplementary Materials

The following content was not necessarily subject to peer review.

A Related Work

In this section, we provide a comprehensive overview of the related work.

Domain-adaptation in online RL. Various approaches have been proposed to tackle domain adaptation in online reinforcement learning, each leveraging different techniques to handle variations in environment dynamics. Imitation learning strategies (Kim et al., 2020; Hejna et al., 2020) utilize expert demonstrations or hierarchical policies to guide the agent in the target domain, while domain randomization methods (Slaoui et al., 2019; Mehta et al., 2020) train agents across diverse simulated environments to build robustness against variations. Representation learning (Xing et al., 2021) extracts domain-invariant features to facilitate transfer, and system identification approaches (Clavera et al., 2018) learn latent parameters of the environment's dynamics to adapt policies online. Data-filtering techniques (Xu et al., 2023a) selectively incorporate experience from different domains to reduce negative transfer, and reward modification based on learned classifiers (Eysenbach et al., 2021) helps align rewards when discrepancies in dynamics lead to misaligned feedback signals.

RL with multiple MDPs. Contextual MDPs (CMDPs) formalize the RL problem with multiple environments as MDPs controlled by a variable known as a "context" (Hallak et al., 2015). Different contexts define different types of problems (Kirk et al., 2023). We focus on the case where the context is a binary task ID determining the dynamics. Thanks to its generality, the CMDP can encapsulate a wide range of RL problems, such as multi-task RL (Zhang et al., 2020; Li et al., 2020; Sodhani et al., 2021) and meta-RL (Zintgraf et al., 2021; Dorfman et al., 2021). Depending on the observability of the context, the solution to the RL problem within CMDPs differs. We can utilize the information in policy training if the context is observable. For example, acquiring a representation of the environment using self-supervised learning (Sodhani et al., 2021; Humplik et al., 2019; Achiam et al., 2018; Li et al., 2020) is common in addressing this objective. In offline RL, MBML (Multi-task Batch RL with Metric Learning) employed metric learning to acquire a robust representation of discrete contexts in an offline setting (Li et al., 2020). Unlike these approaches, our method considers settings where only a subset of the data has observable contexts.

CMDPs with unobservable contexts are also known as Hidden-Parameter (HiP)-MDPs (Doshi-Velez & Konidaris, 2016; Perez et al., 2020). In HiP-MDPs, previous works typically focused on training an inference model for the context from histories of multiple time steps (Rakelly et al., 2019; Zintgraf et al., 2021; Yoo et al., 2022; Dorfman et al., 2021; Ackermann et al., 2024). Since we consider transition-based datasets without trajectory information, such methods are not applicable in our setting.

Unlabeled data in RL. In previous work, "unlabeled data" refers to two settings: reward-unlabeled data and data with the quality of the behavioral policy unknown. In the first case, the unlabeled data consist of transitions without rewards (Xu & Denil, 2021; Zolna et al., 2020; Yu et al., 2022). Several studies have attempted to learn the reward function from reward-unlabeled data using the PU learning technique and then utilize this learned reward function in subsequent RL routines (Xu & Denil, 2021; Zolna et al., 2020). In the offline multi-task RL literature, Yu et al. (2022) explored conservatively using reward-unlabeled data, i.e., setting the reward of the unlabeled transitions to zero. In our study, the label corresponds to a specific domain, while they regard the reward as a label. In the second case, the unlabeled data is a mixture of transitions from policies of unknown quality. In offline RL, previous works attempted to extract high-quality data from unlabeled data using PU learning (Wang et al., 2023; Yan et al., 2023). In our setting, labels correspond to specific domains, not the quality of the behavioral policy.

B Details of Experimental Setup

TD3+BC	
Critic Learning Rate	3×10^{-4}
Actor Learning Rate	3×10^{-4}
Discount Factor	0.99
Target Update Rate	5×10^{-3}
Policy Noise	0.2
Policy Noise Clipping	(-0.5, 0.5)
Policy Update Frequency	Variable
TD3+BC Hyperparameter α	2.5
Actor Hidden Dims	(256, 256)
Critic Hidden Dims	(256, 256)
IQL	
Critic Learning Rate	3×10^{-4}
Actor Learning Rate	3×10^{-4}
Discount Factor	0.99
Expectile	0.7
Temperature	3.0
Target Update Rate	5×10^{-3}
Actor Hidden Dims	(256, 256)
Critic Hidden Dims	(256, 256)

Table 5: Hyperparameters for TD3+BC and IQL.

