Efficient Online Reinforcement Learning with Offline Data

Philip J. Ball [*1] **Laura Smith** [*2] **Ilya Kostrikov** [*2] **Sergey Levine** [2]

Abstract

Sample efficiency and exploration remain major challenges in online reinforcement learning (RL). A powerful approach that can be applied to address these issues is the inclusion of offline data, such as prior trajectories from a human expert or a sub-optimal exploration policy. Previous methods have relied on extensive modifications and additional complexity to ensure the effective use of this data. Instead, we ask: *can we simply apply existing off-policy methods to leverage offline data when learning online?* In this work, we demonstrate that the answer is yes; however, a set of minimal but important changes to existing off-policy RL algorithms are required to achieve reliable performance. We extensively ablate these design choices, demonstrating the key factors that most affect performance, and arrive at a set of recommendations that practitioners can readily apply, whether their data comprise a small number of expert demonstrations or large volumes of sub-optimal trajectories. We see that correct application of these simple recommendations can provide a $\mathbf{2.5}\times$ improvement over existing approaches across a diverse set of competitive benchmarks, with no additional computational overhead. We have released our code here: `github.com/ikostrikov/rlpd`.

1. Introduction

Deep reinforcement learning (RL) has achieved success in a number of complex domains, such as Atari (Mnih et al., 2015) and Go (Silver et al., 2016), as well as real-world applications like Chip Design (Mirhoseini et al., 2021) and Human Preference Alignment (Ouyang et al., 2022). In many of these settings, strong RL performance is predicated on having large amounts of online interaction with an environment, which is usually made feasible through the use of simulators. In real-world problems, however, we are often confronted with scenarios where samples are expensive, and furthermore, rewards are sparse, often exacerbated by high dimensional state and action spaces.

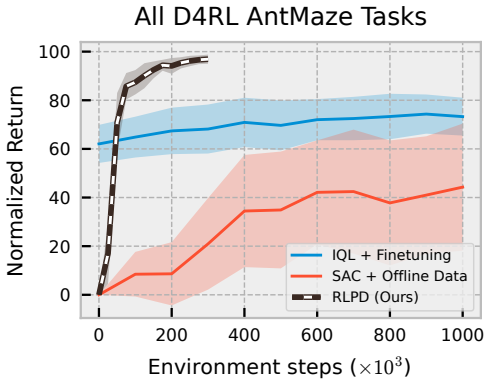


Figure 1. Our approach, RLPD, extends standard off-policy RL and achieves reliable state-of-the-art online performance on a number of tasks using offline data. Here we show the difficult D4RL AntMaze domain (10 seeds, 1 std. shaded), averaged over all 6 tasks. We run RLPD for 300k steps due to early convergence.

One promising way to resolve this issue is via the inclusion of data generated by a previous policy or human expert when training deep RL algorithms (often referred to as *offline data* (Levine et al., 2020)), as evidenced theoretically (Wagenmaker & Pacchiano, 2022; Song et al., 2023) and in real-world examples by Cabi et al. (2019); Nair et al. (2020); Lu et al. (2021). This can alleviate challenges due to sample efficiency and exploration by providing the algorithm with an initial dataset to "kick-start" the learning process, either in the form of high-quality expert demonstrations, or even low-quality but high-coverage exploratory trajectories. This also provides us with an avenue to leverage large pre-collected datasets in order to learn useful policies.

Some prior work has focused on using this data through pre-training, while other approaches introduce constraints when training online to handle issues with distribution shift. However, each approach has its drawbacks, such as additional training time and hyperparamters, or limited improvement beyond the behavior policy respectively. Taking a step back, we note that *standard off-policy algorithms* should be able to take advantage of this offline data, and furthermore issues

[*]Equal contribution [1]University of Oxford [2]UC Berkeley. Correspondence to: Philip J. Ball <ball@robots.ox.ac.uk>, Laura Smith <smithlaura@berkeley.edu>, Ilya Kostrikov <kostrikov@berkeley.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

with distribution shift should be alleviated in this setting, as we can explore the environment online. Thus far however, such methods have seen limited success in this problem setting. Therefore, in this work, we ask the following question: *can we simply apply existing off-policy methods to leverage offline data when learning online, without offline RL pre-training or explicit imitation terms that privilege the prior offline data?*

