Localized Dynamics-Aware Domain Adaptation for Off-Dynamics Offline Reinforcement Learning

Zhangjie Xia[1]*, **Yu Yang**[2]*, **Pan Xu**[2]
[1]New York University [2]Duke University
zx1357@nyu.edu, {yu.yang, pan.xu}@duke.edu

Abstract

Off-dynamics offline reinforcement learning (RL) aims to learn a policy for a target domain using limited target data and abundant source data collected under different transition dynamics. Existing methods typically address dynamics mismatch either globally over the state space or via pointwise data filtering; these approaches can miss localized cross-domain similarities or incur high computational cost. We propose Localized Dynamics-Aware Domain Adaptation (LoDADA), which exploits localized dynamics mismatch to better reuse source data. LoDADA clusters transitions from source and target datasets and estimates cluster-level dynamics discrepancy via domain discrimination. Source transitions from clusters with small discrepancy are more likely to be retained compared with those from clusters with large discrepancy. This yields a fine-grained and scalable data selection strategy that avoids overly coarse global assumptions and expensive per-sample filtering. We provide theoretical insights and extensive experiments across environments with diverse global and local dynamics shifts. Results show that LoDADA consistently outperforms state-of-the-art off-dynamics offline RL methods by better leveraging localized distribution mismatch. Our code is available at https://github.com/panxulab/LoDADA.

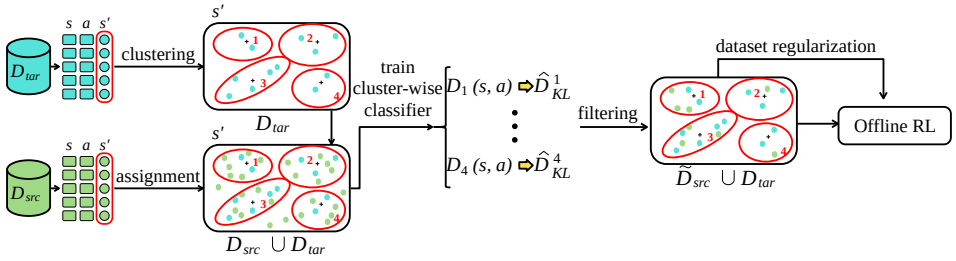


Figure 1: **An overview of our proposed framework.** We first perform K-means clustering on the next state for the target dataset and then assign source transitions to clusters via their next state. We estimate the local KL divergence between source and target dynamics in each cluster using cluster-wise classifiers, and selectively retain source transitions for downstream offline RL algorithms. We further introduce a dataset regularization term to ensure policy consistency with the target dataset.

1 Introduction

The goal of reinforcement learning (RL) [16, 39] is to learn a policy for optimal decision making by interacting with an environment and receiving reward feedback. However, direct interaction with the

*Equal Contribution

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

environment (the *target domain*) can be dangerous or time-consuming in real-world applications such as autonomous driving [17] and healthcare [5]. As a result, practitioners often collect experience in a structurally similar *source domain* and then transfer the learned policy. A key challenge is that a policy trained in the source domain may perform poorly when deployed in the target domain due to mismatched transition dynamics. This issue is usually referred to as *off-dynamics RL* [8, 15]. In this paper, we study a practical setting, *off-dynamics offline RL* [22, 41], where the agent cannot interact with the environment online and must instead learn from a limited target dataset together with a larger source dataset collected under different dynamics.

There has been growing interest in off-dynamics offline RL, with prior work that can be broadly categorized into three types: (i) data augmentation methods that modify rewards in the source data using dynamics-aware regularization [22, 31, 41, 12]; (ii) data filtering approaches that select source transitions based on their similarity to target transitions [44, 42, 33]; and more recently (iii) data generation methods that use source data to learn target dynamics and generate synthetic target samples [18, 13, 45]. However, most methods treat source-target dynamics mismatch globally over the state-action space, which can be overly conservative in regions where the domains align and overly optimistic where they differ substantially. For example, methods such as [22] train a single domain classifier over (s, a, s') tuples across the full dataset, which is challenging with limited target data. A notable exception is pointwise data filtering via optimal transport [33], but its computational cost scales with dataset size, limiting its practicality [6]. Therefore, existing approaches can be data-inefficient and often fail to exploit fine-grained local similarities that would enable more robust and efficient transfer from the source to the target domain.

Our proposed method, **Lo**calized **D**ynamics-**A**ware **D**omain **A**daptation (**LoDADA**), addresses these challenges in a novel way. Unlike prior approaches that rely on global or pointwise assumptions, LoDADA clusters source and target next states via K-means to identify localized regions in the state space where the two datasets are well aligned. We discard source samples that are far from cluster centroids, retaining those that better match the target data. Within each cluster, we train a classifier to distinguish source from target samples and use its outputs to estimate a local KL divergence between source and target transition dynamics. Clusters with large estimated KL divergence, indicating inconsistency between source and target transitions, are treated as less reliable. Instead of discarding entire clusters, LoDADA retains samples from all clusters to preserve state-space coverage, while prioritizing transitions from clusters with smaller estimated KL divergence. During the policy optimization stage, we further introduce a regularization term that penalizes deviations between the learned policy and a behavior policy which is fit to the target dataset, encouraging the learned policy to remain close to the target-domain's behavior.

By selectively retaining source transitions based on cluster-level KL estimates, LoDADA reduces the bias induced by dynamics mismatch and enables more effective use of source data. This framework provides a principled way to balance selective filtering and cross-domain adaptation under localized dynamics shifts. After the filtering step, we optimize the policy using Implicit Q-Learning (IQL) [19] as a backbone, augmented with a regularization term that encourages the learned policy to stay close to the target behavior policy. Our main contributions are summarized as follows:

- We identify a key limitation of existing off-dynamics offline RL methods: they address dynamics mismatch either globally or pointwise, which can lead to data inefficiency and degraded performance. To address this, we propose LoDADA , which performs cluster-wise filtering by estimating local source-target dynamics discrepancy and retaining the most target-consistent source transitions.
- We provide a theoretical analysis that establishes return lower bounds for the policy learned by LoDADA , offering performance guarantees in the target domain under dynamics mismatch.
- We empirically evaluate our method in four MuJoCo environments in the offline setting with different types and levels of dynamics shifts and demonstrate the effectiveness of our method with the 15.5% aggregate improvement over the strongest baselines in gravity and friction shifts, 32.7% on morphology shift and 29.3% on local perturbation. We further show the superior performance of LoDADA on challenging navigation tasks in AntMaze and manipulation tasks in Adroit.

2 Related Work

Domain adaptation in RL. Domain adaptation in RL focuses on adapting policies across domains with differing transition dynamics. Prior work include policy and value function transfer which

reuse knowledge from a source domain to accelerate learning in a target domain [1, 44], and robust MDPs which optimize policies against the worst-case dynamics within a predefined uncertainty set [25, 15, 11]. However, these approaches often require carefully specified uncertainty sets and may fail to cover the true target dynamics. More importantly, they tend to be overly conservative due to the rigid assumption that there is no access to target data at all. More recent work have explored online RL domain adaptation using reward augmentation from dynamics mismatch [8, 12] and representation deviation [31]. In contrast, our method tackles domain adaptation in a purely offline setting through a clustering-based framework, enabling more flexible and efficient reuse of source data under substantial dynamics shifts.

Off-dynamics offline RL. Off-dynamics offline RL aims to leverage an abundant source dataset together with a limited target dataset to improve policy learning in the target domain where online interactions are infeasible. Prior works approached this problem by aligning source and target distributions via importance weighting [35, 20], reward augmentation [22], using adversarial domain adaptation to reduce the mismatch in state-action dynamics [2, 43], or filtering source data based on similarity to the target [24, 42, 33]. Another line of work adopts a dynamics model that is learned in the target domain to enable more principled reuse of source transitions or to help learn a robust Bellman operator under mismatched dynamics [46, 13, 38]. These methods often struggle with severe distribution shifts: importance weighting suffers from high variance under limited coverage, adversarial adaptation can be unstable in practice, heuristic filtering may reduce sample efficiency, and model-based methods are sensitive to model bias. Additionally, people study robust Markov decision processes, aiming to improve policy performance in an unseen target environment [36, 7, 4, 26, 29, 27, 28]. Since they assume that no target-domain data are accessible during training, these approaches rely on carefully designed uncertainty sets and may yield overly conservative policies, which may be unrealistic in practical settings. In contrast, our method balances robustness and coverage by retaining useful source experiences while downweighting unreliable clusters.

Notations: $\mathbb{H}(X)$ is the entropy of random variable X. $z = s \oplus a$ is the vector concatenation of state s and action a, which can be viewed as an identity representation of (s, a).

3 Preliminaries

RL problems are typically formulated as a Markov Decision Process (MDP), specified by a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \rho_0)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P(s'|s, a)$ is the transition kernel, $r(s, a)$ is the reward function given state-action pair (s, a), $\gamma \in [0, 1)$ is the discounting factor, and ρ_0 is the initial state distribution. In the off-dynamics offline setting, we consider two MDPs: a source domain $\mathcal{M}_{\text{src}} = (\mathcal{S}, \mathcal{A}, P_{\text{src}}, r, \gamma, \rho_0)$ and a target domain $\mathcal{M}_{\text{tar}} = (\mathcal{S}, \mathcal{A}, P_{\text{tar}}, r, \gamma, \rho_0)$. The two domains differ only in their transition dynamics, i.e., $P_{\text{src}}(s'|s, a) \neq P_{\text{tar}}(s'|s, a)$, while sharing the same reward function $r_{\text{src}}(s, a) = r_{\text{tar}}(s, a)$. In the offline setting, the agent cannot interact with the environment and must learn from static datasets [21]. Suppose we are given a source dataset $D_{\text{src}} = \{(s, a, s', r)_{\text{src}}\}$ and a smaller target dataset $D_{\text{tar}} = \{(s, a, s', r)_{\text{tar}}\}$. The goal is to learn a policy π that maximizes the expected return in the target domain, $\max_\pi \mathbb{E}_{\pi, P_{\text{tar}}}\left[\sum_{t=0}^{\infty} \gamma^t r_{\text{tar}}(s_t, a_t)\right]$, by leveraging the mixed dataset $D_{\text{mix}} = D_{\text{src}} \cup D_{\text{tar}}$. Typically we have $|D_{\text{tar}}| \ll |D_{\text{src}}|$.

4 Methodology

In this section, we analyze the target-domain performance gap between the optimal target policy and the policy learned from a filtered source dataset. This analysis motivates a clustering-based procedure to estimate local KL divergence for filtering source transitions, which we then integrate into policy optimization. Finally, we summarize the complete algorithm for off-dynamics offline RL. Due to space limitations, all proofs are deferred to Appendix B.

4.1 Theoretical Insights

The goal of data filtering in the off-dynamics setting is to create a filtered source dataset $\tilde{D}_{src}$ out of the source dataset D_{src} such that the dynamics discrepancy between $\tilde{D}_{src}$ and the target dataset D_{tar} is minimized. Following [31], we have the following proposition.

Proposition 4.1. For any (s, a), denote its representation as z, and suppose $s'_{\text{src}} \sim P_{\mathcal{M}_{\text{src}}}(\cdot|z)$, $s'_{\text{tar}} \sim P_{\mathcal{M}_{\text{tar}}}(\cdot|z)$. Then measuring the representation deviation between the source domain and the target domain is equivalent to measuring the dynamics mismatch between two domains, i.e., $D_{\text{KL}}\big(P(z|s'_{\text{src}})\|P(z|s'_{\text{tar}})\big) = D_{\text{KL}}\big(P(s'_{\text{src}}|z)\|P(s'_{\text{tar}}|z)\big) + \mathbb{H}(s'_{\text{src}}) - \mathbb{H}(s'_{\text{tar}})$.

The difference between Proposition 4.1 and Theorem 4.3 of [31] lies in the direction of the KL divergence. We derive the KL divergence from the source representation distribution to the target representation distribution (rather than the reverse). This choice matches our goal of characterizing how far the policy learned from filtered source data can be from the target-optimal policy (Theorem 4.3). Since KL divergence is asymmetric, we reproduce the proof in Appendix B for completeness.

Since the entropy terms are constants for fixed source/target domains, minimizing dynamics deviation is equivalent to minimizing representation deviation. Concretely, let $\mathbb{H}(s'_{\text{src}}) - \mathbb{H}(s'_{\text{tar}}) = B$ be a constant. If the representation deviation is bounded by ϵ, i.e., $D_{\text{KL}}(P(z|s'_{\text{src}})\|P(z|s'_{\text{tar}})) \leq \epsilon$, then the dynamics deviation $D_{\text{KL}}(P(s'_{\text{src}}|z)\|P(s'_{\text{tar}}|z))$ is bounded by $\epsilon - B$.

The following assumption from [8] quantifies the performance gap that the optimal target policy π^* may suffer when evaluated under source dynamics.

Assumption 4.2. Let $\pi^* = \arg\max_\pi \mathbb{E}_p\big[\sum \gamma^t r(s_t, a_t)\big]$ be the reward-maximizing policy in the target domain. Then the expected reward in source and target domains differs by at most $2R_{\max}\sqrt{1 - e^{B-\epsilon}}$:

$$\left|\mathbb{E}_{\pi^*, P_{\text{src}}}\left[\sum_t \gamma^t r(s_t, a_t)\right] - \mathbb{E}_{\pi^*, P_{\text{tar}}}\left[\sum_t \gamma^t r(s_t, a_t)\right]\right| \leq 2R_{\max}\sqrt{1 - e^{B-\epsilon}},$$

where $B = \mathbb{H}(s'_{\text{src}}) - \mathbb{H}(s'_{\text{tar}})$.

Now we present an upper bound for the performance gap in the target domain between the policy learned from the filtered source dataset and the optimal target policy.

Theorem 4.3 (Offline performance bound)**.** Let $\hat{\pi}^*_{\tilde{D}_{src}}$ be the policy that maximizes the reward in the **filtered** source dataset $\tilde{D}_{src}$, $\pi^*_{D_{tar}}$ be the policy that maximizes the reward in the **unfiltered** target dataset D_{tar}, and π^* be the policy that maximizes the reward in the target domain. Assume that π^* satisfies Assumption 4.2 and let C be the offline RL performance bound without dynamics shift, i.e., $\mathbb{E}_{P_{\text{tar}}, \pi^*}\big[\sum \gamma^t r(s_t, a_t)\big] - \mathbb{E}_{P_{\text{tar}}, \pi^*_{D_{tar}}}\big[\sum \gamma^t r(s_t, a_t)\big] \leq C$. Then $\hat{\pi}^*_{\tilde{D}_{src}}$ receives near-optimal reward on the target domain:

$$\mathbb{E}_{P_{\text{tar}}, \hat{\pi}^*_{\tilde{D}_{src}}}\left[\sum_t \gamma^t r(s_t, a_t)\right] \geq \mathbb{E}_{P_{\text{tar}}, \pi^*}\left[\sum_t \gamma^t r(s_t, a_t)\right] - \underbrace{C}_{\text{offline algorithm error}} - \frac{4R_{\max}}{1-\gamma}\underbrace{\sqrt{1 - e^{B-\epsilon}}}_{\text{dynamics deviation}}$$
$$- \frac{2R_{\max}}{1-\gamma}\underbrace{\mathbb{E}_{\substack{\hat{\pi}^*_{\tilde{D}_{src}} \\ \rho_{\mathcal{M}_{\text{src}}}, P_{\text{src}}}}\Big[D_{\text{TV}}\big(\hat{\pi}^*_{\tilde{D}_{src}}(\cdot|s')\|\pi^*_{D_{tar}}(\cdot|s')\big)\Big]}_{\text{filtering deviation}}.$$

Remark 4.4. Compared with the performance bounds from [42] and [33], our theorem offers two key advantages: (1) *Guarantee against the true optimal target policy:* The bound compares the learned policy's return in the target domain directly to the optimal target policy, ensuring the guarantee is stated against what is best achievable in the target environment. (2) *Avoids reliance on the source behavior policy:* Prior works often compare against the source behavior policy, implicitly favoring imitation of the source dataset. In contrast, our bound measures policy deviation against the policy trained on the target dataset, which better reflects the objective of filtering.

4.2 Clustering-Based KL Estimation

Theorem 4.3 motivates a clustering-based filtering strategy. In particular, the bound suggests that effective filtering should jointly reduce the *dynamics deviation* and the *filtering deviation*. Thus, to reduce the policy discrepancy between $\hat{\pi}^*_{\tilde{D}_{\text{src}}}$ and $\pi^*_{D_{\text{tar}}}$, the filtered source dataset $\tilde{D}_{\text{src}}$ should match the target dataset D_{tar} as measured by the KL divergence of the dynamics mismatch.

We can view dynamics mismatch equivalently as the discrepancy of representation based on Proposition 4.1. Concretely, we propose a clustering-based estimator for the representation divergence

$D_{\mathrm{KL}}\big(P(z|s'_{\mathrm{src}})\|P(z|s'_{\mathrm{tar}})\big)$. The idea is to cluster similar next states so that within each cluster, differences between source and target state-action pairs primarily reflect *localized* dynamics mismatch.

Given two datasets $D_{\mathrm{src}} = \{(s_i, a_i, s'_i)\}_{i=1}^{N_{\mathrm{src}}}$ and $D_{\mathrm{tar}} = \{(s_j, a_j, s'_j)\}_{j=1}^{N_{\mathrm{tar}}}$, we first run K-means on the next states s' in the target dataset. We then assign source transitions to the nearest cluster based on their next states, while discarding those that are far away from the average diameter of all target clusters. This yields K disjoint clusters $\{\mathcal{N}^n\}_{n=1}^{K}$, where each cluster $\mathcal{N}^n$ contains target transitions and retained source transitions with similar next states. We then pair each s' in $\mathcal{N}^n$ with its corresponding (s, a), forming $\mathcal{N}^n = \{(z_i, s'_i)\}_{i=1}^{N_0^n+N_1^n}$ with $z_i = s_i \oplus a_i$ as an identity representation of the state-action pairs, where N_0^n and N_1^n are the numbers of source and target samples in $\mathcal{N}^n$ respectively. We assign binary labels within each cluster: target samples z_i^{tar} are labeled as class 1 and source samples z_i^{src} as class 0. The goal is to train a cluster-specific classifier $D_n(z) = P(y = 1|z)$ using the following cross-entropy loss

$$L = -\mathbb{E}_{z\sim P(z|s'_{\mathrm{tar}})}[\log D_n(z)] - \mathbb{E}_{z\sim P(z|s'_{\mathrm{src}})}[\log(1 - D_n(z))]. \tag{4.1}$$

By Bayes' rule, the classifier output satisfies

$$D_n(z) = \frac{P(z|y = 1)P(y = 1)}{P(z|y = 1)P(y = 1) + P(z|y = 0)P(y = 0)}. \tag{4.2}$$

Within each cluster, the next states s' are similar by construction. We therefore assume that the representation z primarily reflects source domain and target domain label information, i.e., $P(z|y = 1) \approx P(z|s'_{\mathrm{tar}})$ and $P(z|y = 0) \approx P(z|s'_{\mathrm{src}})$. Define a density ratio as $w(z) = P(z|s'_{\mathrm{tar}})/P(z|s'_{\mathrm{src}})$. Plug this approximation into Equation (4.2) and we get

$$D_n(z) = \frac{Aw_n(z)}{Aw_n(z) + (1 - A)} \implies \hat{w}_n(z) = \frac{(1 - A)D_n(z)}{A(1 - D_n(z))},$$

where $A = P(y = 1) = N_1^n/(N_1^n + N_0^n)$ and $1 - A = P(y = 0) = N_0^n/(N_1^n + N_0^n)$. Since the KL divergence can be written as $D_{\mathrm{KL}}(P(z|s'_{\mathrm{src}})\|P(z|s'_{\mathrm{tar}})) = \mathbb{E}_{z\sim P(z|s'_{\mathrm{src}})}[\log 1/w(z)]$, we can estimate it using source samples:

$$\hat{D}_{\mathrm{KL}}^n = \frac{1}{N_0^n}\sum_{i=1}^{N_0^n} \log \frac{1}{\hat{w}_n(z_i^{\mathrm{src}})}, \tag{4.3}$$

where $\hat{D}_{\mathrm{KL}}^n$ is a *cluster* estimate of the KL divergence using cluster-specific classifier and $\log 1/\hat{w}_n(z_i^{\mathrm{src}}) =: d_i$ is a *point* estimate of KL divergence.