B.1 PU Learning

Explanation of $\mathrm{TED}^{\mathrm{n}}$ (Garg et al., 2021). Here, we briefly explain the $\mathrm{TED}^{\mathrm{n}}$ (Garg et al., 2021) we used in our experiments. $\mathrm{TED}^{\mathrm{n}}$ consists of two subroutines for the mixture proportion estimation, Best Bin Estimation (BBE), and for PU learning, Conditional Value Ignoring Risk (CVIR). They iterate these subroutines. Given the estimated mixture proportion $\hat{\alpha}$ by BBE, CVIR first discards $\hat{\alpha}$ samples from unlabeled data based on the output probability of being positive from the current classifier f. The discarded samples are seemingly positive data. The classifier is then trained using the labeled positive data and the remaining unlabeled data. On the other hand, in BBE, we estimate the mixture proportion using the output of the classifier f with the samples in the validation dataset as inputs.

Training and evaluation. The PU learning method $\mathrm{TED}^{\mathrm{n}}$ involved two phases: warm-up and main training. We assigned 10 epochs for the warm-up step and 100 epochs for the main training step. We utilized a 3-layer MLP with ReLU for the classifier's network architecture. In our method, the trained classifier was then frozen and shared across different random seeds of offline RL training with identical data generation configurations, such as the positive-to-negative and unlabeled ratios. We reported the average and standard deviation of the test accuracy over 5 random seeds.

B.2 Offline RL

For offline RL, we learned a policy with 1 million update steps. For both TD3+BC (Fujimoto & Gu, 2021) and IQL (Kostrikov et al., 2022) we used the same hyperparameters for all baselines and settings (Table 5). We evaluated the offline RL agent using the normalized score provided by D4RL (Fu et al., 2020). To evaluate the offline RL routine's algorithmic stability, we trained with 10 different random seeds. For each seed, we calculated the average normalized score over 10 episodes. We reported the overall mean and 95% confidence interval from these averaged scores.

B.3 Baselines

Here, we provide a detailed explanation of the Domain-Adaptation baselines.

Algorithm 2 DARA

Require: Target offline data $\mathcal{D}_{\mathrm{t}}$ and source offline data $\mathcal{D}_{\mathrm{s}}$ and η.

1: Learn classifier $q_{\mathrm{sas}} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ and $q_{\mathrm{sa}} : \mathcal{S} \times \mathcal{A} \to [0, 1]$ from $\mathcal{D}_{\mathrm{t}}$ and $\mathcal{D}_{\mathrm{s}}$.

2: For all (s, a, r, s') in $\mathcal{D}_{\mathrm{s}}$:

$$\Delta r(s, a, s') = \log \frac{q_{\mathrm{sas}}(\mathrm{source}|s, a, s')}{q_{\mathrm{sas}}(\mathrm{target}|s, a, s')} + \log \frac{q_{\mathrm{sa}}(\mathrm{source}|s, a)}{q_{\mathrm{sa}}(\mathrm{target}|s, a)} \tag{2}$$

$$r \leftarrow r - \eta \Delta r \tag{3}$$

3: Learn policy with $\mathcal{D}_{\mathrm{t}} \cup \mathcal{D}_{\mathrm{s}}$.

Algorithm 3 IGDF: Info-Gap Data Filtering Algorithm

Require: Source offline data $\mathcal{D}_{\mathrm{s}}$, target offline data $\mathcal{D}_{\mathrm{t}}$,

1: Initialize policy π, value function Q, encoders $\phi(s, a), \psi(s')$,

2: data filter ratio ξ, importance ratio α, batch size B.

3: **// Contrastive Representation Learning**

4: Optimize the contrastive objective in Eq. (6) to train the encoder networks $\phi(s, a)$ and $\psi(s')$.