Through a set of thorough experiments on a collection of widely studied benchmarks, we show that the answer to this question is yes. However, naïvely applying existing online off-policy RL algorithms can result in comparatively poor performance, as we see in Figure 1 comparing 'SAC + Offline Data' with 'IQL + Finetuning'. Instead, a minimal set of key *design choices* must be taken into consideration to ensure their success. Concretely, we first introduce a remarkably simple approach to sampling the offline data, which we call "symmetric sampling", that performs well over a large variety of domains with no hyperparameter tuning. Then, we see that in complex settings (e.g., sparse reward, low volume of offline data, high dimensionality, etc.), it is vital that value functions are prevented from over-extrapolation. To this end, we provide a novel perspective on how Layer Normalization (Ba et al., 2016) implicitly prevents catastrophic value over-extrapolation, thereby greatly improving sample-efficiency and stability in many scenarios, while being a minimal modification to existing approaches. Then, to improve the rate at which the offline data are utilized, we incorporate and compare the latest advances in sample-efficient model-free RL, and find that large ensembles are remarkably effective across a variety of domains. Finally, we identify and provide evidence that key design choices in recent RL literature are in fact environment sensitive, showing the surprising result that environments which share similar properties in fact require entirely different choices, and recommend a *workflow for practitioners* to accelerate their application of our insights to new domains.

We demonstrate that our final approach, which we call **RLPD** (**R**einforcement **L**earning with **P**rior **D**ata) outperforms previously reported results, as we see in Figure 1, on many competitive domains, sometimes by $\mathbf{2.5}\times$. Crucially, as our changes are minimal, we maintain the attractive properties of online algorithms, such as ease of implementation and computational efficiency. Furthermore, we see the generality of our approach, which achieves strong performance across a number of diverse offline datasets, from those containing limited expert demonstrations, through to data comprised of high-volume sub-optimal trajectories.

We believe that our insights are valuable to the community and practitioners. We show that online off-policy RL algorithms can be remarkably effective at learning with offline data. However, we show their reliable performance is predicated on several key design choices, namely the way the offline data are sampled, a crucial way of normalizing the critic update, and using large ensembles to improve sample efficiency. While the individual ingredients of RLPD are refreshingly simple modifications on existing RL components, we show that their combination delivers state-of-the-art performance on a number of popular online RL with offline data benchmarks, exceeds the performance of significantly more complex prior methods, and generalizes to a number of different types of offline data, whether it be expert demonstrations or sub-optimal trajectories. We have released RLPD here: `github.com/ikostrikov/rlpd`.

2. Related work

Offline RL pre-training. We note connections to offline RL (Ernst et al., 2005; Fujimoto et al., 2019; Levine et al., 2020); many prior works perform offline RL, followed by online fine-tuning (Hester et al., 2018; Kalashnikov et al., 2018; Nair et al., 2020; Lee et al., 2021; Kostrikov et al., 2022). Notably, Lee et al. (2021) also considers large ensembles and multiple gradient-step per timestep regimes when learning online. However, our approach uses a significantly simpler sampling mechanism with no hyperparameters and does not rely on costly offline pre-training, which introduces yet *additional* hyperparameters. We also emphasize that our normalized update is *not an offline RL method*—we do not perform any offline pre-training but run online RL from scratch with offline data included in a replay buffer.

Constraining to prior data. An alternative to the offline RL pre-training paradigm is to explicitly constrain the online agent updates such that it exhibits behavior that resembles the offline data (Levine & Koltun, 2013; Fox et al., 2016; Hester et al., 2018; Nair et al., 2018a; Rajeswaran et al., 2018; Rudner et al., 2021). Particularly relevant to our approach is work by Rajeswaran et al. (2018), which augments a policy gradient update with a weighted update that explicitly includes *demonstration* data. In contrast, we use a sample-efficient off-policy paradigm, and *do not* perform any pre-training. Also similar to our work is that by Nair et al. (2018a), who also use an off-policy algorithm with a fixed offline replay buffer. However, we do not restrict the policy using a behavior cloning term, and do not reset to demonstration states. Moreover, we note that these approaches generally require the offline data to be high quality (i.e., 'learning from demonstration data' (Asada & Hanafusa, 1979; Schaal, 1996)), while our approach is, importantly, agnostic to the quality of the data.