4.3 Policy Optimization

We now describe the policy optimization module of our approach. In offline RL, a key challenge is preventing out-of-distribution (OOD) actions at deployment time. Advantage-weighted methods [37] mitigate this issue by weighting behavior cloning with estimated advantages, encouraging the policy to imitate dataset actions with higher values. In off-dynamics RL, however, naively imitating the mixed dataset, especially source transitions collected under mismatched dynamics, can degrade target-domain performance. To address this, we (i) filter the source dataset using KL estimates and (ii) use the (normalized) pointwise KL estimates to adaptively reweight learning. Intuitively, source transitions with smaller estimated KL divergence are more consistent with the target dynamics and therefore should have higher influence during optimization. We minimize the following loss function:

$$\mathcal{L}_Q = \mathbb{E}_{D_{tar}}\big[\big(Q_\theta - \mathcal{T}Q_\theta\big)^2\big] + \mathbb{E}_{(s,a,s',d)\sim\tilde{D}_{src}}\big[\exp\big(-\alpha\cdot\hat{d}_i\big)\big(Q_\theta - \mathcal{T}Q_\theta\big)^2\big], \tag{4.4}$$

where α is a hyperparameter and $\hat{d}_i = \frac{d_i - \max_i d_i}{\max_i d_i - \min_i d_i}$ for $i \in \{1, \ldots, |\tilde{D}_{src}|\}$ is the normalized pointwise KL estimate. After selectively filtering source transitions, where we retain more source transitions in clusters with small KL estimates and less in clusters with large KL estimates, the retained source samples are expected to be more target-consistent and cover the whole state space. Since the target dataset is relatively small and can also be noisy due to its data quality, assigning slightly higher aggregate weight to the filtered source data can improve data efficiency by leveraging abundant transitions.

In addition to KL-based reweighting, we introduce a policy-regularization term that encourages the learned policy to stay close to the support of the target dataset. Following [33], we train a conditional

Table 1: Performance comparison on MuJoCo tasks under a medium-level offline dataset with dynamic shifts in gravity and friction at levels 0.1, 0.5, 2.0, 5.0. Source domains remain unchanged; target domains are shifted. We report normalized target-domain scores (mean ± std over five seeds). Best and second-best scores are highlighted in green and blue, respectively.

Env	Shift Level	BOSA	IQL	DARA	IGDF	OTDF	DmC	LoDADA
HalfCheetah Gravity	0.1	9.31±1.94	12.90±1.01	9.62±4.27	7.65±6.63	12.17±3.58	13.91±3.57	**14.09±3.83**
	0.5	39.92±5.63	43.68±2.96	40.87±5.46	39.73±4.62	37.12±4.88	42.24±2.86	**45.00±2.95**
	2.0	28.47±1.23	**31.12±0.3**	30.49±0.54	30.16±1.5	25.38±3.12	26.91±2.07	28.02±1.67
	5.0	11.27±1.77	20.87±21.45	32.42±18.54	30.08±34.02	24.01±17.88	40.21±29.28	**71.11±1.63**
HalfCheetah Friction	0.1	12.53±3.61	23.69±16.46	**26.39±11.35**	20.56±11.22	10.31±1.68	9.26±0.49	9.97±0.47
	0.5	63.65±9.14	68.26±1.42	68.41±3.19	66.30±1.79	66.32±0.46	**68.59±0.34**	68.51±0.44
	2.0	45.53±2.01	42.68±2.64	46.17±1.72	46.22±1.27	45.84±0.34	45.48±0.54	**48.19±0.75**
	5.0	38.26±10.82	48.97±4.68	45.31±9.83	**54.75±5.6**	38.36±9.82	29.52±4.12	54.65±1.56
Ant Gravity	0.1	**25.58±2.21**	11.03±1.24	12.53±1.11	13.55±2.13	18.82±5.51	18.52±3.06	22.38±1.28
	0.5	17.75±2.48	10.53±1.06	9.08±0.88	11.34±1.89	16.77±3.13	**25.26±2.60**	18.61±2.79
	2.0	37.17±0.96	38.26±3.09	38.54±2.19	41.59±4.63	37.11±0.74	42.87±2.97	**45.82±1.33**
	5.0	33.48±1.70	31.51±1.97	32.51±1.71	34.44±2.77	34.58±0.25	38.62±2.13	**48.29±3.68**
Ant Friction	0.1	**58.95±0.71**	55.12±0.24	55.56±0.46	54.88±0.60	53.38±0.42	51.37±0.94	53.76±1.37
	0.5	**59.72±3.57**	58.92±0.80	59.28±0.80	55.64±5.03	58.20±0.84	48.29±6.57	55.54±3.80
	2.0	20.18±3.79	17.54±2.47	19.84±3.20	19.63±4.15	32.33±3.45	59.13±5.39	**62.27±2.23**
	5.0	9.88±1.17	7.79±0.31	7.78±0.26	13.97±7.68	13.17±8.28	12.14±1.86	**28.62±3.43**
Walker2d Gravity	0.1	18.75±12.02	20.12±5.74	16.04±7.60	24.95±16.19	63.77±6.43	64.66±5.42	**71.41±1.99**
	0.5	40.09±2.37	29.72±16.02	42.05±10.52	37.36±5.49	32.47±7.04	35.45±5.89	**44.65±1.49**
	2.0	8.91±2.28	32.20±1.05	25.69±10.67	34.89±6.73	37.23±2.16	39.54±2.08	**40.27±1.77**
	5.0	5.25±0.50	5.44±0.08	5.42±0.29	5.66±0.32	4.92±0.08	5.11±0.11	**8.55±3.85**
Walker2d Friction	0.1	7.88±1.88	5.65±0.06	5.72±0.23	7.21±1.80	**30.40±7.54**	18.24±6.50	20.21±7.90
	0.5	53.42±20.06	67.91±7.03	68.92±11.43	70.83±4.22	69.06±9.44	**83.32±3.03**	81.41±6.47
	2.0	39.06±17.36	**72.91±0.37**	65.40±7.13	62.55±10.36	59.93±14.53	62.15±6.13	62.01±4.36
	5.0	5.89±3.42	5.38±0.09	5.28±0.19	5.21±0.36	14.98±9.15	14.75±2.89	**15.93±3.09**
Hopper Gravity	0.1	27.82±13.41	23.4±11.62	13.10±0.98	13.89±4.70	33.41±1.92	35.73±0.93	**37.68±6.32**
	0.5	17.74±7.93	17.78±6.85	14.68±5.24	19.81±9.56	35.22±5.94	15.28±2.43	**43.72±9.72**
	2.0	13.18±2.10	14.71±1.67	14.94±1.08	14.71±0.87	**19.82±1.77**	18.30±1.55	16.99±3.74
	5.0	7.97±0.61	7.78±0.37	7.81±0.48	8.03±0.50	8.78±0.03	8.77±0.02	**9.28±0.26**
Hopper Friction	0.1	25.55±2.69	26.13±4.24	24.16±4.50	24.02±5.00	24.76±1.21	25.57±3.22	**28.06±7.97**
	0.5	25.86±2.17	24.75±3.58	24.84±4.63	29.12±3.01	29.89±6.17	27.42±2.15	**33.57±1.31**
	2.0	10.32±0.06	10.15±0.03	10.15±0.06	10.35±0.60	**10.52±0.29**	10.31±0.12	10.44±0.13
	5.0	6.92±1.97	7.92±0.06	7.84±0.04	7.90±0.17	8.07±0.04	**8.08±0.08**	8.04±0.05
Total		826.26	894.82	886.84	916.98	1007.10	1045.00	**1207.05**

variational autoencoder (CVAE), denoted by $\hat{\pi}_b^{\text{tar}}(a|s)$, to model the target behavior policy. The policy is then optimized according to

$$\mathcal{L}_\pi^{\text{reg}} = \mathcal{L}_\pi - \lambda \mathbb{E}_{s\sim\tilde{D}_{\text{src}}\cup D_{\text{tar}}}\left[\log \hat{\pi}_b^{\text{tar}}\big(\pi(\cdot|s)|s\big)\right], \tag{4.5}$$

where $\mathcal{L}_\pi$ is the policy loss of the underlying offline RL algorithm, and λ controls the strength of policy regularization. We use IQL [19] as the backbone optimization algorithm.

Formally, we layout the general framework and detailed implementation of LoDADA in Figure 1 and Algorithm 1 in Appendix D.6 respectively.

5 Experiment

In this section, we conduct extensive experiments to evaluate the performance of LoDADA on the ODRL benchmark for off-dynamics reinforcement learning [32]. We begin with MuJoCo locomotion tasks (HalfCheetah, Ant, Walker2d, and Hopper) under multiple types and levels of dynamics shifts. In addition, we evaluate LoDADA on navigation tasks in AntMaze and manipulation tasks in Adroit to demonstrate effectiveness beyond standard locomotion benchmarks. Finally, we conduct ablation studies to better understand the contributions of individual components, and the sensitivity to key hyperparameters. We also conduct experiments to compare the runtime complexity of LoDADA with that of other data filtering strategies to demonstrate the computational efficiency of our algorithm. We further conduct experiments to analyze LoDADA's sensitivity to target data coverage and quality to understand the potential limitations of LoDADA .

5.1 Experimental Setup

Task and Environments. We evaluate LoDADA on the ODRL benchmark [32] across MuJoCo environments with different dynamics shifts. Each environment includes three types of dynamics shifts: gravity, friction and morphology shifts. The source dataset is collected in the original MuJoCo environment from D4RL [10], and the target dataset is collected by the behavior policy in the

corresponding environment with dynamics shifts. All datasets are collected using SAC [14] trained to a medium performance level in either the original or dynamics-shifted environment. We use the datasets provided by ODRL, where the source dataset contains 1 million transitions while the target one has 5000 transitions. To construct dynamics-shifted target environments, we follow ODRL and modify MuJoCo simulator parameters for gravity, friction and morphology. For gravity, we rescale the global gravity magnitude while keeping its direction unchanged. For friction, we rescale the friction parameters. In both cases, we consider four shift levels $\{0.1, 0.5, 2.0, 5.0\}$, where the original gravity or friction is multiplied by the corresponding factor. Morphology shifts are created by changing the size of specific limbs or the torso of the robot. In the main experiments, both source and target datasets are collected by medium-level behavior policies following ODRL.

In addition to the locomotion tasks, we evaluate on AntMaze problems with different map layouts following ODRL. We focus on the medium-sized AntMaze setting and use 6 different map layouts as target environments, denoted as $\{1, 2, 3, 4, 5, 6\}$. We use the *play* and *diverse* datasets as source datasets, both drawn from D4RL [10], while the target dataset is collected by an agent trained directly in the target environment and provided by ODRL. The diverse dataset consists of trajectories that start from random locations and reach random goals, whereas the play dataset contains trajectories that start from hand-picked locations and reach a fixed hand-picked goal. Consequently, these two source datasets exhibit different state coverage. Finally, we evaluate LoDADA on manipulation tasks from the Adroit benchmark. We consider the *pen* and *door* environments. Similar to the MuJoCo experiments, we construct target environments by introducing kinematic and morphological shifts. For kinematic shifts, we restrict the rotation ranges of hand joints, referred to as *broken-joint*. For morphological shifts, we shrink the sizes of fingers, referred to as *shrink-finger*. We consider shift levels of medium and hard in our experiments. The source dataset is drawn from D4RL at the human level, while the target dataset is obtained from ODRL. Additional details can be found in Appendix D.

Baselines. We compare LoDADA against several strong baseline methods including DARA [22], BOSA [23], IQL [19], IGDF [42], OTDF [33], and DmC [40]. We first consider methods that learn from whole combined source and target datasets. DARA applies reward augmentation learned from the discrepancy between source and target dynamics to the source dataset to mimic the trajectories in the target environment. IQL is an offline RL algorithm that learns a value function and an advantage-weighted policy to achieve higher performance, which demonstrates strong performance in offline RL tasks trained on mixed datasets without explicit domain adaptation. We also compare with the data filtering methods. BOSA enforces support constraints with two objectives to mitigate OOD state–action pairs and OOD dynamics while filtering unreliable transitions. IGDF learns a contrastive representation to estimate a mutual information-based domain gap for filtering source transitions. OTDF is the most recent state-of-the-art method which uses optimal transport to match source–target transitions and performs non-parametric filtering based on the optimal transport cost. DmC uses nearest-neighbor estimation to measure domain discrepancies and guides a diffusion model to generate additional target aligned transitions.

Implementation Details. For LoDADA , we fix the filtering ratios at $(90\%, 80\%, 70\%)$ for low-, medium-, and high-KL clusters across all tasks. We select the number of clusters K from $\{30, 50\}$ and the policy regularization weight λ from $\{0.1, 0.5, 1.0\}$ for MuJoCo and Adroit tasks. For AntMaze, we select K from $\{5, 10, 15, 20\}$ and λ from $\{0.5, 1.0, 5.0\}$. Additional implementation details and hyperparameters are provided in Table 6 in Appendix D.

5.2 Results on MuJoCo Locomotion Tasks

We first summarize the performance of LoDADA and the baseline methods on MuJoCo locomotion tasks under various dynamics shifts, including gravity, friction, morphology shifts, and local perturbations. We evaluate performance using the normalized score $\textit{Normalized Score} = \frac{\textit{score} - \textit{random score}}{\textit{expert score} - \textit{random score}} \times 100$, where the *random score* is achieved by a random policy and the *expert score* is achieved by SAC [14] trained to the expert level in the target domain.

Results on gravity and friction shifts. As shown in Table 1, LoDADA achieves a total score of 1207.05, yielding a 15.5% improvement over the second-best method DmC [40], a 19.9% improvement over OTDF [33], and a 34.9% improvement over the backbone algorithm IQL [19]. Across the 32 evaluation tasks, LoDADA achieves best (18) or second-best (7) performance in 25 out of 32 tasks and remains competitive on the remaining 7 tasks. **These results indicate that LoDADA improves target-domain performance by effectively leveraging localized filtering.**

Table 2: Performance on localized perturbation tasks for Hopper. We report normalized target-domain scores over five seeds.

Env	# Groups	Shift Level	BOSA	DARA	IQL	IGDF	OTDF	DmC	LoDADA
Hopper Localized Perturbation	3	0.1-0.5-2.0	7.84±2.84	8.70±6.67	8.14±6.31	13.08±10.44	33.96±4.57	30.61±10.01	**40.77±5.04**
		0.2-1.0-4.0	11.91±2.60	9.61±5.56	11.97±5.99	8.35±3.56	32.58±6.89	29.53±5.73	**37.51±9.16**
		0.3-1.5-6.0	11.85±3.19	8.11±3.54	10.17±5.50	6.59±2.59	31.77±3.04	27.87±5.27	**33.41±2.72**
	5	0.1-0.2-0.3-0.4-0.5	10.42±3.55	6.37±1.67	6.92±1.85	15.81±9.78	22.25±8.74	**40.35±8.19**	40.30±4.61
		0.2-0.4-0.6-0.8-1.0	11.63±1.87	14.07±5.65	9.82±4.29	11.24±7.86	21.42±9.11	34.36±4.82	**36.06±4.59**
		0.3-0.6-0.9-1.2-1.5	13.97±5.15	11.14±4.01	12.01±2.06	15.63±10.74	29.95±5.01	18.79±5.32	**35.02±5.70**

The policy optimization module in LoDADA builds on IQL. Compared to vanilla IQL, we introduce (i) adaptive weighting in critic learning and (ii) an additional CVAE-based behavior-cloning regularizer. As shown in Table 1, these modifications yield significant gains over IQL on 27 out of 32 tasks. **These observations suggest that combining localized filtering with target-aware regularization improves adaptation to the target domain.**

Additional MuJoCo evaluations. We further evaluate LoDADA on MuJoCo morphology shifts and the setting with medium-level source data and expert-level target data. Detailed results are reported in Table 7 in Appendix E.1 and in Table 12 in Appendix E.5. For morphology shifts, LoDADA achieves the best score in 13 out of 16 tasks and improves over the second-best method by 32.7%. When the target dataset is collected by an expert policy, LoDADA also achieves the highest aggregate score. These results show that our method remains effective beyond the standard medium-level setting.

5.3 Experiments on Localized Perturbations

In this experiment, we evaluate LoDADA under *locally perturbed* dynamics. Unlike the previous setting with *global* shifts (e.g., gravity changes that affect transitions uniformly), now the dynamics vary across regions of the state space. This setting is more challenging because the distribution mismatch is heterogeneous and cannot be well addressed by methods that rely on global assumptions, such as DARA [22] and IGDF [42].

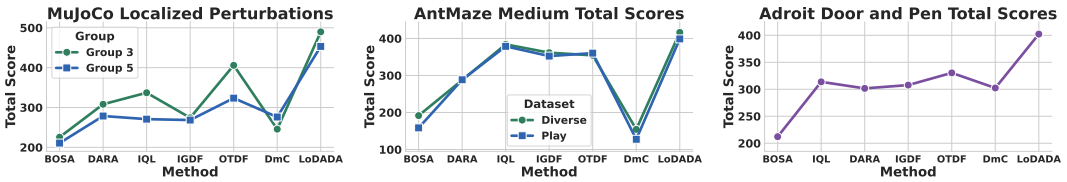
We keep the target domain unchanged and perturb only the source dataset to simulate localized variations. Policies are trained on the perturbed source dataset and evaluated in the unchanged target domain. For the target dataset, we randomly sample 5,000 transitions from D4RL [10] without introducing any dynamics shift. For the source dataset, we start from the original D4RL dataset and partition the next-state space into 15 regions using K-means. We then divide the regions into M groups and inject Gaussian noise into state variables, with all samples in the same group sharing the same noise variance. We evaluate performance using the normalized score defined above. We consider all four MuJoCo environments (Ant, Hopper, HalfCheetah, Walker2d) and set $M \in \{3, 5\}$. When $M = 3$, the noise variances are $\{0.1, 0.5, 2.0\}$, $\{0.2, 1.0, 4.0\}$, and $\{0.3, 1.5, 6.0\}$. When $M = 5$, the variances are $\{0.1, 0.2, 0.3, 0.4, 0.5\}$, $\{0.2, 0.4, 0.6, 0.8, 1.0\}$, and $\{0.3, 0.6, 0.9, 1.2, 1.5\}$.

We summarize the aggregate results in Figure 2a and partial results in Table 2. More detailed results are deferred to Table 8 in Appendix E.2 due to space limit. As shown in Figure 2a, LoDADA achieves the best total score under both group settings. Across all localized perturbation tasks, LoDADA obtains a total score of 943.09, improving over the second-best method OTDF by 29.3% and over IQL by 55.1%. These results demonstrate that LoDADA is effective under localized perturbations, where cluster-aware filtering becomes more beneficial than under global shifts.

5.4 Results on AntMaze and Adroit

In this section, we present additional results on navigation and manipulation tasks described in Section 5.1. We summarize the aggregate performance on AntMaze and Adroit in Figure 2, and defer the detailed numerical results to the appendix.

