Debiased Offline Representation Learning for Fast Online Adaptation in Non-stationary Dynamics

Xinyu Zhang [*1 2] **Wenjie Qiu** [*1 2] **Yi-Chen Li** [*1 2 3] **Lei Yuan** [1 2 3] **Chengxing Jia** [1 2 3]
Zongzhang Zhang [1 2] **Yang Yu** [1 2 3]

Abstract

Developing policies that can adapt to non-stationary environments is essential for real-world reinforcement learning applications. Nevertheless, learning such adaptable policies in offline settings, with only a limited set of pre-collected trajectories, presents significant challenges. A key difficulty arises because the limited offline data makes it hard for the context encoder to differentiate between changes in the environment dynamics and shifts in the behavior policy, often leading to context misassociations. To address this issue, we introduce a novel approach called **D**ebiased **O**ffline **R**epresentation learning for fast online **A**daptation (**DORA**). DORA incorporates an information bottleneck principle that maximizes mutual information between the dynamics encoding and the environmental data, while minimizing mutual information between the dynamics encoding and the actions of the behavior policy. We present a practical implementation of DORA, leveraging tractable bounds of the information bottleneck principle. Our experimental evaluation across six benchmark MuJoCo tasks with variable parameters demonstrates that DORA not only achieves a more precise dynamics encoding but also significantly outperforms existing baselines in terms of performance.

1. Introduction

In Reinforcement Learning (RL), the agent attempts to find an optimal policy that maximizes the cumulative reward obtained from the environment (Sutton & Barto, 2018). However, online RL training typically requires millions of interaction steps (Kaiser et al., 2020; Wu & Zhang, 2023), posing a risk to safety and cost constraints in real-world scenarios (Garcıa & Fernández, 2015). Different from online RL, offline RL (Fujimoto et al., 2019; Wu et al., 2019; Kumar et al., 2020; Ran et al., 2023) aims to learn an optimal policy, exclusively from datasets pre-collected by certain behavior policies, without further interactions with the environment. Showing great promise in turning datasets into powerful decision-making machines, offline RL has attracted wide attention recently (Levine et al., 2020).

Previous offline RL methods commonly assume that the learned policy will be deployed to environments with stationary dynamics (Kumar et al., 2020), while unavoidable perturbations in real world will lead to *non-stationary* dynamics (Choi, 2000). For example, the friction coefficient of the ground changes frequently when a robotic vacuum walks on the floor covered by different surfaces. Policies trained for stationary dynamics will be unable to deal with non-stationary ones since the optimal behaviors depend on the certain dynamic. Nevertheless, training an adaptable policy from offline datasets for online adaptation in non-stationary dynamics is overlooked in current RL studies. In this paper, we focus on the setting where the environment dynamics, such as gravity or damping of the controlled robot, change unpredictably within an episode.

Offline Meta Reinforcement Learning (OMRL), which trains a generalizable meta policy from a multi-task dataset generated in different dynamics, offers a potential solution to handle non-stationary dynamics. Among existing OMRL methods, the gradient-based approaches (Lin et al., 2022b) may not be ideal solutions, as extra gradient updates are likely to cause drastic performance degradation due to the instability of policy gradient methods (Haarnoja et al., 2018). In comparison, context-based OMRL methods (Yuan & Lu, 2022; Li et al., 2021) extract task-discriminative information from trajectories by learning a context encoder. However, these methods still face severe challenges when encountering non-stationary dynamics. Firstly, the representations from the learned encoder may exhibit a biased correlation with the data-collecting behavior policy (Yuan & Lu, 2022).

[*]Equal contribution [1]National Key Laboratory for Novel Software Technology, Nanjing University, China [2]School of Artificial Intelligence, Nanjing University, China [3]Polixir Technologies. Correspondence to: Zongzhang Zhang <zzzhang@nju.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

As a consequence, it will fail to correctly identify the environment dynamics when the learned policy, instead of the behavior policy, is used to collect context during online adaptation. Secondly, existing OMRL methods generally require collecting short trajectories before evaluation during meta-testing phrase (Li et al., 2021), which is not allowed in non-stationary dynamics because the changes of dynamics are unknown to the agent.

To tackle the aforementioned issues, we propose **DORA** (abbreviation of **D**ebiased **O**ffline **R**epresentation learning for fast online **A**daptation). DORA employs a context encoder that uses the most recent state-action pairs to infer the current dynamics. In order to learn a debiased task representation, we respectively derive a lower bound to maximize the mutual information between the dynamics encoding and environmental data, and an upper bound to minimize mutual information between the dynamics encoding and the actions of the behavior policy, following the Information Bottleneck (IB) principle. Specifically, the lower bound is derived to urge the encoder to capture the task-relevant information using InfoNCE (Oord et al., 2018). And the upper bound is for debiasing representations from behavior policy, formalized as the Kullback-Leibler (KL) divergence between the representations with and without behavior policy information of each timestep. The contextual policy is then trained by an offline RL algorithm, such as CQL (Kumar et al., 2020). Experiment results on six MuJoCo tasks with three different changing parameters demonstrate that DORA remarkably outperforms existing OMRL baselines. Additionally, we illustrate that the learned encoder can swiftly identify and adapt to frequent changes in environment dynamics. We summarize our main contributions as below:

- We propose an offline representation learning method for non-stationary dynamics, which enables fast adaptation to tasks with frequent dynamics changes.
- We derive a novel objective for offline meta learning, which trains the encoder to reduce the interference of the behavior policy to correctly identify dynamics.
- Experimentally, our method achieves better performance in unseen dynamics on-the-fly without pre-collecting trajectories, compared with baselines.

2. Preliminaries

Reinforcement Learning We consider the infinite-horizon Markov Decision Process (MDP) (Sutton & Barto, 2018), which can be formulated as a tuple $M = \langle \mathcal{S}, \mathcal{A}, P, r, d_0, \gamma \rangle$. Here, $\mathcal{S}$ and $\mathcal{A}$ represent the state space and action space, respectively. Let Δ_X be the set of probability measures over any space X. We use $P : \mathcal{S} \times \mathcal{A} \to \Delta_{\mathcal{S}}$ to denote the environment transition function, or equivalently, dynamics[1]. $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function. d_0 denotes the distribution of initial states. $\gamma \in [0, 1)$ is the discount factor. The discounted cumulative reward starting from time step t is defined as $G_t = \sum_{k=0}^{\infty} \gamma^k r\left(s_{t+k}, a_{t+k}\right)$, where G_0 is known as return. The policy $\pi : \mathcal{S} \to \Delta_{\mathcal{A}}$ specifies a distribution over the action space $\mathcal{A}$, given any state $s \in \mathcal{S}$. RL aims to find an optimal policy π^* which maximizes the expected return $J_M(\pi)$, i.e., $\pi^* \in \arg\max_\pi J_M(\pi) := \mathbb{E}[G_0]$. When executing the policy $\pi(a|s)$ starting from s_t, the value function can be formulated as $V^\pi(s) = \mathbb{E}_\pi\left[G_0 | s_0 = s\right]$. Similarly, The action value function is $Q^\pi(s, a) = \mathbb{E}_\pi\left[G_0 \mid s_0 = s, a_0 = a\right]$.

Context-based OMRL OMRL combines offline RL (Kumar et al., 2020) and meta RL (Rakelly et al., 2019), aiming to learn a meta policy from offline datasets collected from multiple tasks. The goal of OMRL is to learn a meta policy π_θ with θ denoting the learnable parameters, which can generalize to unseen tasks not appearing in the training dataset. The N tasks for meta training $\mathcal{M} = \{M_i\}_{i=1}^N$ are sampled from the same task distribution P_{train}. We assume that all tasks share the same state space and action space, with the only difference lying in dynamics. For each task M_i, we have an offline dataset D_i of trajectories collected by an unknown behavior policy π_i^b. For simplicity, we may omit the subscript i. In this paper, we use $\tau_{t-H:t} = \{a_{t-H}, s_{t-H+1} \ldots, a_{t-1}, s_t\}$ to denote a short offline trajectory of length H for dynamics identification. During OMRL training, the meta policy π_θ is trained on all the offline datasets. In the testing phase, the agent encounters unseen tasks over P_{test}, and the meta policy needs to generalize to these tasks with limited interactions. As mentioned before, the objective of OMRL is to maximize the expected return in any task sampled from P_{test}, i.e.,

$$\max_\theta J(\pi_\theta) := \mathbb{E}_{M \sim P_{\text{test}}}\left[J_M\left(\pi_\theta\right)\right]. \tag{1}$$

Context-based OMRL utilizes an encoder p_ϕ, where ϕ is the learnable parameters, to output a m-dimensional task representation $z \in \mathbb{R}^m$. In experiments shown in Section 5, we set $m = 2$. The task representation could be regarded as auxiliary information to help the offline policy π_θ to identify the current testing task. For clarity, we use $\pi_\theta(a|s, z)$, $V^\pi(s, z)$, $Q^\pi(s, a, z)$ thereafter to respectively denote the meta policy, meta value function, and meta action value function which takes the task representation z as an additional input. Compared with the online context-based encoder, the offline-trained encoder is more susceptible to behavior policies due to the limited state and action space coverage of the datasets, which is detrimental in context-based OMRL.

[1] In this paper, we use "transition function" and "dynamics" interchangeably.