Unconstrained methods with prior data. Prior work has also considered ways of incorporating offline data without any constraints. Some methods focus on initializing a replay buffer with offline data (Večerík et al., 2017; Hester et al.,

2018), while other works have utilized a balanced sampling strategy to handle online and offline data (Nair et al., 2018b; Kalashnikov et al., 2018; Hansen et al., 2022; Zhang et al., 2023). Most recently, Song et al. (2023) presented a theoretical analysis of such approaches, showing that balanced sampling is important both in theory and practice. In our experiments, we also show that balanced sampling helps online RL with offline data; however, directly using this approach on a range of benchmark tasks is insufficient, and other design decisions that we present are critical in achieving good performance across all tasks.

3. Preliminaries

We consider problems that can be formulated as a Markov Decision Process (MDP) (Bellman, 1957), described as a tuple $(\mathcal{S}, \mathcal{A}, \gamma, p, r, d_0)$ where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space and $\gamma \in (0, 1)$ is the discount factor. The dynamics are governed by a transition function $p(s'|s, a)$; there is a reward function $r(s, a)$ and initial state distribution $d_0(s)$. The goal of RL is then to maximize the expected sum of discounted rewards: $\mathbb{E}_\pi \left[\sum_{t=1}^{\infty} \gamma^t r(s_t, a_t)\right]$.

In this work, we focus on RL while having access to offline datasets $\mathcal{D}$ (Levine et al., 2020), a collection of (s, a, r, s') tuples generated from a particular MDP. A key property of offline datasets is they usually do not provide complete state-action coverage, i.e., $\{s, a \in \mathcal{D}\}$ is a small subset of $\mathcal{S} \times \mathcal{A}$. Due to this lack of on-policy coverage, methods using function approximation may over-extrapolate values when learning on this data, leading to a pronounced effect on learning performance (Fujimoto et al., 2019).

4. Online RL with Offline Data

As outlined in Section 3, we consider the standard RL setting with the addition of a pre-collected dataset. In this work, we aim to design a general approach that is agnostic to the *quality and quantity* of this pre-collected data. For instance, this data could take the form of a handful of human demonstrations, or swathes of sub-optimal, exploratory data. Furthermore, we wish to make recommendations that are agnostic to the nature of the problem setting, such as whether the observations are state or pixel-based, or whether the rewards are sparse or dense.

To this end, we present an approach based on off-policy model-free RL, without pre-training or explicit constraints, which we call **RLPD** (**R**einforcement **L**earning with **P**rior **D**ata). As discussed in Section 4.5, we base our algorithm design on SAC (Haarnoja et al., 2018a;b), though in principle these design choices may improve other off-policy RL approaches. First, we propose a simple mechanism for incorporating the prior data. Then, we identify a pathology that exists when naïvely applying off-policy methods to this problem setting, and propose a simple and minimally invasive solution. After, we improve the rate the offline data are utilized by incorporating the latest approaches in sample-efficient RL. Finally, we highlight common design choices in recent deep RL that are in fact environment sensitive, and should be adjusted accordingly by practitioners.

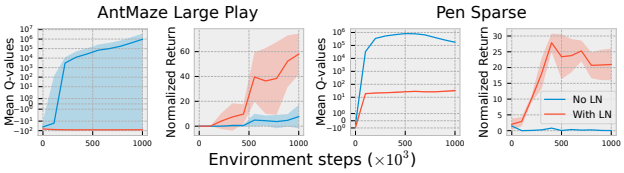


Figure 2. Using SAC with our symmetric sampling method can result in instabilities due to diverging Q-values; with LayerNorm in the critic this disappears, improving performance.

4.1. Design Choice 1: A Simple and Efficient Strategy to Incorporate Offline Data

We start with a simple approach that incorporates prior data which adds no computational overhead, yet is agnostic to the nature of the offline data. We call this 'symmetric sampling', whereby for each batch we sample 50% of the data from our replay buffer, and the *remaining 50% from the offline data buffer*, resembling the scheme used by Ross & Bagnell (2012). As we will see in later sections, this sampling strategy is surprisingly effective across a variety of scenarios, and we extensively ablate various elements of this scheme (see Section 5.1). However, applying this approach to canonical off-policy methods, such as SAC (Haarnoja et al., 2018a), does not yield strong performance, as we see in Figure 1, and further design choices must be taken into consideration.