AntMaze. As shown in Figure 2b, LoDADA achieves the best aggregate performance on AntMaze-medium tasks under both source datasets. This shows that localized filtering can retain source transitions that better match the target map layouts. Compared with prior methods, LoDADA better preserves useful transitions in structured navigation environments, where local state-space similarity is important. These results show that LoDADA generalizes beyond locomotion tasks to navigation problems. Detailed results are reported in Table 9 and Table 10 in Appendix E.3.



(a) MuJoCo localized perturbation. (b) AntMaze-medium. (c) Adroit Door and Pen.

Figure 2: Performance on MuJoCo localized perturbation, AntMaze, and Adroit tasks. Left: total scores on MuJoCo localized perturbation tasks with 3 and 5 perturbed region groups. Middle: total scores on AntMaze-medium tasks using diverse and play source datasets. Right: total scores on Adroit Door and Pen tasks under morphology and kinematic shifts.

Adroit. As shown in Figure 2c, LoDADA achieves the highest aggregate score on Adroit manipulation benchmarks. This shows that our method remains effective under morphology and kinematic shifts in high-dimensional manipulation tasks. In these tasks, global source-target matching can be unreliable because different hand configurations and object interactions may create non-uniform dynamics mismatch, while localized dynamics-aware filtering keeps source transitions that are locally closer to the target dynamics. Detailed Door and Pen results are reported in Table 11 in Appendix E.4.

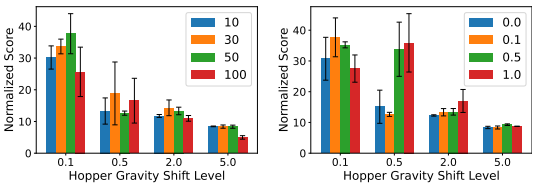
5.5 Ablation Study

We conduct eight ablation studies to provide deeper insights into LoDADA and the effect of key hyperparameters in locomotion tasks. For demonstration of hyperparameter sensitivity, we use the Hopper environment in gravity shift with four shift levels (0.1, 0.5, 2.0, 5.0), focusing on the number of clusters K and the policy-regularization weight λ. Due to the space limit, we defer detailed results to Appendix E.

Ablation 1: Number of clusters K. K controls the level of locality of the state space partition used in our cluster-aware filtering mechanism. A smaller K induces coarser clusters, which may merge heterogeneous transition dynamics, while a large K can lead to overly fine partitions with insufficient samples per cluster. To study this trade-off, we sweep $K \in \{10, 30, 50, 100\}$ and run LoDADA in the hopper-gravity environment with medium-level source and target datasets. As shown in Figure 3a, performance degrades when K is either too small (*e.g.*, 10) or too large (*e.g.*, 100), while intermediate values achieve consistently better results. **These results indicate that a moderate level of locality is critical for effective cluster-aware filtering, balancing dynamics homogeneity and data sufficiency.** As a result, we use $K = 30$ or 50 for MuJoCo and Adroit tasks. Adroit results are summarized in Table 19 in Appendix E.10. For AntMaze, we use smaller cluster numbers due to the structured map, as discussed in Table 18 in Appendix E.10.

Ablation 2: Policy regularization weight λ. λ controls how strongly the learned policy is regularized toward the target behavior policy. We sweep $\lambda \in \{0, 0.1, 0.5, 1.0\}$ in the Hopper-Gravity setting. As shown in Figure 3b, removing this regularization ($\lambda = 0$) leads to poor performance, while the best value can vary across shift levels. Overall, $\lambda = 0.5$ provides a strong default across MuJoCo settings.

Ablation 3: Cluster-level sample coverage. Since cluster-level KL estimation depends on target samples in each cluster, we report representative sample counts and classifier convergence diagnostics in Table 13 in Appendix E.6. The examined clusters contain samples from both domains, and the local classifiers converge early, suggesting that the KL estimates are stable in these settings.



(a) Ablation study on K. (b) Ablation study on λ.

Figure 3: Parameter sensitivity study.

Ablation 4: Filtering ratios. We fix the filtering ratios as $(\xi_1, \xi_2, \xi_3) = (90\%, 80\%, 70\%)$ across all environments. We evaluate more aggressive filtering ratios on both MuJoCo and AntMaze tasks. As shown in Table 14 and Table 15, the default setting achieves the highest performance. We visualize the filtering results on AntMaze in Figure 4. The visualization illustrates that LoDADA filters source data that are less aligned with the target data, while more aggressive filtering ratios remove more data and reduce state space coverage.

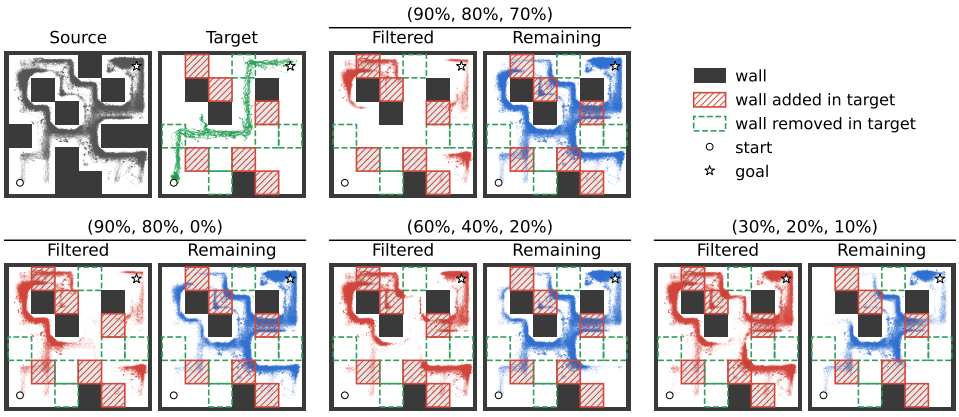


Figure 4: Visualization of source data filtering on AntMaze Play Map 2 under different filtering ratios. Red and green regions indicate walls added and removed in the target domain. Red and blue points represent filtered and retained source transitions. The default setting in our navigation experiments uses $(\xi_1, \xi_2, \xi_3) = (90\%, 80\%, 70\%)$.

Ablation 5: Component contributions. Table 16 in Appendix E.8 studies the effect of removing KL-based critic weighting, removing CVAE policy regularization, and replacing local discriminator based filtering with L2-distance filtering. The full method achieves the best aggregate performance, showing that these components are complementary. Table 17 in Appendix E.9 further shows that adding the same CVAE regularizer to baseline methods does not fully explain the gain of LoDADA .

Ablation 6: Runtime comparison. We report the wall-clock runtime of different filtering methods in Table 20 in Appendix E.11. LoDADA has a higher pre-optimization cost than IGDF because it trains local classifiers, but it is much faster than OTDF, which requires expensive optimal-transport computation. Together with the performance results, this shows that LoDADA provides a practical trade-off between computational cost and final policy performance.

Ablation 7: Target dataset size. We investigate the effectiveness of LoDADA with different target data availability. Table 21 in Appendix E.12 evaluates LoDADA with progressively reduced target dataset sizes in locomotion tasks. The results show that LoDADA remains relatively stable when up to 50% of target data are removed, with more noticeable degradation under extreme sparsity settings when only 10%-20% of target data remain.

Ablation 8: Target dataset quality. We further investigate the performance of LoDADA under different target data quality. Table 22 in Appendix E.12 compares LoDADA using medium, random-medium, and random target datasets. Results suggest that LoDADA's performance generally degrades as target data quality decreases due to the limited target-policy coverage provided by low-quality data, highlighting the importance of reliable target supervision for effective adaptation.

6 Conclusion and Limitations

In this work, we studied off-dynamics offline RL through a localized dynamics-aware data filtering method. We provided a theoretical performance bound that motivates us to perform data filtering from a local perspective and introduce a dataset regularization term to encourage the learned policy to align with the target behavior. Building on this insight, we proposed LoDADA and conducted extensive experiments under various types and levels of both global and local dynamics shifts. Experiment results demonstrate that LoDADA outperforms existing baselines across a wide range of tasks, with particularly strong gains under locally perturbed dynamics. Our method has some limitations. It relies on cluster-level KL estimation, so performance may naturally degrade when target coverage is extremely sparse, target data quality is poor, or source-target overlap is very small. Future work includes exploring the effectiveness of LoDADA in real-world environments and exploring more principled clustering mechanisms to better capture fine-grained state-action structures. In addition, our proposed LoDADA is orthogonal to the data generation methods and can be combined with them to unleash the power of synthetic methods in solving dynamics shift problems.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive comments and valuable feedback. YY and PX are supported in part by the National Science Foundation under Grant Nos. DMS-2323112 and IIS-2541960 (CAREER), and by the Whitehead Scholars Program at the Duke University School of Medicine. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any funding agency.

References

[1] David Abel, Yuu Jinnai, Sophie Yue Guo, George Konidaris, and Michael Littman. Policy and value transfer in lifelong reinforcement learning. In *International conference on machine learning*, pages 20–29. PMLR, 2018. 3

[2] Paul Barde, Julien Roy, Wonseok Jeon, Joelle Pineau, Chris Pal, and Derek Nowrouzezahrai. Adversarial soft advantage fitting: Imitation learning without policy optimization. *Advances in Neural Information Processing Systems*, 33:12334–12344, 2020. 3

[3] Clément L Canonne. A short note on an inequality between kl and tv. *arXiv preprint arXiv:2202.07198*, 2022. 16

[4] Pierre Clavier, Laixi Shi, Erwan Le Pennec, Eric Mazumdar, Adam Wierman, and Matthieu Geist. Near-optimal distributionally robust reinforcement learning with general l_p norms. In *Advances in Neural Information Processing Systems*, volume 37, pages 1750–1810, 2024. 3

[5] Antonio Coronato, Muddasar Naeem, Giuseppe De Pietro, and Giovanni Paragliola. Reinforcement learning for intelligent healthcare applications: A survey. *Artificial intelligence in medicine*, 109:101964, 2020. 2

[6] Pavel Dvurechensky, Alexander Gasnikov, and Alexey Kroshnin. Computational optimal transport: Complexity by accelerated gradient descent is better than by sinkhorn's algorithm. In *International conference on machine learning*, pages 1367–1376. PMLR, 2018. 2

[7] Laurent El Ghaoui and Arnab Nilim. Robust solutions to markov decision problems with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005. 3

[8] Benjamin Eysenbach, Shreyas Chaudhari, Swapnil Asawa, Sergey Levine, and Ruslan Salakhutdinov. Off-dynamics reinforcement learning: Training for transfer with domain classifiers. In *International Conference on Learning Representations*, 2021. 2, 3, 4, 16, 20

[9] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2019. 14

[10] 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. 6, 7, 8

[11] Jingwen Gu, Yiting He, Zhishuai Liu, and Pan Xu. Policy regularized distributionally robust markov decision processes with linear function approximation. *arXiv preprint arXiv:2510.14246*, 2025. 3

[12] Yihong Guo, Yixuan Wang, Yuanyuan Shi, Pan Xu, and Anqi Liu. Off-dynamics reinforcement learning via domain adaptation and reward augmented imitation. In *Advances in Neural Information Processing Systems*, volume 37, pages 136326–136360, 2024. 2, 3

[13] Yihong Guo, Yu Yang, Pan Xu, and Anqi Liu. MOBODY: Model-based off-dynamics offline reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. 2, 3

[14] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018. 7

[15] Yiting He, Zhishuai Liu, Weixin Wang, and Pan Xu. Sample complexity of distributionally robust off-dynamics reinforcement learning with online interaction. In *Forty-second International Conference on Machine Learning*, 2025. 2, 3

[16] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996. 1

[17] 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. 2

[18] Lingkai Kong, Haichuan Wang, Tonghan Wang, GUOJUN XIONG, and Milind Tambe. Composite flow matching for reinforcement learning with shifted-dynamics data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 2

[19] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021. 2, 6, 7, 19

[20] Jongmin Lee, Wonseok Jeon, Byungjun Lee, Joelle Pineau, and Kee-Eung Kim. Optidice: Offline policy optimization via stationary distribution correction estimation. In *International Conference on Machine Learning*, pages 6120–6130. PMLR, 2021. 3

[21] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020. 3

[22] Jinxin Liu, Hongyin Zhang, and Donglin Wang. Dara: Dynamics-aware reward augmentation in offline reinforcement learning. *arXiv preprint arXiv:2203.06662*, 2022. 2, 3, 7, 8, 20

[23] 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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 13945–13953, 2024. 7, 19

[24] Yao Liu, Adith Swaminathan, Alekh Agarwal, and Emma Brunskill. Provably good batch off-policy reinforcement learning without great exploration. *Advances in neural information processing systems*, 33:1264–1274, 2020. 3

[25] Zhishuai Liu, Weixin Wang, and Pan Xu. Upper and lower bounds for distributionally robust off-dynamics reinforcement learning. *arXiv preprint arXiv:2409.20521*, 2024. 3

[26] Zhishuai Liu and Pan Xu. Distributionally robust off-dynamics reinforcement learning: Provable efficiency with linear function approximation. In *International Conference on Artificial Intelligence and Statistics*, pages 2719–2727. PMLR, 2024. 3

[27] Zhishuai Liu and Pan Xu. Minimax optimal and computationally efficient algorithms for distributionally robust offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2024. 3

[28] Zhishuai Liu and Pan Xu. Linear mixture distributionally robust markov decision processes. In *Advances in Neural Information Processing Systems*, 2025. 3

[29] Miao Lu, Han Zhong, Tong Zhang, and Jose Blanchet. Distributionally robust reinforcement learning with interactive data collection: Fundamental hardness and near-optimal algorithms. In *Advances in Neural Information Processing Systems*, 2024. 3

[30] Yuping Luo, Huazhe Xu, Yuanzhi Li, Yuandong Tian, Trevor Darrell, and Tengyu Ma. Algorithmic framework for model-based deep reinforcement learning with theoretical guarantees. In *International Conference on Learning Representations*, 2019. 17

[31] Jiafei Lyu, Chenjia Bai, Jing-Wen Yang, Zongqing Lu, and Xiu Li. Cross-domain policy adaptation by capturing representation mismatch. In *International Conference on Machine Learning*, pages 33638–33663. PMLR, 2024. 2, 3, 4, 14, 17

[32] Jiafei Lyu, Kang Xu, Jiacheng Xu, Jing-Wen Yang, Zongzhang Zhang, Chenjia Bai, Zongqing Lu, Xiu Li, et al. Odrl: A benchmark for off-dynamics reinforcement learning. *Advances in Neural Information Processing Systems*, 37:59859–59911, 2024. 6, 17, 18

[33] Jiafei Lyu, Mengbei Yan, Zhongjian Qiao, Runze Liu, Xiaoteng Ma, Deheng Ye, Jing-Wen Yang, Zongqing Lu, and Xiu Li. Cross-domain offline policy adaptation with optimal transport and dataset constraint. In *The Thirteenth International Conference on Learning Representations*, 2025. 2, 3, 4, 5, 7, 21, 28

[34] Marlos C Machado, Marc G Bellemare, and Michael Bowling. A laplacian framework for option discovery in reinforcement learning. In *International Conference on Machine Learning*, pages 2295–2304. PMLR, 2017. 14

[35] Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. *Advances in neural information processing systems*, 32, 2019. 3

[36] Arnab Nilim and Laurent El Ghaoui. Robustness in markov decision problems with uncertain transition matrices. In *Advances in Neural Information Processing Systems*, volume 16, 2003. 3

[37] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019. 5

[38] Zhongjian Qiao, Rui Yang, Jiafei Lyu, Xiu Li, Zhongxiang Dai, Zhuoran Yang, Siyang Gao, and Shuang Qiu. Dual-robust cross-domain offline reinforcement learning against dynamics shifts. In *The Fourteenth International Conference on Learning Representations*, 2026. 3

[39] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998. 1

[40] Linh Le Pham Van, Minh Hoang Nguyen, Duc Kieu, Hung Le, Hung The Tran, and Sunil Gupta. DmC: Nearest neighbor guidance diffusion model for offline cross-domain reinforcement learning. In *European Conference on Artificial Intelligence*, volume 413 of *Frontiers in Artificial Intelligence and Applications*, pages 2331–2338. IOS Press, 2025. 7, 23

[41] Ruhan Wang, Yu Yang, Zhishuai Liu, Dongruo Zhou, and Pan Xu. Return augmented decision transformer for off-dynamics reinforcement learning. *Transactions on Machine Learning Research*, 2026. 2

[42] 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. 2, 3, 4, 7, 8, 21, 28

[43] Haoran Xu, Xianyuan Zhan, Jianxiong Li, and Honglei Yin. Offline reinforcement learning with soft behavior regularization. *arXiv preprint arXiv:2110.07395*, 2021. 3

[44] 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, 2023. 2, 3

[45] Yu Yang, Yihong Guo, Anqi Liu, and Pan Xu. Cross-domain energy-guided diffusion generation for off-dynamics reinforcement learning. *arXiv preprint arXiv:2605.24810*, 2026. 2

[46] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020. 3

A Additional Related Works

Clustering-based methods in RL. Clustering-based methods in RL exploit structure in the state-action space by grouping similar samples into clusters, thereby reducing variance and improving generalization. Recent approaches use clustering for representation learning, such as learning latent embeddings that capture environment dynamics and then clustering them to discover reusable behaviors or intermediate objectives [34, 9]. These approaches often rely on heuristic similarity metrics, which may not accurately reflect the underlying transition dynamics. Our method leverages clustering not only to structure the data, but also integrates distributional alignment within clusters, ensuring that the grouped samples are both coherent and dynamically relevant to the target domain.

B Proof of the Main Theoretical Results

For simplicity, when we say a policy π is trained on the filtered source domain, we mean it satisfies $\mathbb{E}_{a\sim\pi(a|s),s'\sim p(s'|s,a)}\left[\sum_t D_{\mathrm{KL}}\left(P_{\mathrm{src}}\left(s'_{t+1}\mid z_t\right)\|P_{\mathrm{tar}}\left(s'_{t+1}\mid z_t\right)\right)\right]\leq\epsilon-B$, where $B=\mathbb{H}\left(s'_{\mathrm{src}}\right)-\mathbb{H}\left(s'_{\mathrm{tar}}\right), z_t=s_t\oplus a_t$.

B.1 Proof of Proposition 4.1

Proof of Proposition 4.1. This proof is similar to that of Theorem 4.3 in [31]. However, we derive this result by calculating the KL divergence between the source representation distribution and the target representation distribution, rather than vice versa. This is because our proposed method LoDADA requires characterizing the distance between the policy learned from the filtered source data and the target optimal policy, as demonstrated in Theorem 4.3. Due to the asymmetry of KL divergence, we reproduce the proof here for the completeness of our work.