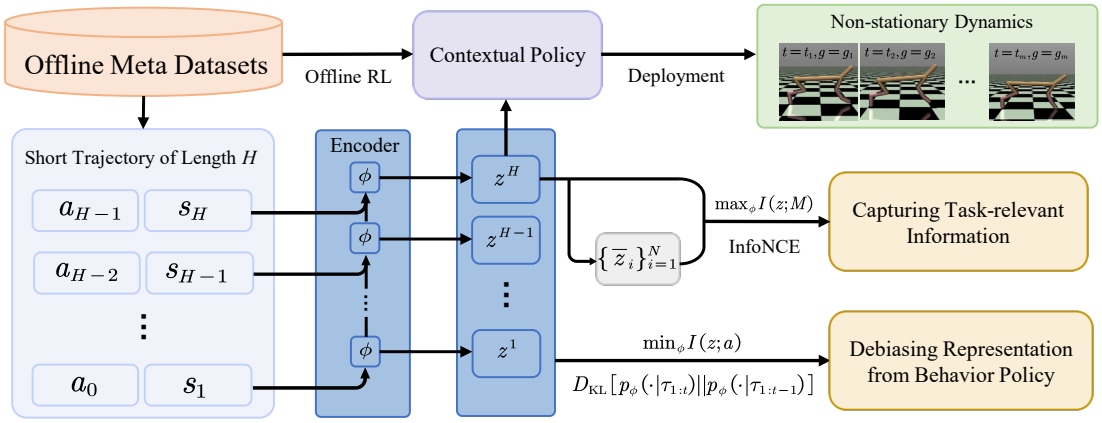


Figure 1. The DORA framework. The encoder utilizes recent state-action pairs to maintain a set of representations $\{z^1, \cdots, z^H\}$ and z^H updates the moving average task encodings $\{\bar{z}_i\}_{i=1}^N$. All these representations are then used to optimize the encoder. The contextual policy is trained through offline RL on the datasets, where each transition is labeled with its representation by the learned encoder.

Information Bottleneck In information theory, the IB concept involves extracting essential information by constraining the information flow, thereby finding a delicate equilibrium between preserving target-related information and achieving efficient compression (Tishby et al., 2000). This problem can be framed as the minimization of mutual information I between the source random variable X and its representation $\hat{X}$ for efficient compression, alongside the maximization of mutual information between $\hat{X}$ and the target Y to retain target-related information (Tishby & Zaslavsky, 2015). The pursuit of an optimal $\hat{X}$ is cast as the following maximization problem:

$$\max_{\hat{X}} I(\hat{X};Y) - \beta I(X;\hat{X}). \tag{2}$$

Here, $I(\hat{X};Y) = \int\int p(\hat{x},y)\log\left(\frac{p(\hat{x},y)}{p(\hat{x})p(y)}\right) d\hat{x}\, dy$, where $p(\hat{x},y)$ is the joint probability density function, and $p(\hat{x})$ and $p(y)$ are marginal probability density functions. The hyper-parameter β represents the tradeoff between compression and fitting. IB enables the context encoder to learn abstract and high-level representations, which helps to improve the generalization ability of the meta policy on unseen tasks (Sohn et al., 2015; Chen et al., 2018).

3. Method

In this section, we present the DORA framework, which learns a dynamics-sensitive trajectory encoder that mitigates biases from the behavior policy and effectively adapts to non-stationary dynamics. In Section 3.1, we adhere to the information bottleneck principle to formulate the objective for offline meta-encoder learning. Subsequently, in Section 3.2 and Section 3.3, we convert IB into two tractable losses. Section 3.4 elaborates on the proposed DORA framework (see Figure 1) including its implementation details.

3.1. From IB to Debiased Representations

Due to the fact that the offline dataset, generated by fixed behavior policies, exhibits limited coverage of the state and action space, the encoder tends to inaccurately treat the behavior policy as a prominent feature for task identification. Such a predicament can result in the decline of the performance of the contextual policy, given its dependency on the encoder's responsiveness to shifts in environmental dynamics. Thus it is crucial for the encoder to discern tasks based on the dynamics rather than the behavior policy.

We follow the IB principle to tackle this issue. Specifically, our approach involves maximizing the mutual information between representations and tasks to encapsulate dynamics-relevant information, while simultaneously minimizing the mutual information between representations and actions of behavior policies to alleviate biases stemming from the behavior policy π^b. Given a training task $M \sim P_{\text{train}}$ and the offline trajectory τ collected on M with trajectory length H, this idea can be formulated as:

$$\max_\phi I(z;M) - \beta I(z;a), \tag{3}$$

where $z \sim p_\phi(\cdot|\tau)$, $a \sim \pi^b(\cdot|s)$ and $s \in \mathcal{S}$.

3.2. Contrastive Dynamics Representation Learning

The first term in Equation (3), $I(z;M)$, is used for the learned representation to effectively encapsulate dynamics-relevant information. Since directly optimizing $I(z;M)$ is intractable, we derive a lower bound of it as shown in Theorem 3.1, which formulates the problem of maximizing $I(z;M)$ as a contrastive learning objective. By training the encoder to distinguish between positive and negative samples, the encoder acquires the capability to generate rep-

resentations that correctly embed the dynamics information.

Theorem 3.1. *Denote a set of N tasks as $\mathcal{M}$, in which each task M_i is sampled from the same training task distribution P_{train}. Let random variables $M \in \mathcal{M}$, τ be a trajectory collected in M, $z \sim p_\phi(\cdot|\tau)$, and $p(z)$ be the prior distribution of z, then we have*

$$I(z;M) \geq \mathbb{E}_{M,\tau,z}\left[\log \frac{\frac{p_\phi(z|\tau)}{p(z)}}{\sum_{M_i\in\mathcal{M}} \frac{p_\phi(z|\tau^i)}{p(z)}}\right] + \log N,$$

where τ^i is a trajectory collected in task M_i, $z_i \sim p_\phi(\cdot|\tau^i)$, and $i \in \{1, 2, \cdots, N\}$.

Due to space limitations, we defer the proof of Theorem 3.1 to Appendix A.1. In practice, to approximate $p_\phi(z|\tau)/p(z)$, we choose the radius basis function $S(z_i, z_j) = \exp(-||z_i - z_j||^2/r)$, which measures the similarity between the two representations z_i and z_j (Mnih & Teh, 2012). More details about this approximation can be found in Appendix A.2. Here, r is a hyper-parameter. We then obtain the distortion loss $\mathcal{L}_{\text{Dist}}$ as follows:

$$\mathcal{L}_{\text{Dist}}(\phi) = -\sum_{\substack{M_i\in\mathcal{M},\\ \tau^i\in D_i}}\left[\log\left(\frac{S(z_i,\bar{z}_i)}{\sum_{j=1}^{N} S(z_i,\bar{z}_j)}\right)\right], \quad (4)$$

where z_i is encoded by the encoder p_ϕ from trajectory τ^i. $\bar{z}_i$ denotes the average task representation for task i, updated by $\bar{z}_i \leftarrow \lambda z_i + (1-\lambda)\bar{z}_i$, where $\lambda \in (0, 1)$ is a hyper-parameter. Intuitively, by grouping trajectories from the same task while distinguishing those from different tasks, the distortion loss will help the encoder to extract dynamics-relevant information.

3.3. Debiasing Representation from Behavior Policy

Ideally, the learned representation should be minimally influenced by the behavior policy. Hence, as shown in the second term in Equation (3), we minimize the mutual information $I(z;a)$ between representations and actions of behavior policy. However, it is intractable to precisely calculate $I(z;a)$. We thus consider deriving an upper bound of it. Firstly, we have the following theorem.

Theorem 3.2. *Given a training task $M \in \mathcal{M}$, a trajectory τ collected in M, and $z \sim p_\phi(\cdot|\tau)$, we have*

$$I(z;a) \leqslant \mathbb{E}_a\left[D_{\text{KL}}\left[p(\cdot \mid a)\,\|t(\cdot)\right]\right], \quad (5)$$

where $t(z)$ is an arbitrary distribution over $\mathbb{R}^m$ and D_{KL} is the Kullback-Leibler divergence (Kullback & Leibler, 1951).

The proof of Theorem 3.2 is deferred to Appendix A.1. To convert Equation (5) into a tractable optimization objective, we choose $p_\phi(\cdot|\tau_{1:t-1})$ to be the prior distribution $t(z)$ at $t-1$ in practice. Since $\tau_{1:t}$ has two additional terms (a_{t-1} and s_t) compared with $\tau_{1:t-1}$, we surprisingly find that $p(\cdot|a)$ in Equation (5) can be cast to $p_\phi(\cdot|\tau_{1:t})$. This is because a_{t-1} generated by $\pi^b(\cdot|s_{t-1})$ carries the information of the behavior policy. Besides, s_t, which is sampled from $P(\cdot|s_{t-1}, a_{t-1})$, is independent of the behavior policy. We thus get the following debias loss $\mathcal{L}_{\text{Debias}}$:

$$\mathcal{L}_{\text{Debias}}(\phi) = \sum_{\substack{M_i\in\mathcal{M},\\ \tau^i_{1:H}\in D_i,\\ t\in\{2,\cdots,H\}}} D_{\text{KL}}\left[p_\phi(\cdot|\tau^i_{1:t})\|p_\phi(\cdot|\tau^i_{1:t-1})\right], \quad (6)$$

For task M_i, we can get a trajectory segment $\tau_{1:H}$ with length H given a full trajectory randomly sampled from D_i.