4.2. Design Choice 2: Layer Normalization Mitigates Catastrophic Overestimation

Standard off-policy RL algorithms query the learned Q-function for out-of-distribution (OOD) actions, which might not be defined during learning. Consequently, there can be significant overestimation of actual values due to the use of function approximation (Thrun & Schwartz, 1993). In practice, this phenomenon leads to training instabilities and possible *divergence* when the critic is trying the catch up with a constantly increasing value.

In particular, we find this to be the case when naïvely applying our symmetric sampling approach for complex tasks (see Figure 2). Critic divergence is a well-studied problem, particularly in the *offline* regime, where the policy cannot generate new experience. In our problem setting, however, we *can* sample from the environment. Therefore, instead of creating a mechanism that explicitly discourages OOD actions, which can be viewed as anti-exploration (Rezaeifar et al., 2022), we instead need to simply ensure that the

learned functions do not extrapolate in an unconstrained manner. To this end, we show that Layer Normalization (LayerNorm) (Ba et al., 2016) can bound the extrapolation of networks but, crucially, *does not* explicitly constrain the policy to remain close to the offline data. This in turn does not discourage the policy from exploring unknown and *potentially valuable* regions of the state-action space. In particular, we demonstrate that LayerNorm bounds the values and empirically prevents catastrophic value extrapolation. Concretely, consider a Q-function Q parameterized by θ, w, applying LayerNorm and intermediate representation $\psi_\theta(\cdot, \cdot)$. For any a and s we can say[1]:

$$\begin{aligned}\|Q_{\theta,w}(s,a)\| &= \|w^T \text{relu}(\psi_\theta(s,a))\| \\ &\leq \|w\| \|\text{relu}(\psi_\theta(s,a))\| \leq \|w\| \|\psi(s,a)\| \\ &\leq \|w\|\end{aligned}$$

Therefore, as a result of *Layer Normalization*, the Q-values are bounded by the norm of the weight layer, even for actions outside the dataset. Thus, the effect of erroneous action extrapolation is greatly mitigated, as their Q-values are unlikely to be significantly greater than those already seen in the data. Indeed, referring back to Figure 2, we see that introducing LayerNorm into the critic greatly improves performance through mitigating critic divergence.

To illustrate this, we generate a dataset with inputs x distributed in a circle with radius 0.5 and labels $y = \|x\|$. We study how a standard two-layer MLP with ReLU activations (common in deep RL) extrapolates outside of the data distribution, and the effect of adding LayerNorm. In Figure 3, the standard parameterization leads to unbounded extrapolation outside of the support, while LayerNorm bounds the values, greatly reducing the effect of uncontrolled extrapolation.

4.3. Design Choice 3: Sample Efficient RL

We now have an online approach leveraging offline data that also suppresses extreme value extrapolation, whilst maintaining the freedom of an unconstrained off-policy method. However, a benefit of offline and constrained approaches is that they have an explicit mechanism to efficiently incorporate prior data, such as through pre-training (Hester et al., 2018; Lee et al., 2021), or an auxiliary supervision term (Nair et al., 2018a; Rudner et al., 2021) respectively. In our case, the incorporation of prior data is implicit through the use of online Bellman backups over offline transitions. Therefore, it is imperative that these Bellman backups are performed as sample-efficiently as possible.

One way to achieve this is to increase the number of updates we perform per environment step (also referred to as update-to-data (UTD) ratio), allowing the offline data to become "backed-up" more quickly. However, as highlighted in recent literature in online RL, this can create issues in the optimization process and ironically *reduce sample efficiency*, due to statistical over-fitting (Li et al., 2022). To ameliorate this, prior work has suggested a number of regularization approaches, such as simple L2 normalization (Večerík et al., 2017), Dropout (Srivastava et al., 2014; Hiraoka et al., 2022) and random ensemble distillation (Chen et al., 2021). In this work, we settle on the latter approach of random ensemble distillation; we will demonstrate through ablations that this performs strongest, particularly on sparse reward tasks.

[1]For simplicity, we consider LayerNorm without bias terms. This does not change the analysis, as it is a constant.