We would like to establish a connection between the representation deviations in the two domains and the dynamics discrepancies between the two domains. Since the source and target MDPs share the same state space, s'_{src} and s'_{tar} take values in the same measurable space, although they are sampled from different transition dynamics. In the proof, $P(z,s'_{\mathrm{src}},s'_{\mathrm{tar}})$ denotes an auxiliary joint coupling whose marginals satisfy

$$\int_{\mathcal{S}}P(z,s'_{\mathrm{src}},s'_{\mathrm{tar}})ds'_{\mathrm{tar}}=P(z,s'_{\mathrm{src}}),\qquad\int_{\mathcal{S}}P(z,s'_{\mathrm{src}},s'_{\mathrm{tar}})ds'_{\mathrm{src}}=P(z,s'_{\mathrm{tar}}).$$

This construction does not assume that the source and target transition dynamics are the same; it only allows the two next-state variables to be written on a common product space, so that the additional integral over the unused variable marginalizes out. To achieve this, we introduce the concept of mutual information:

$$\begin{aligned}
&h\left(z;s'_{\mathrm{src}},s'_{\mathrm{tar}}\right)\\
&=I\left(z;s'_{\mathrm{src}}\right)-I\left(z;s'_{\mathrm{tar}}\right)\\
&=\int_{\mathcal{Z}}\int_{\mathcal{S}}P\left(z,s'_{\mathrm{src}}\right)\log\frac{P\left(z,s'_{\mathrm{src}}\right)}{P(z)P\left(s'_{\mathrm{src}}\right)}dzds'_{\mathrm{src}}-\int_{\mathcal{Z}}\int_{\mathcal{S}}P\left(z,s'_{\mathrm{tar}}\right)\log\frac{P\left(z,s'_{\mathrm{tar}}\right)}{P(z)P\left(s'_{\mathrm{tar}}\right)}dzds'_{\mathrm{tar}}\\
&=\int_{\mathcal{Z}}\int_{\mathcal{S}}P\left(z,s'_{\mathrm{src}}\right)\log\frac{P\left(z|s'_{\mathrm{src}}\right)}{P(z)}dzds'_{\mathrm{src}}-\int_{\mathcal{Z}}\int_{\mathcal{S}}P\left(z,s'_{\mathrm{tar}}\right)\log\frac{P\left(z|s'_{\mathrm{tar}}\right)}{P(z)}dzds'_{\mathrm{tar}}\\
&=\int_{\mathcal{Z}}\int_{\mathcal{S}}\int_{\mathcal{S}}P(z,s'_{\mathrm{src}},s'_{\mathrm{tar}})\log\frac{P(z|s'_{\mathrm{src}})}{P(z)}\,dz\,ds'_{\mathrm{src}}\,ds'_{\mathrm{tar}}\\
&\quad-\int_{\mathcal{Z}}\int_{\mathcal{S}}\int_{\mathcal{S}}P(z,s'_{\mathrm{tar}},s'_{\mathrm{src}})\log\frac{P(z|s'_{\mathrm{tar}})}{P(z)}\,dz\,ds'_{\mathrm{tar}}\,ds'_{\mathrm{src}}\\
&=\int_{\mathcal{Z}}\int_{\mathcal{S}}\int_{\mathcal{S}}P\left(z,s'_{\mathrm{src}},s'_{\mathrm{tar}}\right)\log\frac{P\left(z|s'_{\mathrm{src}}\right)}{P\left(z|s'_{\mathrm{tar}}\right)}dzds'_{\mathrm{src}}ds'_{\mathrm{tar}}\\
&=D_{\mathrm{KL}}\left(P\left(z|s'_{\mathrm{src}}\right)\|P\left(z|s'_{\mathrm{tar}}\right)\right).
\end{aligned}$$

Here the last line should be understood as an averaged conditional KL divergence:

$$D_{\mathrm{KL}}\left(P(z|s'_{\mathrm{src}})\|P(z|s'_{\mathrm{tar}})\right):=\mathbb{E}_{s'_{\mathrm{src}},s'_{\mathrm{tar}}}\left[D_{\mathrm{KL}}\left(P(z|s'_{\mathrm{src}})\|P(z|s'_{\mathrm{tar}})\right)\right].$$

For each fixed pair $(s'_{\text{src}}, s'_{\text{tar}})$, the inner KL is taken with respect to the numerator distribution $P(z|s'_{\text{src}})$, while the outer expectation averages this conditional KL under the auxiliary coupling defined above.At the same time, we also have

$$
\begin{aligned}
&h\left(z; s'_{\text{src}}, s'_{\text{tar}}\right) \\
&= I\left(z; s'_{\text{src}}\right) - I\left(z; s'_{\text{tar}}\right) \\
&= \int_{\mathcal{Z}} \int_{\mathcal{S}} P\left(z, s'_{\text{src}}\right) \log \frac{P\left(z, s'_{\text{src}}\right)}{P(z) P\left(s'_{\text{src}}\right)} \, dz \, ds'_{\text{src}} - \int_{\mathcal{Z}} \int_{\mathcal{S}} P\left(z, s'_{\text{tar}}\right) \log \frac{P\left(z, s'_{\text{tar}}\right)}{P(z) P\left(s'_{\text{tar}}\right)} \, dz \, ds'_{\text{tar}} \\
&= \int_{\mathcal{Z}} \int_{\mathcal{S}} P\left(z, s'_{\text{src}}\right) \log \frac{P\left(s'_{\text{src}}|z\right)}{P\left(s'_{\text{src}}\right)} \, dz \, ds'_{\text{src}} - \int_{\mathcal{Z}} \int_{\mathcal{S}} P\left(z, s'_{\text{tar}}\right) \log \frac{P\left(s'_{\text{tar}}|z\right)}{P\left(s'_{\text{tar}}\right)} \, dz \, ds'_{\text{tar}} \\
&= \int_{\mathcal{Z}} \int_{\mathcal{S}} \int_{\mathcal{S}} P\left(z, s'_{\text{src}}, s'_{\text{tar}}\right) \log \frac{P\left(s'_{\text{src}}|z\right)}{P\left(s'_{\text{src}}\right)} \, dz \, ds'_{\text{src}} \, ds'_{\text{tar}} \\
&\quad - \int_{\mathcal{Z}} \int_{\mathcal{S}} \int_{\mathcal{S}} P\left(z, s'_{\text{tar}}, s'_{\text{src}}\right) \log \frac{P\left(s'_{\text{tar}}|z\right)}{P\left(s'_{\text{tar}}\right)} \, dz \, ds'_{\text{tar}} \, ds'_{\text{src}} \\
&= \int_{\mathcal{Z}} \int_{\mathcal{S}} \int_{\mathcal{S}} P\left(z, s'_{\text{src}}, s'_{\text{tar}}\right) \log \frac{P\left(s'_{\text{src}}|z\right)}{P\left(s'_{\text{tar}}|z\right)} \, dz \, ds'_{\text{src}} \, ds'_{\text{tar}} \\
&\quad - \int_{\mathcal{S}} P\left(s'_{\text{src}}\right) \log P\left(s'_{\text{src}}\right) \, ds'_{\text{src}} + \int_{\mathcal{S}} P\left(s'_{\text{tar}}\right) \log P\left(s'_{\text{tar}}\right) \, ds'_{\text{tar}} \\
&= D_{\text{KL}}\left(P\left(s'_{\text{src}}|z\right) \parallel P\left(s'_{\text{tar}}|z\right)\right) + \mathbb{H}\left(s'_{\text{src}}\right) - \mathbb{H}\left(s'_{\text{tar}}\right).
\end{aligned}
$$

Similarly, the dynamics KL term is an averaged conditional KL:

$$
D_{\text{KL}}\left(P(s'_{\text{src}}|z) \| P(s'_{\text{tar}}|z)\right) := \mathbb{E}_z\left[D_{\text{KL}}\left(P_{\text{src}}(s'|z) \| P_{\text{tar}}(s'|z)\right)\right],
$$

where the inner KL is taken over the same next-state variable $s' \in \mathcal{S}$ with respect to the numerator distribution $P_{\text{src}}(s'|z)$. One can see that the defined function is also connected to the dynamics discrepancy term $D_{\text{KL}}\left(P\left(s'_{\text{src}}|z\right) \parallel P\left(s'_{\text{tar}}|z\right)\right)$ and two entropy terms. Nevertheless, we observe that the source domain and the target domain are specified and fixed, and their state distributions are also fixed, indicating that the entropy terms are constants. So in the end we have

$$
\underbrace{D_{\text{KL}}\left(P\left(z|s'_{\text{src}}\right) \| P\left(z|s'_{\text{tar}}\right)\right)}_{\text{representation deviation}} = \underbrace{D_{\text{KL}}\left(P\left(s'_{\text{src}}|z\right) \| P\left(s'_{\text{tar}}|z\right)\right)}_{\text{dynamics deviation}} + \underbrace{\mathbb{H}\left(s'_{\text{src}}\right) - \mathbb{H}\left(s'_{\text{tar}}\right)}_{\text{constants}}.
$$

This completes the proof. □

B.2 Proof of Theorem 4.3

Proof of Theorem 4.3. For simplicity, denote $\tilde{r} = \sum \gamma^t r\left(s_t, a_t\right)$. Since It is infeasible to directly interact an online policy π^* with an offline policy $\hat{\pi}^*_{\tilde{D}_{src}}$ in different domains, we introduce π^*_{src} that maximizes the reward in the filtered source domain. Then we have

$$
\begin{aligned}
\mathbb{E}_{P_{\text{tar}}, \pi^*}(\tilde{r}) - \mathbb{E}_{P_{\text{tar}}, \hat{\pi}^*_{\tilde{D}_{src}}}(\tilde{r}) = &\underbrace{\mathbb{E}_{P_{\text{tar}}, \pi^*}(\tilde{r}) - \mathbb{E}_{P_{\text{src}}, \pi^*}(\tilde{r})}_{(a)} + \underbrace{\mathbb{E}_{P_{\text{src}}, \pi^*}(\tilde{r}) - \mathbb{E}_{P_{\text{src}}, \pi^*_{\text{src}}}(\tilde{r})}_{(b)} \\
&+ \underbrace{\mathbb{E}_{P_{\text{src}}, \pi^*_{\text{src}}}(\tilde{r}) - \mathbb{E}_{P_{\text{tar}}, \pi^*_{\tilde{D}tar}}(\tilde{r})}_{(c)} + \underbrace{\mathbb{E}_{P_{\text{tar}}, \pi^*_{\tilde{D}tar}}(\tilde{r}) - \mathbb{E}_{P_{\text{tar}}, \hat{\pi}^*_{\tilde{D}_{src}}}(\tilde{r})}_{(d)}.
\end{aligned}
$$

First note that term (b) is 0 because π^*_{src} is the optimal policy in the filtered source domain which should be identical to the target domain, meaning π^*_{src} is almost the same as π^* and they should achieve the same reward in the same domain. Term (a) is bounded above by $2R_{\max}\sqrt{1 - e^{B-\epsilon}}$ by Assumption 4.2. To bound term (c), we first need the following result by applying Theorem C.1:

$$
\mathbb{E}_{P_{\text{src}}, \pi^*_{\text{src}}}(\tilde{r}) - \mathbb{E}_{P_{\text{tar}}, \pi^*}(\tilde{r}) \leq \mathbb{E}_{P_{\text{src}}, \pi^*_{\text{src}}}(\tilde{r}) - \mathbb{E}_{P_{\text{tar}}, \pi^*_{\text{src}}}(\tilde{r}) \leq 2R_{\max}\sqrt{1 - e^{B-\epsilon}}.
$$

Next we can use this and the performance bound of usual offline RL algorithm to find a bound for term (c):

$$
\mathbb{E}_{P_{\text{src}}, \pi^*_{\text{src}}}(\tilde{r}) \leq 2R_{\max}\sqrt{1 - e^{B-\epsilon}} + \mathbb{E}_{P_{\text{tar}}, \pi^*}(\tilde{r}) \leq 2R_{\max}\sqrt{1 - e^{B-\epsilon}} + C + \mathbb{E}_{P_{\text{tar}}, \pi^*_{\tilde{D}tar}}(\tilde{r}).
$$

Rearrange the terms can we find the upper bound for term (c). Finally we will deal with term (d):

$$(d) = \underbrace{\mathbb{E}_{P_{\text{tar}},\pi^*_{\tilde{D}_{tar}}}(\tilde{r}) - \mathbb{E}_{P_{\text{src}},\pi^*_{\tilde{D}_{tar}}}(\tilde{r})}_{I_1} + \underbrace{\mathbb{E}_{P_{\text{src}},\pi^*_{\tilde{D}_{tar}}}(\tilde{r}) - \mathbb{E}_{P_{\text{src}},\hat{\pi}^*_{\tilde{D}_{src}}}(\tilde{r})}_{I_2} + \underbrace{\mathbb{E}_{P_{\text{src}},\hat{\pi}^*_{\tilde{D}_{src}}}(\tilde{r}) - \mathbb{E}_{P_{\text{tar}},\hat{\pi}^*_{\tilde{D}_{src}}}(\tilde{r})}_{I_3}.$$

We bound I_2 by applying Theorem C.3:

$$\begin{aligned}
I_2 &\le \left|\mathbb{E}_{P_{\text{src}},\hat{\pi}^*_{\tilde{D}_{src}}}(\tilde{r}) - \mathbb{E}_{P_{\text{src}},\pi^*_{\tilde{D}_{tar}}}(\tilde{r})\right| \\
&= \left|\frac{1}{1-\gamma}\mathbb{E}_{\rho^{\hat{\pi}^*_{\tilde{D}_{src}}}_{\mathcal{M}_{\text{src}}}(s,a),s'\sim P_{\text{src}}}\left[\mathbb{E}_{a'\sim\hat{\pi}^*_{\tilde{D}_{src}}}\left[Q^{\pi^*_{\tilde{D}_{tar}}}_{\mathcal{M}_{\text{src}}}(s',a')\right] - \mathbb{E}_{a'\sim\pi^*_{\tilde{D}_{tar}}}\left[Q^{\pi^*_{\tilde{D}_{tar}}}_{\mathcal{M}_{\text{src}}}(s',a')\right]\right]\right| \\
&\le \frac{1}{1-\gamma}\mathbb{E}_{\rho^{\hat{\pi}^*_{\tilde{D}_{src}}}_{\mathcal{M}_{\text{src}}}(s,a),s'\sim P_{\text{src}}}\left|\sum_{a'\in\mathcal{A}}\left(\hat{\pi}^*_{\tilde{D}_{src}}(a'|s') - \pi^*_{\tilde{D}_{tar}}(a'|s')\right)Q^{\pi^*_{\tilde{D}_{tar}}}_{\mathcal{M}_{\text{src}}}(s',a')\right| \\
&\le \frac{R_{\max}}{1-\gamma}\mathbb{E}_{\rho^{\hat{\pi}^*_{\tilde{D}_{src}}}_{\mathcal{M}_{\text{src}}}(s,a),s'\sim P_{\text{src}}}\sum_{a'\in\mathcal{A}}\left|\left(\hat{\pi}^*_{\tilde{D}_{src}}(a'|s') - \pi^*_{\tilde{D}_{tar}}(a'|s')\right)\right| \\
&= \frac{2R_{\max}}{1-\gamma}\mathbb{E}_{\rho^{\hat{\pi}^*_{\tilde{D}_{src}}}_{\mathcal{M}_{\text{src}}}(s,a),s'\sim P_{\text{src}}}\left[D_{\text{TV}}\left(\hat{\pi}^*_{\tilde{D}_{src}}(\cdot|s')\,\|\,\pi^*_{\tilde{D}_{tar}}(\cdot|s')\right)\right].
\end{aligned}$$

We bound I_1 by applying Theorem C.2:

$$\begin{aligned}
I_1 &\le \left|\mathbb{E}_{P_{\text{src}},\pi^*_{\tilde{D}_{tar}}}(\tilde{r}) - \mathbb{E}_{P_{\text{tar}},\pi^*_{\tilde{D}_{tar}}}(\tilde{r})\right| \\
&= \left|\frac{\gamma}{1-\gamma}\mathbb{E}_{\rho^{\pi^*_{\tilde{D}_{tar}}}_{\mathcal{M}_{\text{src}}}(s,a)}\left[\mathbb{E}_{s'\sim P_{\text{src}}}\left[V^{\pi^*_{\tilde{D}_{tar}}}_{\mathcal{M}_{\text{tar}}}(s')\right] - \mathbb{E}_{s'\sim P_{\text{tar}}}\left[V^{\pi^*_{\tilde{D}_{tar}}}_{\mathcal{M}_{\text{tar}}}(s')\right]\right]\right| \\
&= \left|\frac{\gamma}{1-\gamma}\mathbb{E}_{\rho^{\pi^*_{\tilde{D}_{tar}}}_{\mathcal{M}_{\text{src}}}(s,a)}\left[\int_{\mathcal{S}}\left(P_{\mathcal{M}_{\text{src}}}(s' \mid s,a) - P_{\mathcal{M}_{\text{tar}}}(s' \mid s,a)\right)V^{\pi^*_{\tilde{D}_{tar}}}_{\mathcal{M}_{\text{tar}}}(s')\,ds'\right]\right| \\
&\le \frac{\gamma}{1-\gamma}\mathbb{E}_{\rho^{\pi^*_{\tilde{D}_{tar}}}_{\mathcal{M}_{\text{src}}}(s,a)}\left[\left|\int_{\mathcal{S}}\left(P_{\mathcal{M}_{\text{src}}}(s' \mid s,a) - P_{\mathcal{M}_{\text{tar}}}(s' \mid s,a)\right)\right| \times \left|V^{\pi^*_{\tilde{D}_{tar}}}_{\mathcal{M}_{\text{tar}}}(s')\right| ds'\right] \\
&\le \frac{2\gamma}{1-\gamma}R_{\max}\left[D_{\text{TV}}\left(P_{\mathcal{M}_{\text{src}}}(\cdot \mid s,a)\|P_{\mathcal{M}_{\text{tar}}}(\cdot \mid s,a)\right)\right] \\
&\le \frac{2\gamma}{1-\gamma}R_{\max}\sqrt{1-e^{B-\epsilon}}.
\end{aligned}$$

Similarly for I_3, we get the same upper bound. Combining all bounds together, we get the desired result. □

C Useful Lemmas

Lemma C.1. Let policy π be trained on the filtered source domain, and $R_{\max}$ be the maximum (entropy-regularized) return of any trajectory. Then the following inequality holds:

$$\left|\mathbb{E}_{\pi,P_{\text{src}}}\left[\sum\gamma^t r(s_t,a_t)\right] - \mathbb{E}_{\pi,P_{\text{tar}}}\left[\sum\gamma^t r(s_t,a_t)\right]\right| \le 2R_{\max}\sqrt{1-e^{B-\epsilon}}.$$

This lemma proves that policies trained on the filtered source domain satisfy the same condition as the optimal policy (Assumption 4.2).

Proof. This proof is similar to that of Lemma B.2 in [8], and the only difference is that we use B-H bound for TV distance instead of Pinsker's inequality in the last step. We repeat the proof here for readers' reference. We know $z = f(s,a)$ is injective, and apply Holder's inequality and Bretagnolle–Huber bound, which is never worse than Pinsker's inequality [3], to obtain the desired result:

$$\left|\mathbb{E}_{P_{\text{src}}}\left[\sum\gamma^t r(s_t,a_t)\right] - \mathbb{E}_{P_{\text{tar}}}\left[\sum\gamma^t r(s_t,a_t)\right]\right|$$

$$
\begin{aligned}
&= \left| \sum_{\tau} \left(P_{\text{src}}(\tau) - P_{\text{tar}}(\tau) \right) \left(\sum \gamma^t r\left(s_t, a_t\right) \right) \right| \\
&\leq \left\| \sum \gamma^t r\left(s_t, a_t\right) \right\|_{\infty} \cdot \left\| P_{\text{src}}(\tau) - P_{\text{tar}}(\tau) \right\|_1 \\
&\leq \left(\max_{\tau} \sum \gamma^t r\left(s_t, a_t\right) \right) \cdot 2\sqrt{1 - e^{-D_{\text{KL}}(P_{\text{src}}(\tau) \| P_{\text{tar}}(\tau))}} \\
&\leq 2R_{\max}\sqrt{1 - e^{B-\epsilon}}.
\end{aligned}
$$

This completes the proof. □

Lemma C.2 (Telescoping lemma, Lemma 4.3 from [30])**.** . Denote $\mathcal{M}_1 = (\mathcal{S}, \mathcal{A}, P_1, r, \gamma)$ and $\mathcal{M}_2 = (\mathcal{S}, \mathcal{A}, P_2, r, \gamma)$ as two MDPs that only differ in their transition dynamics. Then for any policy π, we have

$$
J_{\mathcal{M}_1}(\pi) - J_{\mathcal{M}_2}(\pi) = \frac{\gamma}{1-\gamma} \mathbb{E}_{\rho^{\pi}_{\mathcal{M}_1}(s,a)} \left[\mathbb{E}_{s' \sim P_1} \left[V^{\pi}_{\mathcal{M}_2}\left(s'\right) \right] - \mathbb{E}_{s' \sim P_2} \left[V^{\pi}_{\mathcal{M}_2}\left(s'\right) \right] \right].
$$

Lemma C.3. Denote $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$ as the underlying MDP. Suppose we have two policies π_1, π_2, then the performance difference of these policies in the MDP gives:

$$
J_{\mathcal{M}}\left(\pi_1\right) - J_{\mathcal{M}}\left(\pi_2\right) = \frac{1}{1-\gamma} \mathbb{E}_{\rho^{\pi_1}_{\mathcal{M}}(s,a), s' \sim P} \left[\mathbb{E}_{a' \sim \pi_1} \left[Q^{\pi_2}_{\mathcal{M}}\left(s', a'\right) \right] - \mathbb{E}_{a' \sim \pi_2} \left[Q^{\pi_2}_{\mathcal{M}}\left(s', a'\right) \right] \right].
$$

Proof. This is Lemma B.3 in [31], please check the proof there. □

D Experiment Details

In this section, we provide more details for the experiment settings and hyperparameters in Section 5.