Algorithm 1 DORA Training

Input: context encoder p_ϕ, offline datasets $\{D_i\}_{i=1}^N$, short trajectory length H, batch size U, contextual policy π_θ, contextual action function Q_ψ, learning rate α
// Context Encoder Training
for step = 1, 2, $\cdots$ **do**
  for $u = 1$ to U **do**
    Randomly sample a task $i \in \{1, 2, \cdots, N\}$
    Sample a trajectory τ^u with length H from D_i randomly
  end for
  Let $\mathcal{B}_1 = \{\tau^u\}_{u=1}^U$
  Infer representation z_u from τ^u with p_ϕ, $\forall \tau^u \in \mathcal{B}_1$
  Update the moving average task representation $\{\bar{z}_i\}_{i=1}^N$
  Get encoder loss $\mathcal{L}_{\text{DORA}}(\phi)$ from Equation (7)
  Update encoder p_ϕ with $\phi \leftarrow \phi - \alpha\nabla_\phi\mathcal{L}_{\text{DORA}}(\phi)$
end for
// Policy Training
for step = 1, 2, $\cdots$ **do**
  Randomly get a minibatch $\mathcal{B}_2 = \{\tau^u\}_{u=1}^U$
  Infer representation z_u from τ^u with p_ϕ, $\forall \tau^u \in \mathcal{B}_2$
  Augment the states of minibatch $\mathcal{B}_2$ with $\{z_u\}_{u=1}^U$
  Update π_θ and Q_ψ with CQL (Kumar et al., 2020) using $\{z_u\}_{u=1}^U$ on $\mathcal{B}_2$
end for

3.4. A Practical Implementation

We take the Recurrent Neural Network (RNN) as the backbone of the encoder (Cho et al., 2014). The encoder receives a historical sequence of state-action pairs to generate the representation. By restricting the length of the RNN, we can adjust the amount of historical trajectory information retained in the encoder (Luo et al., 2022), which is helpful when the dynamic changes. Specifically, let H be the history length, then the representation at step t is denoted as $z_i^t \sim p_\phi(\cdot|\tau_{t-H:t})$, where $\tau_{t-H:t} = \{a_{t-H}, s_{t-H+1}, \ldots, a_{t-1}, s_t\}$. If the history length is less

than H, we will use the zero padding trick (Cho et al., 2014). The overall architecture is shown in Figure 1.

During the meta-training process, the encoder p_ϕ is updated using the following loss:

$$\mathcal{L}_{\text{DORA}}(\phi) = \mathcal{L}_{\text{Dist}}(\phi) + \beta \mathcal{L}_{\text{Debias}}(\phi). \tag{7}$$

After the encoder is trained to convergence, we train the meta policy via CQL (Kumar et al., 2020). During meta-testing, the encoder infers the task representation z based on the trajectories collected up to date. At each step, the current representation z and state s are together input into the meta policy, yielding a policy adapted to the current dynamics. Different from existing approaches (Yuan & Lu, 2022; Li et al., 2021), our approach does not require to pre-collect trajectories before policy evaluation, which achieves fast online adaptation on-the-fly. To sum up, the pseudocodes of training and testing are illustrated in Algorithm 1 and Appendix B, respectively. We release the code at Github[2].

4. Related Work

In this section, we introduce works related to our framework. Section 4.1 provides an overview of related works in offline RL. Section 4.2 shows how online RL methods handle non-stationary dynamics. Finally, Section 4.3 delves into the realm of OMRL, exploring relevant research and focusing on the challenges and strategies in this domain.

4.1. Offline RL

Offline RL learns policies from the dataset generated by certain behavior policies without online interaction with the environment, which helps avoid the safety concerns of online RL. The key challenge of offline RL lies in the distribution shift problem (Levine et al., 2020), thus algorithms in this domain typically impose constraints on the training policy to keep it close to the behavior policy. Many model-free approaches (Fujimoto et al., 2019; Wu et al., 2019; Kumar et al., 2020; Ghosh et al., 2022; Ran et al., 2023) introduce regularization terms by considering the divergence between the learned policy and the behavior policy to constrain policy deviation. CQL (Kumar et al., 2020) directly learns a conservative action value function to alleviate the overestimation problem. Model-based methods (Yu et al., 2020; Kidambi et al., 2020; Yu et al., 2021; Swazinna et al., 2022; Jia et al., 2024a) pre-train an environment model from the dataset, then use the model to generate out-of-dataset predictions for state-action transitions.

4.2. RL in Non-stationary Dynamics

To overcome the unavoidable perturbations in real world, recent years have seen an increasing number of research into RL in non-stationary dynamics, including continual RL and meta RL methods. Continual learning approaches aim to enable agents to continuously learn when faced with unseen tasks and avoid forgetting old tasks (Khetarpal et al., 2022). Policy consolidation (Kaplanis et al., 2019) integrates the policy network with a cascade of hidden networks and uses history to regularize the current policy to handle changing dynamics. The meta learning methods (Nagabandi et al., 2019), including gradient-based methods and context-based methods, adapt to new tasks quickly by leveraging experience from training tasks. Specifically, gradient-based meta RL methods (Finn et al., 2017) update the policy with gradients in the testing tasks, which are not suitable for non-stationary dynamics since real-world tasks may not allow for extra updates of the gradient. Context-based meta RL infers task-relevant information by learning a context encoder. In these methods, PEARL (Rakelly et al., 2019) employs variational inference to learn a context encoder. ESCP (Luo et al., 2022) utilizes an RNN-based context encoder to rapidly perceive changes in dynamics, enabling fast adaptation to new environments.

However, both continual RL and online context-based RL methods require online interactions, which are infeasible in the offline setting. As far as we know, there is currently no method capable of learning an effective adaptive policy in non-stationary dynamics in the paradigm of offline RL.

4.3. OMRL

OMRL extends meta RL to the paradigm of offline setting, aiming to generalize from experience of training tasks and facilitate efficient adaptation to unseen testing tasks. Gradient-based OMRL methods (Lin et al., 2022b) require a few interactions to adapt to unseen tasks, but the costly online gradient updates may not be allowed in the real world. In comparison, context-based OMRL methods employ a context encoder for task inference, showing potential for faster adaptation. FOCAL (Li et al., 2021) uses distance metric learning loss to distinguish different tasks. Recently, the transformer-based approaches (Xu et al., 2022; Lin et al., 2022a) leverage Transformer networks for autoregressive training on blended offline data from multiple tasks, and construct policy by planning or setting return-to-go. As mentioned in our work, another key challenge for OMRL is that the encoder is required to accurately identify environment dynamics while debiasing representations from behavior policy. To alleviate this problem, CORRO (Yuan & Lu, 2022) employs contrastive learning to train context encoders and trains dynamic models to generate new samples. Nonetheless, these works still suffer from the entanglement of task dynamics and behavior policy experimentally. Besides, existing works need to pre-collect trajectories before evaluation in meta-testing phase (Gao et al., 2023; Zhou et al., 2024; Jia et al., 2024b), which are not able to handle

[2]https://github.com/Xinyuz26/DORA

Table 1. Performance on the MuJoCo tasks in **stationary dynamics**. **Top:** Average normalized return ± standard deviation over 5 random seeds in testing tasks with **IID dynamics**. **Bottom:** Average normalized return ± standard deviation over 5 random seeds in testing tasks with **OOD dynamics**.

Environment	Offline ESCP	FOCAL	CORRO	Prompt-DT	**DORA (Ours)**
Cheetah-gravity	78.22 ± 21.13	51.99 ± 10.44	56.70 ± 15.48	49.59 ± 18.44	**86.31** ± 16.45
Pendulum-gravity	96.55 ± 17.24	74.95 ± 12.55	76.82 ± 11.39	34.71 ± 15.95	**100.08** ± 0.01
Walker-gravity	52.82 ± 21.01	17.22 ± 12.05	44.49 ± 24.67	26.09 ± 4.38	**66.87** ± 21.64
Hopper-gravity	**78.70** ± 19.17	34.10 ± 11.12	76.73 ± 14.19	42.44 ± 22.19	74.68 ± 17.34
Cheetah-dof	93.90 ± 24.49	39.36 ± 9.27	53.80 ± 13.90	43.99 ± 16.23	**97.85** ± 12.30
Cheetah-torso	56.21 ± 12.37	39.35 ± 6.33	51.93 ± 16.22	45.30 ± 5.60	**61.60** ± 1.19

Environment	Offline ESCP	FOCAL	CORRO	Prompt-DT	**DORA (Ours)**
Cheetah-gravity	64.30 ± 24.69	42.84 ± 12.14	38.72 ± 14.78	30.19 ± 7.74	**70.05** ± 17.02
Pendulum-gravity	86.02 ± 12.26	29.53 ± 14.41	57.21 ± 18.36	20.89 ± 10.99	**98.09** ± 13.95
Walker-gravity	35.64 ± 20.12	12.69 ± 7.08	29.02 ± 19.60	18.91 ± 11.78	**43.33** ± 14.21
Hopper-gravity	**71.96** ± 24.40	12.96 ± 4.63	16.70 ± 7.73	33.52 ± 10.82	71.52 ± 21.37
Cheetah-dof	73.60 ± 22.73	27.36 ± 12.27	43.08 ± 16.09	30.77 ± 10.46	**77.48** ± 12.25
Cheetah-torso	56.50 ± 11.65	38.13 ± 6.50	50.89 ± 6.69	34.04 ± 4.83	**61.57** ± 1.34

Table 2. Average normalized return ± standard deviation on the MuJoCo tasks in **non-stationary dynamics** over 5 random seeds.