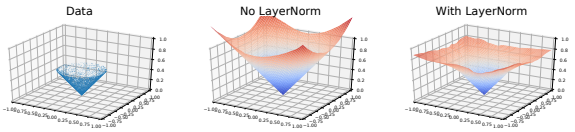


Figure 3. We fit data (left) with a two-layer MLP without LayerNorm (center) and with LayerNorm (right). LayerNorm bounds the values and prevents catastrophic overestimation.

We also note that value over-fitting issues exist when performing TD-learning from images (Cetin et al., 2022). Therefore in these settings, we further include random shift augmentations (Kostrikov et al., 2021; Yarats et al., 2022).

4.4. Per-Environment Design Choices

Having highlighted the 3 key design choices for our approach that can be applied generally to all environments and offline datasets, we now to turn our attention to design choices that are commonly taken for granted, but can in fact be environment-sensitive. It is well documented that deep RL algorithms are sensitive to implementation details (Henderson et al., 2018; Engstrom et al., 2020; Andrychowicz et al., 2021; Furuta et al., 2021). As a result, many works in deep RL require per-environment hyperparameter tuning. Given the huge variety of tasks we consider in our experiments, we believe it is important to contribute to this discourse, and highlight that certain design choices, which are often simply inherited from previous implementations, should in fact be *carefully reconsidered*, and may explain why off-policy methods have not been competitive thus far on the problems we consider. We therefore take a view that, given the well-documented sensitivity of deep RL, it is important to demonstrate a critical path of design choices to consider when assessing new environments, and provide a *workflow* to simplify this process for practitioners.

Clipped Double Q-Learning (CDQ). Value-based methods combined with function approximation and stochastic optimization suffer from estimation uncertainty, which, in combination with the maximization objective of Q-learning, leads to value overestimation (van Hasselt et al., 2016) (as also discussed in Section 4.2). In order to mitigate this

issue, Fujimoto et al. (2018) introduce Clipped Double Q-Learning (CDQ) which involves taking a minimum of an ensemble of two Q-functions for computing TD-backups. They define the targets for updating the critics as follows:

$$y = r(s,a) + \gamma \min_{i=1,2} Q_{\theta_i}(s', a') \text{ where } a' \sim \pi(\cdot|s').$$

However, this corresponds to fitting target Q-values that are 1 std. below the actual target values, and recent work (Moskovitz et al., 2021) suggests that this design choice may not be universally useful as it can be too conservative. Therefore, this is important to reconsider, especially outside of the domains for which it was originally designed, such as sparse reward tasks prevalent in our problem setting.

Maximum Entropy RL. MaxEnt RL augments the traditional return objective with an entropy term:

$$\max_{\pi} \mathbb{E}_{s\sim\rho^{\pi}, a\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t (r_t + \alpha \mathcal{H}(\pi(a|s)))\right].$$

This corresponds to maximizing the discounted reward and expected policy entropy at each time step. The motivation for these approaches is centered around robustness and exploration (i.e., maximize reward while behaving as randomly as possible). Approaches relying on this objective are empirically impressive (Haarnoja et al., 2018a;b; Chen et al., 2021; Hiraoka et al., 2022). We, therefore, believe this design choice is of interest in the context of online fine-tuning, where rewards are often sparse and require exploration.

Architecture. Network architecture can have a significant impact on deep RL performance, and the same architecture that can be optimal in one environment can be sub-optimal in another (Furuta et al., 2021). To simplify the search space, we consider the impact of having 2 or 3 layers in the actor and critic, which have been shown to affect performance, even on canonical tasks (Ball & Roberts, 2021).

4.5. RLPD: Approach Overview

Here we present pseudo-code for our approach, highlighting in Green elements that are important to our approach, and in Purple, environment-specific design choices.

The key factors of RLPD reside in lines 1 and 13 of Algorithm 1 with adopting LayerNorm, large ensembles, sample efficient learning and a symmetric sampling approach to incorporate online and offline data. For environment specific choices, we recommend the following as a starting point:

- *Line 3:* Subset 2 critics
- *Line 16:* Remove entropy
- *Line 1:* Utilize a *deeper* 3 layer MLP

As a pragmatic workflow, we recommend ablating these Purple design choices first, and in the order stated above.

