Principled Offline RL in the Presence of Rich Exogenous Information

Riashat Islam * 1 2 3 **Manan Tomar** * 4 2 **Alex Lamb** 3 **Yonathan Efroni** 5 **Hongyu Zang** 6 **Aniket Didolkar** 7 3 **Dipendra Misra** 3 **Xin Li** 6 **Harm Van Seijen** 2 **Remi Tachet Des Combes** 2 **John Langford** 3

Abstract

Learning to control an agent from offline data collected in a rich pixel-based visual observation space is vital for real-world applications of reinforcement learning (RL). A major challenge in this setting is the presence of input information that is hard to model and irrelevant to controlling the agent. This problem has been approached by the theoretical RL community through the lens of *exogenous information*, i.e., any control-irrelevant information contained in observations. For example, a robot navigating in busy streets needs to ignore irrelevant information, such as other people walking in the background, textures of objects, or birds in the sky. In this paper, we focus on the setting with visually detailed exogenous information and introduce new offline RL benchmarks that offer the ability to study this problem. We find that contemporary representation learning techniques can fail on datasets where the noise is a complex and time-dependent process, which is prevalent in practical applications. To address these, we propose to use multi-step inverse models to learn Agent-Centric Representations for Offline-RL (ACRO). Despite being simple and reward-free, we show theoretically and empirically that the representation created by this objective greatly outperforms baselines.

1. Introduction

Effective real-world applications of reinforcement learning or sequential decision-making must cope with exogenous information in sensory data. For example, visual datasets of a robot or car navigating in busy city streets might contain information such as advertisement billboards, birds in the sky, or other people crossing the road. Parts of the observation (such as birds in the sky) are irrelevant for controlling the agent, while other parts (such as people crossing along the navigation route) are extremely relevant. How can we effectively learn a representation of the world that extracts just the relevant information for controlling the agent while ignoring irrelevant information?

Real-world tasks are often more easily solved with fixed offline datasets since operating from offline data enables thorough testing before deployment, which can ensure safety, reliability, and quality in the deployed policy (Lange et al., 2012; Ebert et al., 2018; Kumar et al., 2019; Jaques et al., 2019; Levine et al., 2020). The Offline-RL setting also eliminates the need to address exploration and planning, which come into play during data collection.[1] Although approaches from representation learning have been studied in the online case, yielding improvements, exogenous information has proved to be empirically challenging. In this paper, we therefore ask the question: is it possible to learn distraction-invariant representations from rich observations in offline RL?

Approaches for discovering small tabular MDPs ($\leq$500 discrete latent states) or linear control problems invariant to exogenous information have been introduced (Dietterich et al., 2018; Efroni et al., 2021; 2022a;b) before. However, the planning and exploration techniques in these algorithms are difficult to scale. A key insight that Lamb et al. (2022) uncovered is the usefulness of multi-step action prediction for learning exogenous-invariant representation. However this work was limited to settings where the endogenous dynamic is tabular and contains small amount of latent states. Further, they did not use the learned representations to solve a downstream task.

We propose to learn *Agent-Centric Representations for Offline-RL (ACRO)* using multi-step inverse models, which predict actions given current and future observations, as in Figure 2. ACRO avoids the problem of learning distrac-

*Equal contribution [1]McGill University, Quebec AI Institute [2]Microsoft Research, Montreal [3]Microsoft Research, New York [4]University of Alberta [5]Meta, New York [6]Beijing Institute of Technology, Beijing [7]University of Montreal, Quebec AI Institute. Correspondence to: Riashat Islam <riashat.islam@mail.mcgill.ca>, Manan Tomar <manan.tomar@gmail.com>, Alex Lamb <lambalex@microsoft.com>, John Langford <jcl@microsoft.com>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

[1] This elimination, however, can make offline RL more difficult if the wrong data is collected.

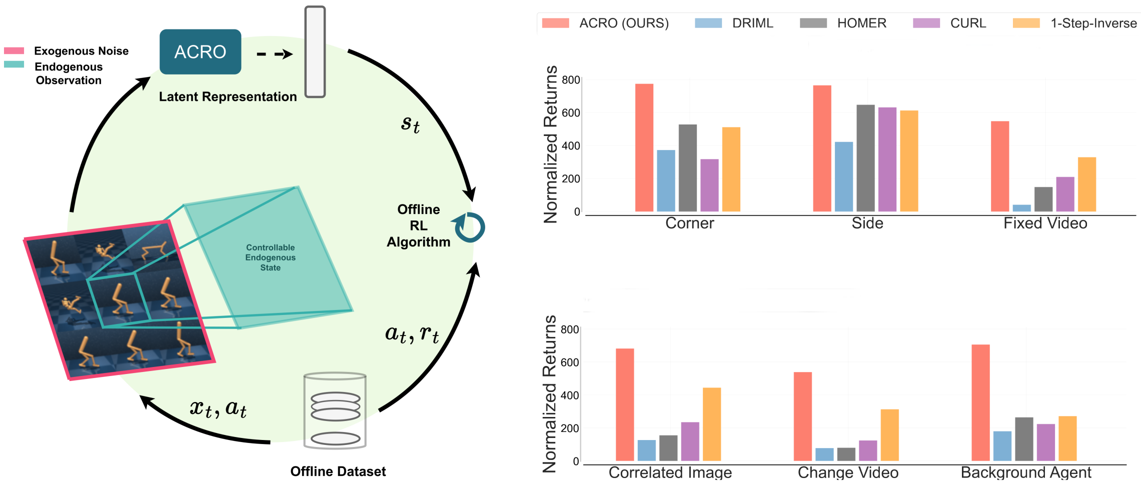


Figure 1: **Left: Representation Learning for Visual Offline RL in Presence of Exogenous Information**. We propose ACRO, that recovers the agent-centric latent representations from visual data which includes uncontrollable irrelevant information, such as observations of other agents acting in the same environment. **Right: Normalized Results Summary**. ACRO learns to ignore the observations of task irrelevant agents, while baselines tend to capture such exogenous information. We use different offline datasets with varying levels of exogenous information (Section 4) and find that baseline methods consistently under-perform w.r.t. ACRO, as is supported by our theoretical analysis. Experimental results are normalized (averaged) across domains and different types of datasets (expert, medium-expert and medium).

tors because they are not predictive of the agent's actions. This property even holds for temporally-correlated exogenous information. At the same time, we show that multi-step inverse models capture all the information that is sufficient for controlling the agent while being entirely reward-free, which we refer to as the agent-centric representation. Our first contribution is to show that ACRO outperforms all current baselines on datasets from policies of varying quality and stochasticity. Figure 1 gives an illustration of ACRO along with a summary of experimental findings.

A second core contribution of this work is to develop and release several new offline-RL benchmarks designed to have especially challenging exogenous information. In particular, we focus on *diverse temporally-correlated* exogenous information with datasets where (1) every episode has a different video playing in the background, (2) the same STL-10 image is placed to the side or corner of the observation throughout the episode, and (3) the observation consists of the views of nine independent agents but the actions only control one of them (see Fig. 1). Task (3) is challenging since the agent that is controllable must be learned from data.

Finally, we also introduce a new theoretical analysis (Section 2.2) that explores the connection between exogenous noise in the learned representation and the success of Offline-RL. In particular, we show that Bellman completeness is achieved from the agent-centric representation of ACRO while representations which include exogenous noise may not verify it. Bellman completeness has been previously shown to be a sufficient condition for the convergence of offline RL methods based on Bellman error minimization (Munos, 2003; Munos & Szepesvári, 2008). This highlights the challenges of Offline RL with exogenous information at the observation level.

2. ACRO: Agent-Centric Representations for Offline-RL

2.1. Preliminaries

We consider a Markov Decision Process (MDP) setting for modeling systems with both relevant and irrelevant components (also referred as exogenous block MDP in Efroni et al. (2021)). This MDP consists of a set of observations, $\mathcal{X}$; a set of latent states, $\mathcal{Z}$; a set of actions, $\mathcal{A}$; a transition distribution, $T(z' \mid z, a)$; an emission distribution $q(x \mid z)$; a reward function $R : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$; and a start state distribution $\mu_0(z)$. We also assume that the support of the emission distributions of any two latent states are disjoint. The latent state is decoupled into two parts $z = (s, e)$ where $s \in \mathcal{S}$ is the agent-centric state and $e \in \mathcal{E}$ is the exogenous state. For $z, z' \in \mathcal{Z}, a \in \mathcal{A}$ the transition function is decoupled as $T(z' \mid z, a) = T(s' \mid s, a)T_e(e' \mid e)$, and the reward only depends on (s, a). These definitions imply that there exist mappings $\phi_\star : \mathcal{X} \to \mathcal{S}$ and $\phi_{\star,e} : \mathcal{X} \to \mathcal{E}$ from observations to the corresponding agent-centric and exogenous and uncontrollable latent states. As in Lamb et al. (2022), we assume that the agent-centric dynamics is deterministic. The agent interacts with the environment, generating a latent state, observation and action sequence, $(z_1, x_1, a_1, z_2, x_2, a_2, \cdots,)$ where $z_1 \sim \mu(\cdot)$ and $x_t \sim q(\cdot \mid z_t)$. The agent does not observe the latent states $(z_1, z_2, \cdots)$, instead receiving only the observations $(x_1, x_2, \cdots)$. The agent chooses actions using a policy distribution $\pi(a \mid x)$. A policy is an *exo-free policy* if it is not a function of the exogenous noise. Formally, for any x_1 and x_2, if $\phi_\star(x_1) = \phi_\star(x_2)$, then $\pi(\cdot \mid x_1) = \pi(\cdot \mid x_2)$.

We consider learning representations from an offline dataset $\mathcal{D} = (\mathcal{X}, \mathcal{A})$ consisting of sequences of N observations $\mathcal{X} = (x_1, x_2, x_3, ..., x_N)$ and the corresponding actions $\mathcal{A} = (a_1, a_2, a_3, ..., a_N)$. We are in the rich-observation setting, *i.e.*, observation $x_t \in \mathbb{R}^m$ is sufficient to decode z_t. Our focus is on pre-training an encoder $\phi : \mathbb{R}^m \to \mathbb{R}^d$ on $\mathcal{D}$ such that the frozen representation $s_t = \phi(x_t)$ is suitable for offline policy optimization. In our setting, we assume that the reward function is free of exogenous noise, and only depends on the endogenous part of the observation space.

2.2. Benefits of Exogenous Invariant Representation in Offline RL

Due to its importance to practical applications, the offline RL setting has been extensively studied by the theoretical community. The majority of provable value-based offline RL algorithms follow a Bellman error minimization approach (Munos, 2003; Munos & Szepesvári, 2008; Antos et al., 2008), in line with the techniques used in practice. The common representational assumptions needed to derive these results are: (A1) the function class contains the optimal Q function (realizability), (A2) the data distribution is sufficiently diverse (concentrability), and (A3) Bellman completeness (Munos & Szepesvári, 2008). This last condition states that the function class can properly represent the Bellman backup of any function it contains.

Definition 2.1 (Bellman Completeness)**.** We say that a function class $\mathcal{F}$ is Bellman complete if it is closed under the Bellman operator. That is, for any $f \in \mathcal{F}$ it holds that $\mathcal{T}f \in \mathcal{F}$, where $(\mathcal{T}f)(x,a) \equiv R(x,a) + \mathbb{E}_{x' \sim T(x'|x,a)}[\max_{a'} f(x', a')]$ for all $(x,a) \in \mathcal{X} \times \mathcal{A}$.

Chen & Jiang (2019) conjectured that (A1) and (A2) alone are not sufficient for sample efficient offline RL, and, recently, Foster et al. (2021) established a lower bound proving this claim. Thus, the representational requirements needed for offline RL are more intricate than in supervised learning.

With these observations in mind, we highlight a key advantage of the agent-centric representation $\phi_\star$ relatively to other representations in the offline RL setting. Namely, we show one can construct a Bellman complete function class on top of $\phi_\star$, while some representations that include exogenous information provably violate Bellman completeness. To formalize these claims, we denote by $\mathcal{Q}_\mathcal{S} = \{(s,a) \mapsto [0,1] : (s,a) \in \mathcal{S} \times \mathcal{A}\}$ the set of Q-functions defined over $\mathcal{S}$, and for a given representation ϕ, we let $\mathcal{F}(\phi) = \{(s,a) \mapsto Q(\phi(s), a) : Q \in \mathcal{Q}_\mathcal{S}, (s,a) \in \mathcal{S} \times \mathcal{A}\}$ denote the set of Q-functions defined on top of ϕ. The following proposition states that the Agent-Centric representation leads to a Bellman complete function class (all proofs in Appendix A.1/ Appendix A.2).

Proposition 2.2 (ACRO Representation is Bellman Complete)**.** *$\mathcal{F}(\phi_\star)$ is Bellman complete.*

Next, we show that there exists a representation strictly more expressive than ACRO (*i.e.*, one that includes exogenous information and all the agent-centric information) which, surprisingly, violates the Bellman completeness property.

Proposition 2.3 (Exogenous Information May Violate Bellman Completeness)**.** *There exists ϕ which is a refinement*[2] *of $\phi_\star$ such that $\mathcal{F}(\phi)$ is not Bellman complete.*

This proposition implies that exogenous information being included in the representation may break the Bellman completeness assumption, which is a requirement for establishing the convergence of offline RL algorithms based on Bellman error minimization. From this perspective, additional information in the representation may deteriorate the performance of offline RL. Conversely, a coarser representation may trivially violate the realizability assumption A1: such a representation may merge states on which the optimal Q-function differs, preventing it from being realized.

Together, these observations motivate the experimental pipeline used this work: learn the agent-centric representation by optimizing Equation 30, then perform offline RL on top of it. In doing so, we obtain a representation that is sufficient for optimal performance, and yet filters the exogenous information which can (i) be impossible to exactly model, and (ii) hurt the offline RL performance.

2.3. Proposed Method: ACRO

Learning Representations: To learn representations that discard exogenous information, we leverage prior works from the theoretical RL community and train a multi-step inverse action prediction model, which captures long-range dependencies thanks to its conditioning on distant future observations. This leads to the ACRO objective, namely to predict the action conditioning on $\phi(x_t)$ and $\phi(x_{t+k})$. Note that even though we are conditioning on the future observation, we only predict the first action instead of the sequence of actions up to the k-th timestep, as the former is easier to learn (k is sampled up to a maximum span of K).

Our proposed method, which we call *Agent-Centric Representations for Offline-RL* (ACRO), optimizes the following

[2] Let $\mathcal{X}$ be a finite set of elements. Given a partition P of $\mathcal{X}$ let its induced equivalence relation be denoted by $\sim_P$. A partition P_1 is finer than P_2 if for any $x_1, x_2 \in \mathcal{X}$ such that $x_1 \sim_{P_1} x_2$ it also holds that $x_1 \sim_{P_2} x_2$.

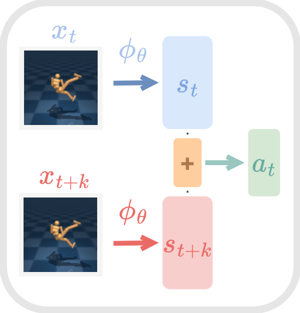


Figure 2: **ACRO**. is a multi-step inverse model that predicts the first action conditioned on the current state and the future state. **+** denotes concatenation.

objective based on a multi-step inverse model:

$$\phi_\star \in \arg\max_{\phi\in\Phi} \mathop{\mathbb{E}}_{\substack{t\sim U(0,N),\\ k\sim U(0,K)}} \log\left(\mathbb{P}(a_t \mid \phi(x_t), \phi(x_{t+k}))\right) \quad (1)$$

This approach (Figure 2) is motivated by two desiderata: (i) ignoring exogenous information and (ii) capturing the latent state that is necessary for control. The invariance lemma (Efroni et al., 2021) (see Appendix B.1 for proof) states that optimal action predictor models can be obtained without dependence on exogenous noise, when the data-collection policy is assumed not to depend on it either.

At the same time, prior work has shown that single-step inverse models of action prediction can fail to capture the full agent-centric latent state (Efroni et al., 2021; Lamb et al., 2022; Hutter & Hansen, 2022). One type of counter-example for single-step inverse models stems from a failure to capture long-range dependencies. For example, in an empty gridworld, a pair of positions that are two or more spaces apart can be mapped to the same representation without increasing the loss of a one-step inverse model. Another simple counter-example involves a problem where the last action the agent took is recorded in the observation, in which case the encoder can simply retrieve that action directly while ignoring all other information (although recording all recent actions in the observation is an issue for multi-step inverse models). The use of multi-step inverse models resolves both of these counter-examples and is able to learn the full agent-centric state (Efroni et al., 2021). Detailed theoretical analysis on ACRO provided in appendix B, with specific counterexamples to one-step models in B.4.