Environment	Offline ESCP	FOCAL	CORRO	Prompt-DT	**DORA (Ours)**
Cheetah-gravity	65.42 ± 6.54	49.96 ± 6.41	54.08 ± 8.54	23.52 ± 7.75	**71.61** ± 7.56
Pendulum-gravity	85.92 ± 16.37	8.66 ± 5.42	69.66 ± 14.80	11.36 ± 5.86	**97.11** ± 16.91
Walker-gravity	27.34 ± 10.75	17.91 ± 10.76	33.62 ± 14.86	14.49 ± 2.81	**40.46** ± 19.04
Hopper-gravity	**54.29** ± 25.39	21.09 ± 7.99	53.26 ± 20.36	15.91 ± 10.13	49.74 ± 23.74
Cheetah-dof	80.72 ± 8.49	37.27 ± 13.76	52.41 ± 13.99	20.24 ± 12.66	**91.83** ± 10.57
Cheetah-torso	53.99 ± 16.09	42.23 ± 15.01	45.58 ± 15.98	33.79 ± 10.51	**60.13** ± 13.65

changing dynamics since the changes are unknown to the agent. From the information bottleneck perspective, we tackle these problems by developing a novel model-free framework that efficiently debiases the representations from the behavior policy and swiftly adapts to non-stationary dynamics.

5. Experiments

In this section, we conduct the experiments to answer the following questions:

- How does the learned encoder and contextual policy of DORA perform in unseen tasks of in-distribution (IID) and out-of-distribution (OOD) dynamics?
- How well does DORA identify and adapt to online non-stationary dynamics?
- Are the learned representations of DORA debiased from the behavior policy?

In Section 5.1, we introduce the environments and baselines. In Section 5.2, we compare the performance of different algorithms in IID, OOD, and non-stationary dynamics. In Section 5.3 and Section 5.4, we visualize the representations in IID dynamics and study the encoder's sensitivity to non-stationary dynamics. In Section 5.5, we study whether the learned representations are debiased from behavior policy. In Section 5.6, we ablate multiple design choices in the encoder training process.

5.1. Environments and Baselines

Task Description. We choose MuJoCo tasks for experiments, including `HalfCheetah-v3`, `Walker2d-v3`, `Hopper-v3`, and `InvertedDoublePendulum-v2`, which are common benchmarks in offline RL (Todorov et al., 2012).

Changing Dynamics. In order to generate different non-stationary dynamics, we change the physical parameters of the environments, namely `gravity`, `dof-damping`, and `torso-mass`. In **IID** dynamics, the parameter is uniformly sampled from the same distribution used for sampling training tasks. In **OOD** dynamics, the parameter is uniformly sampled from a distribution of the 20% range outside the IID dynamics. The parameter of **non-stationary dynamics** is sampled from the union of IID and OOD dynamics and changes every 50 timesteps. For convenience, we will ab-

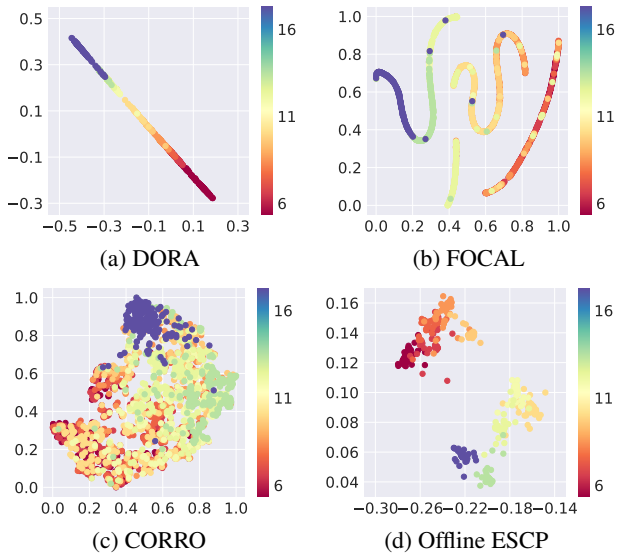


Figure 2. Representation visualization in `Cheetah-gravity` tasks with IID dynamics. The points are the (projected) representations in a 2D latent space, with the color indicating the real parameters of dynamics.

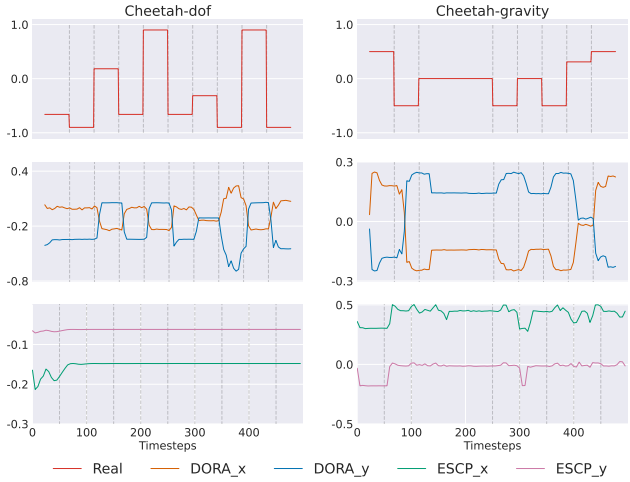


Figure 3. Representation tracking in a single trajectory in non-stationary dynamics. `Real` represents the normalized real parameters of unseen dynamics, `DORA_x` and `DORA_y` are the coordinates of the DORA's representations in the 2D latent space, and the same applies to offline ESCP. **Left:** In `Cheetah-dof`. **Right:** In `Cheetah-gravity`.

breviate the experimental tasks as `Cheetah-gravity`, `Pendulum-gravity`, `Walker-gravity`, `Hopper-gravity`, `Cheetah-dof`, and `Cheetah-torso`.

Offline Data Collection. For each training dataset, we use SAC (Haarnoja et al., 2018) to train a policy for each training task independently to make sure the behavior policies are different in every single-task dataset. The offline dataset is then collected from the replay buffer. We gather 200,000 transitions for each single-task dataset, except for the tasks of `Pendulum-gravity`, which comprises 40,000 transitions.

Baselines: We compare DORA with 4 baselines. **CORRO** (Yuan & Lu, 2022), **FOCAL** (Li et al., 2021), and **Prompt-DT** (Xu et al., 2022) are prominent OMRL baselines in recent years. ESCP (Luo et al., 2022) is an online meta RL approach that effectively adapts to sudden changes. In order to operate this algorithm in the offline setting, we develop the **offline ESCP** as a baseline.

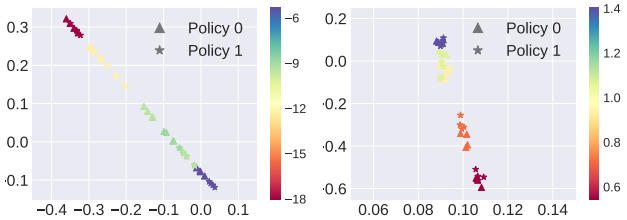
For all baselines and our method, a history trajectory of length 8 is used to infer task representations. The debiased loss weight β of our method is adjusted for different tasks. More details about the environments, offline datasets, and baselines in our experiments can be found in Appendix C.

5.2. Performance

We conduct experiments in IID, OOD, and non-stationary dynamics to compare the normalized return of the meta CQL policies learned with encoders from different algorithms. During testing, all algorithms are not permitted to pre-collect trajectories on new tasks, which means the encoder has to use short history trajectories generated by contextual policy to infer representations. As shown in Table 1, DORA outperforms the baselines in 5 of the total 6 MuJoCo tasks in both IID and OOD dynamics. Although offline ESCP performs better in `Hopper-gravity`, its performance suffers from a significant drop of 8.6% from IID to OOD tests, while the drop of DORA is 4.2%. The poor performance of Prompt-DT reveals that the transformer struggles to unleash its fitting and generalization capabilities in the absence of a well-defined prompt. DORA also excels at handling non-stationary dynamics, and its lead in most of the non-stationary environments is stronger than that in stationary environments, as illustrated in Table 2. These results suggest that DORA is effective in dealing with unseen non-stationary dynamics. As all baselines share the same contextual policies, it can thus be inferred that our encoder is more sensitive to the environment dynamics.

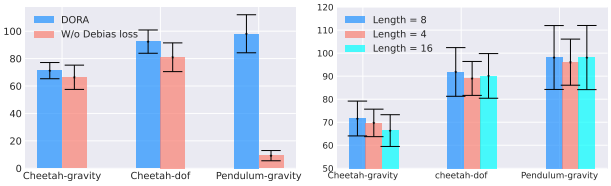
5.3. Representations in IID Dynamics

In this section, we compare the latent representations of DORA with other context-based OMRL baselines in `Cheetah-gravity` with IID dynamics, which are visualized in Figure 2. The representations of DORA and offline ESCP are encoded into two-dimensional vectors, while the representations of CORRO and FOCAL are encoded into higher dimensions and projected into 2D space via t-SNE (Van der Maaten & Hinton, 2008), following the approach used in the original paper. For each algorithm, we sample



(a) Cheetah-gravity (b) Cheetah-dof

Figure 4. Visualization on task representations generated with 2 different context-collection policies in 5 unseen dynamics. Points of different shapes represent different policies. **Left:** In `Cheetah-gravity`. **Right:** In `Cheetah-dof`.



(a) Ablation on debias loss (b) Ablation on history length

Figure 5. Ablation studies: Average normalized return of DORA on 3 environments over 5 random seeds. The error bar stands for the standard deviation. **Left:** DORA with and without the debias loss. **Right:** RNN history lengths of 4, 8, and 16.