5: **// Data Filtering algorithm**

6: **for** each gradient step **do**

7: Sample a batch $b_{\mathrm{src}} := \{(s, a, r, s')\}^{\frac{B}{2}\xi}$ from $\mathcal{D}_{\mathrm{s}}$

8: Sample a batch $b_{\mathrm{tar}} := \{(s, a, r, s')\}^{\frac{B}{2}}$ from $\mathcal{D}_{\mathrm{t}}$

9: Select the top-ξ samples from b_{src} ranked by $h(s, a, s') == \exp(\phi(s, a)^{\top} \psi(s'))$

10: Combine the top-ξ samples from b_{src} with all samples from b_{tar}

11: Optimize the value function Q_θ via Eq. (8)

12: Learn the policy $\pi(a \mid s)$ via offline RL algorithms

13: **end for**

DARA. Here, we explain the Domain-Adaptation (DA) baseline used in Section 4. For domain adaptation in offline RL, we utilized the Dynamics-Aware Reward Augmentation (DARA) (Liu et al., 2022). In domain adaptation in offline RL, we focus on the performance in a target domain $\mathcal{M}_{\mathrm{t}}$ with a limited amount of target domain data $\mathcal{D}_{\mathrm{t}}$. To address this scarcity, domain adaptation uses data $\mathcal{D}_s$ from the source domain $\mathcal{M}_{\mathrm{s}}$. DARA modifies the source domain data's reward using a trained domain classifier and then utilizes this data with the modified reward for offline RL. Lacking full domain labels in PUORL, we treated the positive data $\mathcal{D}_{\mathrm{p}}$ as target domain data and the domain-unlabeled data $\mathcal{D}_{\mathrm{u}}$ as source domain data, training the classifier with 5000 steps with batch size 256. We set $\eta = 0.1$ following original paper (Liu et al., 2022).

IGDF. IGDF (Liu et al., 2022) is a method that uses the information of the source domain to improve the performance of the target domain. IGDF filters the source domain data using encoder networks trained with contrastive learning with target domain data as positive samples and source domain data as negative samples. Similar to DARA, this method is also plug-and-play. We set the representation dimension to 64 and trained the encoder with 7000 steps with batch size 256. The data filter ratio ξ is set to 0.75 following the original paper (Wen et al., 2024).

C Extention to Reward Shift

To extend PUORL in for reward shift, we define the positive and negative MDPs as follows: positive MDP $\mathcal{M}_{\mathrm{p}} := (\mathcal{S}, \mathcal{A}, P, \rho, r_{\mathrm{p}}, \gamma)$, which we target for and negative MDPs $\{\mathcal{M}_{\mathrm{n}}^k := (\mathcal{S}, \mathcal{A}, P, \rho, r_{\mathrm{n}}^k, \gamma)\}_{k=1}^N$, which share the same state and action spaces and dynamics. For each MDP, there exist fixed behavioral policies: π_{p} for positive MDP and π_{n}^k for negative MDPs. They induce the stationary distributions over the state-action pair denoted as $\mu_{\mathrm{p}}(s,a)$ and $\mu_{\mathrm{n}}^k(s,a)$ for all $k \in \{1, \ldots, N\}$. We define $\mu_{\mathrm{n}}(s,a) := \sum_{k=1}^N \eta_k \mu_{\mathrm{n}}^k(s,a)$, where $\eta_k \in [0,1], \sum_{k=1}^N \eta_k = 1$ is the MDP-mixture proportion.

We are given two datasets:

- **Positive data**: explicitly labeled target-domain transitions, $\mathcal{D}_{\mathrm{p}} := \{(s_i, a_i, r_i, s_i', +1)\}_{i=1}^{n_{\mathrm{p}}}$. These transitions are i.i.d. samples from $\mu_{\mathrm{p}}(s,a)$, r_{p}, and P.
- **Domain-unlabeled data**: a mixture of positive and negative-domain transitions, $\mathcal{D}_{\mathrm{u}} := \{(s_i, a_i, r_i, s_i')\}_{i=1}^{n_{\mathrm{u}}}$. These transitions are i.i.d. samples from $\mu_{\mathrm{u}}(s,a) := \alpha_{\mathrm{p}}\mu_{\mathrm{p}}(s,a) + \alpha_{\mathrm{n}}\mu_{\mathrm{n}}(s,a)$, r_{n}, and P.

Instead of taking transition, (s,a,s'), we take (s,a,r) to train the classifier with PU learning based on the reward shift.