We emphasize here that even though inverse models of action prediction have appeared in past literature (as discussed in related works), they are often proposed for the purposes of exploration and reward bonus. In contrast, we propose to learn the multi-step inverse model to explicitly uncover a representation that contains only the agent-centric, endogenous part of the state. Recently, Lamb et al. (2022) proposed a multi-step inverse model where the learnt representation $\phi(\cdot)$ is regularized, so that $\phi(\cdot)$ discards irrelevant details from observations x. This was accomplished by using vector-quantization on the encoder's output, forcing discrete latent states to be learnt for constructing a tabular MDP for latent recovery. In contrast, ACRO is not limited to tabular settings and learns a continuous endogenous latent state without a bottleneck. The learnt pre-trained representation $\phi(\cdot)$ is used for policy optimization in offline RL. More details of our algorithm are discussed in Appendix E.3.

Offline RL: Given the learnt representation ϕ, we can then use any existing offline RL algorithm. The performance on the downstream task depends on the robustness of the learnt representation ϕ. For our experiments, we build off from the open source code base accompanying the v-d4rl benchmark (Lu et al., 2022b). We implement the pre-trained representation objectives in a model-free setting, where we use **TD3 + BC** as the baseline offline RL algorithm (**?**). The policy improvement objective for the baseline RL algorithm is given by: $L_\theta(\mathcal{D}) = -\mathbb{E}_{x_t\sim\mathcal{D}_\phi}\left[Q_\psi(s_t, \tilde{a}_t)\right]$ where $s_t = \phi(\text{aug}(x_t))$ is the encoded augmented visual observation, $\tilde{a}_t = \pi_\theta(s_t) + \epsilon$ (action with clipped noise to smooth targets, $\epsilon \sim \text{clip}(\mathcal{N}(0,\sigma^2), -c, c)$). The critic $Q(s, \pi(s))$ is evaluated by a TD loss, and we use re-parameterized gradients through the critic for policy improvement step. The overall loss for policy improvement using the encoded $\mathcal{D}_\phi$ is:

$$\mathcal{L}_\theta(\mathcal{D}) = -\mathbb{E}_{s_t,a_t\sim\mathcal{D}_\phi}\left[\lambda Q_\psi(s_t, \mathbf{a}_t) - (\pi_\theta(s_t) - \mathbf{a}_t)^2\right] \quad (2)$$

For pixel based visual observations, recent work (Lu et al., 2022b) used TD3 algoroithm along with **DrQ + BC**, where it additionally applies the data augmentations on pixel based inputs. DrQ passes the gradients of the critic to learn the encoder, and there are no separate or explicit representation losses other than the critic estimation, for training the encoder in DrQ. We use the same offline experiment pipeline from (Lu et al., 2022a), where representations are pre-trained with ACRO.

2.4. Does ACRO also remove task relevant information?

We produced a small analysis to demonstrate that the agent-centric representation captures information about objects that the agent can affect. Let us consider two variants of a gridworld environment in which actions move the agent in the four directions (left, right, up, and down), and the agent is able to move a block without the block itself having any effect on the agent. In a push variant, the agent moves the block when it moves towards it (moving off the edge of the gridworld wraps onto the other side, to prevent the block from getting trapped in the corners). In a pull variant, the agent moves the block along with itself unless it moves to the outer edge of the grid, which causes the agent to drop

Table 1: **Overview of Properties**. of prior works on representation learning in RL, in particular their robustness to exogenous information. The comparison to ACRO aims to be as generous as possible to the baselines. ✗ is used to indicate a known counterexample for a given property. We compare the properties (i) Time-Independent Exogenous Invariant (ii) Reward-Free (iii) Exogenous Invariant (iv) Non-Expert Policy (v) Full Agent-Centric Representation.

Algorithms	TD3 (DrQ)	CURL	DRIML	DBC	AE	1-Step Inverse	Behavior Cloning	BYOL Explore	**ACRO** (Ours)
Time-Ind. Exo. Inv.	✓	✗	✓	✓	✗	✓	✓	✓	✓
Reward Free	✗	✓	✓	✗	✓	✓	✓	✓	✓
Exogenous Invariant	✗	✗	✗	✓	✗	✓	✓	?	✓
Non-Expert Policy	✓	✓	✓	✓	✓	✓	✗	✓	✓
Full Rep.	✓	✗	✓	✗	✓	✗	✓	✗	✓

the block. We trained ACRO on 500000 samples on each of these environments, and found that the learnt representation captures the state of both the agent and the block, while discarding exogenous noise. Detailed experimental results extensive discussion included in Appendix D.1.

This demonstrates that the agent-centric representation is more extensive than might be imagined at first glance. Anything that influences the actions taken by the policy needs to be captured by the representation in order to predict the first action it took from a pair of representations. As an additional illustration, consider the example of a robot hand grabbing a block, and a pair of states $(s_t, s_{t'})$, where s_t corresponds to the state before the object was grasped and $s_{t'}$ after. The agent-centric representation needs to include information that pertains to the block's position and orientation; otherwise, it would be impossible to predict the actions governing the robot's motion (in the direction of the object) or how its joints are adjusting to grab the object. Note that this would not be required for a simple one-step inverse model, as the robot hand's joint positions in successive states are sufficient to infer what action was taken.

3. Related Work

In Table 1, we list prior works and whether they verify various properties, in particular invariance to exogenous information. An extended discussion on related works is provided in Appendix C.

Inverse Dynamics Models. One-Step Inverse Models predict the action taken conditioned on the previous and resulting observations. This is invariant to exogenous noise but fails to capture the agent-centric latent state (Efroni et al., 2021), as previously discussed in Section 2.3. This can result from inability to capture long-range dependencies or could result from trivial prediction of actions using a dashboard displaying the last action taken, such as the brakelight which turns on after the break is applied on a car (De Haan et al., 2019). Behavior Cloning predicts actions given current state and may also condition on future returns. This is invariant to exogenous noise but can struggle with non-expert policies and generally fails to learn agent-centric latent state. Inverse models predicting sequences of actions, like GLAMOR (Paster et al., 2020) considers an online setting where they learn an action sequence as a sequential multi-step inverse model and rollout via random shooting and re-scoring, using both the inverse-model accuracies and an action-prior distribution. Additional discussion and experimental results comparing ACRO with action sequence prediction, are provided in appendix B.3.

Contrastive Methods. CURL (Augmentation Contrastive, (Laskin et al., 2020)) learns a representation which is invariant to a class of data augmentations while being different across random example pairs. Depending on what augmentations and datasets are used, the learnt representations would generally learn exogenous noise and also fail to capture agent-centric latent states (which could be removed by some augmentations). HOMER and DRIML (Time Contrastive, (Misra et al., 2020), (Mazoure et al., 2020)) learns representations which can discriminate between adjacent observations in a rollout and pairs of random observations. This has been proven to not be invariant to exogenous information and neither can capture the agent-centric latent state (Efroni et al., 2021).

Predictive Models. Autoencoders learn to reconstruct an observation through a representation bottleneck. Generative modeling approaches usually capture all information in the input space which includes both exogenous noise and the agent-centric latent state (Hafner et al., 2019). Wang et al. (2022a;b) showed that a generative model of transition in the observation space can decompose the space into agent-centric state and exogenous information. While this does, in principle, eventually achieve an Agent-Centric representation, it comes at the cost of learning the exogenous representation and its dynamics before discarding the information. BYOL-Explore (Guo et al., 2022) achieved impressive empirical results in online exploration by predicting future representations based on past representations and actions. While this approach can ignore exogenous information, there is no guarantee that it will do so, nor that it will learn the full agent-centric state.

RL with Exogenous Information. Several prior works study the RL with exogenous information problem. In Dietterich et al. (2018); Efroni et al. (2022a;b), the authors consider specific representational assumptions on the underlying model, such as linear dynamics or factorized representation of the exogenous information in observations. Our work focuses on the rich observation setting where the representation itself should be learned. Efroni et al. (2021)

proposes a deterministic path planning algorithm for being invariant to exogenous noise. Unlike their approach which requires interaction with the environment using a tabular-MDP, ACRO is a purely offline algorithm. Lastly, the work of Lamb et al. (2022) suggests an endogenous latent state recovery algorithm through the use of a discretization bottleneck. Their approach is designed to work for MDPs with tabular and small endogenous state space. In contrast, ACRO recovers a continuous counterpart of the endogenous latent state space directly, without the need to construct a tabular-MDP. Hence, it is applicable for larger scale problems. Furthermore, we focus on reward optimization, not only on latent state discovery.

4. Experiments: Offline RL with Exogenous Information

This section provides extensive analysis of representation learning from visual offline data under rich exogenous information (Figure 3). Our experiments aim to understand the effect of exogenous information and if ACRO can truly learn the agent-centric state and thus improve performance in visual offline RL. To this end, we evaluate ACRO against several state of the art representation learning baselines across two axes of added exogenous information: *Temporal Correlation* and *Diversity*, hence characterizing the level of difficulty systematically. We find that under exogenous information in offline RL, the performance of several state of the art representation learning objectives can degrade dramatically.

Two particular challenges in the datasets we explore are the temporal correlation and diversity in the exogenous noise. *Temporal Correlation:* Exogenous noise which lacks temporal correlation (time-independent noise) is relatively easy to filter out in the representation, especially in tasks where the agent-centric latent state has strong temporal correlation. *Diversity:* Similarly for the other axis, if exogenous noise is more diverse, it has a greater impact on the complexity of the subsequently learned representation. For example, if there are only two possible distracting background images, in the worst case the cardinality of a discrete representation is only doubled. On the other hand if there are thousands of possible distracting background images, then the effect on the complexity of representation would be far greater. We primarily categorize our novel visual offline datasets into *three categories* (Figure 3 in appendix provides observations under different exogenous distractors):

- **Easy-Exo**. Exogenous noise with low-diversity and no time correlation. **a)** Visual offline datasets from v-d4rl benchmark (Lu et al., 2022a) without any background distractors; **b)** Distractor setting (Lu et al., 2022b) with a single fixed exogenous image in the background.

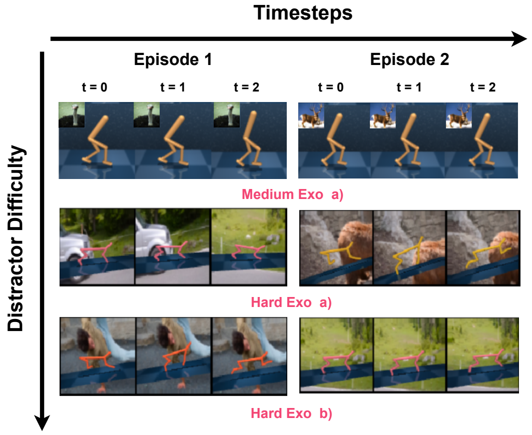


Figure 3: **Examples of Different Categories of Exogenous Information**. Further details of different exogenous information in offline datasets, with visual examples, are provided in Appendix E.1.1.

- **Medium-Exo**. Exogenous noise with either low-diversity or simple time-correlation. **a)** Exogenous image placed in the corner of agent observations, changes per episode; **b)** Exogenous image placed on the side of agent observations, changes per episode; **c)** A single fixed exogenous video playing in the background.
- **Hard-Exo**. Exogenous noise with both high-diversity and rich temporal correlation. **a)** Exogenous image in the background which changes per episode; **b)** Exogenous video in the background which changes per episode; **c)** Exogenous observations of nine agents placed in a grid, but the actions only control one of the agents (see Figure 1).

Experiment Setup. We provide details of each Exogenous Datasets in Appendix E.1.1, along with descriptions for the data collection process in Appendix E.2. Following Fu et al. (2020); Lu et al. (2022a), we release these datasets for future use by the RL community. All experiments involve pre-training the representation, and then freezing it for use in an offline RL algorithm. We use TD3 + BC as the downstream RL algorithm, along with data augmentations (Kostrikov et al., 2020). Experiment setup and implementation details are discussed in Appendix E.3. Additional experimental results are also provided in appendix D.

Baselines. We compare *five* baselines, which are standard for learning representations of visual data. The baselines we consider are: (i) two temporal contrastive learning methods, DRIML (Mazoure et al., 2020) and HOMER (Misra et al., 2020); (ii) a data augmentation method, DrQ (Kostrikov et al., 2020), and a spatial contrastive approach, CURL (Laskin et al., 2020); and (iii) inverse dynamics model learning, *i.e.*, 1-step inverse action prediction (Pathak et al., 2017). We do not con-

Table 2: **Easy-Exo**. Comparison of different representation methods on the standard v-d4rl benchmark, without additional exogenous information. ACRO consistently outperforms baseline methods in visual offline data. Performance plots in Appendix Figure 16. 10 seeds and std. dev. reported.

Environment	Dataset	ACRO	DRIML	HOMER	DrQv2	CURL	1-Step Inverse
Cheetah-Run	Expert	451.0 ± 3.9	330.2 ± 2.9	227.8 ± 1.6	256.9 ± 2.2	213.0 ± 0.6	239.9 ± 0.4
	Medium-Expert	466.0 ± 3.2	399.2 ± 2.5	390.7 ± 1.3	388.1 ± 3.5	328.4 ± 2.1	299.3 ± 0.6
	Medium	528.7 + 0.8	508.5 ± 0.7	518.1 ± 0.4	488.3 ± 0.5	377.0 ± 0.8	400.3 ± 0.4
	Medium-Replay	416.9 ± 0.9	233.3 ± 1.2	333.2 ± 1.2	381.5 ± 1.6	279.4 ± 1.8	272.3 ± .6
Walker-Walk	Expert	924.5 ± 2.2	485.10 ± 4.9	670.55 ± 4.1	888.6 ± 6.0	800.36 ± 2.5	831.5 ± 3.4
	Medium-Expert	914.6 ± 1.8	438.3 ± 3.3	774.5 ± 2.5	906.6 ± 0.9	724.6 ± 4.5	651.8 ± 4.0
	Medium	486.7 ± 0.2	469.4 ± 0.5	485.1 ± 0.7	425.6 ± 1.6	429.0 ± 2.0	389.4 ± 1.1
	Medium-Replay	277.8 ± 0.5	204.3 ± 3.4	318.9 ± 4.0	308.5 ± 1.5	234.8 ± 2.4	146.7 ± 0.7
Humanoid-Walk	Expert	79.9 ± 1.1	17.5 ± 0.1	21.6 ± 0.4	34.1 ± 0.3	28.5 ± 0.2	25.4 ± 0.1
	Medium-Expert	142.4 ± 1.2	26.8 ± 0.2	31.8 ± 0.1	70.8 ± 0.5	63.2 ± 0.9	56.3 ± 0.5
	Medium	103.8 ± 1.8	35.1 ± 0.3	53.8 ± 0.4	96.4 ± 0.9	40.6 ± 0.4	46.7 ± 0.1
	Medium-Replay	197.8 ± 0.5	92.6 ± 0.3	102.7 ± 0.6	121.0 ± 0.4	77.8 ± 0.8	100.7 ± 1.1
Average		415.8	270.0	327.4	363.9	299.7	288.4

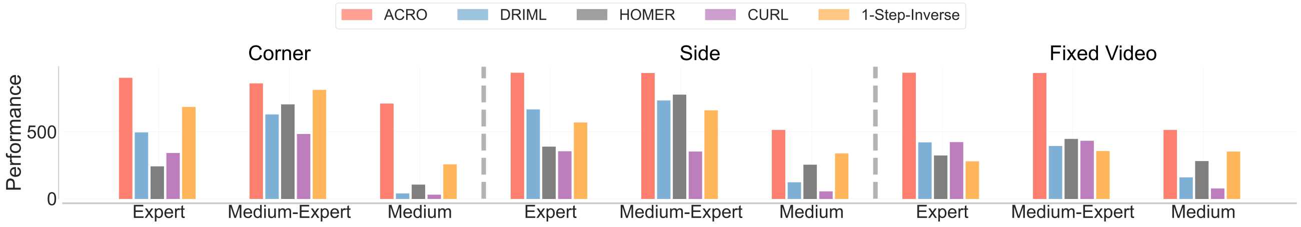


Figure 4: **Medium-Exo Results**. Performance comparison of ACRO with several other baselines, with varying levels of exogenous information settings, either from STL10 dataset (Coates et al., 2011) or fixed video distractors in background during offline data collection. Normalized (averaged) performance plots across different control domains (Humanoid, HalfCheetah and Walker) as we vary the type of offline data collecting policies.