200 short trajectories from the tasks with IID dynamics. The color of the encodings represents the real value of the changing dynamics parameter, and the color bar is on the right of each sub-figure correspondingly. The visualization results suggest that DORA's representations appear as a straight line with parameters of dynamics gradually increasing or decreasing from one end to the other. In contrast, representations of CORRO and FOCAL are much more cluttered. Notably, although ESCP extracts linear-shape representations in the online setting, the offline ESCP fails to achieve the same effect. The shape of representations indicates that DORA's encoder extracts more informative encodings and distinguishes the dynamics better than other baselines. We also visualize the encodings in OOD dynamics in Appendix D, which indicates DORA's representations can be extended to unseen OOD dynamics.

5.4. Representation Tracking in Changing Dynamics

In order to study the encoder's sensitivity to changing dynamics, we track the changes of representations during the adaptation in part of a single trajectory, which is shown in Figure 3. *Real* represents the normalized real parameters of unseen dynamics, x and y are the coordinates of the representations in the two-dimensional latent space. The environment dynamics change every 50 timesteps. It is evident to find that DORA's encoding promptly follows almost every sudden change of dynamics, while offline ESCP generates few responses. As illustrated by these experimental results, DORA's encoder infers robust representations and responds to non-stationary dynamics swiftly and precisely.

5.5. Debiased Representation Visualization

In this part, we study whether the learned representations are debiased from behavior policy. We first randomly choose the datasets of training tasks and train a Behavior Cloning (BC, Ross & Bagnell (2010)) policy using each chosen dataset. Then these trained BC policies are used to roll out trajectories of H timesteps in different unseen IID and OOD dynamics. Subsequently, we adopt DORA's encoder to generate representations with these short trajectories, which are visualized in Figure 4. The experimental results show that representations inferred from trajectories generated by different policies still cluster together in the same task dynamics. Moreover, in comparison with the results of DORA in Figure 2, these representations are encoded to similar coordinates according to the real parameter of dynamics in Figure 4. These results powerfully demonstrate that DORA learns a dynamic-sensitive encoder, which takes the environment dynamics rather than behavior policy as a feature to generate debiased representations.

5.6. Ablation Studies

We first conduct ablation experiments to study the interdependence of the distortion loss and the debias loss. In Figure 5a, the ablation experiments in 3 randomly chosen environments show that the debias loss significantly improves DORA. Notably, DORA can not converge without the distortion loss, so this ablated version is not included in the bar chart. Besides, to assess the impact of RNN history length on DORA's performance, we compared instances with RNN lengths of 4, 8, and 16. The experimental results in Figure 5b demonstrate that DORA is not sensitive to the RNN history length. Additionally, we conducted sensitivity tests on the hyper-parameter β in Appendix F.2.

6. Summary

In this paper, we consider the online adaptation issue in non-stationary dynamics from a fully offline perspective. We propose a novel offline representation learning framework DORA, in which the context encoder can efficiently identify changes in task dynamics and the meta policy is capable of rapidly adapting to online non-stationary dynamics. We follow the information bottleneck principle to formalize the offline representation learning problem. To conquer this problem, we derive a lower bound for the encoder to capture task-relevant information and an upper bound to debias the representation from behavior policy. We com-

pare DORA with other OMRL baselines in environments of IID, OOD, and non-stationary dynamics. Experimental results show that DORA debiases the representation from behavior policy and exhibits better performance compared with baselines. Considering that our method is a model free approach, DORA did not augment the offline dataset. In our future work, we hope to expand DORA into a model-based framework to achieve data efficient representation learning.

Impact Statement

Our work focuses on offline representation learning. We anticipate that its societal impact lies in advancing the field of machine learning. As our work primarily involves theoretical advancements and algorithmic innovations, we do not foresee specific ethical or safety concerns associated with our contributions. The potential societal impact of our work lies in fostering more efficient and adaptable offline RL policies, which can benefit various applications across industries. We believe that the implications of our research align with well-established ethical standards in the field of machine learning.

Acknowledgements

This work is supported by the National Science and Technology Major Project (2022ZD0114804), the National Science Foundation of China (62276126), and the Natural Science Foundation of Jiangsu (BK20221442). The authors would like to thank Fanming Luo, Chen-Xiao Gao, and the anonymous reviewers for their support and helpful discussions on improving the paper.

References

An, G., Moon, S., Kim, J.-H., and Song, H. O. Uncertainty-Based Offline Reinforcement Learning with Diversified Q-Ensemble. In *Advances in Neural Information Processing Systems*, 2021.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision Transformer: Reinforcement Learning via Sequence Modeling. In *Advances in Neural Information Processing Systems*, 2021.

Chen, R. T., Li, X., Grosse, R. B., and Duvenaud, D. K. Isolating Sources of Disentanglement in Variational Autoencoders. In *Advances in Neural Information Processing Systems*, 2018.

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Conference on Empirical Methods in Natural Language Processing*, 2014.

Choi, P.-M. *Reinforcement Learning in Nonstationary Environments*. Hong Kong University of Science and Technology (Hong Kong), 2000.

Finn, C., Abbeel, P., and Levine, S. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *International Conference on Machine Learning*, 2017.

Fujimoto, S., Meger, D., and Precup, D. Off-Policy Deep Reinforcement Learning without Exploration. In *International Conference on Machine Learning*, 2019.

Gao, C.-X., Wu, C., Cao, M., Kong, R., Zhang, Z., and Yu, Y. ACT: Empowering Decision Transformer with Dynamic Programming via Advantage Conditioning. In *AAAI Conference on Artificial Intelligence*, 2024.

Gao, Y., Zhang, R., Guo, J., Wu, F., Yi, Q., Peng, S., Lan, S., Chen, R., Du, Z., Hu, X., Guo, Q., Li, L., and Chen, Y. Context Shift Reduction for Offline Meta-Reinforcement Learning. In *Advances in Neural Information Processing Systems*, 2023.

Garcıa, J. and Fernández, F. A Comprehensive Survey on Safe Reinforcement Learning. *Journal of Machine Learning Research*, 16:1437–1480, 2015.

Ghosh, D., Ajay, A., Agrawal, P., and Levine, S. Offline RL Policies should be Trained to be Adaptive. In *International Conference on Machine Learning*, 2022.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *International Conference on Machine Learning*, 2018.

Jia, C., Gao, C., Yin, H., Zhang, F., Chen, X.-H., Xu, T., Yuan, L., Zhang, Z., Zhou, Z.-H., and Yu, Y. Policy Rehearsing: Training Generalizable Policies for Reinforcement Learning. In *International Conference on Learning Representations*, 2024a.

Jia, C., Zhang, F., Li, Y.-C., Gao, C.-X., Liu, X.-H., Yuan, L., Zhang, Z., and Yu, Y. Disentangling Policy from Offline Task Representation Learning via Adversarial Data Augmentation. In *International Conference on Autonomous Agents and Multiagent Systems*, 2024b.

Kaiser, L., Babaeizadeh, M., Milos, P., Osinski, B., Campbell, R. H., Czechowski, K., Erhan, D., Finn, C., Kozakowski, P., Levine, S., Mohiuddin, A., Sepassi, R., Tucker, G., and Michalewski, H. Model Based Reinforcement Learning for Atari. In *International Conference on Learning Representations*, 2020.

Kaplanis, C., Shanahan, M., and Clopath, C. Policy Consolidation for Continual Reinforcement Learning. In *International Conference on Machine Learning*, 2019.

Khetarpal, K., Riemer, M., Rish, I., and Precup, D. Towards Continual Reinforcement Learning: A Review and Perspectives. *Journal of Artificial Intelligence Research*, 75: 1401–1476, 2022.

Kidambi, R., Rajeswaran, A., Netrapalli, P., and Joachims, T. MOReL : Model-Based Offline Reinforcement Learning. In *Advances in Neural Information Processing Systems*, 2020.

Kullback, S. and Leibler, R. A. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1): 79–86, 1951.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative Q-Learning for Offline Reinforcement Learning. In *Advances in Neural Information Processing Systems*, 2020.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. *arXiv preprint arXiv:2005.01643*, 2020.

Li, L., Yang, R., and Luo, D. FOCAL: Efficient Fully-Offline Meta-Reinforcement Learning via Distance Metric Learning and Behavior Regularization. In *International Conference on Learning Representations*, 2021.

Lin, R., Li, Y., Feng, X., Zhang, Z., Fung, X. H. W., Zhang, H., Wang, J., Du, Y., and Yang, Y. Contextual Transformer for Offline Meta Reinforcement Learning. *arXiv preprint arXiv:2211.08016*, 2022a.

Lin, S., Wan, J., Xu, T., Liang, Y., and Zhang, J. Model-Based Offline Meta-Reinforcement Learning with Regularization. In *International Conference on Learning Representations*, 2022b.

Luo, F.-M., Jiang, S., Yu, Y., Zhang, Z., and Zhang, Y.-F. Adapt to Environment Sudden Changes by Learning a Context Sensitive Policy. In *AAAI Conference on Artificial Intelligence*, 2022.

Mnih, A. and Teh, Y. W. A Fast and Simple Algorithm for Training Neural Probabilistic Language Models. In *International Conference on Machine Learning*, 2012.

Nagabandi, A., Clavera, I., Liu, S., Fearing, R. S., Abbeel, P., Levine, S., and Finn, C. Learning to Adapt in Dynamic, Real-World Environments through Meta-Reinforcement Learning. In *International Conference on Learning Representations*, 2019.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation Learning with Contrastive Predictive Coding. *arXiv preprint arXiv:1807.03748*, 2018.