D.1 Locomotion Tasks

In this section, we demonstrate the detailed modification in the MuJoCo `xml` file to construct the target environment. In the locomotion tasks, we follow the ODRL, providing three dynamics shifts including gravity, friction and morphology shift. In the gravity and friction shift, we consider the shift level with $\{0.1, 0.5, 2.0, 5.0\}$. In the morphology shift, we consider the *medium* and *hard* level shift.

Gravity Shift. Following the ODRL benchmark [32], we modify the gravity of the environment by editing the gravity attribute. For example, the gravity of the HalfCheetah in the target is modified to 0.5 times the gravity in the source domain with the following code.

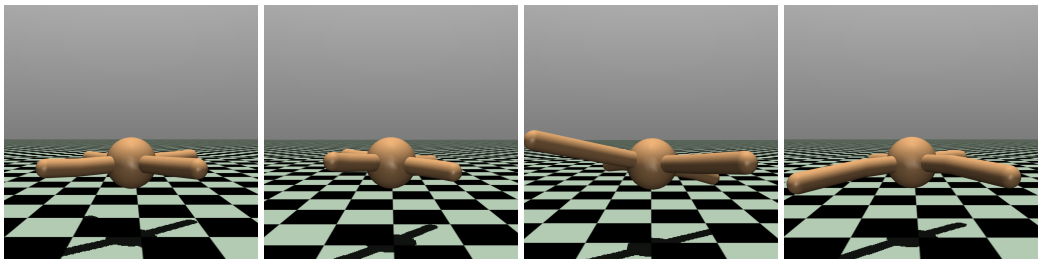
```
# gravity
<option gravity="0 0 -4.905" timestep="0.01"/>
```

Friction Shift. The friction shift is generated by modifying the friction attribute in the geom elements. The frictional components are adjusted to 0.1, 0.5, 2.0, 5.0 times the frictional components in the source domain, respectively.

Morphology Shift. The morphology shift is achieved by modifying the size of specific limbs or torsos of the simulated robot in MuJoCo without altering the state space and action space. Visualizations can be found in Figure 5, Figure 6, Figure 7 and Figure 8.

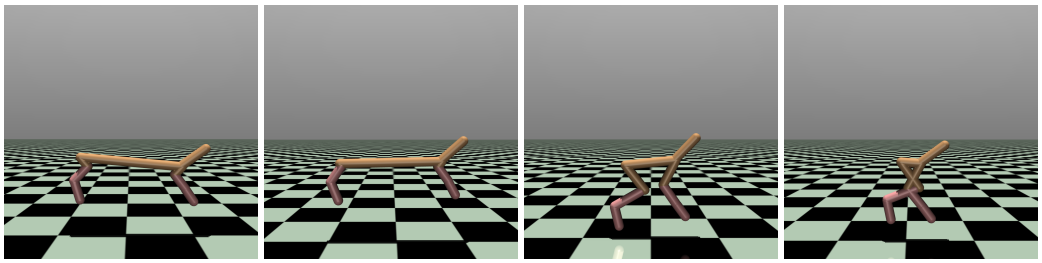
D.2 Navigation Tasks

In this section, we present the detailed modification in the AntMaze map to construct the target environment. In this setting, we follow the ODRL benchmark and consider the medium size with six different target maps. Visualization of the map can be found in Figure 9.



(a) alllegs (M) (b) alllegs (H) (c) halflegs (M) (d) halflegs (H)

Figure 5: Visualization of morphology shift environments for Ant.



(a) thigh (M) (b) thigh (H) (c) torso (M) (d) torso (H)

Figure 6: Visualization of morphology shift environments for HalfCheetah.

D.3 Manipulation Tasks

In this section, we discuss the detailed modifications in the Adroit environment. In this setting, we follow the ODRL benchmark using the *pen* and *door* as the tasks. The *pen* task is to control the 24-DoF shadow hand robot to twirl a pen, and the *door* task is to open a door. We consider the two types of dynamics shifts under this environment, denoted as the kinematic shift and the morphology task. We only consider the *medium* and *hard* level shift. Visualizations can be found in Figure 10 and Figure 11.

Kinematic Shift. The kinematic shift in the Adroit robot hand occurs at all finger joints of the index finger and the thumb. We make constraints on the range of torsos. We use two types of shift levels, denoted as medium and hard level for the kinematic shift.

Morphology Shift. The morphology shift in the Adroit robot hand occurs on the fingers. We shrink the sizes of these fingers with the medium and hard levels, which multiply by 0.25 and 0.125 for the size of the fingers, respectively.

D.4 Datasets

Note that we follow D4RL and adopt the *normalized score* metric to better characterize the agent's performance across different tasks. The normalized score in the target domain is defined as:

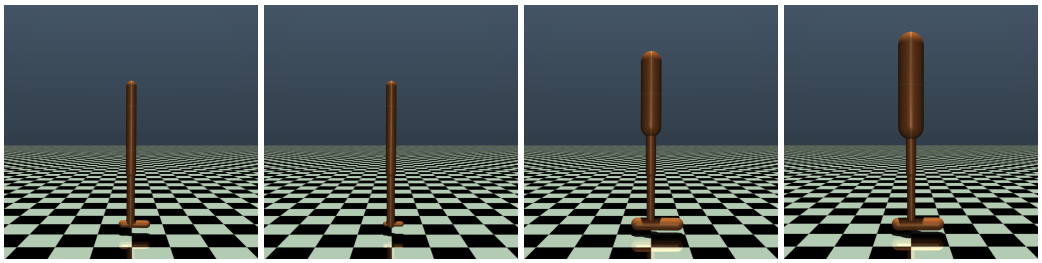
$$\mathrm{NS} = \frac{J - J_r}{J_e - J_r} \times 100,$$

where J denotes the return achieved by the agent in the target domain, and J_r, J_e represent the returns obtained by the random and expert policies in the same domain, respectively.

We summarize the reference scores of J_r and J_e under different dynamics shift scenarios in Table 3, Table 4 and Table 5.

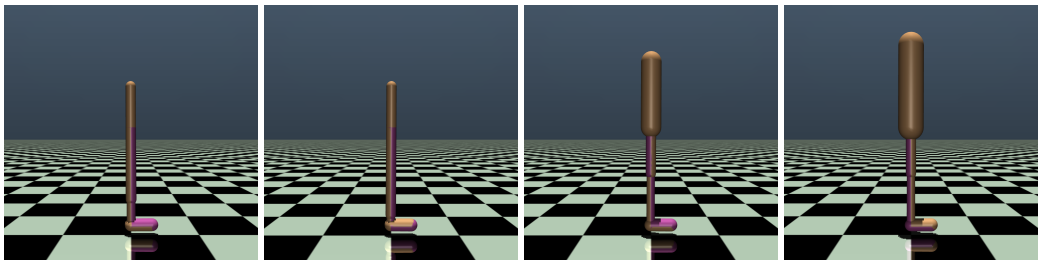
D.5 Technical Details about Baseline Algorithms

In this section, we describe the implementation details of the baseline methods, following the ODRL benchmark [32]. Furthermore, we list the hyperparameter setup used for all methods in Table 6.



(a) foot (M) (b) foot (H) (c) torso (M) (d) torso (H)

Figure 7: Visualization of morphology shift environments for Hopper.



(a) leg (M) (b) leg (H) (c) torso (M) (d) torso (H)

Figure 8: Visualization of morphology shift environments for Walker2d.

BOSA [23]. BOSA deals with the OOD state–action pairs through a supported policy optimization and addresses the OOD dynamics issue through a supported value optimization by data filtering. Specifically, the policy is updated by maximizing the following objective function:

$$\mathcal{L}_{\text{actor}} = \mathbb{E}_{s\sim D_{\text{src}}\cup D_{\text{trg}},\, a\sim\pi_\phi(s)}\big[Q(s,a)\big], \quad \text{s.t.} \quad \mathbb{E}_{s\sim D_{\text{src}}\cup D_{\text{trg}}}\big[\hat{\pi}_{\theta_{\text{offline}}}(\pi_\phi(s)\mid s)\big] > \epsilon.$$

Here, ϵ is the threshold, $\hat{\pi}_{\theta_{\text{offline}}}$ is the learned policy for the combined offline dataset. The value critic is updated with

$$\begin{aligned}\mathcal{L}_{\text{critic}} = {} & \mathbb{E}_{(s,a)\sim D_{\text{src}}}\big[Q(s,a)\big] \\ & + \mathbb{E}_{(s,a,r,s')\sim D_{\text{src}}\cup D_{\text{trg}},\, a'\sim\pi_\phi(\cdot|s')}\Big[I\left(\hat{p}_{\text{trg}}(s'\mid s,a) > \epsilon'\right)\big(Q_{\theta_i}(s,a) - y\big)^2\Big],\end{aligned}$$

where $I(\cdot)$ is the indicator function, $\hat{p}_{\text{trg}}(s'\mid s,a) = \operatorname{argmax}\mathbb{E}_{(s,a,s')\sim D_{\text{trg}}}\big[\log\hat{p}_{\text{trg}}(s'\mid s,a)\big]$ is the estimated target domain dynamics and ϵ' is the threshold. The above objective is transformed to a relaxed Lagrangian form for optimization. Both the behavior policy and the dynamics models are modeled by the CVAE. We implement BOSA by following the instructions in the original paper. BOSA is trained for 500K gradient steps in practice by drawing samples from both the source domain dataset and the target domain dataset.

IQL [19]. IQL learns the state value function and state–action value function simultaneously by minimizing the following expectile regression loss function

$$\mathcal{L}_V = \mathbb{E}_{(s,a)\sim D_{\text{src}}\cup D_{\text{trg}}}\big[L_2^\tau\big(Q_\theta(s,a) - V_\psi(s)\big)\big],$$

where $L_2^\tau(u) = |\tau - I(u<0)|\|u\|^2$, $I(\cdot)$ is the indicator function, and θ is the target network parameter. The state–action value function is then updated by

$$\mathcal{L}_Q = \mathbb{E}_{(s,a,r,s')\sim D_{\text{src}}\cup D_{\text{trg}}}\Big[\big(r(s,a) + \gamma V_\psi(s') - Q_\theta(s,a)\big)^2\Big].$$

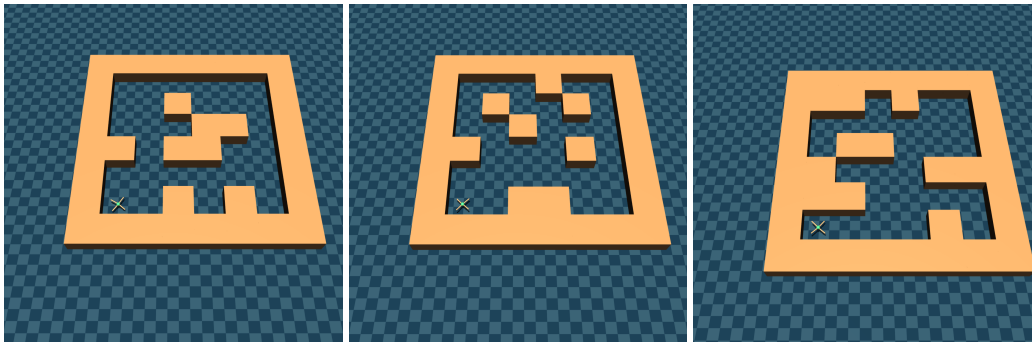
The advantage function is $A(s,a) = Q(s,a) - V(s)$. The policy is optimized by the advantage-weighted behavior cloning

$$\mathcal{L}_{\text{actor}} = \mathbb{E}_{(s,a)\sim D_{\text{src}}\cup D_{\text{trg}}}\big[\exp\big(\beta\cdot A(s,a)\big)\log\pi_\phi(a\mid s)\big],$$

where β is the inverse temperature coefficient. We implement IQL by following its official codebase and also train the IQL agent on offline datasets from both domains for 500K gradient steps.

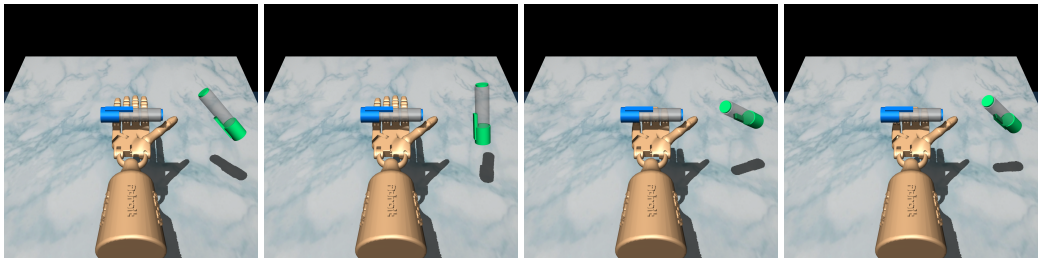


(a) antmaze-medium-1 (b) antmaze-medium-2 (c) antmaze-medium-3



(d) antmaze-medium-4 (e) antmaze-medium-5 (f) antmaze-medium-6

Figure 9: Visualization of the AntMaze map structure. We consider medium size map with 6 different map layouts.



(a) broken-joint (M) (b) broken-joint (H) (c) shrink-finger (M) (d) shrink-finger (H)

Figure 10: Visualization of morphology and kinematic shift environments for Adroit Pen.

DARA [22]. DARA is the offline version of DARC [8]. It trains two domain classifiers $q_{\theta_{\text{SAS}}}(\text{target} \mid s_t, a_t, s_{t+1})$, $q_{\theta_{\text{SA}}}(\text{target} \mid s_t, a_t)$ with the following objectives

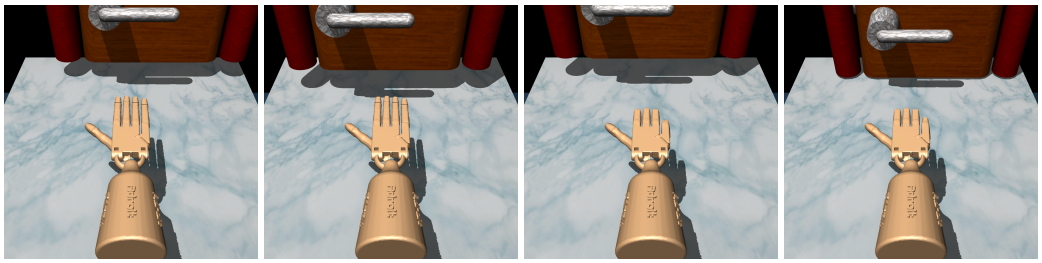
$$\mathcal{L}(\theta_{\text{SAS}}) = \mathbb{E}_{D_{\text{tar}}}\left[\log q_{\theta_{\text{SAS}}}(\text{target} \mid s_t, a_t, s_{t+1})\right] + \mathbb{E}_{D_{\text{src}}}\left[\log\left(1 - q_{\theta_{\text{SAS}}}(\text{target} \mid s_t, a_t, s_{t+1})\right)\right],$$
$$\mathcal{L}(\theta_{\text{SA}}) = \mathbb{E}_{D_{\text{tar}}}\left[\log q_{\theta_{\text{SA}}}(\text{target} \mid s_t, a_t)\right] + \mathbb{E}_{D_{\text{src}}}\left[\log\left(1 - q_{\theta_{\text{SA}}}(\text{target} \mid s_t, a_t)\right)\right],$$

to estimate the dynamics gap

$$\log \frac{P_{\mathcal{M}_{\text{tar}}}(s_{t+1} \mid s_t, a_t)}{P_{\mathcal{M}_{\text{src}}}(s_{t+1} \mid s_t, a_t)}$$

between the source domain and the target domain. DARA approximates this term by leveraging the trained classifiers and proposes to modify the source domain rewards as follows

$$\hat{r}_{\text{DARA}} = r - \lambda \times \delta r,$$
$$\delta r(s_t, a_t) = -\log \frac{q_{\theta_{\text{SAS}}}(\text{target} \mid s_t, a_t, s_{t+1})\, q_{\theta_{\text{SA}}}(\text{source} \mid s_t, a_t)}{q_{\theta_{\text{SAS}}}(\text{source} \mid s_t, a_t, s_{t+1})\, q_{\theta_{\text{SA}}}(\text{target} \mid s_t, a_t)},$$



(a) broken-joint (M) (b) broken-joint (H) (c) shrink-finger (M) (d) shrink-finger (H)

Figure 11: Visualization of morphology and kinematic shift environments for Adroit Door.

where λ is an important hyperparameter that controls the strength of the reward penalty, which is set to be 0.1 by default. After that, DARA is trained on data from both domains for 500K gradient steps. We implement DARA by following the original paper and clip the reward penalty term to lie in $[-10, 10]$. We use IQL as the base algorithm for DARA.

IGDF [42]. IGDF captures the dynamics gap between the source domain and the target domain through contrastive learning. It trains a score function $h(\cdot)$ using $(s, a, s'_{\text{tar}}) \sim D_{\text{tar}}$ from the target domain as positive samples, and mixed transition (s, a, s'_{src}) as negative samples, where $(s, a) \sim D_{\text{tar}}$, $s'_{\text{src}} \sim D_{\text{src}}$. The score function is optimized via the following contrastive learning objective:

$$\mathcal{L}_{\text{contrastive}} = -\mathbb{E}_{(s,a,s'_{\text{tar}})}\mathbb{E}_{S'^-}\left[\log \frac{h(s,a,s'_{\text{tar}})}{\sum_{s'\sim S'^- \cup s'_{\text{tar}}} h(s,a,s')}\right],$$

where S'^- is a collection of the next states in negative samples. Practically, IGDF adopts two neural networks $\phi(s,a), \psi(s')$ to learn representations of state-action pairs and states, and approximate the score function with a linear parameterization of them:

$$h(s,a,s') = \exp\left(\phi(s,a)^\top \psi(s')\right).$$

Based on the measured score function, IGDF proposes to filter out source domain data when training value functions:

$$\mathcal{L}_{\text{critic}} = \tfrac{1}{2}\,\mathbb{E}_{D_{\text{tar}}}\left[(Q_\theta - \mathcal{T}Q_\theta)^2\right] + \tfrac{1}{2}\,\alpha \cdot h(s,a,s')\,\mathbb{E}_{(s,a,s')\sim D_{\text{src}}}\left[\mathbf{1}\left(h(s,a,s') > h_{\xi\%}\right)(Q_\theta - \mathcal{T}Q_\theta)^2\right],$$

where α is the importance coefficient for weighting the TD error of the source domain data, and ξ is the data selection ratio akin to that in OTDF. We run IGDF by using its official codebase and use IQL as its backbone.