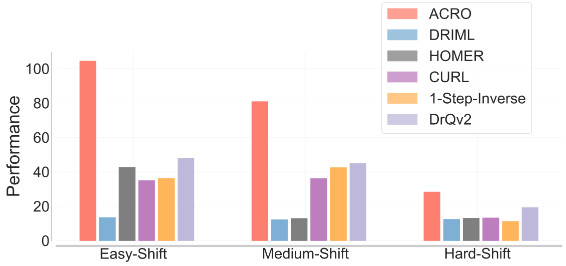


Figure 5: **Normalized results**. across two domains from the v-d4rl distractor suite with varying levels (easy, medium and hard categories) of data shift severity (Lu et al., 2022b).

sider baselines such as SPR (Schwarzer et al., 2020) and SGI (Schwarzer et al., 2021) which work well on the ALE Atari100K benchmark but not on control benchmarks (Tomar et al., 2021). We also include Atari results in Appendix D.2 where representations are pre-trained using ACRO and used over a Decision Transformer (Chen et al., 2021b).

4.1. Easy-Exogenous Information Offline Datasets

Table 2 summarizes results from the v-d4rl benchmark with visual offline data (Lu et al., 2022b). We label this as Easy-Exo since the dataset only contains a blank background without any additional exogenous noise being added. We find that ACRO learns a good agent-centric latent representation from pixel data with no apparent noise in observations, and can lead to effective performance improvements through pre-training representations. Extending results of Easy-Exo with static uncorrelated image background distractors from the v-d4rl benchmark, we see that the performance significantly decreases for all methods, while ACRO can strongly outperform all baselines, with the smallest drop in performance. Figure 5 shows normalized results across two different datasets and domains from v-d4rl. The distractors in this case belong to varying degree of shifts in the data distribution, according to (Lu et al., 2022b).

4.2. Medium-Exogenous Information Offline Datasets

Figure 4 shows normalized results across three domains (cheetah-run, walker-walk, humanoid-walk) for the Medium-Exo setting. Among these, the fixed background video is the hardest task. Most methods underperform on data collected from a medium policy, compared to medium-expert and expert policies. However, ACRO consistently outperforms all methods across datasets and distractor settings. Note the high variability in performance of baselines when changing the type of exogenous information (from corner, to side, to fixed video), while in contrast, ACRO performs similarly for all three settings. This suggests that baseline methods do not learn exogenous-free robust representations, while ACRO remains impervious to it.

4.3. Hard-Exogenous Information Offline Datasets

With correlated exogenous noise in the form of either images or video, we observe that baseline representation objectives can be remarkably broken. Figure 6 shows normalized performance comparisons across different types of datasets (expert, medium-expert, medium) for three different types of HARD-EXO settings. Comparatively, ACRO can be more robust to the hard exogenous distractors, even though as the HARD-EXO types increase in difficulty, the maximum performance reached by all methods can degrade. Among the three HARD-EXO settings, changing video distractors in background during data collection seems to be the hardest, leading to performance drops for most methods. This suggests there is a strong correlation issue between the representation and the video pixels, which breaks when the episode changes, hence leading to the worst scores across the three settings. However, ACRO remains comparatively robust and outperforms all baselines across all the HARD-EXO settings.

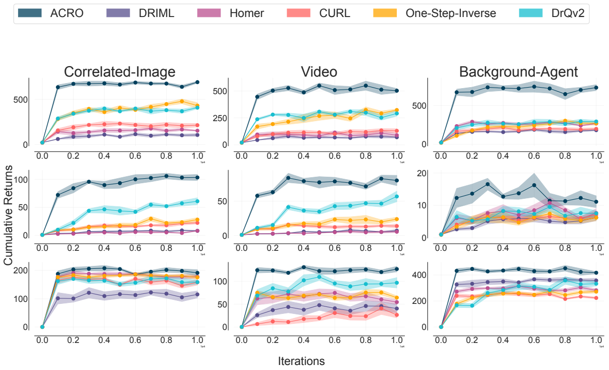


Figure 6: **HARD-EXO Results**. Normalized performance across three datasets: medium, expert, and medium-expert. **First Column**. Time correlated exogenous distractor in background; **Second Column**. Video distractors that changes per episode in background; **Third Column**. Multiple background agent observations as distractors are placed in a grid of agent observation space. Normalized performance plots across different data collecting policies (expert, medium-expert and medium) for three different domains (Cheetah, Humanoid and Walker).

4.4. Method Ablations

We compare ACRO with three of its variations, 1) when $k = 1$, i.e. a standard one-step inverse model; 2) with $\mathbf{x}_{t+k}$ not provided as input, i.e. simply training the representation with a behavior cloning loss; and 3) when $m(k)$ the timestep embedding, is additionally provided as input. Ablations are shown over three different policies: random, medium-replay and expert in Table 3. Appendix D.4 includes results of ACRO on role of data coverage in offline datasets.

Table 3: **Ablations for different policies**. The highlighted cells indicate where each variant fails to match ACRO's performance, hence showing that each component of ACRO is essential for consistently good performance. 5 seeds and std. dev. reported.

ENVIRONMENT	RANDOM	MEDIUM-REPLAY	EXPERT
ACRO	82.9 ± 5.5	228.8 ± 50.1	525.8 ± 89.0
K=1	94.7 ± 7.9	241.0 ± 9.9	187.5 ± 33.8
ONLY x_t	0.5 ± 0.1	229.4 ± 64.7	496.8 ± 100.2
WITH k	43.1 ± 49.5	251.8 ± 15.3	302.2 ± 29.1

For both random and medium-replay policies, $k = 1$ leads to similar results when k is randomly chosen from 1 to 15. ACRO performs much better under an expert policy. We conjecture that the benefits of larger k can only be realized when the policy is of high enough quality to preserve information over long time horizons. Additionally, training a behavior cloning loss performs similarly to ACRO for the medium-replay and expert datasets. However, when the actions come from a random policy, ACRO performs much better, while the behavior cloning ablation collapses completely. This result is analyzed theoretically in Appendix B.2, which shows that ACRO is equivalent to behavior cloning under a deterministic and fixed expert policy, but should be much better otherwise. Adding a k embedding generally degrades performance, although the effect is inconsistent. These results suggest that ACRO is a more well rounded and robust objective than other variants. In appendix D.3 we also include further ablations on how ACRO performs, when compared to predicting a sequence of actions.

4.5. Visualizing Reconstructions from the Decoder

Visualizing Reconstructions. Having learnt a representation, we can train a decoder over it to minimize the reconstruction loss given the original observation. Such reconstructions would therefore measure how much information in the original observation is preserved in the representation, and thus act as a metric for evaluating the quality of representations. We compare such reconstructions in Figure 7 for the cheetah domain where the exogenous noise comes from a video playing in the background. Notably, ACRO is able to remove most background information while keeping the relevant body pose information intact. On the other hand, DRIML performs contrastive comparisons between states in a given trajectory and is not able to remove exogenous information quite as well. DRQ is able to remove exogenous noise but is unable to learn the agent-centric state. Besides such qualitative differences in the learnt representations, we provide quantitative results showing how ACRO learns to remove exogenous information while retaining endogenous information in Appendix D.5.

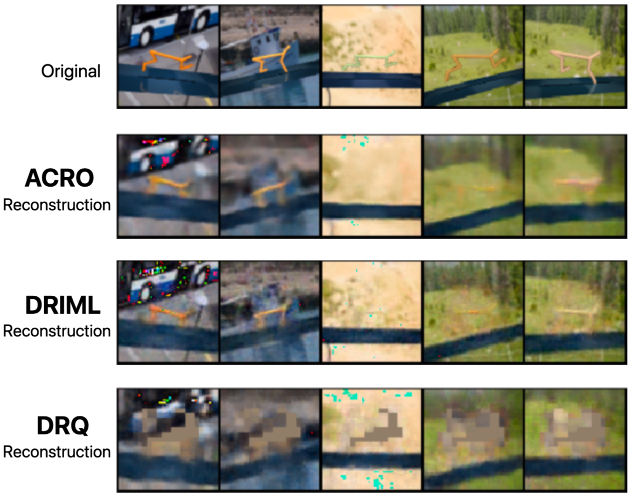


Figure 7: **Reconstructions**. from a decoder with a static background image per episode: **Top-Bottom**: Original, ACRO, DRIML, DrQ. Ideally we want to see that the background is blurry in the reconstruction, demonstrating removal of exogenous noise. This is observed to some extent with both ACRO and DrQ, but not DRIML. At the same time, we want to see that the agent is still visible in the reconstruction, which is mostly the case for ACRO and DRIML, but not DrQ

5. Discussion

In this work, we introduced offline RL datasets with varying difficulties of exogenous information in the observations. Our results showcase that existing representation learning methods can significantly degrade performance for certain types of exogenous noise. We presented ACRO, a pretraining objective for offline RL based on a multi-step inverse prediction model, and showed it is far more robust to exogenous information, both theoretically and empirically.

Limitations and Future Work. Since ACRO does not require reward information for learning representations, it is natural to wonder if data from multiple datasets (e.g., combining random and medium-replay) or different domains (e.g., a transfer learning setting) can be used to train a stronger representation than when using a single dataset. Additionally, under varying transition dynamics across tasks, a model-based counterpart of ACRO might be worth studying. For the domains and tasks considered in this work, it would be interesting to quantify how accurately ACRO recovers the underlying endogenous latent states, while using the latent structure for solving the task objective.

Acknowledgement

The authors would like to thank Akshay Krishnamurthy, Dylan Foster, Romain Laroche, Michael Janner, Anirudh Goyal and Doina Precup for valuable feedback on the draft. Hongyu Zang and Xin Li were partially supported by NSFC under Grant 62276024, 92270125, and U19B2020.

References

Agarwal, R., Schuurmans, D., and Norouzi, M. An optimistic perspective on offline reinforcement learning. In International Conference on Machine Learning, 2020.

An, G., Moon, S., Kim, J., and Song, H. O. Uncertainty-based offline reinforcement learning with diversified q-ensemble. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, pp. 7436–7447, 2021.

Antos, A., Szepesvári, C., and Munos, R. Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. Machine Learning, 71(1):89–129, 2008.

Buckman, J., Gelada, C., and Bellemare, M. G. The importance of pessimism in fixed-dataset policy optimization. In 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net, 2021.

Castro, P. S., Kastner, T., Panangaden, P., and Rowland, M. Mico: Improved representations via sampling-based state similarity for markov decision processes. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, pp. 30113–30126, 2021.

Chen, C., Chen, X., Toyer, S., Wild, C., Emmons, S., Fischer, I., Lee, K., Alex, N., Wang, S. H., Luo, P., Russell, S., Abbeel, P., and Shah, R. An empirical investigation of representation learning for imitation. In Vanschoren, J. and Yeung, S. (eds.), Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual, 2021a.

Chen, J. and Jiang, N. Information-theoretic considerations in batch reinforcement learning. In International Conference on Machine Learning, pp. 1042–1051. PMLR, 2019.

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 Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), Advances in Neural Information Processing Systems, volume 34, pp. 15084–15097. Curran Associates, Inc., 2021b. URL https://proceedings.neurips.cc/paper/2021/file/7f489f642a0ddb10272b5c31057f0663-Paper.pdf.

Coates, A., Ng, A., and Lee, H. An analysis of single-layer networks in unsupervised feature learning. In Gordon, G., Dunson, D., and Dudík, M. (eds.), Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, volume 15 of Proceedings of Machine Learning Research, pp. 215–223, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/coates11a.html.

De Haan, P., Jayaraman, D., and Levine, S. Causal confusion in imitation learning. Advances in Neural Information Processing Systems, 32, 2019.

Dietterich, T., Trimponias, G., and Chen, Z. Discovering and removing exogenous state variables and rewards for reinforcement learning. In International Conference on Machine Learning, pp. 1262–1270. PMLR, 2018.

Ebert, F., Finn, C., Dasari, S., Xie, A., Lee, A. X., and Levine, S. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. CoRR, abs/1812.00568, 2018. URL http://arxiv.org/abs/1812.00568.

Efroni, Y., Misra, D., Krishnamurthy, A., Agarwal, A., and Langford, J. Provably filtering exogenous distractors using multistep inverse dynamics. In International Conference on Learning Representations, 2021.

Efroni, Y., Foster, D. J., Misra, D., Krishnamurthy, A., and Langford, J. Sample-efficient reinforcement learning in the presence of exogenous information. In Loh, P. and Raginsky, M. (eds.), Conference on Learning Theory, 2-5 July 2022, London, UK, volume 178 of Proceedings of Machine Learning Research, pp. 5062–5127. PMLR, 2022a. URL https://proceedings.mlr.press/v178/efroni22a.html.

Efroni, Y., Kakade, S., Krishnamurthy, A., and Zhang, C. Sparsity in partially controllable linear systems. In International Conference on Machine Learning, pp. 5851–5860. PMLR, 2022b.

Foster, D. J., Krishnamurthy, A., Simchi-Levi, D., and Xu, Y. Offline reinforcement learning: Fundamental barriers for value function approximation. arXiv preprint arXiv:2111.10919, 2021.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4rl: Datasets for deep data-driven reinforcement learning. arXiv preprint arXiv:2004.07219, 2020.

Fujimoto, S. and Gu, S. S. A minimalist approach to offline reinforcement learning. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, pp. 20132–20145, 2021.

Fujimoto, S., Meger, D., and Precup, D. Off-policy deep reinforcement learning without exploration. In Chaudhuri, K. and Salakhutdinov, R. (eds.), Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, volume 97 of Proceedings of Machine Learning Research, pp. 2052–2062. PMLR, 2019.

Ghosh, D. and Bellemare, M. G. Representations for stable off-policy reinforcement learning. In Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event, volume 119 of Proceedings of Machine Learning Research, pp. 3556–3565. PMLR, 2020.

Guo, Z. D., Thakoor, S., Pîslar, M., Pires, B. A., Altché, F., Tallec, C., Saade, A., Calandriello, D., Grill, J.-B., Tang, Y., Valko, M., Munos, R., Azar, M. G., and Piot, B. Byol-explore: Exploration by bootstrapped prediction. arXiv preprint arXiv: Arxiv-2206.08332, 2022.

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. Dream to control: Learning behaviors by latent imagination. arXiv preprint arXiv:1912.01603, 2019.

Hutter, M. and Hansen, S. Uniqueness and complexity of inverse mdp models. arXiv preprint arXiv:2206.01192, 2022.

Jaques, N., Ghandeharioun, A., Shen, J. H., Ferguson, C., Lapedriza, À., Jones, N., Gu, S., and Picard, R. W. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. CoRR, abs/1907.00456, 2019. URL http://arxiv.org/abs/1907.00456.

Kidambi, R., Rajeswaran, A., Netrapalli, P., and Joachims, T. Morel: Model-based offline reinforcement learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), Advances in Neural Information

Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.

Kostrikov, I., Yarats, D., and Fergus, R. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. arXiv preprint arXiv:2004.13649, 2020.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009.

Kumar, A., Fu, J., Soh, M., Tucker, G., and Levine, S. Stabilizing off-policy q-learning via bootstrapping error reduction. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pp. 11761–11771, 2019.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.

Lamb, A., Islam, R., Efroni, Y., Didolkar, A., Misra, D., Foster, D., Molu, L., Chari, R., Krishnamurthy, A., and Langford, J. Guaranteed discovery of controllable latent states with multi-step inverse models. arXiv preprint arXiv:2207.08229, 2022.

Lange, S., Gabel, T., and Riedmiller, M. A. Batch reinforcement learning. In Reinforcement Learning, 2012.

Laroche, R., Trichelair, P., and des Combes, R. T. Safe policy improvement with baseline bootstrapping. In Chaudhuri, K. and Salakhutdinov, R. (eds.), Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, volume 97 of Proceedings of Machine Learning Research, pp. 3652–3661. PMLR, 2019.

Laskin, M., Srinivas, A., and Abbeel, P. Curl: Contrastive unsupervised representations for reinforcement learning. In International Conference on Machine Learning, pp. 5639–5650. PMLR, 2020.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. CoRR, abs/2005.01643, 2020. URL https://arxiv.org/abs/2005.01643.

Lu, C., Ball, P. J., Rudner, T. G., Parker-Holder, J., Osborne, M. A., and Teh, Y. W. Challenges and opportunities in offline reinforcement learning from visual observations. arXiv preprint arXiv:2206.04779, 2022a.

Lu, C., Ball, P. J., Rudner, T. G. J., Parker-Holder, J., Osborne, M. A., and Teh, Y. W. Challenges and opportunities in offline reinforcement learning from visual observations. CoRR, abs/2206.04779, 2022b. doi: 10.48550/arXiv.2206.04779. URL https://doi.org/10.48550/arXiv.2206.04779.

Mazoure, B., Tachet des Combes, R., Doan, T. L., Bachman, P., and Hjelm, R. D. Deep reinforcement and infomax learning. Advances in Neural Information Processing Systems, 33:3686–3698, 2020.

Misra, D., Henaff, M., Krishnamurthy, A., and Langford, J. Kinematic state abstraction and provably efficient rich-observation reinforcement learning. In International conference on machine learning, pp. 6961–6971. PMLR, 2020.