Rakelly, K., Zhou, A., Finn, C., Levine, S., and Quillen, D. Efficient Off-Policy Meta-Reinforcement Learning via Probabilistic Context Variables. In *International Conference on Machine Learning*, 2019.

Ran, Y., Li, Y., Zhang, F., Zhang, Z., and Yu, Y. Policy Regularization with Dataset Constraint for Offline Reinforcement Learning. In *International Conference on Machine Learning*, 2023.

Ross, S. and Bagnell, D. Efficient Reductions for Imitation Learning. In *International Conference on Artificial Intelligence and Statistics*, 2010.

Sohn, K., Lee, H., and Yan, X. Learning Structured Output Representation using Deep Conditional Generative Models. In *Advances in Neural Information Processing Systems*, 2015.

Sutton, R. and Barto, A. *Reinforcement Learning: An Introduction*. MIT press, 2018.

Swazinna, P., Udluft, S., and Runkler, T. User-Interactive Offline Reinforcement Learning. In *International Conference on Learning Representations*, 2022.

Tishby, N. and Zaslavsky, N. Deep Learning and the Information Bottleneck Principle. In *IEEE Information Theory Workshop*, 2015.

Tishby, N., Pereira, F. C., and Bialek, W. The Information Bottleneck Method. *arXiv preprint arXiv: physics/0004057*, 2000.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A Physics Engine for Model-Based Control. In *International Conference on Intelligent Robots and Systems*, 2012.

Van der Maaten, L. and Hinton, G. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(11), 2008.

Wu, C. and Zhang, Z. Surfing Information: The Challenge of Intelligent Decision-Making. *Intelligent Computing*, 2:0041, 2023.

Wu, Y., Tucker, G., and Nachum, O. Behavior Regularized Offline Reinforcement Learning. *arXiv preprint arXiv:1911.11361*, 2019.

Xu, M., Shen, Y., Zhang, S., Lu, Y., Zhao, D., Tenenbaum, J., and Gan, C. Prompting Decision Transformer for Few-Shot Policy Generalization. In *International Conference on Machine Learning*, 2022.

Yu, T., Thomas, G., Yu, L., Ermon, S., Zou, J. Y., Levine, S., Finn, C., and Ma, T. MOPO: Model-Based Offline Policy Optimization. In *Advances in Neural Information Processing Systems*, 2020.

Yu, T., Kumar, A., Rafailov, R., Rajeswaran, A., Levine, S., and Finn, C. COMBO: Conservative Offline Model-Based Policy Optimization. In *Advances in Neural Information Processing Systems*, 2021.

Yuan, H. and Lu, Z. Robust Task Representations for Offline Meta-Reinforcement Learning via Contrastive Learning. In *International Conference on Machine Learning*, 2022.

Zhou, R., Gao, C.-X., Zhang, Z., and Yu, Y. Generalizable Task Representation Learning for Offline Meta-Reinforcement Learning with Data Limitations. In *AAAI Conference on Artificial Intelligence*, 2024.

A. Proofs and Explanations

A.1. Proof of Theorems

Theorem 3.1. *Denote a set of N tasks as $\mathcal{M}$, in which each task M_i is sampled from the same training task distribution P_{train}. Let random variables $M \in \mathcal{M}$, τ be a trajectory collected in M, $z \sim p_\phi(\cdot|\tau)$, and $p(z)$ be the prior distribution of z, then we have*

$$I(z;M) \geq \mathbb{E}_{M,\tau,z}\left[\log \frac{\frac{p_\phi(z|\tau)}{p(z)}}{\sum_{M_i\in\mathcal{M}} \frac{p_\phi(z|\tau^i)}{p(z)}}\right] + \log N,$$

where τ^i is a trajectory collected in task M_i, $z_i \sim p_\phi(\cdot|\tau^i)$, and $i \in \{1, 2, \cdots, N\}$.

Proof. We follow the proof of Yuan & Lu (2022), as follows:

$$\begin{aligned}
I(z;M) &= \mathbb{E}_{\mathcal{M},z}\left[\log \frac{p(M \mid z)}{p(M)}\right] \\
&\overset{(a)}{=} \mathbb{E}_{\mathcal{M},z}\left[\log \mathbb{E}_\tau\left[\frac{p_\phi(z \mid \tau)}{p(z)}\right]\right] \\
&\geq \mathbb{E}_{\mathcal{M},z}\mathbb{E}_\tau\left[\log \frac{p_\phi(z \mid \tau)}{p(z)}\right] \\
&= \mathbb{E}_{\mathcal{M},\tau,z}\left[\log \frac{p_\phi(z \mid \tau)}{p(z)}\right] \\
&= -\mathbb{E}_{\mathcal{M},\tau,z}\left[\log\left(\frac{p\left(z\right)}{p_\phi\left(z \mid \tau\right)}N\right)\right] + \log N \\
&\overset{(b)}{\geq} -\mathbb{E}_{\mathcal{M},\tau,z}\left[\log\left(1 + \frac{p\left(z\right)}{p_\phi\left(z \mid \tau\right)}(N-1)\right)\right] + \log N \\
&= -\mathbb{E}_{\mathcal{M},\tau,z}\left[\log\left(1 + \frac{p\left(z\right)}{p_\phi\left(z \mid \tau\right)}(N-1)\underset{M_i\in\mathcal{M}\setminus\{M\}}{\mathbb{E}}[\frac{p_\phi\left(z \mid \tau^i\right)}{p\left(z\right)}]\right)\right] + \log N \\
&\approx -\mathbb{E}_{\mathcal{M},\tau,z}\left[\log\left(1 + \frac{p\left(z\right)}{p_\phi\left(z \mid \tau\right)}\sum_{M_i\in\mathcal{M}\setminus\{M\}}\frac{p_\phi\left(z \mid \tau^i\right)}{p\left(z\right)}\right)\right] + \log N \\
&= \mathbb{E}_{\mathcal{M},\tau,z}\left[\log\left(\frac{\frac{p_\phi(z|\tau)}{p(z)}}{\frac{p_\phi(z|\tau)}{p(z)} + \sum_{M_i\in\mathcal{M}\setminus\{M\}}\frac{p_\phi(z|\tau^i)}{p(z)}}\right)\right] + \log N \\
&= \mathbb{E}_{\mathcal{M},\tau,z}\left[\log\left(\frac{\frac{p_\phi(z|\tau)}{p(z)}}{\sum_{j=1}^{N}\frac{p_\phi(z|\tau^i)}{p(z)}}\right)\right] + \log N.
\end{aligned}$$

Here, $\mathcal{M}\setminus\{M\}$ is a set of tasks in $\mathcal{M}$ except the task M. The equality at (a) is derived from the following expression:

$$\frac{p(z \mid M)}{p(z)} = \int \frac{p(\tau \mid M)p(z \mid \tau, M)}{p(z)}\mathrm{d}\tau = \int \frac{p(\tau)p_\phi(z \mid \tau)}{p(z)}\mathrm{d}\tau = \mathbb{E}_\tau\left[\frac{p_\phi(z \mid \tau)}{p(z)}\right].$$

The inequality at (b) can be proved as follows:

$$
\begin{aligned}
&\mathbb{E}_{\tau,z}\left[\log\left(1+\frac{p(z)}{p(z\mid\tau)}(N-1)\right)\right]-\mathbb{E}_{\tau,z}\left[\log\left(\frac{p(z)}{p(z\mid\tau)}N\right)\right]\\
&=\mathbb{E}_{\tau,z}\left[\log\left(\frac{1}{N}\cdot\frac{p(z\mid\tau)}{p(z)}+\frac{N-1}{N}\cdot 1\right)\right]\\
&\overset{(c)}{\geqslant}\mathbb{E}_{\tau,z}\left[\frac{1}{N}\log\frac{p(z\mid\tau)}{p(z)}+\frac{N-1}{N}\log 1\right]\\
&=\frac{1}{N}\mathbb{E}_{\tau,z}\left[\log\frac{p(z\mid\tau)}{p(z)}\right]\\
&=\frac{1}{N}I(z;\tau)\\
&\geqslant 0
\end{aligned}
$$

The inequality at (c) is derived from Jensen's Inequality. Thus, the proof is completed. □

Theorem 3.2. *Given a training task $M\in\mathcal{M}$, a trajectory τ collected in M, and $z\sim p_\phi(\cdot|\tau)$, we have*

$$
I(z;a)\leqslant\mathbb{E}_a\left[D_{\mathrm{KL}}\left[p\left(\cdot\mid a\right)\|t(\cdot)\right]\right], \tag{5}
$$

where $t(z)$ is an arbitrary distribution over $\mathbb{R}^m$ and D_{KL} is the Kullback-Leibler divergence (Kullback & Leibler, 1951).