OTDF [33]. OTDF performs off-dynamics offline policy adaptation by aligning the transition dynamics of the source and target domains using optimal transport. It first solves the optimal transport problem using

$$W(u,u') = \min_{\mu\in\mathcal{M}} \sum_{t=1}^{|D_{\text{src}}|}\sum_{t'=1}^{|D_{\text{tar}}|} C(u_t, u'_{t'})\,\mu_{t,t'},$$

given cost function C. Suppose the optimal coupling obtained is μ^*, then it defines the deviation between a source domain data point and the target domain dataset as:

$$d(u_t) = -\sum_{t'=1}^{|D_{\text{tar}}|} C(u_t, u'_{t'})\,\mu^*_{t,t'}, u_t = (s_t^{\text{src}}, a_t^{\text{src}}, (s'_{\text{src}})_t) \sim D_{\text{src}}.$$

After computing and normalizing d, it trains the value function via

$$\mathcal{L}_Q = \mathbb{E}_{(s,a,s')\sim D_{\text{tar}}}\left[(Q_\theta - \mathcal{T}Q_\theta)^2\right] + \mathbb{E}_{(s,a,s')\sim D_{\text{src}}}\left[exp(\hat{d})\cdot \mathbf{1}(d > d_{\xi\%})\,(Q_\theta - \mathcal{T}Q_\theta)^2\right],$$

Table 3: The referenced minimum and maximum scores for MuJoCo datasets under various dynamics shift scenarios. These reference scores are used to compute normalized performance in the target domain.

Task Name	Dynamics Shift Type	Reference Min Score (J_r)	Reference Max Score (J_e)
HalfCheetah	Gravity - 0.1	-280.18	2466.85
	Gravity - 0.5	-280.18	9509.15
	Gravity - 2.0	-280.18	9509.15
	Gravity - 5.0	-280.18	3756.24
	Friction - 0.1	-280.18	41696.55
	Friction - 0.5	-280.18	7357.07
	Friction - 2.0	-280.18	11255.97
	Friction - 5.0	-280.18	10199.33
	Local	-280.18	12135.0
Hopper	Gravity - 0.1	-26.34	3234.3
	Gravity - 0.5	-26.34	3234.3
	Gravity - 2.0	-26.34	3234.3
	Gravity - 5.0	-26.34	3234.3
	Friction - 0.1	-26.34	3234.3
	Friction - 0.5	-26.34	3234.3
	Friction - 2.0	-26.34	3234.3
	Friction - 5.0	-26.34	3234.3
	Local	-20.272305	3234.3
Walker2d	Gravity - 0.1	10.08	2074.90
	Gravity - 0.5	10.08	5194.71
	Gravity - 2.0	10.08	5056.45
	Gravity - 5.0	10.08	3665.39
	Friction - 0.1	10.08	3360.18
	Friction - 0.5	10.08	4229.35
	Friction - 2.0	10.08	5180.04
	Friction - 5.0	10.08	4988.84
	Local	1.629008	4592.3
Ant	Gravity - 0.1	-325.6	2782.09
	Gravity - 0.5	-325.6	4317.07
	Gravity - 2.0	-325.6	6705.12
	Gravity - 5.0	-325.6	6226.89
	Friction - 0.1	-325.6	7938.96
	Friction - 0.5	-325.6	8301.34
	Friction - 2.0	-325.6	5167.38
	Friction - 5.0	-325.6	4545.02
	Local	-325.6	3879.7

where $\hat{d}$ is normalized d and $\mathbf{1}(\cdot)$ is the indicator function. It then trains a Conditional Variational Autoencoder (CVAE) using loss function

$$\mathcal{L}_{\text{CVAE}} = \mathbb{E}_{(s,a)\sim D_{\text{tar}},\, z\sim E_\nu(s,a)}\Big[(a - D_\varsigma(s,z))^2\Big] + D_{\text{KL}}\big(E_\nu(s,a)\,\|\,\mathcal{N}(0,I)\big),$$

where $E_\nu(s,a)$ is the encoder and $D_\varsigma(s,z)$ is the decoder, to model the target dynamics and generates multiple latent transitions. Finally, they train the policy via

$$\hat{\mathcal{L}}_\pi = \mathcal{L}_\pi - \beta\,\mathbb{E}_{s\sim D_{\text{src}}\cup D_{\text{tar}}}\left[\log\left(\sum_{i=1}^{M}\exp\left(\log\hat{\pi}^i_{\text{tar}}(\pi(\cdot|s)\mid s)\right)\right)\right],$$

where $\hat{\pi}^i_{\text{tar}}$ denotes the i-th constructed Gaussian behavior policy in the target domain. A data selection mechanism filters source samples whose generated transitions closely match the target. We run OTDF using its official codebase.

Table 4: The referenced minimum and maximum scores for AntMaze medium datasets under various dynamics shift scenarios. These reference scores are used to compute normalized performance in the target domain.

Task Name	Dynamics Shift Type	Reference Min Score (J_r)	Reference Max Score (J_e)
AntMaze-medium	1	0.0	1.0
	2	0.0	1.0
	3	0.0	1.0
	4	0.0	1.0
	5	0.0	1.0
	6	0.0	1.0

Table 5: The referenced minimum and maximum scores for Adroit datasets under various dynamics shift scenarios. These reference scores are used to compute normalized performance in the target domain.

Task Name	Dynamics Shift Type	Reference Min Score (J_r)	Reference Max Score (J_e)
Pen	broken-joint-medium	-12.17	6408.38
	broken-joint-hard	-12.17	6408.38
	shrink-finger-medium	-12.17	6408.38
	shrink-finger-hard	-12.17	6408.38
Door	broken-joint-medium	-52.34	2880.57
	broken-joint-hard	-52.34	2880.57
	shrink-finger-medium	-52.34	2880.57
	shrink-finger-hard	-52.34	2880.57

DmC [40]. DmC combines nearest-neighbor-based domain discrepancy estimation with diffusion-based data augmentation. For each source transition $x = (s, a, s')$, it estimates the domain discrepancy as

$$\rho_k(x) = \log \nu_{k,\text{tar}}(x) - \log \nu_{k,\text{src}}(x),$$

where $\nu_{k,\text{tar}}$ and $\nu_{k,\text{src}}$ denote the k-th nearest-neighbor distances from x to the target and source datasets, respectively. The scores are shifted as $\hat{\rho}_k(x) = \rho_k(x) - \min_{x'} \rho_k(x')$ and converted into weights $w_k(x) = 1/(1 + \hat{\rho}_k(x))$, where higher weights indicate better alignment with the target data. DmC then trains a diffusion model on source transitions, using these proximity weights as conditioning signals to generate target-aligned transitions. The generated transitions are added to the source dataset for offline RL training. During critic training, DmC filters and reweights source transitions according to

$$\omega(x) = w_k(x)\mathbf{1}(w_k(x) \geq w_{k,\xi\%}),$$

where $w_{k,\xi\%}$ is the weight threshold determined by the filtering percentile. Using IQL as the backbone, DmC further applies a CVAE-based regularizer to keep the learned policy close to the target behavior policy.

D.6 Technical Details about LoDADA

The LoDADA algorithm (Localized Dynamics-Aware Domain Adaptation), summarized in Algorithm 1, leverages clustering and divergence-based filtering to improve off-dynamics policy learning. First, target domain data are clustered by next-state variables to form clusters, and source samples are assigned to clusters based on proximity to centroids within a diameter threshold. For each cluster, a classifier estimates KL divergence between source and target distributions, and clusters are ranked accordingly. Source samples are then filtered adaptively based on the divergence ranking of their assigned clusters, which are divided into three classes according to their KL divergence: low, medium, and high. Clusters in the low-divergence class, which are most similar to the target

dataset, admit the largest proportion ξ_1 of their source samples; clusters in the medium-divergence class admit a moderate proportion ξ_2; and clusters in the high-divergence class—indicating greater dissimilarity—admit only a small fraction ξ_3.

After filtering, each retained source sample is assigned an importance weight derived from its normalized KL estimate, so that samples from more similar clusters have higher influence during training, while samples from less similar clusters are down-weighted. During training, mini-batches from both domains are used to update the value function, compute target values, and optimize the state-action value function with divergence-weighted losses. Finally, the target network is updated, and the policy is optimized with CVAE-based regularization, enabling effective adaptation across domains.

We use fixed filtering ratios $(\xi_1, \xi_2, \xi_3) = (90\%, 80\%, 70\%)$ across all environments and do not tune them separately for each task. This three-tier schedule keeps the filtering rule simple and reproducible, while retaining a nonzero fraction of samples from high-KL clusters to preserve state-space coverage. For the number of clusters K and the policy regularization weight λ, we tune them within small task-dependent candidate sets. For MuJoCo locomotion and Adroit manipulation tasks, we use $K \in \{30, 50\}$ and $\lambda \in \{0.1, 0.5, 1.0\}$. For AntMaze, we use a smaller cluster range $K \in \{5, 10, 15\}$ and a larger regularization range $\lambda \in \{0.5, 1.0, 5.0\}$, because useful source-target overlap is concentrated around structured map regions. Additional sensitivity results on filtering ratios, AntMaze, and Adroit are reported in Appendix E.7 and Appendix E.10.

D.7 Compute Infrastructure

We run the experiment on a single GPU: NVIDIA RTX A5000-24564MiB with 8-CPUs: AMD Ryzen Threadripper 3960X 24-Core. Each experiment requires 12GB RAM and require 20GB available disk space for storage of the data.

E More Experimental Results

In this section, we provide more experimental results that are omitted from the main text due to space limitations.

E.1 Results under morphology shifts

As shown in Table 7, our method improves the total score to 597.88, yielding a 32.7% improvement over the second best approach OTDF, and a 63% improvement over backbone algorithm IQL. Across the 16 evaluation tasks (8 environments × 2 shift levels), LoDADA achieves best performance in 13 out of 16 tasks. These results indicate that our proposed approach consistently enhances robustness under morphology shifts. Relative to the baseline IQL, our method delivers consistent and significant performance gains across 13 out of 16 tasks.

E.2 Results on Localized Perturbations

We provide the detailed setup and numerical results for the localized perturbation experiments. In this setting, the target domain remains unchanged, while the source dataset is perturbed to simulate region-dependent dynamics mismatch. Policies are trained on the perturbed source dataset together with the limited target dataset and evaluated in the unchanged target domain.

For the target dataset, we randomly sample 5,000 transitions from D4RL without adding dynamics perturbations. For the source dataset, we first partition the next-state space into 15 regions using K-means. We then divide these regions into M groups and inject Gaussian noise into state variables, where all samples in the same group share the same noise variance. We consider $M \in \{3, 5\}$. When $M = 3$, the noise variance groups are $\{0.1, 0.5, 2.0\}$, $\{0.2, 1.0, 4.0\}$, and $\{0.3, 1.5, 6.0\}$. When $M = 5$, the noise variance groups are $\{0.1, 0.2, 0.3, 0.4, 0.5\}$, $\{0.2, 0.4, 0.6, 0.8, 1.0\}$, and $\{0.3, 0.6, 0.9, 1.2, 1.5\}$.

As shown in Table 8, LoDADA achieves a total score of 943.09, corresponding to a 29.3% improvement over the second-best approach OTDF and a 55.1% improvement over IQL. Across the 24 evaluation tasks (4 environments × 6 shift levels), LoDADA achieves best (18) or second-best (3)

Algorithm 1 LoDADA : Localized Dynamics-Aware Domain Adaptation

Input: Source domain data D_{src}, target domain data D_{tar}
Initialize: Policy π_ϕ, value networks V_ψ, Q_θ, target Q function $Q_{\theta'}$, batch size B, number of clusters K, diameter threshold δ, critic importance ratio α, regularization weight λ, filtering ratio ξ_1, ξ_2, ξ_3, target update rate η, CVAE with encoder $E_\nu(s,a)$ and decoder $D_\varsigma(s,z)$, number of sampled latent variables M
Train CVAE policy to model the behavior policy in the target domain dataset using

$$\mathcal{L}_{\text{CVAE}} = \mathbb{E}_{(s,a)\sim\mathcal{D}_{\text{tar}},\, z\sim E_\nu(s,a)} \left[\|a - D_\varsigma(s,z)\|_2^2 + D_{\text{KL}}\left(E_\nu(s,a) \| \mathcal{N}(0,I)\right) \right].$$

Cluster D_{tar} into $\{\mathcal{N}^i\}_{i=1}^K$ based on s', with centroid c^i and diameter d^i
for `each source sample` $(s,a,s') \in D_{src}$ **do**
  Find nearest centroid c^i among $\{\mathcal{N}^i\}_{i=1}^K$
  if dist$(s', c^i) \le \delta \cdot \frac{1}{K}\sum_{i=1}^K d^i$ **then**
    Assign (s,a,s') to cluster $\mathcal{N}^i$
  end if
end for
for i=1, 2..., K **do**
  Train classifiers $D_n(z)$ based on Equation (4.1)
  Calculate cluster estimate of KL divergence $\hat{D}_{KL}^n$ based on Equation (4.3)
end for
Sort $\{\mathcal{N}^i\}_{i=1}^K$ based on $\hat{D}_{KL}^n$ from smallest to largest and get $\{\mathcal{N}^{(j)}\}_{j=1}^K$
for each cluster $\mathcal{N}^{(j)}$, $j = 1, \ldots, K$ **do**
  Rank point estimates $\{d_i^j\}_{i=1}^{|\mathcal{N}_0^{(j)}|}$ of sampled source data
  Admit the top $\xi_{g(j)}\%$ samples, where $g(j) = \begin{cases} 1, & j \le \lceil K/3 \rceil, \\ 2, & \lceil K/3 \rceil < j \le \lceil 2K/3 \rceil, \\ 3, & j > \lceil 2K/3 \rceil. \end{cases}$
end for
Normalize point estimate of KL divergence via $\hat{d}_i = \frac{d_i - \max_i d_i}{\max_i d_i - \min_i d_i}, i \in \{1, ..., |\tilde{D}_{src}|\}$
for `each gradient step` **do**
  Sample a mini-batch b_{src} of size $B/2$ from $\tilde{D}_{src}$ and b_{tar} of size $B/2$ from D_{tar}
  Update the state value function V_ψ via: $\mathcal{L}_V = \mathbb{E}_{(s,a)\sim\tilde{D}_{src}\cup D_{tar}} \left[L_2^\tau \left(Q_{\theta'}(s,a) - V_\psi(s) \right) \right]$
  Compute the target value via: $y = r + \gamma V_\psi(s')$
  Optimize the state-action value function Q_θ on $b_{\text{src}} \cup b_{\text{tar}}$ via:

$$\mathcal{L}_Q = \mathbb{E}_{D_{tar}}[(Q_\theta - y)^2] + \mathbb{E}_{(s,a,s',d)\sim\tilde{D}_{src}}[\exp(-\alpha \cdot \hat{d}_i)(Q_\theta - y)^2].$$

  Update the target network via: $\theta' \leftarrow \eta\theta + (1-\eta)\theta'$
  Decode M actions from the CVAE and construct Gaussian distributions $\{\hat{\pi}_i^{\text{tar}}(\cdot \mid s)\}_{i=1}^M$.
  Compute the advantage A and optimize the policy π_ϕ on $b_{\text{src}} \cup b_{\text{tar}}$ using

$$\mathcal{L}_\pi = -\mathbb{E}_{(s,a)\sim\tilde{D}_{\text{src}}\cup\mathcal{D}_{\text{tar}}} \left[\exp(A) \log \pi_\phi(a \mid s) \right] - \lambda \mathbb{E}_{s\sim\tilde{D}_{\text{src}}\cup\mathcal{D}_{\text{tar}}} \left[\log \left(\sum_{i=1}^M \hat{\pi}_{\text{tar}}^i \left(\pi_\phi(\cdot \mid s) \mid s \right) \right) \right].$$

end for

performance in 21 out of 24 tasks. In contrast, DARA and IGDF often underperform in this setting due to their reliance on global mismatch assumptions. These results demonstrate that LoDADA is particularly effective under localized perturbations, where cluster-aware filtering becomes more beneficial than under global shifts.

E.3 Results on Antmaze

In this section, we present additional experiment results on the AntMaze benchmark using the play and diverse source dataset. We follow the same evaluation procedure as in the diverse dataset setting and report normalized target environment performance across 6 AntMaze medium environments.The results are summarized in Table 9 and Table 10. As shown in Table 9, our proposed LoDADA achieves the best overall performance under the play dataset, with a total score of 398.8, outperforming all baseline methods. Compared to the second-best method, LoDADA improves the total score by 5.3%, demonstrating its robustness under limited state coverage in the source dataset. Our method achieves the best performance on 3 out of 6 target maps and ranks second on 2 additional maps. Specifically, LoDADA performs best on maps $\{2, 3, 4\}$ and achieves second-best performance on maps $\{5, 6\}$. As shown in Table 10, LoDADA achieves the best overall performance in the diverse dataset with a total score of 416.4, outperforming the second-best model by 8.4%. Our method achieves the best performance in 5 out of 6 maps and ranks second on the remaining one graph. Together, these results demonstrate the strong performance of our proposed LoDADA across different AntMaze environments, consistently outperforming baselines in both source dataset settings. They highlight the effectiveness of LoDADA in adapting to challenging navigation environments, while previous methods struggle to identify and utilize source data that are relevant to target navigation, leading to less consistent performance across different maze layouts.

E.4 Results on Adroit tasks

We report the performance of our method and baseline approaches on the Adroit manipulation benchmarks in Table 11. As shown in Table 11, our proposed LoDADA consistently outperforms all baseline methods across both the Pen and Door tasks. Overall, LoDADA achieves the highest total score of 402.24, yielding a substantial improvement over the second-best method (OTDF) with a total score of 330.41. This demonstrates the strong robustness of our approach under severe morphology and kinematic mismatches in real-world application. These results indicate that LoDADA generalizes well beyond navigation tasks and achieves strong performance on challenging manipulation benchmarks. By leveraging localized dynamics-aware filtering, our method is able to retain source transitions that align well with the target dynamics, while prior methods relying on global assumption struggle under morphology and kinematic shifts.

E.5 Results under Medium-Expert setting

The main experiments follow the standard ODRL setting, where both source and target datasets are collected by medium-level policies. We further examine whether LoDADA remains effective when the target dataset is replaced by an expert-level dataset. Specifically, we evaluate a medium-source and expert-target setting, where the source dataset remains medium-level while the target dataset is collected by an expert policy. This setting allows us to test whether the proposed filtering mechanism can still identify useful source transitions when the available target data comes from a stronger policy.

We evaluate this setting on HalfCheetah-Friction and Ant-Gravity across four shift levels. As shown in Table 12, LoDADA achieves the highest aggregate score of 328.60, improving over the strongest baseline IGDF with 267.82 by 22.7%. LoDADA obtains the best performance in 7 out of 8 settings. The gains are especially clear on Ant-Gravity, where LoDADA outperforms all baselines at every shift level. On HalfCheetah-Friction, LoDADA performs best in three out of four settings, while IGDF performs better on HalfCheetah-Friction-0.1. Overall, these results show that LoDADA remains effective when the target dataset is collected by an expert policy, suggesting that our localized filtering mechanism is not limited to the standard medium-level target-data setting.