Munos, R. Error bounds for approximate policy iteration. In ICML, volume 3, pp. 560–567, 2003.

Munos, R. and Szepesvári, C. Finite-time bounds for fitted value iteration. Journal of Machine Learning Research, 9(5), 2008.

Nachum, O. and Yang, M. Provable representation learning for imitation with contrastive fourier features. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, pp. 30100–30112, 2021.

Nadjahi, K., Laroche, R., and Tachet des Combes, R. Safe policy improvement with soft baseline bootstrapping. In Proceedings of the 17th European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD), 2019.

Paster, K., McIlraith, S. A., and Ba, J. Planning from pixels using inverse dynamics models. arXiv preprint arXiv:2012.02419, 2020.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In Precup, D. and Teh, Y. W. (eds.), Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, volume 70 of Proceedings of Machine Learning Research, pp. 2778–2787. PMLR, 2017. URL http://proceedings.mlr.press/v70/pathak17a.html.

Rajeswaran, A., Kumar, V., Gupta, A., Vezzani, G., Schulman, J., Todorov, E., and Levine, S. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In Kress-Gazit, H., Srinivasa, S. S., Howard, T., and Atanasov, N. (eds.), Robotics: Science and Systems XIV, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, June 26-30, 2018, 2018.

Schwarzer, M., Anand, A., Goel, R., Hjelm, R. D., Courville, A., and Bachman, P. Data-efficient reinforcement learning with self-predictive representations. arXiv preprint arXiv:2007.05929, 2020.

Schwarzer, M., Rajkumar, N., Noukhovitch, M., Anand, A., Charlin, L., Hjelm, R. D., Bachman, P., and Courville, A. C. Pretraining representations for data-efficient reinforcement learning. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, pp. 12686–12699, 2021.

Simão, T. D., Laroche, R., and Tachet des Combes, R. Safe policy improvement with estimated baseline bootstrapping. In Proceedings of the 19th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), 2020.

Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., Casas, D. d. L., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A., et al. Deepmind control suite. arXiv preprint arXiv:1801.00690, 2018.

Tomar, M., Mishra, U. A., Zhang, A., and Taylor, M. E. Learning representations for pixel-based control: What matters and why? arXiv preprint arXiv:2111.07775, 2021.

Uehara, M., Zhang, X., and Sun, W. Representation learning for online and offline RL in low-rank mdps. CoRR, abs/2110.04652, 2021. URL https://arxiv.org/abs/2110.04652.

Van Den Oord, A., Vinyals, O., et al. Neural discrete representation learning. Advances in neural information processing systems, 30, 2017.

Wang, T., Du, S., Torralba, A., Isola, P., Zhang, A., and Tian, Y. Denoised mdps: Learning world models better than the world itself. In International Conference on Machine Learning, pp. 22591–22612. PMLR, 2022a.

Wang, Z., Xiao, X., Xu, Z., Zhu, Y., and Stone, P. Causal dynamics learning for task-independent state abstraction. arXiv preprint arXiv:2206.13452, 2022b.

Yang, M. and Nachum, O. Representation matters: Offline pretraining for sequential decision making. In Meila, M. and Zhang, T. (eds.), Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event, volume 139 of Proceedings of Machine Learning Research, pp. 11784–11794. PMLR, 2021. URL http://proceedings.mlr.press/v139/yang21h.html.

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 Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.

Zang, H., Li, X., and Wang, M. Simsr: Simple distance-based state representations for deep reinforcement learning. In Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022, pp. 8997–9005. AAAI Press, 2022.

Zang, H., Li, X., Yu, J., Liu, C., Islam, R., des Combes, R. T., and Laroche, R. Behavior prior representation learning for offline reinforcement learning. In The Eleventh International Conference on Learning Representations, 2023. URL https://openreview.net/forum?id=hQ4K9Bf4G2B.

Zhang, A., McAllister, R., Calandra, R., Gal, Y., and Levine, S. Learning invariant representations for reinforcement learning without reconstruction. arXiv preprint arXiv:2006.10742, 2020.

Zhang, A., McAllister, R. T., Calandra, R., Gal, Y., and Levine, S. Learning invariant representations for reinforcement learning without reconstruction. In 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net, 2021a. URL https://openreview.net/forum?id=-2FCwDKRREu.

Zhang, C., Kuppannagari, S. R., and Prasanna, V. K. BRAC+: improved behavior regularized actor critic for offline reinforcement learning. In Balasubramanian, V. N. and Tsang, I. W. (eds.), Asian Conference on Machine Learning, ACML 2021, 17-19 November 2021, Virtual Event, volume 157 of Proceedings of Machine Learning Research, pp. 204–219. PMLR, 2021b.

Appendix

A. Benefits of Exogenous Invariant Representation in Offline RL

A.1. Proof of Proposition 2.2.

We need to show that for any $f \in \mathcal{F}(\phi_\star)$ and x, a, it holds that

$$R(x,a) + \mathbb{E}_{x' \sim T(\cdot \mid x,a)}[\max_{a'} f(x',a')] \tag{3}$$

is contained in $\mathcal{F}(\phi_\star)$. Since the reward is a function of the agent controller representation only, and since $f \in \mathcal{F}(\phi_\star)$, equation 3 can be written as:

$$\begin{aligned}
&R(x,a) + \mathbb{E}_{x' \sim T(\cdot \mid x,a)}[\max_{a'} f(x',a')] \\
&= R(\phi_\star(x),a) + \mathbb{E}_{x' \sim T(\cdot \mid x,a)}[\max_{a'} f(\phi_\star(x'),a')] \\
&= R(\phi_\star(x),a) + \mathbb{E}_{x' \sim T(\cdot \mid \phi_\star(x),a)}[\max_{a'} f(\phi_\star(x'),a')]. \qquad (4)
\end{aligned}$$

The first relation holds since $f \in \mathcal{F}(\phi_\star)$ and by the assumption on the reward function (that it is a function of the endogenous states). The second relation holds by

$$\begin{aligned}
&\mathbb{E}_{x' \sim T(\cdot \mid x,a)}[\max_{a'} f(\phi_\star(x'),a')] \\
&\overset{\text{(a)}}{=} \sum_{s',e'} \sum_{x' \in \text{supp} q(x' \mid s',e')} q(x' \mid \phi_\star(x'), \phi_{\star,e}(x')) T(s' \mid \phi_\star(x),a) T_e(e' \mid \phi_{\star,e}(x)) f(s',a') \\
&\overset{\text{(b)}}{=} \sum_{s'} T(s' \mid \phi_\star(x),a) f(s',a') \sum_{e'} T_e(e' \mid \phi_{\star,e}(x)) \\
&\overset{\text{(c)}}{=} \sum_{s'} T(s' \mid \phi_\star(x),a) f(s',a'),
\end{aligned}$$

where (a) holds by the Ex-BMDP transition model assumption,

$$T(x' \mid x,a) = q(x' \mid \phi_\star(x'), \phi_{\star,e}(x')) T(\phi_\star(x') \mid \phi_\star(x),a) T_e(\phi_{\star,e}(x') \mid \phi_{\star,e}(x)),$$

(b) and (c) hold by marginalizing over x' and e'. This establishes equation 4 and the proposition: the function $R(\phi_\star(x),a) + \mathbb{E}_{x' \sim T(\cdot \mid \phi_\star(x),a)}[\max_{a'} f(\phi_\star(x'),a')]$ is contained within $\mathcal{F}(\phi_\star)$ since it only depends on $\phi_\star$.

A.2. Proof of Proposition 2.3.

Consider an Ex-BMDP with one action a where the agent controller representation is trivial and has a single fixed state (the agent has no ability to affect the dynamics). We will establish a counter-example by constructing a tabular-MDP. Because tabular-MDP is a special case of a more general MDP with continuous states, this will also establish a counterexample for the more general non-tabular setting considered in the paper.

Let the observations and dynamics be given has follows. The observation is a 2-dimensional vector $x = (x(1), x(2))$ where $x(1), x(2) \in \{0,1\}$. The dynamics is deterministic and its time evoluation is given as follows:

$$\begin{aligned}
x_{t+1}(1) &= x_t(1) \oplus x_t(2) \\
x_{t+1}(2) &= x_t(2),
\end{aligned}$$

where $\oplus$ is the XOR operation. In this case, the transition model is given by $T(x' \mid x,a) = T(x' \mid x)$ and $\phi_\star = \{\, s_0\}$ where s_0 is a single state; since the observations are not controllable the controller representation maps all observations to a unique state. Further, assume that the reward function is 0 for all observations.

Assume that $\phi(x) = (x_1)$, i.e., the representation ignores the second feature x_2. This representation is more refined than $\phi_\star$ since the latter maps all observations into the same state. Consider the tabular Q function class on top of this representation

$\mathcal{Q}_{N=2}$, and consider $Q \in \mathcal{Q}_{N=2}$ given as follows

$$\begin{aligned} Q(x_1 = 1) &= 1 \\ Q(x_1 = 0) &= 0. \end{aligned}$$

We now show that $\mathcal{T}Q$ is not contained in $\mathcal{Q}_{N=2}$. According to the construction of the transition model, it holds that

$$\begin{aligned} (TQ)(x_1 = 1, x_2 = 1) &= 0 \\ (TQ)(x_1 = 0, x_2 = 1) &= 1 \\ (TQ)(x_1 = 1, x_2 = 0) &= 1 \\ (TQ)(x_1 = 0, x_2 = 0) &= 0. \end{aligned}$$

This function cannot be represented by a function from $\mathcal{Q}_{N=2}$; we cannot represent (TQ) since it is not a mapping of the form $x_1 \to \mathbb{R}$ by the fact that, e.g.,

$$(TQ)(x_1 = 1, x_2 = 1) \neq (TQ)(x_1 = 1, x_2 = 0).$$

Meaning, it depends on the value of x_2.

B. Theory on Learning Exogenous-Free Representations with ACRO

B.1. Multi-Step Inverse Model Invariance Proof

For any $k > 0$, consider $x, x' \in \mathcal{X}$ such that x and x' are separated by k steps. Both proofs first use bayes theorem, then apply the factorized transition dynamics, and then eliminate terms shared in the numerator and denominator. This proof is essentially the same as lemmas found in (Efroni et al., 2021; Lamb et al., 2022), but is presented here for clarity.

The multi-step inverse model is invariant to exogenous noise. For any exo-free policy $\pi : \mathcal{X} \to \mathcal{A}$, for all $a_t \in \mathcal{A}$, and $(x_t, x_{t+k}) \in \texttt{supp}\, \mathbb{P}_\pi(X_t, X_{t+k})$:

$$\mathbb{P}_\pi(a_t \mid x_t, x_{t+k}) = \mathbb{P}_\pi(a_t \mid \phi_\star(x_t), \phi_\star(x_{t+k})) \tag{5}$$

Proof.

$$\begin{aligned} &\mathbb{P}_{\pi,\mu}(a \mid x', x) \\ &\overset{(a)}{=} \frac{\mathbb{P}_{\pi,\mu}(x' \mid x, a)\mathbb{P}_{\pi,\mu}(a \mid x)}{\sum_{a'} \mathbb{P}_{\pi,\mu}(x' \mid x, a')} \\ &\overset{(b)}{=} \frac{\mathbb{P}_{\pi,\mu}(x' \mid x, a)\pi(a \mid \phi_\star(x)))}{\sum_{a'} \mathbb{P}_{\pi,\mu}(x' \mid x, a')\pi(a' \mid \phi_\star(x))} \\ &\overset{(c)}{=} \frac{q(x' \mid \phi_\star(x'), \phi_{\star,e}(x'))\mathbb{P}_{\pi,\mu}(\phi_\star(x') \mid \phi_\star(x), a)\mathbb{P}_{\pi,\mu}(\phi_{\star,e}(x') \mid \phi_{\star,e}(x))\pi(a \mid \phi_\star(x))}{\sum_{a'} q(x' \mid \phi_\star(x'), \phi_{\star,e}(x'))\mathbb{P}_{\pi,\mu}(\phi_\star(x') \mid \phi_\star(x), a')\mathbb{P}_{\pi,\mu}(\phi_{\star,e}(x') \mid \phi_{\star,e}(x))\pi(a' \mid \phi_\star(x))} \\ &= \frac{\mathbb{P}_{\pi,\mu}(\phi_\star(x') \mid \phi_\star(x), a)\pi(a \mid \phi_\star(x))}{\sum_{a'} \mathbb{P}_{\pi,\mu}(\phi_\star(x') \mid \phi_\star(x), a')\pi(a' \mid \phi_\star(x))}. \end{aligned}$$

□

Relation (a) holds by Bayes' theorem. Relation (b) holds by the assumption that π is uniformly random (in the first proof) or exo-free (in the second proof). Relation (c) holds by the factorization property. Thus, $\mathbb{P}_{\pi,\mu}(a \mid x', x) = \mathbb{P}_{\pi,\mu}(a \mid \phi_\star(x'), \phi_\star(x))$, and is constant upon changing the observation while fixing the agent-centric state.

B.2. Connection Between ACRO and Behavior Cloning

In the special case where all data is collected under a fixed deterministic, exogenous-free policy, ACRO and behavior cloning become equivalent. This case can still be non-trivial, if the start state of the episode is stochastic or if the environment dynamics are stochastic.

Lemma B.1. *Under fixed, deterministic, and exo-free policy $\hat{\pi} : \mathcal{X} \rightarrow \mathcal{A}$, multi-step inverse model is equivalent to behavior cloning. For all $a_t \in \mathcal{A}$, and (x_t, x_{t+k}) such that $\mathbb{P}_{\hat{\pi}}(X_t = x_t, X_{t+k} = x_{t+k}) > 0$ we have:*

$$\mathbb{P}_{\hat{\pi}}(a_t \mid \phi_\star(x_t), \phi_\star(x_{t+k})) = \mathbb{P}_{\hat{\pi}}(a_t \mid \phi_\star(x_t)) \tag{6}$$

The proof of this claim is simply that behavior cloning is already able to predict actions perfectly in this special case, so there can be no benefit to conditioning on future observations.

Proof. By the assumption of the deterministic exo-free policy, we have that $\mathbb{P}_{\hat{\pi}}(a_t = \hat{a}(\phi_\star(x_t)) \mid \phi_\star(x_t)) = 1$ where $\hat{a} : \mathcal{S} \rightarrow A$ is a function mapping the latent state to the action.

Using bayes theorem we write:

$$\mathbb{P}_{\hat{\pi}}(a_t \mid \phi_\star(x_t), \phi_\star(x_{t+k})) = \frac{\mathbb{P}_{\hat{\pi}}(\phi_\star(x_{t+k} \mid \phi_\star(x_t), a_t)\mathbb{P}_{\hat{\pi}}(a_t \mid \phi_\star(x_t))}{\sum_{a''} \mathbb{P}_{\hat{\pi}}(\phi_\star(x_{t+k} \mid \phi_\star(x_t), a'')\mathbb{P}_{\hat{\pi}}(a'' \mid \phi_\star(x_t))} \tag{7}$$

For any examples in the dataset and for all $a' \in A$:

Case 1: $a' = \hat{a}(\phi_\star(x_t))$. It holds that

$$\begin{aligned}
\mathbb{P}_{\hat{\pi}}(a_t \mid \phi_\star(x_t), \phi_\star(x_{t+k})) &= \frac{\mathbb{P}_{\hat{\pi}}(\phi_\star(x_{t+k} \mid \phi_\star(x_t), a_t)\mathbb{P}_{\hat{\pi}}(a_t \mid \phi_\star(x_t))}{\sum_{a''} \mathbb{P}_{\hat{\pi}}(\phi_\star(x_{t+k} \mid \phi_\star(x_t), a'')\mathbb{P}_{\hat{\pi}}(a'' \mid \phi_\star(x_t))} \\
\mathbb{P}_{\hat{\pi}}(a_t \mid \phi_\star(x_t), \phi_\star(x_{t+k})) &= \frac{\mathbb{P}_{\hat{\pi}}(\phi_\star(x_{t+k} \mid \phi_\star(x_t), a_t = a')}{\mathbb{P}_{\hat{\pi}}(\phi_\star(x_{t+k} \mid \phi_\star(x_t), a_t = a')} \\
\mathbb{P}_{\hat{\pi}}(a_t \mid \phi_\star(x_t), \phi_\star(x_{t+k})) &= 1.
\end{aligned}$$

On the other hand, it holds that $\mathbb{P}_{\hat{\pi}}(a_t = a' \mid \phi_\star(x_t)) = 1$ since $a' = \hat{a}(\phi_\star(x_t))$. Hence, for this case, the claim holds true.