Proof.

$$
\begin{aligned}
I(z;a)&=\int\int p\left(z,a\right)\log\frac{p\left(z\mid a\right)}{p(z)}\mathrm{d}z\mathrm{d}a\\
&=\int\int p\left(z,a\right)\log p\left(z\mid a\right)\mathrm{d}z\mathrm{d}a-\int p(z)\log p(z)\mathrm{d}z\\
&\overset{(d)}{\leqslant}\int\int p\left(z,a\right)\log p\left(z\mid a\right)\mathrm{d}z\mathrm{d}a-\int p(z)\log t(z)\mathrm{d}z\\
&=\int\int p\left(a\right)p\left(z\mid a\right)\log\frac{p\left(z\mid a\right)}{t(z)}\mathrm{d}z\mathrm{d}a\\
&=\mathbb{E}_a\left[\int p\left(z\mid a\right)\log\frac{p\left(z\mid a\right)}{t(z)}\mathrm{d}z\right]\\
&=\mathbb{E}_a\left[D_{\mathrm{KL}}\left[p\left(\cdot\mid a\right)\|t(\cdot)\right]\right].
\end{aligned}
$$

The inequality at (d) is derived from $D_{\mathrm{KL}}\left[p(\cdot)\|t(\cdot)\right]\geq 0$. □

A.2. Explanations of the Approximation in Distortion Loss

In this section, we further explain the reason for the approximation of $p(z|\tau)/p(z)$ by $S(z,\bar{z})=\exp(-||z-\bar{z}||^2/r)$. From Bayesian theory, we know that the bigger $\frac{p(z|\tau)}{p(z)}$ is, the more likely z correlates with τ. In practice, it may be intractable to get the marginal probability $p(z)$. Thus, inspired by InfoNCE, we learn a positive real score function $S(z,\tau)$ to approximate $\frac{p(z|\tau)}{p(z)}$. Given N trajectories $\tau^1,\tau^2,\cdots,\tau^N$ sampled from N different dynamics respectively, Theorem 3.1 tells us that, to maximize $I(z;M)$, the encoder is required to predict the context z with τ_i as input and z must have the least correlation with all τ_j, where $j\in\{1,2,\cdots,i-1,i+1,\cdots,N\}$. This gives us an intuition that $S(z,\tau)$ should measure the similarity of z and τ. Practically, we implement S with a Gaussian kernel function (though there can be more choices, such as the cosine similarity used by InfoNCE). Moreover, since τ has an indefinite length, it is difficult to directly calculate the similarity of z and τ. Therefore, in our implementation, we turn to calculate the similarity between z and $\bar{z}$, where $\bar{z}$ is the moving average of all the contexts from the dynamics in which we collect τ.

B. DORA Testing

The pseudocodes of the testing phase of DORA is illustrated in Algorithm 2.

Algorithm 2 DORA Test

Input: testing tasks $\{M_i^{\text{test}}\}_{i=1}^N$, pre-trained context encoder p_ϕ, pre-trained contextual policy $\pi_\theta(s, z)$, episode length T_s
for each task $M_i \in \{M_i^{\text{test}}\}_{i=1}^N$ **do**
  Initialize a history trajectory τ of fixed length H with zeros
  Get the initial state s_0 of M_i, append $(s_0, 0)$ to τ
  for t = 0 to T_s **do**
    Infer z^t from τ with p_ϕ
    Get action $a \sim \pi_\theta(s, z^t)$
    Step in the environment and get the next state s'
    $s \leftarrow s'$
    Append (s, a) to τ
  end for
end for

C. Experiments Details

C.1. Task Description

We choose several MuJoCo tasks for experiments, including `HalfCheetah-v3`, `Walker2d-v3`, `Hopper-v3`, and `InvertedDoublePendulum-v2`, which are common benchmarks in offline RL (Todorov et al., 2012). In `HalfCheetah`, `Walker2d`, and `Hopper`, agents with several degrees of freedom need to learn to move forward as fast as possible. `InvertedDoublePendulum` originates from the classical control problem `Cartpole`, in which the agent needs to push the cart left and right to balance the pole on top of the bottom pole.

C.2. Environment Details

In this section, we provide detailed information about the dynamic-changing environments.

Gravity: `Gravity` is a global parameter in MuJoCo, which mainly affects the agent's fall speed and vertical pressure. The dynamic is sampled by multiplying the default gravity g_0 with $1.5^\mu, \mu \sim U[-a, a]$, where $a = 1.5, 1.8$ for IID and non-stationary dynamics, respectively. We uniformly sample μ from $U[-1.8, -1.5] \cup [1.5, 1.8]$ for OOD dynamics.

Dof-Damping: `Dof-damping` refers to the damping value matrix applied to all degrees of freedom of the agent, serving as linear resistance proportional to velocity. The dynamics are sampled by multiplying the default damping matrix A_0 with $1.5^\mu, \mu \sim U[-a, a]$, where $a = 1.5, 1.8$ for IID and non-stationary dynamics, respectively. We uniformly sample μ from $U[-1.8, -1.5] \cup [1.5, 1.8]$ for OOD dynamics.

Torso-mass: `Torsoz-mass` is the mass of the agent's torso. The dynamic is sampled by multiplying the default mass of torso m_0 with $1.5^\mu, \mu \sim U[-a, a]$, where $a = 1.5, 1.8$ for IID and non-stationary dynamics, respectively. We uniformly sample μ from $U[-1.8, -1.5] \cup [1.5, 1.8]$ for OOD dynamics.

Besides, each environment contains 10 tasks for training and 10 tasks for testing for both IID, OOD, and non-stationary dynamics.

C.3. Details of the Offline Dataset

Maximum and Minimum Returns of Offline Datasets. In Table 3, we explicitly provide the maximum and minimum returns of offline datasets, which are used to calculate the normalized return in the policy evaluation. Specifically, the evaluation return x for a particular is normalized using the formula $\frac{x - x_{\min}}{x_{\max} - x_{\min}} \times 100$, where $x_{\max}$ and $x_{\min}$ denote the corresponding maximum and minimum returns of each dataset, respectively.

C.4. Introduction of Baselines

We compare DORA with 4 baselines which are listed below.

CORRO. CORRO (Yuan & Lu, 2022) designs a bi-level task encoder where the transition encoder is optimized by

Table 3. Maximum and minimum returns of offline datasets.

	Cheetah-gravity	Cheetah-dof	Cheetah-torso_mass	Hopper-gravity	Walker-graavity	Pendulum-gravity
Maximum	9218.48	8703.55	10060.59	3536.97	4630.10	9351.94
Minimal	-460.33	-613.09	-479.86	-1.22	-110.50	26.71

Table 4. Configurations and hyper-parameters used in offline encoder training.

Configurations	Cheetah-gravity	Cheetah-dof	Cheetah-torso	Hopper-gravity	Walker-gravity	Pendulum-gravity
Debias loss weight	0.2	1.0	1.0	0.2	1.0	1.0
Distortion loss weight	1.0	1.0	1.0	1.0	1.0	1.0
History length	8	8	8	8	8	8
Latent space dim	2	2	2	2	2	2
Batch size	256	256	256	256	256	256
Learning rate	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}
Training steps	2×10^{6}	2×10^{6}	2×10^{6}	2×10^{6}	2×10^{6}	4×10^{5}
Radius of radius basis function	200	200	200	200	200	200

contrastive task learning and the aggregator encoder gathers all representations. CORRO uses generative modeling and reward randomization to generate negative pairs for contrastive learning. In our setting, the reward function remains the same across tasks thus we adopt the approach of generative modeling to compare.

FOCAL. FOCAL (Li et al., 2021) uses distance metric loss to learn a deterministic contextual encoder, which is also under the paradigm of contrastive learning. Additionally, FOCAL combines BRAC (Wu et al., 2019) to constrain the bootstrapping error.

Prompt-DT. Prompt-DT (Xu et al., 2022) employs the Decision Transformer (Chen et al., 2021; Gao et al., 2024) to tackle OMRL tasks. It utilizes a short segment of task trajectory containing task-specific information as the prompt input, guiding the transformer to model trajectories from different tasks. For each task dataset, we select the top 5 trajectories with the highest cumulative returns to construct the task prompt. The remaining trajectories are then used for training.

Offline ESCP. ESCP (Luo et al., 2022) is an online meta RL approach that proposes variance minimization and relational matrix determinant maximization in optimizing the encoder to adapt to environment sudden changes. In order to fairly compare ESCP with other algorithms in offline settings, we develop an offline version of ESCP, denoted as offline ESCP. In this variant, online interaction process with the environment is not allowed. Instead, the agent has to sample trajectories from the offline dataset.

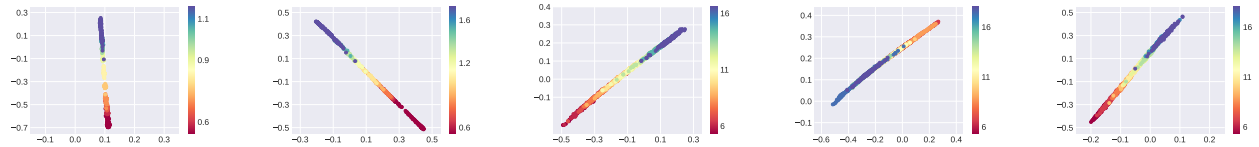
C.5. Configurations

The details of the important configurations and hyper-parameters used to produce the experimental results in this paper are listed in Table 4. Regarding the encoder model, we utilized a linear layer with 128 hidden units, a GRU network with 64 hidden units, and a linear layer with 64 hidden units for parameterization. The encoder's output is scaled using a Tanh function.

D. Representation Visualization of DORA

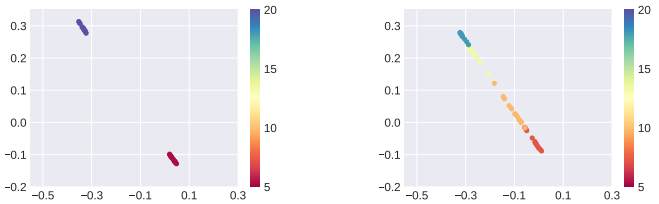
We present the representation visualization figures of other 5 MuJoCo tasks in Figure 6. With the color indicating the real parameters of dynamics, we find that the data points are sorted sequentially based on the real values of the varying parameters, which shows that the task representations capture the information of different task dynamics.