E.6 Cluster-level sample coverage and classifier stability

LoDADA estimates the cluster-level KL divergence by training a classifier within each cluster. A natural concern is whether the limited target dataset provides enough samples for this estimation. We therefore check the per-cluster sample counts and classifier training behavior in two representative tasks: HalfCheetah-Friction-0.5 and Ant-Gravity-0.5. We use $K = 30$ and report three representative clusters for each task.

As shown in Table 13, the cluster sizes are imbalanced, but the examined clusters still contain samples from both domains. The target clusters contain 59 to 251 transitions, and the corresponding source clusters contain 1,431 to 82,304 transitions. Thus, the classifier-based density-ratio estimation is well-defined in these clusters.

To reduce the effect of source-target imbalance within a cluster, we train each cluster classifier with minibatch resampling with replacement. We set the maximum number of classifier-training epochs to 7000 and use early stopping with a patience of 15. In practice, early stopping is triggered after 25 to 54 epochs, far before the maximum epoch budget. The final training losses are also small, suggesting that the cluster classifiers can be optimized stably in these representative settings.

E.7 Sensitivity to filtering ratios

LoDADA uses fixed filtering ratios $(\xi_1, \xi_2, \xi_3) = (90\%, 80\%, 70\%)$ for low-, medium-, and high-KL clusters. These ratios are fixed across all environments and are not tuned for each task. We use this simple three-tier rule to keep enough source data while reducing the influence of clusters with larger estimated dynamics mismatch.

We study three more aggressive variants. The first variant sets $(\xi_1, \xi_2, \xi_3) = (90\%, 80\%, 0\%)$, which completely removes source samples from high-KL clusters. The second variant sets $(\xi_1, \xi_2, \xi_3) = (60\%, 40\%, 20\%)$, which reduces the retention ratios across all clusters. The third variant sets $(\xi_1, \xi_2, \xi_3) = (30\%, 20\%, 10\%)$, which keeps only a small fraction of source samples from all clusters. All other components are kept unchanged.

As shown in Table 14, aggressive filtering is not a reliable default. The hard-exclusion variant can perform slightly better in some Ant-Gravity settings, but it underperforms the default setting on HalfCheetah-Friction. The low-retention variant is worse in most settings, suggesting that keeping too little source data can hurt policy learning. These results suggest that the default soft filtering schedule provides a practical balance between mismatch filtering and state-space coverage, without requiring task-specific tuning of the filtering ratios.

For AntMaze, as shown in Table 15, the default setting achieves the highest average score of 70.40 and performs best on three out of four maps. Although more aggressive filtering slightly improves performance on Diverse map 3, it generally underperforms the default setting. These results suggest that retaining sufficient source data is important for preserving state-space coverage in AntMaze.

E.8 Component ablations and L2-distance filtering

We ablate three components of LoDADA . First, we set $\alpha = 0$ to remove KL-based critic weighting. Second, we set $\lambda = 0$ to remove the CVAE-based policy regularizer. Third, we replace the local discriminator-based filtering score with an L2-distance filtering score. All other components are unchanged.

As shown in Table 16, the full LoDADA achieves the best aggregate score of 316.42. Removing critic weighting reduces the total score to 297.36, and removing the CVAE regularizer further reduces it to 288.65. The L2 filtering variant obtains 298.71, which is competitive in some settings but worse than LoDADA overall. These results show that KL-based critic weighting, target-support regularization, and local discriminator-based filtering all contribute to the final performance.

E.9 Comparison with CVAE-regularized baselines

We further test whether the gain of LoDADA mainly comes from the CVAE-based policy regularizer. To study this factor, we add the same CVAE regularizer to three baselines: IQL, DARA, and IGDF. The original filtering or adaptation mechanism of each baseline is kept unchanged.

As shown in Table 17, adding CVAE regularization can improve some baselines in some settings, but the effect is not stable. For example, IQL+CVAE improves over IQL on HalfCheetah-Friction-0.1, but performs worse on HalfCheetah-Friction-0.5. Similarly, IGDF+CVAE improves over IGDF on Ant-Gravity-5.0, but performs worse on Ant-Gravity-2.0. In contrast, LoDADA achieves the best aggregate score of 316.42, outperforming the strongest CVAE-regularized baseline IQL+CVAE, which obtains 291.14. On Ant-Gravity, LoDADA also outperforms all CVAE-regularized baselines across all shift levels. These results show that CVAE regularization alone does not explain the

performance gain. The improvement comes from combining target-support regularization with localized dynamics-aware filtering.

E.10 Sensitivity analysis on AntMaze and Adroit

We further study the sensitivity of LoDADA to the number of clusters K and the policy regularization weight λ on AntMaze navigation and Adroit manipulation tasks. These environments differ substantially from MuJoCo locomotion: AntMaze has structured map geometry and sparse navigation objectives, while Adroit involves high-dimensional manipulation with morphology and kinematic shifts. Therefore, the best hyperparameters need not be identical across all task families.

For AntMaze, we evaluate smaller cluster numbers $K \in \{5, 10, 15\}$ and larger regularization weights $\lambda \in \{0.5, 1.0, 5.0\}$. As shown in Table 18, performance remains stable across the tested values of K, with all three choices achieving comparable average performance. The regularization weight has a larger effect: $\lambda = 1.0$ provides the most reliable performance across the representative maps, while $\lambda = 0.5$ is often too weak for the structured navigation setting.

For Adroit, we evaluate $K \in \{30, 50\}$ and $\lambda \in \{0.1, 1.0\}$. As shown in Table 19, both cluster numbers lead to comparable performance, and $\lambda = 1.0$ is slightly more stable on average. Overall, these results suggest that LoDADA only requires tuning within a compact task-dependent candidate set rather than a large continuous hyperparameter search.

E.11 Time comparison of data filtering methods

To understand the computational cost of LoDADA compared with other filtering methods, we compare its wall-clock runtime with IGDF [42] and OTDF [33]. We use the Ant-Gravity-0.5 environment and vary the number of source samples while keeping the target dataset fixed. We decompose the runtime into two stages. P1 denotes the pre-optimization stage, which includes the method-specific filtering or scoring procedure before policy optimization. For LoDADA , P1 includes K-means clustering, source-sample assignment, local discriminator training, KL estimation, and source-data filtering. Thus, the reported P1 runtime of LoDADA explicitly includes the cost of training the cluster-level domain classifiers. P2 denotes the downstream policy optimization stage after filtering or scoring.

For fairness, we modify the replay buffer so that it samples without replacement, meaning that it goes over the entire dataset in each epoch. We fix the batch size to 32 and the number of epochs to 7000. As shown in Table 20, IGDF has the smallest P1 runtime, while OTDF has the largest P1 runtime because it computes optimal transport costs between source and target samples. LoDADA is slower than IGDF in P1 due to local discriminator training, but is much faster than OTDF. In P2, the runtime is mainly determined by the downstream policy optimization pipeline and the additional regularization modules. Overall, IGDF is the fastest method but has weaker policy performance, while LoDADA provides a better balance between runtime and final performance.

E.12 Robustness to Target Data

When target data are extremely sparse or collected by a suboptimal behavior policy, estimating localized dynamics discrepancies and learning effective policies becomes more challenging. To evaluate the performance of LoDADA under these conditions, we conduct additional experiments that progressively reduce the amount of target data and replace the original medium-quality target datasets with lower-quality alternatives.

We evaluate LoDADA on locomotion tasks with Ant-Gravity and HalfCheetah-Friction environments. For the limited target data setting, we retain 70%, 50%, 20%, and 10% of the original target dataset. For the low-quality target data setting, we replace the medium target dataset with random-medium or random target data, using 5,000 transitions for each alternative dataset. The number of clusters K and regularization weight λ are tuned for these experiments following the hyperparameters in Table 6. We also adjust the diameter threshold δ to avoid removing all source data during the initial filtering stage when target data are limited.

Limited target data. Table 21 reports the results with progressively smaller target datasets. In most cases, LoDADA remains relatively stable when up to 50% of the target data are removed, with more noticeable performance degradation under extreme sparsity (10% and 20% of the original target

data remain). These results suggest that localized discrepancy estimation remains effective under moderately limited target coverage. An exception is Ant-Gravity-0.5, where performance does not show a consistent decreasing trend. We attribute this behavior to variability in experimental settings rather than to a systematic benefit from reducing the number of target samples.

Low-quality target data. We further evaluate LoDADA using target datasets of different quality levels. As shown in Table 22, performance generally decreases as target data quality decreases, with larger drops when using random data. This is because low-quality target data provide limited state-action coverage and weaker supervision for policy learning. This limitation also applies to other off-dynamics methods that rely on target data for adaptation. Overall, these results show that LoDADA is sensitive to target data quality and benefits from higher-quality target data.

Table 6: Hyperparameters of LoDADA and baselines.

Hyperparameter	Value
Shared	
Actor network	(256, 256)
Critic network	(256, 256)
Learning rate	3×10^{-4}
Optimizer	Adam
Discount factor	0.99
Replay buffer size	10^6
Nonlinearity	ReLU
Target update rate	5×10^{-3}
Source domain batch size	128
Target domain batch size	128
DARA	
Temperature coefficient	0.2
Maximum log std	2
Minimum log std	-20
Classifier network	(256, 256)
Reward penalty coefficient λ	0.1
BOSA	
Temperature coefficient	0.2
Maximum log std	2
Minimum log std	-20
Policy regularization coefficient λ_{policy}	0.1
Transition coefficient $\lambda_{\text{transition}}$	0.1
Threshold parameter ϵ, ϵ'	$\log(0.01)$
Value weight ω	0.1
CVAE ensemble size	1 (behavior), 5 (dynamics)
IGDF	
Representation dimension	{16, 64}
Contrastive encoder network	(256, 256)
Encoder pretraining steps	7000
Importance coefficient	1.0
Data selection ratio ξ	75%
OTDF	
CVAE training steps	10000
CVAE learning rate	0.001
Number of sampled latents M	10
Gaussian std	$\sqrt{0.1}$
Cost function	cosine
Policy coefficient β	0.5
Data selection ratio ξ	80%
IQL	
Temperature coefficient	0.2
Maximum log std	2
Minimum log std	-20
Inverse temperature β	3.0
Expectile τ	0.7
DmC	
Nearest-neighbor index k	4
Data selection ratio ξ	50%
Diffusion training steps	10^5
Guidance score percentile	90–99
Number of synthetic transitions	$\max(\lvert D_{\text{src}} \rvert, 10^6)$
Policy regularization weight λ	0.1
LoDADA	
Number of clusters K	$\{30, 50\}$ for MuJoCo/Adroit; $\{5, 10, 15, 20\}$ for AntMaze
Policy regularization weight λ	$\{0.1, 0.5, 1.0\}$ for MuJoCo/Adroit; $\{0.5, 1.0, 5.0\}$ for AntMaze
Filtering ratio (ξ_1, ξ_2, ξ_3)	$(90\%, 80\%, 70\%)$
Input noise std σ	1.0
Importance weight α	1.0
Diameter threshold δ	1.5
CVAE pretraining gradient steps	10000
CVAE learning rate	0.001

Table 7: Performance comparison on HalfCheetah, Ant, Walker2d, and Hopper environments under **morphology shifts** across 5 seeds. We use **M** and **H** to denote the medium and hard levels of the dynamics shift.

Env	Type	Level	BOSA	IQL	DARA	IGDF	OTDF	DmC	LoDADA
HalfCheetah	morph-thigh	M	25.12±4.33	22.22±1.13	20.68±2.49	19.31±4.35	21.94±6.30	17.65±2.90	**26.42±3.44**
		H	20.74±3.22	15.90±1.47	16.35±0.97	15.57±2.06	22.75±2.76	17.79±1.54	**25.81±5.29**
	morph-torso	M	0.66±0.90	1.95±1.69	1.40±1.66	1.84±0.37	3.38±1.30	2.34±0.52	**9.96±4.07**
		H	16.44±12.82	**30.87±10.00**	29.89±5.88	28.85±3.89	26.06±6.00	17.49±3.11	23.44±4.82
Ant	morph-halflegs	M	64.52±7.46	67.15±4.33	65.82±5.31	69.07±5.76	73.86±0.79	50.16±5.72	**77.48±0.77**
		H	37.21±6.15	54.27±5.60	61.31±5.71	60.08±2.09	59.76±9.27	52.06±3.14	**64.47±4.62**
	morph-alllegs	M	31.09±7.09	30.05±7.41	26.49±4.52	37.87±17.24	32.98±4.91	63.63±2.27	**77.55±3.44**
		H	13.89±1.46	5.65±3.30	3.21±3.26	7.24±1.40	12.84±2.06	11.19±2.47	**24.25±0.02**
Walker2d	morph-torso	M	8.63±3.49	11.44±0.91	14.80±1.69	14.16±2.19	11.47±4.93	21.12±11.43	**29.67±11.80**
		H	1.93±0.53	4.78±1.13	6.86±2.38	5.38±1.34	7.05±0.57	6.92±1.16	**7.76±2.42**
	morph-leg	M	28.10±8.64	40.61±4.78	43.87±6.22	41.71±10.14	45.07±10.09	50.61±5.59	**52.86±8.82**
		H	10.17±1.60	20.54±5.17	20.33±3.49	18.78±3.84	15.13±3.25	22.56±5.66	**59.01±1.76**
Hopper	morph-foot	M	12.87±0.09	**25.62±18.89**	20.05±7.32	14.91±4.52	22.56±8.65	13.06±0.05	13.11±0.05
		H	9.84±0.17	10.72±2.84	25.74±10.70	18.35±8.08	66.78±9.79	18.90±2.91	**67.53±14.03**
	morph-torso	M	13.43±0.37	13.66±0.98	14.70±4.48	13.06±4.33	16.84±1.44	14.27±0.71	**27.81±9.08**
		H	11.28±0.24	11.30±0.44	10.82±0.19	10.66±1.19	**12.23±0.15**	12.03±0.19	10.75±0.89
Total			305.92	366.73	382.32	376.84	450.70	391.78	**597.88**

Table 8: Performance comparison on MuJoCo tasks under a medium-level offline dataset. Target domains remain unchanged; source datasets are shifted by adding local perturbations with varying shift levels across regions. We report normalized target-domain scores (mean ± std over five seeds). Best and second-best scores are highlighted in green and blue, respectively.

Env	# Groups	Shift Level	BOSA	DARA	IQL	IGDF	OTDF	DmC	LoDADA
HalfCheetah	3	0.1-0.5-2.0	27.36±2.22	19.86±4.42	17.57±3.41	14.59±3.13	29.26±3.10	28.84±1.66	**34.97±0.85**
		0.2-1.0-4.0	25.28±3.10	16.94±3.31	16.07±3.45	13.16±4.83	24.21±2.82	22.50±2.78	**28.36±2.76**
		0.3-1.5-6.0	20.75±2.66	6.59±0.45	10.73±2.88	8.73±2.31	22.84±0.54	17.54±3.18	**27.24±4.37**
	5	0.1-0.2-0.3-0.4-0.5	19.30±2.38	12.43±1.95	10.83±1.05	11.27±2.35	28.99±1.85	24.73±4.46	**34.28±2.91**
		0.2-0.4-0.6-0.8-1.0	23.59±5.81	13.18±2.09	14.77±2.41	11.43±3.96	21.41±3.95	19.41±4.10	**25.73±5.38**
		0.3-0.6-0.9-1.2-1.5	**23.95±2.23**	7.10±0.97	8.98±2.61	7.37±3.03	11.53±2.04	18.22±6.33	18.74±1.55
Walker2d	3	0.1-0.5-2.0	5.28±0.54	17.85±10.67	36.44±15.19	14.02±10.20	34.52±11.33	13.45±7.94	**37.45±18.71**
		0.2-1.0-4.0	6.42±1.10	19.79±3.58	15.29±9.44	14.62±8.16	20.66±7.06	16.56±6.16	**31.10±18.00**
		0.3-1.5-6.0	7.16±2.91	9.86±4.44	12.28±8.14	11.14±3.69	16.78±9.39	13.18±5.16	**18.69±10.51**
	5	0.1-0.2-0.3-0.4-0.5	7.17±1.48	8.10±3.46	6.96±2.01	6.99±2.77	33.87±16.17	25.31±10.35	**35.84±16.01**
		0.2-0.4-0.6-0.8-1.0	6.51±4.60	13.90±9.23	12.07±7.55	9.37±2.01	13.87±6.95	16.72±6.93	**24.92±5.96**
		0.3-0.6-0.9-1.2-1.5	6.43±2.53	8.65±3.18	7.91±3.80	8.99±5.44	**18.52±15.46**	16.99±5.77	12.98±3.88
Hopper	3	0.1-0.5-2.0	7.84±2.84	8.70±6.67	8.14±6.31	13.08±10.44	33.96±4.57	30.61±10.01	**40.77±5.04**
		0.2-1.0-4.0	11.91±2.60	9.61±5.56	11.97±5.99	8.35±3.56	32.58±6.89	29.53±5.73	**37.51±9.16**
		0.3-1.5-6.0	11.85±3.19	8.11±3.54	10.17±5.50	6.59±2.59	31.77±3.04	27.87±5.27	**33.41±2.72**
	5	0.1-0.2-0.3-0.4-0.5	10.42±3.55	6.37±1.67	6.92±1.85	15.81±9.78	22.25±8.74	**40.35±8.19**	40.30±4.61
		0.2-0.4-0.6-0.8-1.0	11.63±1.87	14.07±5.65	9.82±4.29	11.24±7.86	21.42±9.11	34.36±4.82	**36.06±4.59**
		0.3-0.6-0.9-1.2-1.5	13.97±5.15	11.14±4.01	12.01±2.06	15.63±10.74	29.95±5.01	18.79±5.32	**35.02±5.70**
Ant	3	0.1-0.5-2.0	55.51±10.37	69.67±8.59	69.65±10.93	63.79±9.55	70.30±5.24	8.77±4.00	**85.28±6.52**
		0.2-1.0-4.0	31.75±10.39	54.02±5.97	59.71±7.84	55.91±7.68	54.55±6.35	16.44±6.55	**61.51±6.89**
		0.3-1.5-6.0	14.41±4.68	67.13±5.07	**69.03±9.18**	50.24±7.41	34.47±4.88	20.40±7.87	53.54±7.07
	5	0.1-0.2-0.3-0.4-0.5	44.36±8.50	75.03±7.08	66.50±8.95	63.40±10.26	58.51±10.29	36.23±12.33	**78.96±7.34**
		0.2-0.4-0.6-0.8-1.0	30.26±11.22	56.84±5.10	**67.07±4.40**	55.65±13.74	47.99±8.48	14.20±2.07	61.92±10.42
		0.3-0.6-0.9-1.2-1.5	13.14±5.58	**51.81±6.79**	46.98±9.14	51.36±8.39	15.23±4.79	10.84±2.38	48.51±9.77
Total			436.25	586.77	607.87	542.73	729.44	521.84	**943.09**

Table 9: Performance comparison on Antmaze tasks under the play offline dataset with dynamic shifts in the map. Source domains remain unchanged; target domains are shifted. We report normalized target-domain scores (mean ± std over five seeds). Best and second-best scores are highlighted in green and blue, respectively.