Case 2: $a' \neq \hat{a}(\phi_\star(x_t))$. It holds that

$$\begin{aligned}
\mathbb{P}_{\hat{\pi}}(a_t \mid \phi_\star(x_t), \phi_\star(x_{t+k})) &= \frac{\mathbb{P}_{\hat{\pi}}(\phi_\star(x_{t+k} \mid \phi_\star(x_t), a_t)\mathbb{P}_{\hat{\pi}}(a_t \mid \phi_\star(x_t))}{\sum_{a''} \mathbb{P}_{\hat{\pi}}(\phi_\star(x_{t+k} \mid \phi_\star(x_t), a'')\mathbb{P}_{\hat{\pi}}(a'' \mid \phi_\star(x_t))} \\
\mathbb{P}_{\hat{\pi}}(a_t \mid \phi_\star(x_t), \phi_\star(x_{t+k})) &= \frac{0}{\mathbb{P}_{\hat{\pi}}(\phi_\star(x_{t+k} \mid \phi_\star(x_t), a_t = \hat{a}(\phi_\star(x_t)))} \\
\mathbb{P}_{\hat{\pi}}(a_t \mid \phi_\star(x_t), \phi_\star(x_{t+k})) &= 0.
\end{aligned}$$

On the other hand, it holds that $\mathbb{P}_{\hat{\pi}}(a_t = a' \mid \phi_\star(x_t)) = 0$ since $a' \neq \hat{a}(\phi_\star(x_t))$. Hence, for this case the claim also holds true. This concludes the proof since the two distributions are equal in both cases. □

B.3. Predicting the First Action vs. Predicting Action Sequences

In ACRO we only predict the first action from x_t to x_{t+k} rather than predicting the entire action sequence. In an environment with deterministic dynamics, we will prove that these two approaches are asymptotically equivalent. For the proof we will also make a stronger assumption that the dynamics are deterministic in the learned latent space, i.e. for the learned encoder ϕ, there exists a function f such that: $\phi(x_j) = f(\phi(x_t), a_{t:j})$. This assumption will hold for the optimal ϕ, and it is also likely to be empirically true since ϕ is a high-dimensional continuous latent state, thus no two points are likely to have exactly the same representation. In a stochastic environment, the two approaches are different, but we will provide a counter-example to make the case against predicting action sequences.

The ACRO objective optimizes the following

$$\phi_\star \in \arg\max_{\phi \in \Phi} \mathop{\mathbb{E}}_{t \sim U(0,N)} \mathop{\mathbb{E}}_{k \sim U(0,K)} \log\left(\mathbb{P}(a_t \mid \phi(x_t), \phi(x_{t+k}))\right). \tag{8}$$

The k step action sequence prediction approach optimizes:

$$\phi_\star \in \arg\max_{\phi \in \Phi} \mathop{\mathbb{E}}_{t \sim U(0,N)} \mathop{\mathbb{E}}_{k \sim U(0,K)} \log \left(\mathbb{P}(a_t, \ldots a_{t+k} \mid \phi(x_t), \phi(x_{t+k})) \right). \tag{9}$$

B.3.1. Deterministic Dynamics

$$\mathbb{P}(a_{t:t+k} \mid \phi(x_t), \phi(x_{t+k})) = \prod_{j=t}^{t+k} \mathbb{P}(a_j \mid \phi(x_t), \phi(x_{t+k}), a_{t:j}) \tag{10}$$

By the assumption of deterministic dynamics in the latent space:

$$\mathbb{P}(a_{t:t+k} \mid \phi(x_t), \phi(x_{t+k})) = \prod_{j=t}^{t+k} \mathbb{P}(a_j \mid \phi(x_t), a_{t:j}, \phi(x_{t+k})) \tag{11}$$

After applying the markov assumption as we have assumed an MDP:

$$\mathbb{P}(a_{t:t+k} \mid \phi(x_t), \phi(x_{t+k})) = \prod_{j=t}^{t+k} \mathbb{P}(a_j \mid \phi(x_j), \phi(x_{t+k})) \tag{12}$$

Now we can put this back into the k-step action sequence prediction problem:

$$\phi_\star \in \arg\max_{\phi \in \Phi} \mathop{\mathbb{E}}_{t \sim U(0,N)} \mathop{\mathbb{E}}_{k \sim U(0,K)} \log \left(\prod_{j=t}^{t+k} \mathbb{P}(a_j \mid \phi(x_j), \phi(x_{t+k})) \right) \tag{13}$$

$$\phi_\star \in \arg\max_{\phi \in \Phi} \mathop{\mathbb{E}}_{t \sim U(0,N)} \mathop{\mathbb{E}}_{k \sim U(0,K)} \sum_{j=t}^{t+k} \log \left(\mathbb{P}(a_j \mid \phi(x_j), \phi(x_{t+k})) \right) \tag{14}$$

$$\phi_\star \in \arg\max_{\phi \in \Phi} \mathop{\mathbb{E}}_{t \sim U(0,N)} \mathop{\mathbb{E}}_{k \sim U(0,K)} \mathop{\mathbb{E}}_{j \sim U(t,t+k)} \log \left(\mathbb{P}(a_j \mid \phi(x_j), \phi(x_{t+k})) \right) \tag{15}$$

When $N \gg k$, the distributions of t and j converge, and we can write:

$$\phi_\star \in \arg\max_{\phi \in \Phi} \mathop{\mathbb{E}}_{t \sim U(0,N)} \mathop{\mathbb{E}}_{k \sim U(0,K)} \log \left(\mathbb{P}(a_t \mid \phi(x_t), \phi(x_{t+k})) \right) \tag{16}$$

which we can see is the same as the first-action prediction objective that ACRO optimizes.

B.3.2. Stochastic Dynamics

If the environment has stochastic dynamics, predicting future actions to reach a goal state without conditioning on the preceding observations is very difficult. This is because what action needs to be taken may depend on what actually happened in the environment. For example, if I'm playing a video game, and there is a small chance that the game pauses for one minute, the actions will need to depend on whether the pause occurred. We can construct a stochastic environment in which every action following the first action is completely unpredictable. We can imagine an environment where there is some information in the observation space which is set randomly on every step and indicates how the agent's controls are

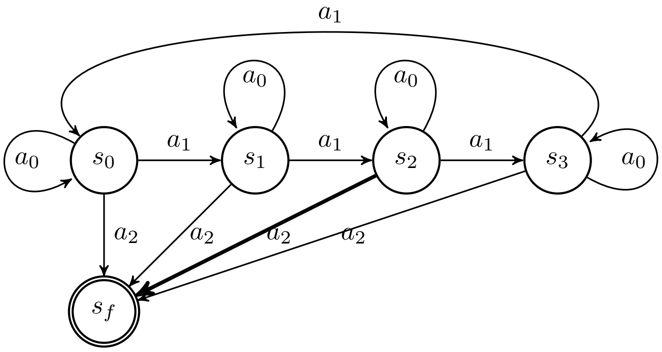


Figure 8: An MDP where one-step models can fail.

randomly permuted for that step. In principle, the first-action predictor can easily use this information to adapt what actions it predicts, and can obtain its original accuracy given sufficient model capacity. On the other hand, the action-sequence predictor ($\mathbb{P}(a_t, \dots a_{t+k} \mid \phi(x_t), \phi(x_{t+k}))$) will only be able to predict the first action well, and can have no better than uniformly random accuracy at predicting the remaining actions. This is because only $\phi(x_t)$ contains the information about what has happened in the environment which is necessary for control, the history of past actions do not contain the necessary information. In this example, predicting the sequence of actions makes the task much noisier, while providing no additional signal for the model.

B.4. Counterexample for One-Step Inverse Models

In this section, we build an MDP and a dataset of trajectories in that MDP where a one-step model will fail to allow the learning of the optimal policy. The MDP can be seen in Figure 8, all transitions are deterministic, and yield 0 reward, apart for action a_2 in state s_2, which gives a reward of 1 (bold arrow in the graph). The final state s_f denotes the termination of the trajectory

We consider the dataset comprised of the following five trajectories, formatted as $\langle s_t, a_t, r_t, s_{t+1} \dots \rangle$:

$$\mathcal{D} = \{ \underbrace{\langle s_0, a_0, 0, s_0, a_0, 0, s_0, a_2, 0, s_f \rangle}_{\tau_1}, \tag{17}$$

$$\underbrace{\langle s_0, a_1, 0, s_1, a_0, 0, s_1, a_2, 0, s_f \rangle}_{\tau_2}, \tag{18}$$

$$\underbrace{\langle s_0, a_1, 0, s_1, a_1, 0, s_2, a_0, 0, s_2, a_2, \mathbf{1}, s_f \rangle}_{\tau_3}, \tag{19}$$

$$\underbrace{\langle s_0, a_1, 0, s_1, a_1, 0, s_2, a_1, 0, s_3, a_0, 0, s_3, a_2, 0, s_f \rangle}_{\tau_4}, \tag{20}$$

$$\underbrace{\langle s_0, a_1, 0, s_1, a_1, 0, s_2, a_1, 0, s_3, a_1, 0, s_0, a_2, 0, s_f \rangle}_{\tau_5} \}. \tag{21}$$

τ_1 loops twice in s_0 and then terminates. The other four trajectories reach s_1, s_2, s_3 and s_0 by taking a_1 a minimal number of times, then loop once (except τ_5), and terminate. We note that this dataset covers the full state and action space.

It is possible to reach a 0 loss with a one-step inverse model p_1 built on top of the following representation ϕ from $\mathcal{S}$ to $\mathcal{X} = \{x_0, x_1, x_f\}$: $\phi(s_0) = \phi(s_2) = x_0$, $\phi(s_1) = \phi(s_3) = x_1$ and $\phi(s_f) = x_f$:

$$p_1(a|x_0, x_0) = \delta_{a=a_0}, \tag{22}$$

$$p_1(a|x_0, x_1) = \delta_{a=a_1}, \tag{23}$$

$$p_1(a|x_0, x_f) = \delta_{a=a_2}. \tag{24}$$

Now, once projected onto $\mathcal{X}$, the dataset becomes:

$$\mathcal{D} = \{ \underbrace{\langle x_0, a_0, 0, x_0, a_0, 0, x_0, a_2, 0, x_f \rangle}_{\tau_1}, \tag{25}$$

$$\underbrace{\langle x_0, a_1, 0, x_1, a_0, 0, x_1, a_2, 0, x_f \rangle}_{\tau_2}, \tag{26}$$

$$\underbrace{\langle x_0, a_1, 0, x_1, a_1, 0, x_0, a_0, 0, x_0, a_2, \mathbf{1}, x_f \rangle}_{\tau_3}, \tag{27}$$

$$\underbrace{\langle x_0, a_1, 0, x_1, a_1, 0, x_0, a_1, 0, x_1, a_0, 0, x_1, a_2, 0, x_f \rangle}_{\tau_4}, \tag{28}$$

$$\underbrace{\langle x_0, a_1, 0, x_1, a_1, 0, x_0, a_1, 0, x_1, a_1, 0, x_0, a_2, 0, x_f \rangle}_{\tau_5} \}. \tag{29}$$

We see that action a_2 in state x_0 has an expected reward of $1/3$, and it is the only state-action pair with a non-zero expected reward. Any reasonable offline RL algorithm applied to this dataset will output a policy with a non-zero probability assigned to that action in x_0. However, executing that policy, *i.e.,* taking action a_2 in state s_0 terminates the episode with 0 reward, which is suboptimal.

On the other hand, an optimal two-step model (and by extension an n-step one) will not collapse states s_0 and s_2 as this would prevent distinguishing the first action taken in τ_1 from the first action taken in τ_3 (corresponding respectively to pairs (s_0, s_0) and (s_0, s_2) after two timesteps). Consequently, an offline RL applied in representation can still learn an optimal policy.

C. Extended Related Work

Representation learning in Offline RL. Representation learning offers an exciting avenue to address the demands of learning compact feature for state by incorporating the auxiliary task of the state feature within the learning task. Empirical studies on representation learning in Offline RL have been first addressed by Yang & Nachum (2021), which evaluate the ability of a broad set of representation learning objectives in the offline dataset and propose Attentive Contrastive Learning (ACL) to improve downstream policy performance. After that, Chen et al. (2021a) investigate whether the auxiliary representation learning objectives that broadly used in NLP or CV domains can help for imitation across different Offline RL tasks. Lu et al. (2022a) further explores the existing challenges for visual observation input with the Offline RL dataset, meanwhile providing simple modifications on several state-of-the-art Offline RL algorithms to establish a competitive baseline. Zang et al. (2023) leverages behavior cloning for state representation learning to improve the downstream Offline policy performance. Another branch of representation learning in Offline RL is theoretical side. Uehara et al. (2021) studies the representation learning in low-rank MDPs with Offline settings and proposes an algorithm that leverages pessimism to learn under a partial coverage condition, Nachum & Yang (2021) develops a representation objective that provably accelerate the sample-efficiency of downstream Offline RL tasks, Ghosh & Bellemare (2020) theoretically shows that the stability of the policy is tightly connected with the geometry of the transition matrix, which can provide stability conditions for algorithms that learn features from the transition matrix of a policy and rewards.

Reward-Dependent Methods. DRQV2 (Kostrikov et al., 2020) learns a value function from offline tuples of observations, rewards, and actions. This could feasibly ignore exogenous noise given a suitable data-collection policy, but will not generally learn the full agent-centric latent state due to a heavy dependence on the reward structure. Bisimulation-based methods (DBC (Zhang et al., 2020), MICO (Castro et al., 2021), and SIMSR (Zang et al., 2022)) learns representations which have similar values under a learned value function. In general, bisimulation is an overly restrictive state abstraction that fails to transfer to different tasks.

Offline RL. The predominant approach to train offline RL agent is regularizing the learned policy to be close to the behavior policy of the offline dataset. This can be implemented by generating the actions that similar to the dataset and restricting the output of the learned policy close to the generated actions (Fujimoto et al., 2019), penalizing the distance between the learned policy and the behavior of the dataset (Kumar et al., 2019; Zhang et al., 2021b), or introducing a pessimism term to regularize the Q function for avoiding high Q value of the out-of-distribution actions (Kumar et al., 2020; Buckman et al., 2021). Some approaches utilize BC as a reference for policy optimization with the baseline methods (Fujimoto & Gu, 2021;

Laroche et al., 2019; Nadjahi et al., 2019; Simão et al., 2020; Rajeswaran et al., 2018). Some other approaches improve the performance by measuring the uncertainty of the model's prediction (Yu et al., 2020; Kidambi et al., 2020; An et al., 2021).

D. Additional Experiment Results

D.1. Manipulation Gridworld Environments

We investigated the ability of ACRO on whether it removes task relevant information as well or not. For this, we constructed a simple manipulation task on gridworlds, as described earlier in the paper. Detailed results are provided in Table 4.

Table 4: Probing Accuracy on the position of the agent, the position of the block, and the exogenous noise after training with ACRO. We found that the position of the agent and the block are both captured by the representation, while exogenous noise is discarded.

Task	Size	Agent Position	Block Position	Exogenous Noise
Push	4x4	100.00	100.00	6.239
Push	5x5	100.00	100.00	3.991
Push	6x6	100.00	100.00	2.752
Push	7x7	97.27	100.00	1.982
Push	8x8	94.53	99.61	1.621
Pull	4x4	100.00	100.00	6.221
Pull	5x5	100.00	100.00	4.000
Pull	6x6	100.00	100.00	2.816
Pull	7x7	100.00	100.00	2.048
Pull	8x8	100.00	100.00	1.566

D.2. Atari Experiments with Exogenous Information

We also consider the setting of Atari. We build on the setup introduced in decision transformers (Chen et al., 2021b). While decision transformers focus on framing the reinforcement learning problem as a sequence modeling problem, we mainly focus on learning representations which can learn to ignore exogenous noise. As such, for the decision transformer, we use trajectories of latent state and action pairs, instead of raw states as input, for modelling the sequence of actions. We use the same 4 games used in (Chen et al., 2021b) - Pong, Qbert, Breakout, and Seaquest. We consider the MEDIUM-EXO setting where a different image is used in each episode and concatenated to the side of each observation. We add randomly sampled images from the CIFAR10 dataset (Krizhevsky & Hinton, 2009) as exogenous noise. Figure 9 shows an example observation from breakout with exogenous noise.

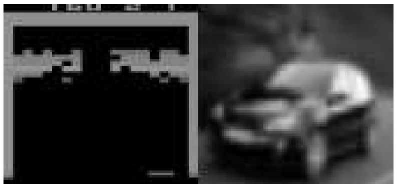


Figure 9: **Example** of an observation from Breakout with a CIFAR10 image on the side.