Moreover, we visualize the encodings of DORA in `Cheetah-gravity` with OOD dynamics, shown in Figure 7. It is evident to find that the representations in OOD dynamics (Figure 7a) are situated at both ends of the encodings in IID dynamics (Figure 7b). Such visualization results indicate that DORA learns generalizable representations, which can be extended to unseen OOD dynamics.



(a) Cheetah-dof (b) Cheetah-torso (c) Pendulum-gravity (d) Walker-gravity (e) Hopper-gravity

Figure 6. Representation visualization of DORA in other 5 MuJoCo tasks.



(a) In OOD dynamics (b) In IID dynamics

Figure 7. Representation visulization in `Cheetah-gravity`. **Left:** In OOD dynamics. **Right:** In IID dynamics.

E. Experiments in the Setting of OMRL

In the general OMRL setting, the algorithms are permitted to use a certain policy to collect context before evaluation. We compare the performance of DODA and other baselines in such a setting and the results are shown in Figure 8. Benefiting from the effective task representations, DORA still exhibits a remarkable performance and outperforms the baselines in 5 of the total 6 MuJoCo tasks.

F. Further Ablation Studies

F.1. Ablation Studies on the Losses

To visually illustrate the impact of the distortion loss and debias loss in DORA, we take the visualization results on `Cheetah-gravity` as an example. Figure 9a shows that the encoder can not extract informative representations without distortion loss. In addition, without the debias loss, representations inferred from tasks of similar dynamics no longer exhibit correlations as shown in Figure 9b.

F.2. Ablation Studies on β

We conducted sensitivity tests on the hyper-parameter β on 3 MuJoCo tasks in Figure 9c, and the results indicate that excessively large or small values of β lead to a decline in policy performance. Although in `Pendulum-gravity`, the case with $\beta = 0.2$ shows better average performance, the policy performance sharply drops when $\beta = 0$ (Figure 5a). These experimental results, on the one hand, demonstrate that when β is too small, the encoder struggles to debias representations from the behavior policy. On the other hand, when β is too large, the diminishing effect of the distortion loss during the optimization process leads to a decrease in policy performance.

F.3. Performance with Less Offline Data Available

We conducted experiments to test the performance of DORA under the conditions of halving the size of each single-task dataset (i.e. 10^5 transitions per single-task dataset) and find DORA still outperforms the other baselines in Table 5.

Table 5. Performance with less offline data available on `Cheetah-gravity` in non-stationary dynamics.

	FOCAL	CORRO	Prompt-DT	Offline ESCP	DORA
Half Datastes	47.43 ± 4.24	51.93 ± 4.84	32.85 ± 7.66	51.75 ± 4.36	$\mathbf{66.28 \pm 7.85}$
Full Datasets	49.96 ± 6.41	54.08 ± 8.54	23.52 ± 7.75	65.42 ± 6.54	$\mathbf{71.61 \pm 7.56}$

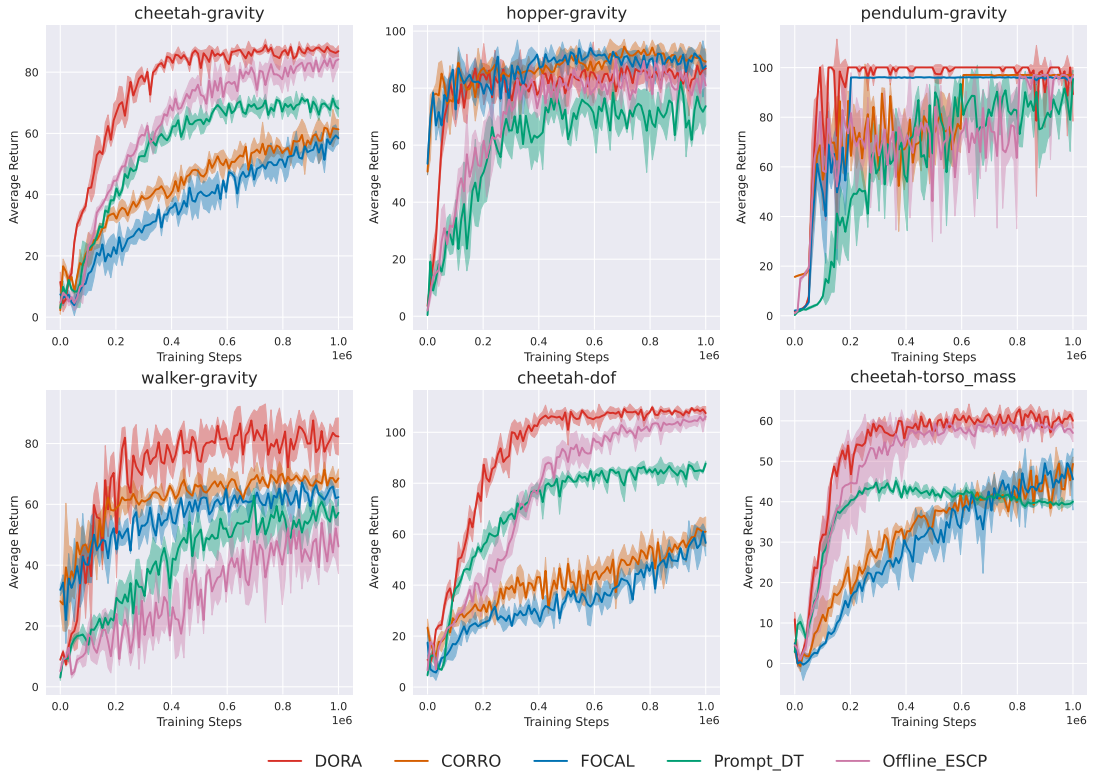


Figure 8. Test returns of DORA against the baselines in IID dynamics in the general OMRL setting. During meta-testing, a pre-collected context can be utilized to infer task representations before evaluation.

Besides, we conducted experiments of varying the number of training tasks to 5, 8, and 10. The experimental results are shown in Table 6. We observe that as the number of tasks decreases, the performance of all algorithms declines. This is because fewer training tasks can affect the algorithms' generalization performance in non-stationary dynamics.

Table 6. Performance with less traning tasks on `Cheetah-gravity` in non-stationary dynamics.

Number of Traning Tasks	5	8	10
DORA's Performance	54.92 $\pm$ 17.71	65.48 $\pm$ 10.58	71.61 $\pm$ 7.56

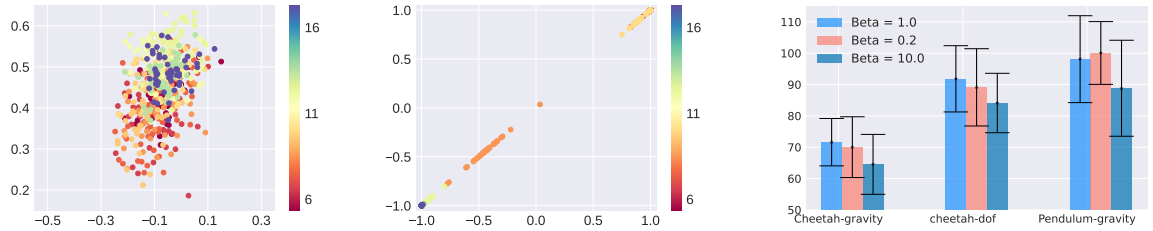
F.4. Ablation Studies on Offline RL Algorithms

Besides CQL, we train the contextual policy of DORA with different offline RL algorithms, including a model-based method COMBO (Yu et al., 2021) and a model-free method EDAC (An et al., 2021). The results are shown in Table 7.

Although DORA (CQL) is not the best performing among several versions, we use CQL as the general offline RL algorithm in our framework for 2 reasons. Firstly, CQL is well-known and widely applied in offline RL. Secondly, this algorithm is relatively simple and easy to implement. Thus, we can easily compare the performance of various baselines upon it.

Table 7. Performance with different offline RL algorithms in non-stationary dynamics.

	DORA (CQL)	DORA (EDAC)	DORA (COMBO)
cheetah-gravity	71.61 $\pm$ 7.56	**73.58** $\pm$ **5.44**	69.16 $\pm$ 7.36
cheetah-dof	91.83 $\pm$ 10.57	89.89 $\pm$ 5.85	**93.04** $\pm$ **7.47**
pendulum-gravity	97.11 $\pm$ 16.91	93.12 $\pm$ 15.35	**100.08** $\pm$ **0.05**



(a) Without distortion loss (b) Without debias loss (c) Ablation studies on β

Figure 9. Further ablation studies. **Left:** DORA without the distortion loss in `Cheetah-gravity`. **Middle:** DORA without debias loss in `Cheetah-gravity`. **Right:** Average normalized return of DORA with $\beta = 0.2$, 1.0, and 10.0 over 5 random seeds. The error bar stands for the standard deviation.

F.5. Performance with different changing rate of non-stationary dynamics

In order to test DORA's performance with different changing rates of non-stationary dynamics, we change the physical parameters of the environments every 10, 30, and 50 timesteps. The results in Table 8 indicate that DORA can adapt to frequent changes in dynamics.

Table 8. Performance with Different Changing Rate of Non-stationary Dynamics.

Tasks \ Change per Step	10	30	50
cheetah-gravity	64.28 ± 4.01	66.62 ± 5.78	71.61 ± 7.56
cheetah-dof	85.09 ± 5.63	87.44 ± 6.48	91.83 ± 10.57
pendulum-gravity	96.89 ± 14.56	96.49 ± 12.44	97.11 ± 16.91