Algorithm 4 Data filtering for the positive domain with reward shift

Initialize classifier parameters ψ of classifier f
Initialize policy parameters θ and value function parameters ϕ
Initialize experience replay buffer $\mathcal{D}_{\mathrm{p}}$ and $\mathcal{D}_{\mathrm{u}}$
Specify epochs K_{PU}, K_{RL}
for iteration $k \in [0, \ldots, K_{\mathrm{PU}}]$ **do** ▷ PU learning routine
    Update ψ on $\mathcal{D}_{\mathrm{p}}$ and $\mathcal{D}_{\mathrm{u}}$ by PU learning with MPE
end for
$\tilde{\mathcal{D}}_{\mathrm{p}} \leftarrow \mathcal{D}_{\mathrm{p}} \cup \{(s,a,r,s') \in \mathcal{D}_{\mathrm{u}} : f_\psi(s,a,r) = +1\}$ ▷ Data filtering
for iteration $k \in [0, \ldots, K_{\mathrm{RL}}]$ **do** ▷ Offline RL routine
    Update θ and ϕ on $\tilde{\mathcal{D}}_{\mathrm{p}}$ by Offline RL method
end for
Output θ and ϕ

Table 6: The average normalized score and 95% confidence interval from 10 seeds in body mass shift (labeled ratio = 0.03) with TD3+BC. Of feasible methods (OLP, Sharing-All, DARA, IGDF, Ours), the best average is in **blue**. The last column (Oracle) is for reference (ratio=0.05).

Body mass shift (0.03)							
Env	**Quality**	**OLP**	**Sharing-All**	**DARA**	**IGDF**	**Ours**	**Oracle**
Hopper	ME/ME	50.0 ± 10.1	52.3 ± 8.1	89.9 ± 13.8	71.1 ± 10.1	$\mathbf{90.4 \pm 9.2}$	98.2 ± 8.4
	ME/R	46.6 ± 8.9	86.8 ± 12.6	73.8 ± 9.6	77.2 ± 6.9	$\mathbf{90.0 \pm 10.9}$	98.2 ± 8.4
	M/M	57.4 ± 13.0	48.1 ± 3.3	$\mathbf{59.8 \pm 3.0}$	56.3 ± 5.1	49.7 ± 3.5	48.9 ± 2.8
	M/R	44.7 ± 6.0	45.4 ± 2.1	55.0 ± 4.7	$\mathbf{59.3 \pm 3.8}$	47.4 ± 2.6	48.9 ± 2.8
Halfcheetah	ME/ME	24.4 ± 1.9	$\mathbf{78.6 \pm 3.6}$	48.3 ± 4.2	45.5 ± 4.7	75.6 ± 8.3	86.9 ± 4.4
	ME/R	25.3 ± 2.5	70.3 ± 7.7	23.6 ± 2.1	22.5 ± 4.2	$\mathbf{82.2 \pm 6.9}$	86.9 ± 4.4
	M/M	43.4 ± 1.6	41.0 ± 0.7	45.9 ± 0.3	46.0 ± 0.3	$\mathbf{48.7 \pm 0.2}$	48.8 ± 0.3
	M/R	45.4 ± 0.5	39.6 ± 8.1	46.7 ± 1.7	45.2 ± 1.9	$\mathbf{48.5 \pm 0.2}$	48.8 ± 0.3
Walker2d	ME/ME	71.7 ± 26.1	87.9 ± 0.9	101.8 ± 8.7	103.9 ± 4.0	$\mathbf{108.7 \pm 0.2}$	108.5 ± 0.4
	ME/R	87.0 ± 13.5	89.2 ± 23.7	45.2 ± 16.6	26.5 ± 15.8	$\mathbf{108.8 \pm 0.3}$	108.5 ± 0.4
	M/M	57.3 ± 11.5	80.9 ± 0.9	67.0 ± 8.6	61.5 ± 12.8	$\mathbf{83.8 \pm 1.1}$	84.6 ± 0.6
	M/R	64.3 ± 6.0	77.2 ± 5.2	47.6 ± 15.8	51.5 ± 10.6	$\mathbf{84.5 \pm 0.6}$	84.6 ± 0.6

Table 7: The average normalized score and 95% confidence interval from 10 seeds in mixture shift (labeled ratio = 0.03) with TD3+BC. Of feasible methods (OLP, Sharing-All, DARA, IGDF, Ours), the best average is in **blue**. The last column (Oracle) is for reference (ratio=0.05).