Decision Transformers use the DQN-Replay dataset (Agarwal et al., 2020) for training. The model is trained using a sequence modeling objective to predict the next action given the past states, actions, and returns-to-go $\hat{R}_c = \sum_{c'=c}^{C} r_c$, where c denotes the timesteps. This results in the following trajectory representation: $\tau = \left(\hat{R}_1, s_1, a_1, \hat{R}_2, s_2, a_2, \hat{R}_3, s_3, a_3, \dots\right)$, where a_c denotes the actions and s_c denotes the states. At test time, the start state s_1 and desired return $\hat{R}_1$ is fed into the model and it autoregressively generates the rest of the trajectory.

Decision Transformer uses a convolutional encoder to encode the observations. We first pretrain this encoder using the proposed ACRO objective. We use the 1-step inverse objective and DRIML as our baselines. After pretraining, we train the decision transformer using the sequence modeling objective keeping the encoder fixed. We present results in Table 5. We can see that ACRO outperforms both the baselines in all games further showing the effectiveness of the proposed approach.

Hyperparameter Details. We keep most of the hyperparameter details same as used in Chen et al. (2021b). They use episodes of fixed length during training - also referred to as the *context length*. We use a context length of 30 for Seaquest and Breakout and 50 for Pong and Qbert. Similar to Chen et al. (2021b), we consider one observation to be a stack of 4 atari frames. To implement the ACRO objective, we sample 8 different values for k and calculate the objective for each value of k, obtaining the final loss by taking the sum across all the sampled values of k. We do not feed the embedding for k in the MLP that predicts the action while computing the ACRO objective.

Table 5: **Atari (MEDIUM-EXO)**. Here we compare ACRO to One-Step-Inverse model and DRIML on various games from the Atari benchmark with exo-noise. We can see that ACRO outperforms the baselines in all but one case. Results averaged across 5 seeds.

Game	1-Step Inv.	DRIML	AC-State	ACRO
Breakout	$3.8_{\pm 0.4}$	$1.0_{\pm 0.0}$	$19.4_{\pm 3.323}$	$20.6_{\pm 3.2}$
Pong	$8.6_{\pm 3.2}$	$-20.0_{\pm 0.0}$	$9.0_{\pm 3.578}$	$11.8_{\pm 3.37}$
Qbert	$536.2_{\pm 233.75}$	$277.8_{\pm 46.24}$	$388.4_{\pm 184.22}$	$657.4_{\pm 271.52}$
Seaquest	$274.0_{\pm 29.61}$	$94.4_{\pm 4.63}$	$984.8_{\pm 80.31}$	$972.4_{\pm 136.09}$

D.3. Experimental Comparison between Single Action vs Multiple Actions Prediction

We empirically investigate how the ACRO algorithm compares if we use multi-action prediction up to the k-th timestep, in comparison to only predicting a single next step action. Since ACRO already conditions on $\phi(x_{t+k})$ for k-th step in the future, it is natural to ask how the performance varies if we predict multiple future actions compared to a single action. Concretely, the ACRO objective optimizes the following

$$\phi_\star \in \arg\max_{\phi \in \Phi} \mathop{\mathbb{E}}_{t \sim U(0,N)} \mathop{\mathbb{E}}_{k \sim U(0,K)} \log \left(\mathbb{P}(a_t \mid \phi(x_t), \phi(x_{t+k}))\right). \tag{30}$$

whereas, we could instead predict the k step action sequence:

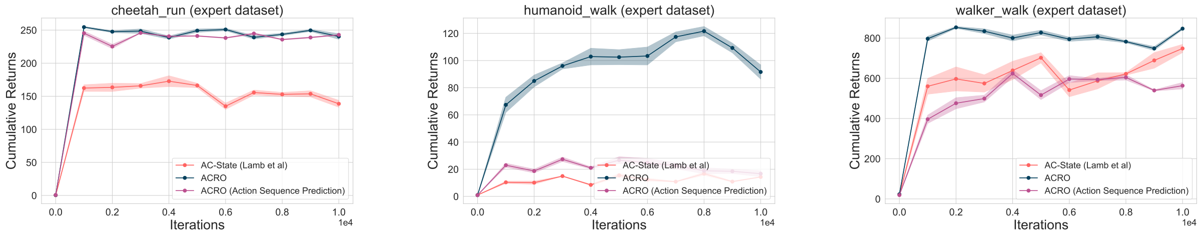
$$\phi_\star \in \arg\max_{\phi \in \Phi} \mathop{\mathbb{E}}_{t \sim U(0,N)} \mathop{\mathbb{E}}_{k \sim U(0,K)} \log \left(\mathbb{P}(a_t, \dots a_{t+k} \mid \phi(x_t), \phi(x_{t+k}))\right). \tag{31}$$

where equation 31 is implemented using an LSTM that outputs the k actions. One reason to prefer predicting just the first action is that it is a simpler model and is computationally cheaper (as it requires just a classifier over actions and not an autoregressive model over sequences like the LSTM). Intuitively, we also felt that predicting the first action to reach a goal would be sufficient, because ultimately every action along the trajectory is still predicted, but conditioned on the observation prior to the action being taken. In an environment with stochastic dynamics, we see this as being better in principle, because the best action to take at a given step is dependent on what has happened in the environment. In a deterministic environment, both approaches are valid in principle. Nonetheless, we agree that it is important to also answer this question experimentally. Figures 10 and 11 show comparison between ACRO and a variation of ACRO predicting multiple actions in the future. We use the same training setup, where all the representations are pre-trained in presence of HARD-EXO noise in observations.

In addition, we also compare ACRO with the AC-State objective (Lamb et al., 2022) which requires an additional information bottleneck based auxilliary objective for removing exogenous information. (Lamb et al., 2022) uses a vector quantization bottleneck (Van Den Oord et al., 2017) in addition to a multi-step inverse dynamics objective for removal of exogenous noise. In contrast, ACRO is much more straightforward, without requiring any fine-tuning with information bottlenecks, and we demonstrate the significance of ACRO compared to both these objectives, as shown in figures 10 and 11

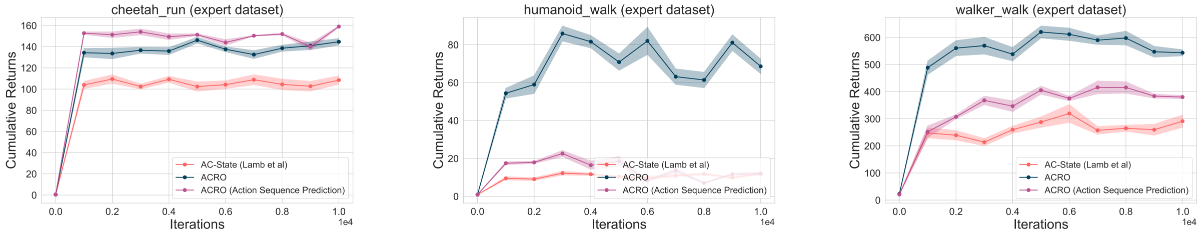
D.4. Data coverage and role of exploration in offline RL

In this section, we investigate the ability of learnt representations from ACRO and other baselines, depending on the dataset coverage in the offline setting. This leads to interesting insights on the role of data and exploration in offline RL, and how the quality of learnt representation depends on coverage data, for learning robust representations. From theoretical insights studying offline RL, it is clear that lack of dataeset coverage would degrade the algorithm. The fact that the concentrability coefficient is finite implies there is a good dataset coverage. In this work, we mostly focused on this problem from its empirical side and established the failure of existing approaches, as well as offering a fix (that works under the dataset



(a) Halfcheetah-expert (b) Humanoid-Walk-expert (c) Walker-Walk-expert

Figure 10: Comparions between ACRO, ACRO with multiple action predicton (ie, predicting an action sequence) and also with AC-State (Lamb et al., 2022) in **time-correlated** HARD-EXO offline datasets



(a) Halfcheetah-expert (b) Humanoid-Walk-expert (c) Walker-Walk-expert

Figure 11: Comparions between ACRO, ACRO with multiple action predicton (ie, predicting an action sequence) and also with AC-State (Lamb et al., 2022) in **changing video** HARD-EXO offline datasets

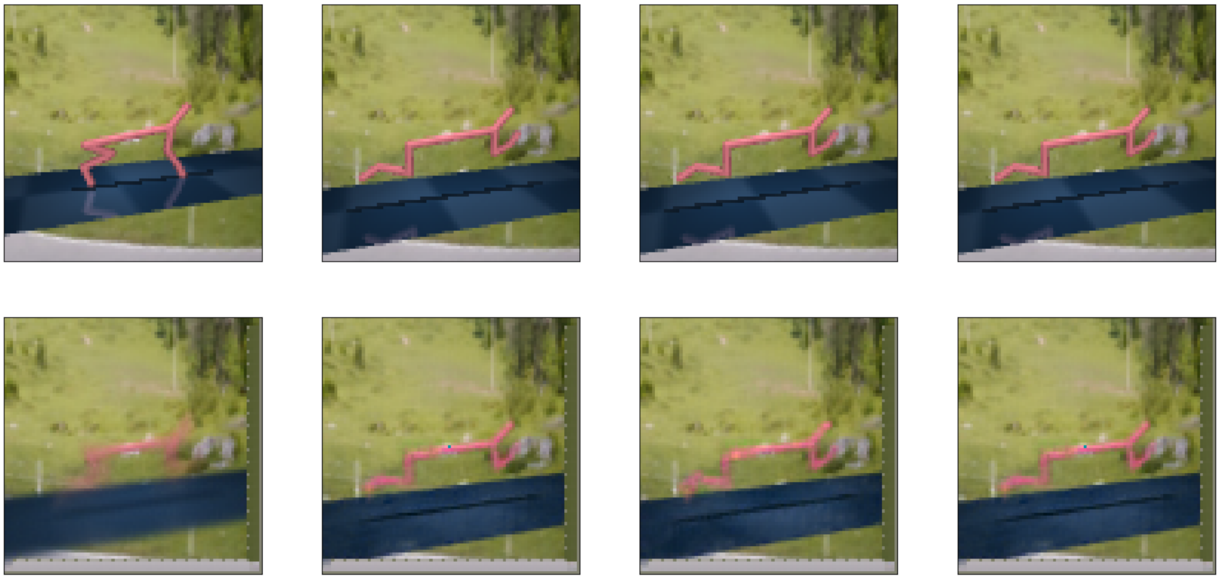


Figure 12: **Reconstructions for Fixed Background Distractor** from a decoder learnt of over two kinds of representations: **Left to Right**: Random features, DrQ, Behavior cloning, and ACRO. **Top Row**: Original observation, **Bottom Row**: Reconstruction.

coverage assumption). Further, the online RL problem with exogenous noise is also of interest. As of now, there is no algorithm with provable guarantees for the general RL problem with exogenous noise (there are some solutions under different sets of assumptions as we elaborated on in the related work section). Our work does not tackle this challenging

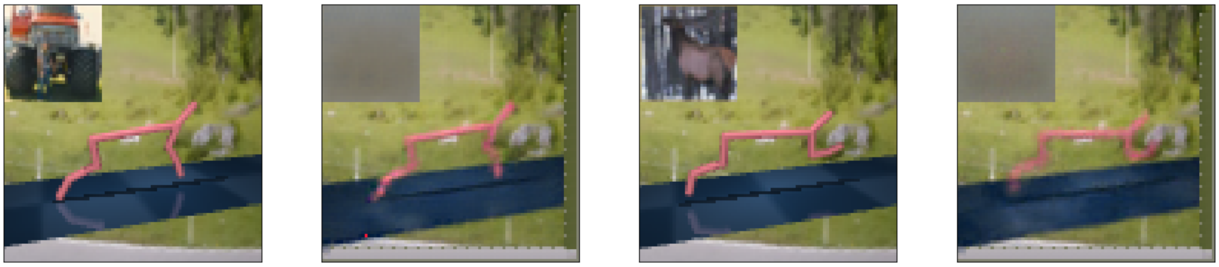


Figure 13: **Reconstructions with CIFAR images in background** from a decoder learnt of two kinds of representations from left to right: (1) Observation, (2) ACRO reconstruction visualization, (3) Observation, (4) ACRO reconstruction visualization.

problem, but study the offline aspects of its; a prevalent problem from a practical perspective.

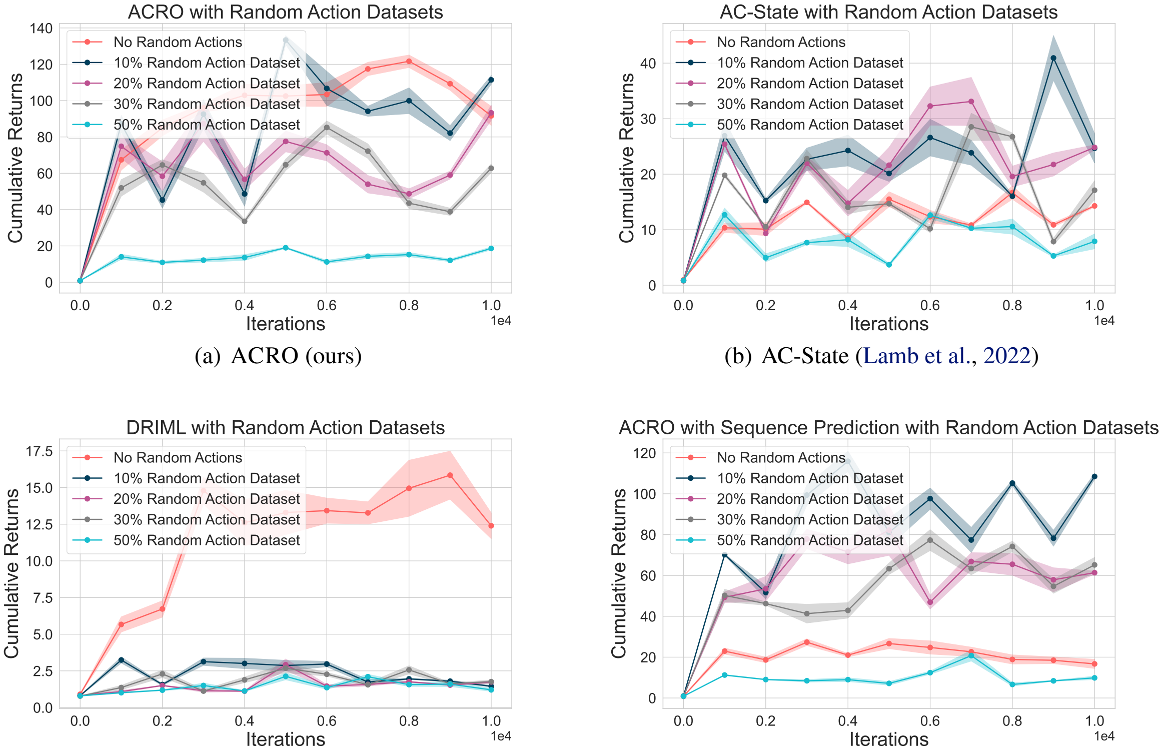
Experiments : We varied the amount of data coverage in the offline datasets, by taking the of times, the data collecting policy takes random actions. We vary the $\%$ of random actions from 10% to 50% where we assume that more randomness in actions taken by an expert policy means higher state space coverage. We follow the same experiment setup as before, and now show how the performance of ACRO (and two other baselines) varies as we have lower to higher coverage in the datasets, as in figure 14.

D.5. Quantitative Analysis

We provide results which show how ACRO representations lead to much more aligned cosine similarities as compared to DRIML and DRQ, when the background distractor is changed. We conduct a simple experiment where 1) the Exo-information is changed while keeping the Endo-information the same, and 2) the Exo-information remains the same while the Endo-information is changed (see Table 6). For each case, we then compute the cosine similarity between representations and report normalized cosine similarities in the Table below. The normalization is w.r.t the base cosine similarity from when the Exo-information is changed while keeping the Endo-information fixed. We see that the ACRO representations consistently have higher cosine similarities, with DrQ performing second best (and even being slightly better than ACRO in one case), while DRIML representations capturing quite a bit of the background information. Note that in terms of policy returns, the ranking between methods is consistent with the cosine similarities, i.e ACRO gets the highest return, followed by DrQ and then DRIML.

Table 6: **Quantitative Analysis** for how cosine similarities differ when keeping exo information fixed while varying endo information and vice versa.

ENVIRONMENT	DRIML	DRQ	ACRO
HARD-EXO b)	0.03	0.35	**0.80**
HARD-EXO a)	0.02	0.05	**0.60**
MEDIUM-EXO c)	0.11	0.53	0.45



(a) ACRO (ours)

(b) AC-State (Lamb et al., 2022)

(c) DRIML (Mazoure et al., 2020)

(d) ACRO with Sequence Prediction

Figure 14: Experimental results comparing different representation learning methods as we have varying amounts of coverage on the datasets. We make the assumption that adding more random actions to the datasets means the dataset has a higher coverage of state and action space. We show that as the % of random actions on datasets increases, the performance of each method degrates, especially the ones like DRIML that are independent of action prediction. In contrast, all other methods that rely on action prediction, suffer less depending on the amount of random actions that may exist in the datasets. This shows a potential for representation objectives based on action prediction that the performance of these methods degrades least even if the dataset quality is poor (where in our case, higher coverage due to more random actions means the dataset is degrading from an expert dataset to more of a random dataset).