Algorithm 1 Online RL with Offline Data (RLPD)

1: Select LayerNorm, Large Ensemble Size E, Gradient Steps G, and architecture.
2: Randomly initialize Critic θ_i (set targets $\theta'_i = \theta_i$) for $i = 1, 2, \ldots, E$ and Actor ϕ parameters. Select discount γ, temperature α and critic EMA weight ρ.
3: Determine number of Critic targets to subset $Z \in \{1, 2\}$
4: Initialize empty replay buffer $\mathcal{R}$
5: Initialize buffer $\mathcal{D}$ with offline data
6: **while** True **do**
7: Receive initial observation state s_0
8: **for** t = 0, T **do**
9: Take action $a_t \sim \pi_\phi(\cdot|s_t)$
10: Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{R}$
11: **for** $g = 1, G$ **do**
12: Sample minibatch b_R of $\frac{N}{2}$ from $\mathcal{R}$
13: Sample minibatch b_D of $\frac{N}{2}$ from $\mathcal{D}$
14: Combine b_R and b_D to form batch b of size N
15: Sample set $\mathcal{Z}$ of Z indices from $\{1, 2, \ldots, E\}$
16: With b, set

$$y = r + \gamma \left(\min_{i\in\mathcal{Z}} Q_{\theta'_i}(s', \tilde{a}')\right), \quad \tilde{a}' \sim \pi_\phi(\cdot|s')$$

17: Add entropy term $y = y + \gamma\alpha \log \pi_\phi(\tilde{a}'|s')$
18: **for** $i = 1, E$ **do**
19: Update θ_i minimizing loss:

$$L = \frac{1}{N}\sum_i (y - Q_{\theta_i}(s,a))^2$$

20: **end for**
21: Update target networks $\theta'_i \leftarrow \rho\theta'_i + (1-\rho)\theta_i$
22: **end for**
23: With b, update ϕ maximizing objective:

$$\frac{1}{E}\sum_{i=1}^{E} Q_{\theta_i}(s, \tilde{a}) - \alpha \log \pi_\phi(\tilde{a}|s), \quad \tilde{a} \sim \pi_\phi(\cdot \mid s)$$

24: **end for**
25: **end while**

5. Experiments

We design our experiments to not only demonstrate the importance of our design choices, but also provide the insights that allow practitioners to quickly adapt RLPD to their problems. As such, we aim to answer the following questions:

1. Is RLPD competitive with prior work despite using *no pre-training nor having explicit constraints*?
2. Does RLPD transfer to *pixel-based* environments?
3. Does LayerNorm *mitigate value divergence*?
4. Does the proposed workflow around environment-specific design choices lead to *reliable performance*?

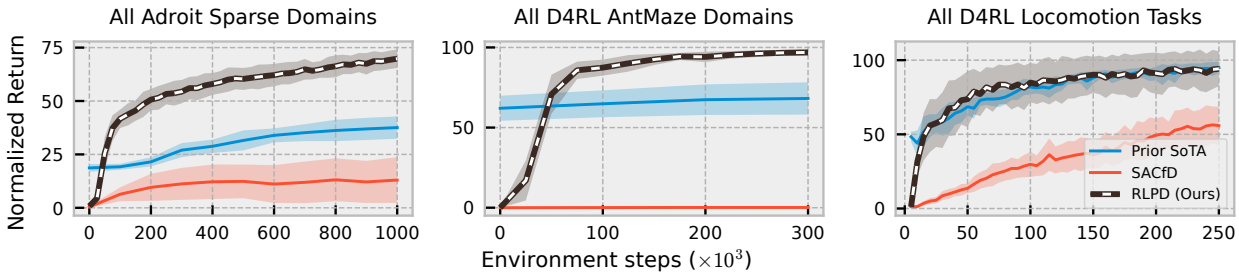


Figure 4. RLPD exceeds prior state-of-the-art performance on a number of different popular benchmarks whilst being significantly simpler. Results are aggregated over **21 different environments** (10 Seeds, 1 std. shaded). In each case, we compare to the prior best known work (IQL + Finetuning in Adroit and AntMaze, Off2On in Locomotion), and SACfD, a canonical off-policy approach using offline data.

For (1), we compare RLPD to those which have been designed to use offline data to accelerate online learning. For (2), we consider an additional suite of tasks to study RLPD's applicability to vision-based domains. Then, for (3) we perform analysis to demonstrate the importance of using LayerNorm. Lastly, for (4) we demonstrate our proposed workflow on the most challenging tasks (see Subsection 4.5).