Mixture shift (0.03)							
Env	**Quality**	**OLP**	**Sharing-All**	**DARA**	**IGDF**	**Ours**	**Oracle**
Hopper	ME/ME	55.9 ± 10.4	68.2 ± 19.1	71.7 ± 8.7	74.3 ± 10.1	$\mathbf{98.1 \pm 8.5}$	96.4 ± 8.2
	ME/R	48.8 ± 7.7	84.6 ± 10.5	80.5 ± 4.3	69.2 ± 12.9	$\mathbf{100.7 \pm 4.1}$	96.4 ± 8.2
	M/M	45.0 ± 8.1	50.4 ± 5.3	55.6 ± 2.6	52.9 ± 4.0	$\mathbf{87.6 \pm 8.7}$	45.9 ± 1.5
	M/R	48.6 ± 1.9	49.6 ± 5.9	57.8 ± 3.2	55.1 ± 3.5	$\mathbf{49.2 \pm 2.0}$	45.9 ± 1.5
Halfcheetah	ME/ME	24.3 ± 4.4	82.1 ± 1.3	41.6 ± 5.4	43.1 ± 8.7	$\mathbf{80.0 \pm 9.0}$	81.3 ± 9.6
	ME/R	21.6 ± 4.9	82.1 ± 6.8	22.6 ± 2.8	24.2 ± 7.1	$\mathbf{67.1 \pm 9.3}$	81.3 ± 9.6
	M/M	35.8 ± 2.1	48.1 ± 1.3	39.8 ± 2.5	40.4 ± 2.9	$\mathbf{48.7 \pm 0.3}$	48.7 ± 0.2
	M/R	32.5 ± 2.0	51.7 ± 1.4	14.7 ± 3.2	16.8 ± 4.4	$\mathbf{48.8 \pm 0.2}$	48.7 ± 0.2
Walker2d	ME/ME	80.3 ± 15.1	100.2 ± 6.7	86.5 ± 19.7	96.4 ± 12.7	$\mathbf{108.3 \pm 0.2}$	108.5 ± 0.4
	ME/R	90.8 ± 14.4	101.8 ± 23.3	72.1 ± 14.4	89.4 ± 20.4	$\mathbf{108.6 \pm 0.3}$	108.5 ± 0.4
	M/M	61.6 ± 8.0	81.8 ± 2.4	62.4 ± 11.3	71.0 ± 7.9	$\mathbf{83.9 \pm 0.8}$	84.8 ± 1.4
	M/R	64.2 ± 7.8	79.1 ± 3.7	66.3 ± 16.7	65.7 ± 16.8	$\mathbf{82.8 \pm 1.7}$	84.8 ± 1.4

D Supplemental result

In this section, we present the supplementary results and discussion to provide additional insights into the main findings.

D.1 Results with TD3+BC with labeled ratio = 0.03

Table 6– 8 show the results of TD3+BC with the labeled ratio = 0.03. For all the results, our method achieves the best performance in almost all the settings, indicating its efficacy in PUORL. Another point to note is that the performance of the domain adaptation baselines is improved compared with the labeled ratio of 0.01, indicating the severe influence of extremely limited labeled target domain data.

D.2 Results with IQL

Here, we provide the experimental results with IQL (Kostrikov et al., 2022). Table 9–11 show the results with the labeled ratio = 0.01. The results show that our method achieves the best performance in 17 out of 26 settings. Overall, the results with hopper are unstable and worse for all methods, indicating that the performance of IQL is sensitive in Hopper with limited data (30% in maximum).

Table 8: The average normalized score and 95% confidence interval from 10 seeds in entire body shift (labeled ratio = 0.03) with TD3+BC. Of feasible methods (OLP, Sharing-All, DARA, IGDF, Ours), the best average is in **blue**. The last column (Oracle) is for reference (ratio=0.05).