Env	Shift Level	BOSA	DARA	IQL	IGDF	OTDF	DmC	LoDADA
Antmaze Medium	1	33.6 ± 8.0	52.4 ± 7.8	**76.4±10.1**	58.0 ± 12.4	56.8 ± 6.87	25.2 ± 4.5	52.8 ± 9.4
	2	25.2 ± 4.6	29.6 ± 6.2	46.8 ± 4.1	42.4 ± 8.2	63.2 ± 7.95	24.0 ± 2.2	**67.6±6.7**
	3	30.0 ± 8.1	56.4 ± 11.1	66.4 ± 6.7	64.8 ± 7.0	69.6 ± 5.90	25.2 ± 6.5	**71.6±3.8**
	4	34.0 ± 8.9	52.8 ± 6.3	48.8 ± 6.9	58.0 ± 1.4	44.8 ± 3.90	20.0 ± 2.2	**71.6±6.8**
	5	22.8 ± 13.9	34.4 ± 9.3	**73.6±8.2**	54.4 ± 6.2	55.2 ± 3.03	24.4 ± 7.1	60.8 ± 7.2
	6	12.8 ± 4.8	62.8 ± 3.0	66.8 ± 8.1	**74.8±4.1**	70.8 ± 5.93	8.8 ± 2.0	74.4 ± 3.0
Total		158.4	288.4	378.8	352.4	360.4	127.6	**398.8**

Table 10: Performance comparison on AntMaze tasks under the diverse source dataset with dynamic shifts in the map. Source domains remain unchanged; target domains are shifted. We report normalized target-domain scores (mean ± std over five seeds) with the mean of 50 episodes. Best and second-best scores are highlighted in green and blue, respectively.

Env	Shift Level	BOSA	DARA	IQL	IGDF	OTDF	DmC	LoDADA
Antmaze Medium	1	43.6 ± 11.2	53.6 ± 9.9	**76.5±8.4**	60.4 ± 9.1	51.6 ± 10.53	33.6 ± 6.6	66.8 ± 4.6
	2	22.4 ± 5.5	33.6 ± 3.3	52.8 ± 7.9	45.2 ± 7.8	56.4 ± 4.34	31.2 ± 2.7	**69.2±8.9**
	3	39.6 ± 11.5	50.8 ± 3.0	57.6 ± 7.7	65.2 ± 6.6	71.6±5.90	25.6 ± 6.4	**73.2±7.3**
	4	40.0 ± 3.2	48.8 ± 8.3	58.4 ± 4.8	60.4 ± 6.2	52.4 ± 8.53	21.2 ± 7.0	**65.2±5.9**
	5	21.2 ± 7.6	41.6 ± 12.7	66.4±8.3	56.8 ± 7.6	57.2 ± 7.16	27.6 ± 7.9	**66.8±4.8**
	6	24.8 ± 14.0	59.2 ± 12.4	72.4 ± 10.7	73.6 ± 1.7	65.2 ± 9.86	15.2 ± 1.6	**75.2±6.7**
Total		191.6	287.6	384.1	361.6	354.4	154.4	**416.4**

Table 11: Performance comparison on the Door and Pen environments under **morphology shifts** and **kinematic shifts** across 5 seeds. We use **M** and **H** to denote the medium and hard levels of the dynamics shift.

Env	Type	Level	BOSA	IQL	DARA	IGDF	OTDF	DmC	LoDADA
Pen	kin-broken-joint	M	42.56±11.31	45.46±8.54	46.17±6.26	50.74±11.20	52.14±6.51	49.39±4.52	**64.31±11.33**
		H	19.36±5.42	24.02±5.16	22.41±4.73	18.62±10.47	20.95±7.23	21.64±4.47	**28.16±18.04**
	morph-shrink-finger	M	10.33±2.79	15.64±2.97	15.33±2.39	13.19±4.41	18.02±5.64	13.34±2.52	**19.88±8.02**
		H	19.47±5.01	19.40±5.05	18.24±2.59	19.05±3.87	20.34±2.85	18.78±3.41	**33.06±11.48**
Door	kin-broken-joint	M	20.65±10.72	40.17±7.10	36.82±4.63	37.76±7.02	39.82±4.75	37.65±3.96	**55.12±12.50**
		H	19.80±15.61	52.37±5.52	49.31±5.53	52.61±4.83	58.24±6.37	51.07±4.33	**59.81±7.23**
	morph-shrink-finger	M	41.05±15.36	54.85±7.79	54.49±5.88	54.73±5.28	59.15±4.89	60.59±5.23	**71.48±8.80**
		H	38.92±18.85	61.84±4.16	58.91±3.05	61.07±6.89	61.74±3.24	49.93±6.75	**70.42±4.26**
Total			212.15	313.75	301.67	307.78	330.41	302.39	**402.24**

Table 12: Performance under the medium-source and expert-target setting. The source dataset is collected by a medium-level behavior policy, while the target dataset is collected by an expert policy. We report normalized target-domain scores over five seeds.

Task	BOSA	IQL	DARA	IGDF	OTDF	LoDADA
HalfCheetah-Friction-0.1	8.27 ± 1.97	22.09 ± 13.30	20.98 ± 10.98	**32.63 ± 6.37**	9.29 ± 0.46	9.64 ± 0.35
HalfCheetah-Friction-0.5	67.24 ± 2.93	68.58 ± 1.05	68.54 ± 0.99	67.16 ± 1.50	66.01 ± 2.85	**68.81 ± 0.30**
HalfCheetah-Friction-2.0	44.18 ± 3.93	45.35 ± 1.46	44.54 ± 2.36	44.07 ± 1.65	43.83 ± 0.55	**45.97 ± 0.48**
HalfCheetah-Friction-5.0	23.22 ± 3.48	17.79 ± 5.20	20.88 ± 6.49	25.30 ± 6.28	22.04 ± 6.76	**32.55 ± 3.08**
Ant-Gravity-0.1	17.26 ± 4.07	12.37 ± 1.07	12.61 ± 0.79	13.96 ± 2.58	16.24 ± 3.25	**26.41 ± 3.75**
Ant-Gravity-0.5	12.21 ± 1.10	10.18 ± 1.04	10.14 ± 1.94	9.13 ± 2.00	12.54 ± 2.01	**24.42 ± 4.36**
Ant-Gravity-2.0	42.50 ± 2.24	37.30 ± 1.31	37.65 ± 2.64	44.30 ± 12.06	37.55 ± 4.44	**75.86 ± 6.85**
Ant-Gravity-5.0	31.74 ± 0.75	31.27 ± 0.57	30.99 ± 0.23	31.27 ± 0.71	35.27 ± 0.37	**44.94 ± 3.81**
Total	246.62	244.93	246.33	267.82	242.77	**328.60**

Table 13: Cluster-level sample coverage and classifier training diagnostics. We report representative clusters from HalfCheetah-Friction-0.5 and Ant-Gravity-0.5 with $K = 30$.

Task	Cluster index	Source samples	Target samples	Stopping epoch	Training loss
HalfCheetah-Friction-0.5	10	2,151	251	42	0.018
HalfCheetah-Friction-0.5	20	29,002	62	31	0.020
HalfCheetah-Friction-0.5	29	82,304	59	46	0.034
Ant-Gravity-0.5	10	2,362	188	54	0.066
Ant-Gravity-0.5	20	1,431	88	25	0.170
Ant-Gravity-0.5	29	32,329	196	38	0.070

Table 14: Sensitivity to filtering ratios. The default LoDADA uses fixed ratios $(\xi_1, \xi_2, \xi_3) = (90\%, 80\%, 70\%)$ across all environments.

Task	$(90\%, 80\%, 0\%)$	$(60\%, 40\%, 20\%)$	$(30\%, 20\%, 10\%)$	LoDADA
HalfCheetah-Friction-0.5	67.67 ± 0.42	67.30 ± 0.45	66.08 ± 2.35	**68.51 ± 0.44**
HalfCheetah-Friction-2.0	47.25 ± 0.50	46.74 ± 0.92	45.59 ± 1.56	**48.19 ± 0.75**
Ant-Gravity-0.5	**21.90 ± 1.19**	18.64 ± 1.79	19.39 ± 6.29	18.61 ± 2.79
Ant-Gravity-2.0	**46.68 ± 2.99**	44.18 ± 2.61	45.09 ± 3.25	45.82 ± 1.33

Table 15: Sensitivity to filtering ratios on AntMaze. We report normalized target-domain scores over five seeds. The default LoDADA uses fixed ratios $(\xi_1, \xi_2, \xi_3) = (90\%, 80\%, 70\%)$.

Task	$(90\%, 80\%, 0\%)$	$(60\%, 40\%, 20\%)$	$(30\%, 20\%, 10\%)$	LoDADA
Diverse map 2	62.80 ± 8.63	64.80 ± 3.92	65.20 ± 4.12	$\mathbf{69.20 \pm 8.90}$
Diverse map 3	69.60 ± 8.80	$\mathbf{73.60 \pm 4.08}$	$\mathbf{73.60 \pm 8.52}$	73.20 ± 7.30
Play map 2	65.60 ± 5.12	57.20 ± 6.88	63.60 ± 5.57	$\mathbf{67.60 \pm 6.70}$
Play map 3	69.20 ± 7.22	68.40 ± 7.09	66.40 ± 5.43	$\mathbf{71.60 \pm 3.80}$
Average	66.80	66.00	67.20	**70.40**

Table 16: Component ablations of LoDADA . We compare the full method with variants that remove KL-based critic weighting ($\alpha = 0$), remove CVAE policy regularization ($\lambda = 0$), or replace local discriminator-based filtering with L2-distance filtering.

Task	$\alpha = 0$	$\lambda = 0$	L2 filter	LoDADA
HalfCheetah-Friction-0.1	9.40 ± 0.82	9.91 ± 0.29	$\mathbf{10.02 \pm 0.78}$	9.97 ± 0.47
HalfCheetah-Friction-0.5	66.17 ± 0.58	$\mathbf{68.86 \pm 0.48}$	64.57 ± 0.69	68.51 ± 0.44
HalfCheetah-Friction-2.0	46.12 ± 0.40	48.03 ± 0.82	45.68 ± 0.50	$\mathbf{48.19 \pm 0.75}$
HalfCheetah-Friction-5.0	51.25 ± 5.11	54.36 ± 2.50	53.07 ± 6.14	$\mathbf{54.65 \pm 1.56}$
Ant-Gravity-0.1	10.15 ± 1.24	10.06 ± 0.23	10.20 ± 0.16	$\mathbf{22.38 \pm 1.28}$
Ant-Gravity-0.5	$\mathbf{19.35 \pm 2.64}$	12.58 ± 1.83	17.22 ± 4.50	18.61 ± 2.79
Ant-Gravity-2.0	45.31 ± 4.19	45.78 ± 1.94	45.45 ± 2.27	$\mathbf{45.82 \pm 1.33}$
Ant-Gravity-5.0	49.61 ± 7.01	39.07 ± 4.48	$\mathbf{52.50 \pm 6.59}$	48.29 ± 3.68
Total	297.36	288.65	298.71	**316.42**

Table 17: Comparison with CVAE-regularized baselines. We add the same CVAE-based policy regularizer used in LoDADA to IQL, DARA, and IGDF.

Task	IQL	IQL+CVAE	DARA	DARA+CVAE	IGDF	IGDF+CVAE	LoDADA
HalfCheetah-Friction-0.1	23.69 ± 16.46	$\mathbf{43.63 \pm 7.56}$	26.39 ± 11.35	40.18 ± 18.79	20.56 ± 11.22	30.66 ± 7.75	9.97 ± 0.47
HalfCheetah-Friction-0.5	68.26 ± 1.42	48.12 ± 10.15	68.41 ± 3.19	46.04 ± 12.35	66.30 ± 1.79	55.69 ± 1.44	$\mathbf{68.51 \pm 0.44}$
HalfCheetah-Friction-2.0	42.68 ± 2.64	$\mathbf{48.34 \pm 0.97}$	46.17 ± 1.72	45.79 ± 3.08	46.22 ± 1.27	46.65 ± 0.89	48.19 ± 0.75
HalfCheetah-Friction-5.0	48.97 ± 4.68	52.97 ± 1.56	45.31 ± 9.83	$\mathbf{55.74 \pm 0.69}$	54.75 ± 5.60	55.70 ± 8.17	54.65 ± 1.56
Ant-Gravity-0.1	11.03 ± 1.24	14.27 ± 1.53	12.53 ± 1.11	12.89 ± 0.85	13.55 ± 2.13	13.37 ± 0.65	$\mathbf{22.38 \pm 1.28}$
Ant-Gravity-0.5	10.53 ± 1.06	12.57 ± 2.04	9.08 ± 0.88	12.24 ± 0.97	11.34 ± 1.89	11.32 ± 0.87	$\mathbf{18.61 \pm 2.79}$
Ant-Gravity-2.0	38.26 ± 3.09	36.81 ± 2.17	38.54 ± 2.19	36.57 ± 2.25	41.59 ± 4.63	37.23 ± 0.74	$\mathbf{45.82 \pm 1.33}$
Ant-Gravity-5.0	31.51 ± 1.97	34.43 ± 1.00	32.51 ± 1.71	35.15 ± 2.17	34.44 ± 2.77	37.60 ± 4.10	$\mathbf{48.29 \pm 3.68}$
Total	274.93	291.14	278.94	284.60	288.75	288.22	**316.42**

Table 18: Hyperparameter sensitivity of LoDADA on AntMaze. We report normalized target-domain scores under different choices of the number of clusters K and the policy regularization weight λ.

Task	$K = 5$	$K = 10$	$K = 15$	$\lambda = 0.5$	$\lambda = 1.0$	$\lambda = 5.0$
Diverse map 2	64.80 ± 3.03	69.20 ± 8.90	63.20 ± 2.28	55.20 ± 8.32	69.20 ± 8.90	62.80 ± 8.07
Diverse map 3	73.20 ± 7.30	65.20 ± 7.56	69.60 ± 8.05	54.80 ± 5.40	73.20 ± 7.30	73.20 ± 7.56
Play map 2	69.60 ± 6.23	67.60 ± 6.70	64.40 ± 5.18	54.00 ± 5.10	67.60 ± 6.70	58.40 ± 13.37
Play map 3	68.80 ± 8.07	71.60 ± 3.80	72.80 ± 7.16	63.20 ± 5.40	71.60 ± 3.80	65.20 ± 10.06
Average	69.10	68.40	67.50	56.80	70.40	64.90

Table 19: Hyperparameter sensitivity of LoDADA on Adroit. We report normalized target-domain scores under different choices of K and λ.

Task	$K = 30$	$K = 50$	$\lambda = 0.1$	$\lambda = 1.0$
Pen morph-shrink (M)	19.88 ± 8.02	18.77 ± 4.22	16.75 ± 9.79	19.88 ± 8.02
Pen morph-shrink (H)	23.89 ± 13.84	33.06 ± 11.48	33.06 ± 11.48	22.76 ± 17.33
Door kin-broken (M)	45.08 ± 12.50	55.12 ± 12.50	42.73 ± 5.53	55.12 ± 12.50
Door kin-broken (H)	61.98 ± 10.07	49.77 ± 16.18	55.67 ± 8.30	61.98 ± 10.07
Average	37.71	39.18	37.05	39.94

Table 20: Runtime comparison in Ant-Gravity-0.5 with different numbers of source samples. P1 denotes the pre-optimization filtering or scoring stage, and P2 denotes the downstream policy optimization stage. For LoDADA , P1 includes clustering, source-sample assignment, local discriminator training, KL estimation, and filtering.

# Sources	IGDF-P1	IGDF-P2	OTDF-P1	OTDF-P2	LoDADA -P1	LoDADA -P2
1000000	41.49s	4570.07s	4597.82s	5988.57s	134.62s	6072.26s
100000	38.71s	4878.31s	628.16s	7671.25s	80.18s	3955.51s
10000	55.64s	4706.42s	126.39s	6384.07s	72.92s	3806.96s

Table 21: Performance of LoDADA with different amounts of target data. Results are reported as mean $\pm$ standard deviation over five seeds.

Task	All target data	70%	50%	20%	10%
HalfCheetah-Friction-0.1	$\mathbf{9.97 \pm 0.47}$	9.64 ± 0.85	9.65 ± 0.30	9.60 ± 0.20	8.79 ± 0.50
HalfCheetah-Friction-0.5	$\mathbf{68.51 \pm 0.44}$	65.50 ± 0.62	64.51 ± 1.18	60.16 ± 1.75	55.80 ± 0.72
HalfCheetah-Friction-2.0	$\mathbf{48.19 \pm 0.75}$	47.05 ± 0.29	46.65 ± 0.41	44.52 ± 0.63	41.93 ± 2.69
HalfCheetah-Friction-5.0	$\mathbf{54.65 \pm 1.56}$	47.43 ± 5.28	49.23 ± 4.36	42.94 ± 4.51	30.15 ± 13.81
Ant-Gravity-0.1	22.38 ± 1.28	13.77 ± 10.39	$\mathbf{22.75 \pm 3.90}$	10.69 ± 1.27	8.85 ± 0.17
Ant-Gravity-0.5	18.61 ± 2.79	21.96 ± 4.40	21.83 ± 2.42	$\mathbf{23.57 \pm 5.47}$	21.53 ± 4.92
Ant-Gravity-2.0	$\mathbf{45.82 \pm 1.33}$	44.89 ± 1.04	42.75 ± 2.50	37.60 ± 3.46	33.60 ± 0.84
Ant-Gravity-5.0	$\mathbf{48.29 \pm 3.68}$	42.60 ± 6.72	36.02 ± 2.40	34.82 ± 0.41	37.00 ± 6.50

Table 22: Performance of LoDADA with target datasets of different quality levels. Results are reported as mean $\pm$ standard deviation over five seeds.

Task	Medium target	Random-medium target	Random target
HalfCheetah-Friction-0.1	$\mathbf{9.97 \pm 0.47}$	8.5 ± 0.7	4.0 ± 0.8
HalfCheetah-Friction-0.5	$\mathbf{68.51 \pm 0.44}$	66.7 ± 1.2	62.0 ± 2.3
HalfCheetah-Friction-2.0	$\mathbf{48.19 \pm 0.75}$	44.7 ± 1.0	40.2 ± 2.0
HalfCheetah-Friction-5.0	$\mathbf{54.65 \pm 1.56}$	21.8 ± 3.0	11.0 ± 2.0
Ant-Gravity-0.1	$\mathbf{22.38 \pm 1.28}$	10.4 ± 0.1	9.9 ± 0.1
Ant-Gravity-0.5	$\mathbf{18.61 \pm 2.79}$	8.9 ± 0.5	7.0 ± 0.4
Ant-Gravity-2.0	$\mathbf{45.82 \pm 1.33}$	34.8 ± 1.1	23.6 ± 4.2
Ant-Gravity-5.0	$\mathbf{48.29 \pm 3.68}$	30.5 ± 0.6	2.1 ± 0.7

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The claims are supported by the theory and experiments in the main text, with additional experimental results in the appendix.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: We discuss limitations in the main text, including dependence on target data coverage, quality and source-target overlap.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes].

Justification: The assumptions and theoretical results are stated in the main text, and the complete proofs and supporting lemmas are provided in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes].

Justification: The main text describes the benchmarks, evaluation protocol, and main experimental setup, while the appendix provides datasets, hyperparameters, baselines, and implementation details.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: We use public benchmark datasets and provide a code package with instructions to reproduce the main experiments. Additional implementation details are provided in the appendix.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: The main text describes the training and evaluation setup, and the appendix provides dataset details, hyperparameters, and baseline implementation details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

   Justification: We report mean and standard deviation over five seeds for the main experimental results and most additional experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: We report the GPU, CPU, memory, and storage requirements in Appendix D.7, and provide runtime comparisons in Appendix E.10.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes].

   Justification: The work uses public offline RL benchmarks and simulation environments, and does not involve human subjects, private data, or high-risk deployment.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A].

    Justification: This is an algorithmic study evaluated on simulated offline RL benchmarks and does not introduce a deployed system or high-risk application.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: The paper does not release high-risk models, scraped datasets or generative systems that require special safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: The existing datasets, benchmarks, and baseline methods used in the paper are cited in the main text and appendix.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A].

    Justification: The paper does not introduce or release new datasets, models, or other new research assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The paper does not involve human subjects, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: The core method, experiments, and theoretical analysis do not use LLMs as a methodological component.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.