E. Detailed Experiment Setup and Full Results

E.1. Exogenous Information Datasets

In this section, we provide a detailed summary of the different types of exogenous information based datasets as demonstrated in Figure 3.

E.1.1. Dataset Details

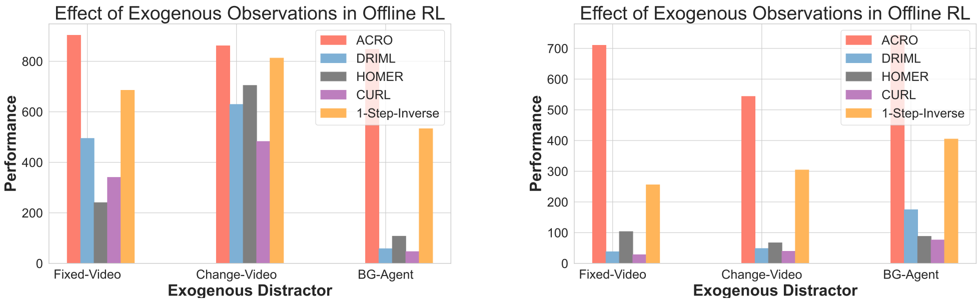


Figure 15: **Summary of Experiment Results on Walker Domain with Expert Dataset:** We vary the type of exogenous distractors present in offline datasets, and evaluate the ability of ACRO for policy learning from provably robust exogenous-free representations, while baseline methods can be prone to the exogenous information present in datasets. From easiest distractor (uncorrelated static images placed in corner or on the side), to corrrelated background exogenous images, and then to fixed or changing video distractors playing in the background, to finally the hardest exogenous information of other random agent information in the agent observation space; we show that ACRO can consistently learn good policies for downstream control tasks, while the ability of baselines to ignore exogenous information dramatically degrades, as we move to hard exogenous information settings. Appendix E.1.5 provides further ablation studies on the different HARD-EXO offline settings, and performance difference for different domains and datasets.

Easy-Exogenous Information (EASY-EXO). A visual RL offline benchmark has been recently proposed in (Lu et al., 2022b), where the authors provided pixel-based offline datasets collected using varying degrees of a soft actor-critic (SAC) policy. Furthermore, (Lu et al., 2022b) proposed a suite of distractor based datasets with different levels of severity in distractor shift, ranging from easy-shift, medium-shift to hard-shift. In the EASY-EXO setting, we first consider pixel-based offline data without and with visual distractors, as shown in Table 2. Experimental results in Table 2 show that even without any exogenous noise, ACRO learns agent-centric latent state representations more accurately than the different state-of-the-art baselines, such that by efficiently decoupling the endogenous state from the exogenous states, policy learning during the offline RL algorithm can lead to significantly better evaluation performance compared to several other baselines.

Medium-Exogenous Information (MEDIUM-EXO). We then consider three different types of medium exogenous information that might appear in visual offline data. To that end, we consider exogenous uncorrelated images from STL-10 image dataset (Coates et al., 2011) that appear on the corner or the side of the agent observation, and the goal of ACRO is to filter out the exogenous information while recovering only the agent state. We consider *three different types of* MEDIUM-EXO *information:*,

- The exogenous image from STL10 dataset appears in the corner of the agent observation. This does not change the observation size of the agent; and we simply add the exogenous image in one corner, which is fixed during an entire episode during the offline data collection. Figure 18 summarizes the result with different exogenous images placed in the corner. ACRO consistently outperforms several other baselines for a range of different datasets, since it can suitably filter out the exogenous part of the agent state.
- A slightly more difficult setting where now the STL10 exogenous image appears on the side of the agent observation space. This augments the agent observation space from $84 \times 84 \times 3$ to $84 \times 84 \times 2 \times 3$ since we consider downsampled STL10 images. Figure 19 summarizes this result comparing ACRO with the baseline representation objectives.
- Finally we consider the distractor setting that has appeared in prior works in online RL (Zhang et al., 2021a) with fixed video distractors playing in the background of agent observation space. For this setting, we specifically re-collect

the dataset following the procedure in (Lu et al., 2022b) where the SAC data collecting agent also sees a fixed video distractor playing in the background. Figure 20 summarizes the result and shows performance plots where ACRO can significantly outperform all the baselines across all different types of exogenous datasets.

Hard Exogenous Information (HARD-EXO). We finally consider three sets of different hard exogenous information settings, and find that these HARD-EXO can remarkably make it difficult for existing state of the art representation objectives to learnt underlying agent-centric states. This setting provides evidence that under suitably constructed exogenous information, which appear highly time correlated during the offline data collection, the baseline methods can fail to capture underlying agent-centric latent states. In contrast, the objective we consider in ACRO, along with the theoretical guarantees for learning a suitable encoder to recover the endogenous states, shows that policy optimization based on the agent-centric latent states can lead to efficient learning in these control tasks. We consider *three different types of* HARD-EXO *information:*

- We first consider time correlated static images appearing in the background of the agent observation. For this setting, during data collection, the agent sees a fixed image in the background for an entire episode, and it changes per episode of data collection. This time correlated exogenous information ensures that the baselines can remarkably get distracted, while ACRO can still be robust to the static image background. Figure 21 summarizes the results and shows that several existing representation baselines can fail due to time correlated static image distractors.
- We then consider an even more difficult HARD-EXO setting where now the video distractors playing in background also changes per episode during data collection. This is a novel setting with video distractors in RL, since we explicitly consider diverse set of background videos which also changes per episode of data collection. Similar to the above, Figure 22 summarizing the results with diverse video distractors in background per episode, shows that this setting can also break the baseline representation learners to recover the agent-centric latent states, while performance of ACRO remains robust to it, since the ACRO objective learns encoder to recover the endogenous agent-centric latent states accurately.
- Finally, we consider the most challenging HARD-EXO where now in addition to the environment observation, the agent additionally sees other random action agent observations. Here, the goal of the agent is to learn representations to identify the *controllable* environment, while other random-action observations are *uncontrollable* or exogenous to the agent. This is quite a difficult task since the agent we are tring to control also sees observations from the same domain, of other agents playing with random actions. The agent-centric agent observation now consists of other agents placed in a 3×3 grid. Figure 24 summarizes the experiment results showing that ACRO significantly outperforms all baseline representation learners.

E.1.2. EASY-EXO: PIXEL-BASED OFFLINE RL FROM V-D4RL BENCHMARKS

Visual Offline Control (V-D4RL) without Distractors. We first verify the effectiveness of learning representations with ACRO without any additional exogenous distractors, and compare with several baselines for learning representations. Figure 16 provides detailed performance curves for Table 2.

V-D4RL with Varying Severity of Distractor Data Shift. We then consider the distractor setting in v-d4rl benchmark (Lu et al., 2022b) with varying levels of distractor difficulty. Here, the exogenous noise is based on background static image distractors inducing a distribution shift in the dataset, depending on the level of difficulty from *easy*, to *medium* to *hard* distractors. As shown in Figure 17, we consider two different domains and find that with varying difficulty levels, ACRO can consistently outperform several state of the art baselines, learning directly from pixel data.

E.1.3. MEDIUM-EXO: STL10 EXOGENOUS IMAGES OR FIXED VIDEO DISTRACTORS

We extend our experimental results with different types of exogenous image distractors in the observation space of the agent. Detailed description of the dataset collection process is provided in Appendix E.2.

Exogenous Image Distractors Placed on the Corner or Side of Agent Observation. We consider two different settings where the agent environment observation is augmented with STL-10 image (Coates et al., 2011) distractors, either placed in the corner or on the side of the environment observation. Here, we consider adding uncorrelated exogenous images where for each pre-training of representations update, the environment observation has image distractors. When placed on the side, it extends the observation space of the agent. Figure 18 and Figure 19 shows results with exogenous images placed in the corner or on the side of the agent observations respectively.

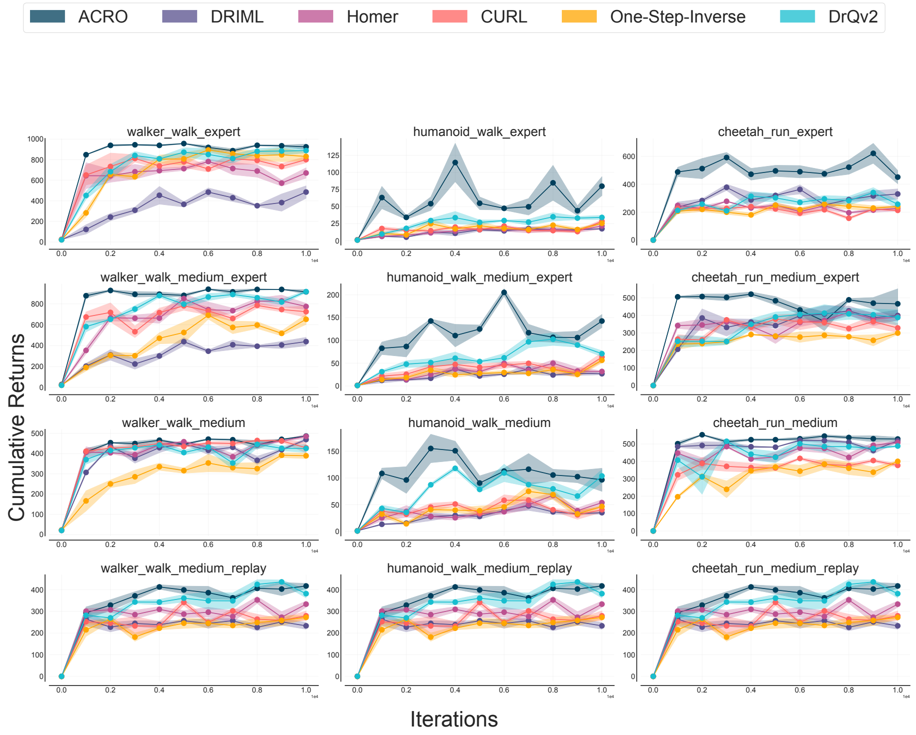


Figure 16: **Easy-Exo-No Distractors Full Results**. Experiments over 6 random seeds. For these experiments, we use the visual offline benchmark from (Lu et al., 2022b) and compare ACRO with several state of the art representation learning objectives. We find that across all tasks, ACRO either outperforms or equally performs as well as the best performing baseline method.

Fixed Video Distractor. We first consider a setting where the exogenous distractor in the background is fixed with a single type of video distraction. Figure 20 shows results with fixed video distractor showing that across several datasets, ACRO can consistently outperform baselines.

E.1.4. Hard-Exo: Time Correlated and Most Diverse Exogenous Distractors

Static Background Image that Changes Per Episode. We further experiment with time correlated exogenous distractors in the background, where during every episode of data collection, we provide a background image to the data collecting policy. We find that in presence of exogenous background distractors, ACRO can still be robust to the exogenous noise, while the existing baselines learning representations are more likely to fail, as shown in Figure 21.

Exogenous Video Distractors that Changes Per Episode. We then consider a more difficult setting where the type of video distractor playing in background changes during every episode. Further details on the data collection with *fixed* and *changing* video distractor in background is provided in the appendix. Figure 22 shows results with changing video distractor, where ACRO can consistently outperform baselines representation learning methods.

Multi-Environment Agent Observations as Exogenous Information. We then consider a setting where the observation space of the agent is augmented with other random agents moving in the environment. For this experiment, we take the observations from the environment the agent wants to control, while the other observations of other agents come from a random-policy dataset, either from the same domain, or from different domains. We do this since the random observations from other agents can be treated as exogenous information to the representation encoder; our experiment results show that ACRO can ignore the exogenous information from other agents, leading to significant performance improvements from the offline RL algorithm based on the exogenous observation dataset. Figure 24 summarizes the results.

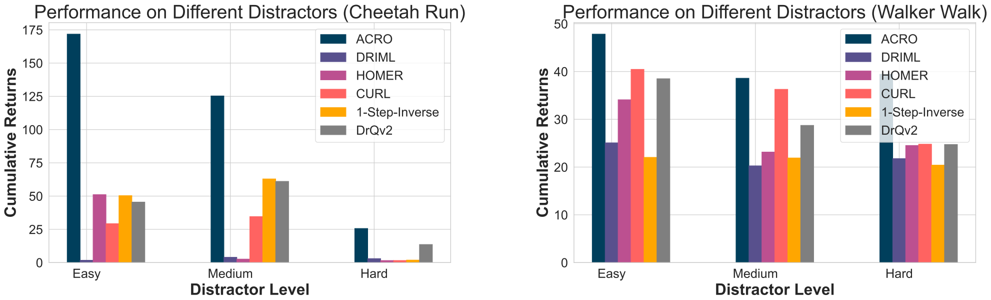


Figure 17: **Easy-Exo-Image Distractors Full Results**. Comparison of ACRO with baselines using the distractor suite of data shift severity from v-d4rl (Lu et al., 2022b) benchmark. We compare results with the two domains and datasets that were released in the v-d4rl benchmark distractor suite.

E.1.5. Ablation Studies - Hard-Exo Offline RL

Figure 23 provides a summary of comparison between different datasets in the Hard-Exo noise setting.

E.2. Data Collection for Offline RL with Exogenous Information

Easy-Exo Datasets. For the Easy-Exo setting with exogenous information, we consider uncorrelated visual distractors in the background of the observation space. For this setting, we extensively use the datasets released from the v-d4rl benchmark (Lu et al., 2022b) for offline RL. We note that the data shift severity in v-d4rl benchmark are only limited to two different domains and two data distributions (medium-expert and random). We experiment with both these datasets, and additionally consider the setting with no uncorrelated static images in the background.

Medium-Exo Datasets. For the Medium-Exo datasets, we collect new offline datasets using a SAC policy, following the same data collection procedure as in the literature from d4rl benchmark (Fu et al., 2020). The main difference is that when collecting new datasets with the SAC policy, we consider variations of different exogenous noise types in the dataset. As discussed earlier, for the Medium-Exo setting, we collect three different exo-types of datasets : **(a)** Exogenous stl10 images placed in the **corner** of the agent observation space. For this setting, during an episode of data collection, at each time step, we sample a new STL-10 image and place in the corner of the agent observations. **(b)** In the **side** exogenous information setting, instead of the corner, we place the exogenous image on the side of the environment observation. This augments the entire agent observaton space. A major difference with (a), however is that, in this setting we consider time correlated exogenous images where each step of SAC policy during an episode sees the same exogenous image, which only changes per episode of data collection. We consider this to be a harder setting compared to changing the exo image at each time-step, since this induced time correlation can make it harder for the representation objectives to be completely robust to the side exogenous information. **(c)** Finally, we conisder a **fixed video** distractor setting, which has been extensively studied in the online control benchmark (Tassa et al., 2018). Prior works have experimented in the online setting with fixed video distractors. We use the same procedure and fixed video distractor as here, except we use a SAC policy for data collection to be used in the offline setting. All these categories, cumulatively are denoted as Medium-Exo in this work. We release datasets and detailed experiment details for all these settings for Medium-Exo based offline RL datasets.