Entire body shift (0.03)							
Env	**Quality**	**OLP**	**Sharing-All**	**DARA**	**IGDF**	**Ours**	**Oracle**
Halfcheetah	ME/ME	23.1 ± 3.9	51.8 ± 5.3	25.7 ± 4.8	28.2 ± 3.2	$\mathbf{82.7 \pm 5.8}$	84.7 ± 4.9
	ME/R	25.6 ± 3.7	28.1 ± 8.1	13.6 ± 3.1	13.4 ± 3.2	$\mathbf{82.1 \pm 7.2}$	84.7 ± 4.9

Table 9: The average normalized score and 95% confidence interval from 10 seeds in body mass shift (labeled ratio = 0.01) with IQL.

Body mass shift		**OLP**	**Sharing-All**	**DARA**	**IGDF**	**PU**	**Oracle**
Hopper	ME/ME	23.9 ± 5.9	38.99 ± 14.1	37.44 ± 7.6	29.75 ± 5.84	$\mathbf{39.72 \pm 8.64}$	54.3 ± 14.9
	ME/R	23.35 ± 4.23	7.79 ± 0.16	7.74 ± 0.25	11.21 ± 5.02	$\mathbf{42.04 \pm 9.95}$	54.3 ± 14.9
	M/M	37.37 ± 4.58	37.06 ± 1.28	35.73 ± 1.17	36.28 ± 7.38	$\mathbf{56.37 \pm 3.83}$	54.2 ± 3.3
	M/R	$\mathbf{33.56 \pm 3.47}$	8.04 ± 0.16	21.68 ± 8.98	12.37 ± 5.4	18.66 ± 8.36	54.3 ± 14.9
Halfcheetah	ME/ME	0.7 ± 0.86	51.53 ± 3.96	54.41 ± 2.17	51.33 ± 3.18	$\mathbf{82.72 \pm 3.57}$	87.3 ± 2.7
	ME/R	-0.11 ± 0.47	26.95 ± 7.54	47.71 ± 7.73	38.39 ± 4.55	$\mathbf{84.83 \pm 4.06}$	87.3 ± 2.7
	M/M	3.89 ± 1.62	37.3 ± 0.2	36.93 ± 0.22	36.43 ± 0.7	$\mathbf{46.33 \pm 0.46}$	46.5 ± 0.1
	M/R	6.31 ± 3.66	41.64 ± 2.38	43.56 ± 0.5	41.18 ± 1.9	$\mathbf{46.57 \pm 0.13}$	46.5 ± 0.1
Walker2d	ME/ME	4.3 ± 2.75	90.68 ± 0.38	88.92 ± 6.82	96.73 ± 7.74	$\mathbf{110.32 \pm 0.78}$	109.1 ± 1.4
	ME/R	6.64 ± 6.22	65.18 ± 12.23	62.04 ± 17.2	66.61 ± 10.58	$\mathbf{88.57 \pm 14.48}$	109.1 ± 1.4
	M/M	14.42 ± 5.57	$\mathbf{82.77 \pm 0.45}$	82.42 ± 0.63	74.85 ± 6.35	73.49 ± 9.69	75.6 ± 5.2
	M/R	4.79 ± 3.4	52.22 ± 7.44	47.96 ± 6.59	$\mathbf{54.33 \pm 10.88}$	50.34 ± 19.19	75.6 ± 5.2

D.3 Classifier Performance

Here, we review the performance of the classifier under the mixture shift. Seeing Table 12–13, we can see that the PU classifier achieved higher than 98% accuracy, demonstrating the efficacy of PU learning under mixture shift and entire body shift.

Table 10: The average normalized score and 95% confidence interval from 10 seeds in mixture shift (labeled ratio = 0.01) with IQL.

Mixture shift		OLP	Sharing-All	DARA	IGDF	PU	Oracle
Hopper	ME/ME	20.43 ± 6.53	24.58 ± 4.47	43.0 ± 10.54	$\mathbf{43.08 \pm 11.08}$	35.28 ± 8.5	54.3 ± 14.9
	ME/R	19.1 ± 3.74	22.08 ± 3.58	$\mathbf{36.36 \pm 7.83}$	28.43 ± 8.29	29.03 ± 6.45	54.3 ± 14.9
	M/M	29.96 ± 3.98	$\mathbf{61.82 \pm 8.82}$	54.86 ± 9.64	51.55 ± 6.13	50.1 ± 3.34	54.2 ± 3.3
	M/R	33.98 ± 3.15	44.78 ± 7.52	$\mathbf{50.47 \pm 2.51}$	47.2 ± 2.13	45.86 ± 1.44	54.2 ± 3.3
Halfcheetah	ME/ME	0.26 ± 0.67	62.46 ± 1.43	56.73 ± 3.36	57.48 ± 4.63	$\mathbf{69.5 \pm 2.97}$	87.3 ± 2.7
	ME/R	1.13 ± 1.09	57.94 ± 7.1	49.84 ± 9.63	51.46 ± 8.41	$\mathbf{72.83 \pm 4.74}$	87.3 ± 2.7
	M/M	5.58 ± 2.32	48.05 ± 0.64	$\mathbf{48.43 \pm 0.42}$	46.36 ± 2.09	46.54 ± 0.19	46.5 ± 0.1
	M/R	7.16 ± 4.39	44.97 ± 0.46	44.84 ± 1.45	43.16 ± 1.04	$\mathbf{46.58 \pm 0.21}$	46.5 ± 0.1
Walker2d	ME/ME	3.28 ± 2.08	93.95 ± 19.68	96.8 ± 13.09	93.13 ± 11.37	$\mathbf{108.46 \pm 2.65}$	109.1 ± 1.4
	ME/R	6.33 ± 3.13	93.38 ± 8.08	89.98 ± 12.61	88.98 ± 13.59	$\mathbf{98.96 \pm 12.36}$	109.1 ± 1.4
	M/M	3.55 ± 2.43	74.68 ± 2.73	$\mathbf{72.85 \pm 5.38}$	64.14 ± 8.58	64.45 ± 13.22	75.6 ± 5.2
	M/R	11.84 ± 7.15	50.24 ± 6.64	60.78 ± 7.39	59.43 ± 7.88	$\mathbf{62.15 \pm 18.06}$	75.6 ± 5.2

Table 11: The average normalized score and 95% confidence interval from 10 seeds in halfcheetah vs walker2d shift (labeled ratio = 0.01) with IQL.

Halfcheetah vs Walker2d		OLP	Sharing-All	DARA	IGDF	PU	Oracle
Halfcheetah	ME/ME	0.28 ± 0.37	40.78 ± 3.42	52.55 ± 4.95	53.96 ± 4.46	$\mathbf{89.31 \pm 1.92}$	87.3 ± 2.7
	ME/R	0.07 ± 0.57	35.64 ± 3.58	36.93 ± 3.33	31.12 ± 4.52	$\mathbf{86.73 \pm 2.91}$	87.3 ± 2.7

Table 12: The results of the PU classifier in the mixture shift with labeled ratio = 0.01 and 0.03. For each setting, we reported the average and standard deviation of the test accuracy over 5 seeds.

Env	Ratio	ME/ME	ME/R	M/M	M/R
Hopper	0.01	98.92 ± 0.54	98.91 ± 0.14	99.33 ± 0.20	99.21 ± 0.08
	0.03	99.44 ± 0.11	99.22 ± 0.09	99.79 ± 0.11	99.42 ± 0.05
Halfcheetah	0.01	99.43 ± 0.10	99.42 ± 0.10	99.38 ± 0.05	99.35 ± 0.03
	0.03	99.63 ± 0.04	99.56 ± 0.03	99.39 ± 0.02	99.32 ± 0.19
Walker2d	0.01	98.49 ± 0.14	98.02 ± 0.16	98.63 ± 0.19	98.05 ± 0.12
	0.03	99.00 ± 0.07	98.83 ± 0.10	99.26 ± 0.08	98.81 ± 0.25

Table 13: The results of the PU classifier in the entire body shift with labeled ratio = 0.01 and 0.03. For each setting, we reported the average and standard deviation of the test accuracy over 5 seeds.

Env	Ratio	ME/ME	ME/R	M/M	M/R
Halfcheetah	0.01	99.76 ± 0.28	99.87 ± 0.15	99.79 ± 0.11	99.74 ± 0.13
	0.03	99.98 ± 0.01	99.93 ± 0.04	99.95 ± 0.06	99.96 ± 0.21