Hard-Exo Datasets. We consider this to be the hardest of the exogenous distractor setting. For this setting, we introduce several new offline benchmarks with different types of exogenous information. **(a) Time Correlated Exogenous Image in the Background** This is a hard distractor type where most existing representation learning baselines can fail. For this, during the data collection with SAC agent, we place a static background image from the STL10 dataset in the background. This image remains fixed for all timesteps within an episode, and we only sample a new exogenous background image at every episode. This makes the exogenous noise to be highly time correlated, where baseline representations are likely to capture the background image in addition to learning an embedding of the environment. **(b) Changing Video Distractors** In this setting, like (a), at every new episode we change the *type* of video distractor that we use. During an episode for different timesteps, the agent already sees correlated noise from the video frames playing in the background. However, since the type of video sequence that plays in background changes, this makes the offline datasets even more difficult to learn from. This setting is inspired by a real world application, where for example, the agent perceiving the world, can see background

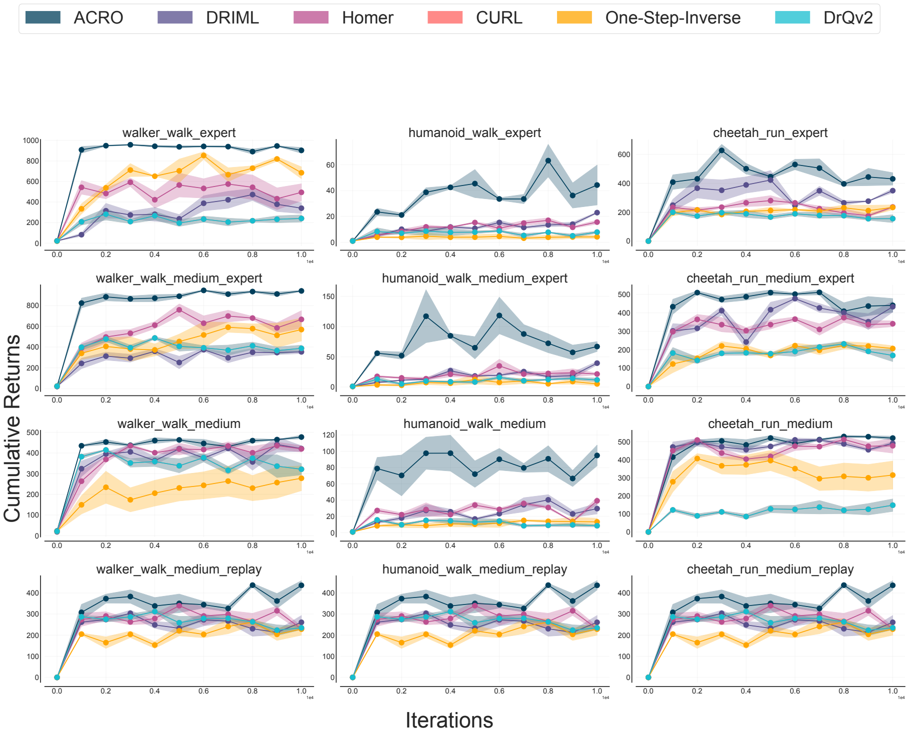


Figure 18: **MEDIUM-EXO-Corner Full Results**. Performance comparison of ACRO with several other baselines.

data from different data distributions (e.g background moving cars compared to people walking in the background). **(c)** Finally, we consider another HARD-EXO dataset, where now during every timestep of data collection, the agent sees other agents playing randomly. Here, we place several other agents, taking random actions, in a grid, and the goal of the agent is to recover only the controllable agent, while being able to ignore the other uncontrollable agents taking random actions, within its observation space. For this setting, the other uncontrollable agents can either be from the same domain (e.g using a Humanoid walker agent for both the endogenous, controllable part and the exogenous, uncontrollable part), or a different setting where the exogenous agents can be from other domains (e.g using a Humanoid agent for the controllable part, while the uncontrollable agents are from a Cheetah agent). We demonstrate both these types of observations either same-exogenous or different-exogenous in Figure 26 and Figure 25 respectively. We also release these datasets as an additional contribution to this work, and hope that future works in offline RL will use these datasets as benchmarks, for learning robust representations.

E.3. Offline RL Experiment Setup and Details

We describe our experiment setup in details. We use the visual d4rl (v-d4rl) benchmark (Lu et al., 2022b) and additionally add exogenous noise to visual datasets, as described earlier. For all our experiments, comparing ACRO with other baseline representation objectives, we pre-train the representations for $100K$ pre-training steps. Given pixel based visual offline data, we use a simple CNN+MLP architecture for encoding obeservations and predicting the ACRO actions. We also use cropping-based data augmentation as in DrQv2 while pre-training the representations for all methods. Specifically, the ACRO encoder uses 4-layers of convolutions, each with a kernel size of 3 and 32 channels. The original observation is of $84 \times 84 \times 9$, corresponding to a 3 channel-observation and a frame stacking of 3. The final encoder layer is an MLP which maps the convolutional output to a representation dimension of 256, giving the output $\phi(x)$. This is followed by a 2-layer MLP (hidden dim-256) that is used to predict the action given a 512 input corresponding to a concatenated s_t and s_{t+k} representations. For ACRO, we sample k from 1 to 15 uniformly. We use ReLU non-linearity and ADAM for optimization all throughout.

In DrQv2, data augmentation is applied only to the images sampled from the replay buffer, and not during the sample collection procedure. Given the pixel based control tasks, where the images are 84×84, DrQ pads each side by 4 pixels

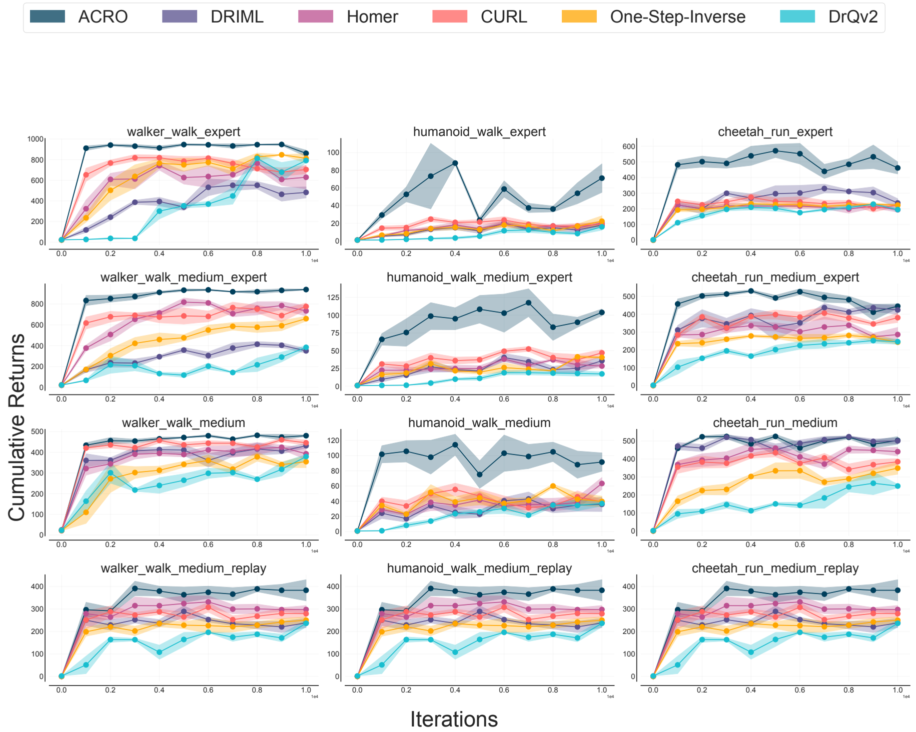


Figure 19: **MEDIUM-EXO-Side Full Results** ACRO can be quite robust to the exogenous images when the exogenous images appear to be similar to the agent in the environment. For example, consider the Cheetah-Run environment with a dog run image on the side, which can be quite distracting to the baseline methods.

(repeating boundary pixels) and then selects a random 84×84 crop, yielding the original image, shifted by 4 pixels. This procedure is repeated every time an image is sampled from the replay buffer; and makes DrQ data augmentation quite effective based on the random shifts alone, without the need for any additional auxiliary losses.

E.4. Background Agent Consecutive Frames Visualization

Figures 25 and 26 shows consecutive frames from the HARD-EXO background agent experiment setting.

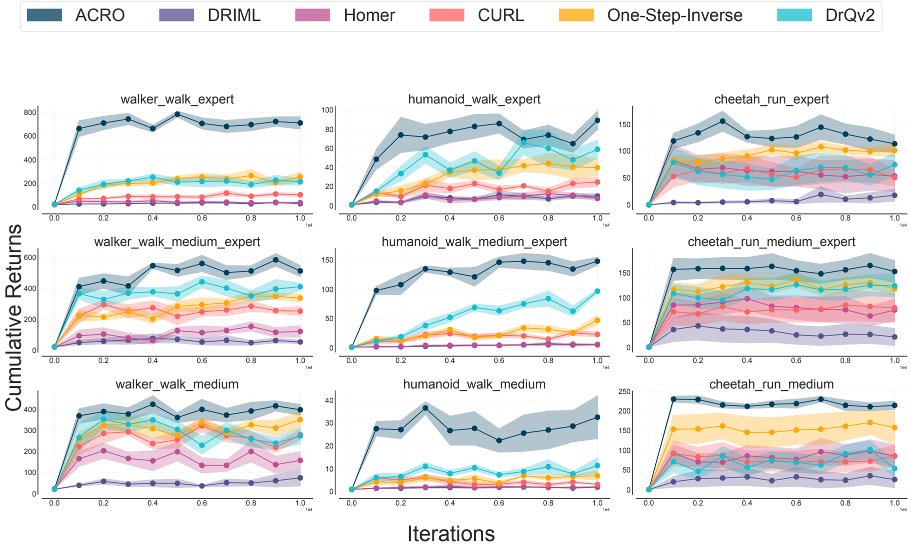


Figure 20: **MEDIUM-EXO-Fixed Video Full Results**. ACRO can outperform baselines with fixed exogenous distractors in background, which is time-correlated in nature. The fixed video distractor setting have often been studied in online RL literature. In this work, we study fixed video distractors in offline RL, where the distractors were present during the offline data collection.

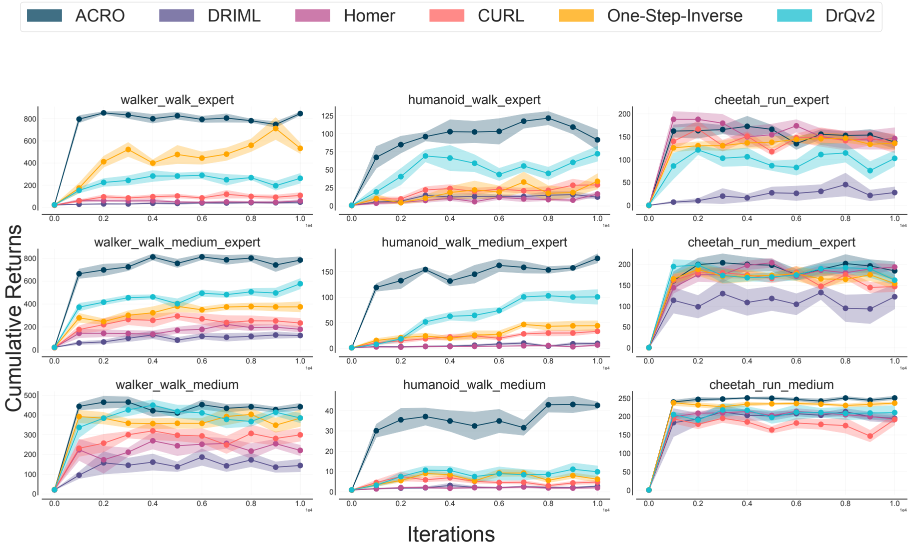


Figure 21: **HARD-EXO-Static Image Full Results**. We find that ACRO can significantly outperform baselines in presence of correlated exogenous static images playing in the background

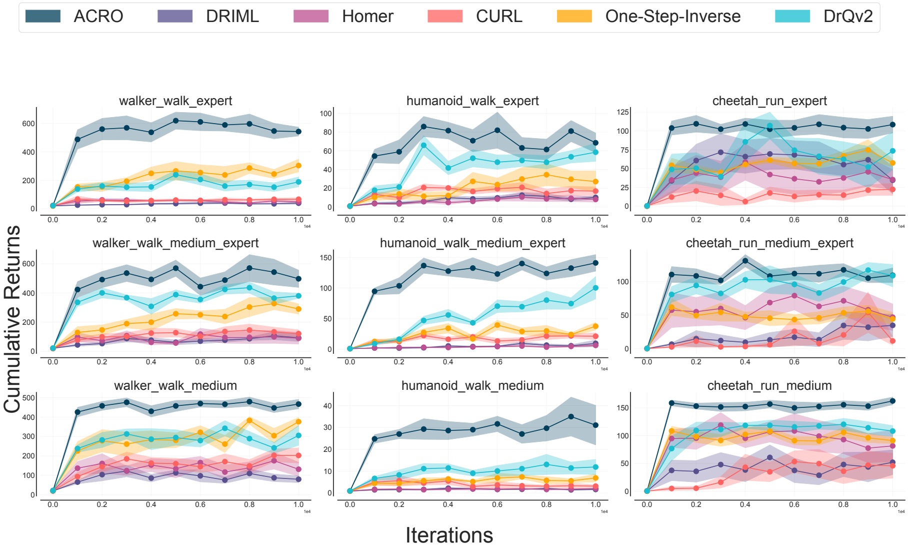


Figure 22: **HARD-EXO-Video Full Results** Across all datasets, ACRO can consistently outperform baseline methods for all types of datasets. When learning representations for offline policy optimization, the ability of ACRO to ignore exogenous information makes it outperform baseine representation objectives in almost all cases. The changing and time correlated video distractors are often hard for baseline methods to ignore, leading to significant performance drops depending on the offline data distribution

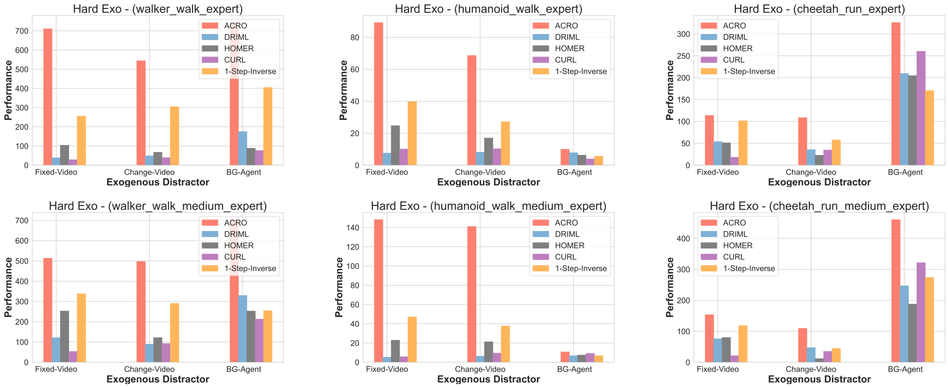


Figure 23: **Summary and Performance Difference between ACRO and baselines in the HARD-EXO offline RL setting**

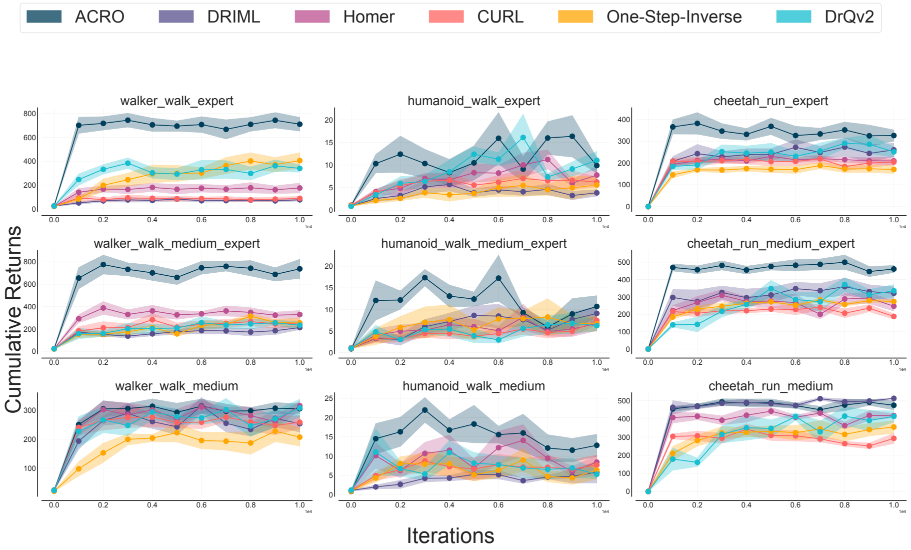


Figure 24: **Hard-Exo-Background Agent Full Results**. Random agent observations placed on the grid, of the entire agent observation space.

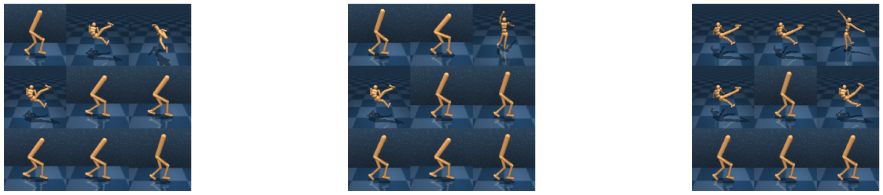


Figure 25: Consecutive timesteps from an episode of the environment where the controllable agent and the background agents are from different domains

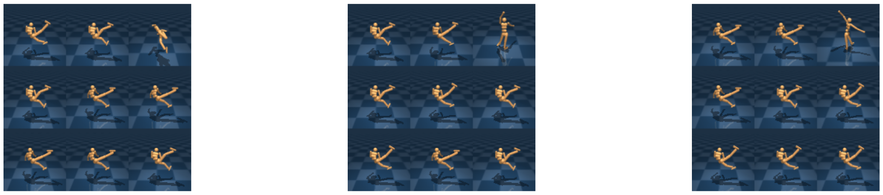


Figure 26: Consecutive timesteps from an episode of the environment where the controllable agent and the background agents are from the same domains