How does RLPD compare? We consider the following **21 tasks** from established benchmarks:

- **Sparse Adroit** (Nair et al., 2020). These 3 dexterous manipulation tasks—pen-spinning, door-opening, ball relocation—are challenging, sparse-reward tasks. The offline data are multi-modal, with a small set of human demonstrations and a large set of trajectories from a behavior-cloned policy trained on this human data. We follow the *rigorous evaluation criteria* of Kostrikov et al. (2022), whereby performance is based on completion speed, rather than success rate. IQL + Finetuning represents the strongest prior work (Kostrikov et al., 2022).
- **D4RL AntMaze** (Fu et al., 2020). These 6 sparse reward tasks require an Ant agent to learn to walk and navigate through a maze to reach a goal. The offline data comprise *only sub-optimal* trajectories that can in principle be "stitched" together. Again, IQL + Fine-tuning represents the strongest prior work (Kostrikov et al., 2022).
- **D4RL Locomotion** (Fu et al., 2020). Lastly, we have 12 dense reward, locomotion tasks featuring offline data with varying levels of expertise. Off2On (Lee et al., 2021) has state-of-the-art performance on this suite of tasks.

For evaluation, we first include **SACfD**, a baseline studied in prior work (Večerík et al., 2017; Nair et al., 2020), which, similar to RLPD, is an off-policy approach that incorporates offline data during training. However, SACfD simply *initializes* the online replay buffer with the offline data. We implement this baseline using SAC without the additional design decisions discussed in subsection 4.5. Then, since there is no *single* prior method that achieves the best performance across all groups of tasks, we compare to the state-of-the-art method specific to each group (as listed above).We refer to this comparison in our plots as **Prior SoTA**. For all experiments in this work, we report the mean and standard deviation across *10 seeds* and aggregate the results across tasks within the three groups listed. For full detailed results broken down by task, see Appendix A, and for more environment details, see Appendix B.1.

We see that RLPD performs strongly, either matching or *significantly exceeding* the best known prior work on these challenging benchmarks(see Figure 4). We reiterate that we present results for RLPD and SACfD without doing *any pre-training*, unlike the Prior SoTA methods. So, while prior work (shown in blue) may achieve strong initial performance, the online improvement is more modest. On the other hand, our method reaches or surpasses this performance in the order of just 10k online samples. Notably, we outperform the best reported performance on the Sparse Adroit 'Door' task by $\mathbf{2.5}\times$. Moreover, to our knowledge, our method is the first to effectively 'solve' *all* AntMaze tasks. Furthermore, we are able to do so in *less than a third* the time-step budget allocated to prior methods.

Does RLPD transfer to pixels? Here we consider the medium and expert locomotion tasks in V-D4RL (Lu et al., 2022), an offline dataset with *only pixel observations*. V-D4RL is particularly challenging as the behavior policies are state-based, which means the data is partially observable in pixel-space. This is most obvious in 'Humanoid Walk', where body parts are visually occluded, thus behavior cloning (BC) can struggle to achieve strong performance. For evaluation, as our method seeks to accelerate online learning with offline data, we focus on data-efficiency—we introduce a challenge that we call "10%DMC", which involves training policies using only 10% of the total timesteps recommended by Yarats et al. (2022). As we use the medium and expert data, we include a **BC** baseline. To evaluate RLPD's ability to efficiently use offline data to boost online learning, we compare to a baseline approach that does not use the offline data. To isolate the effect of the utilization of offline data, we use the same architecture and policy optimizer as our method and label this baseline as **Online** in our plots. Then, to evaluate how this compares to the state-of-the-art sample-efficient RL method from pixels, we compare to **DrQ-v2** (Yarats et al., 2022).

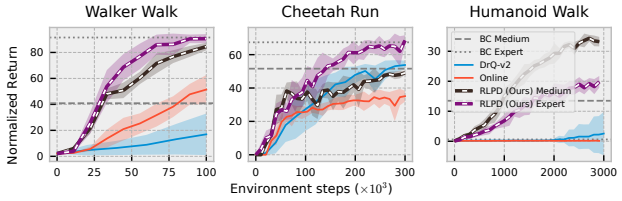


Figure 5. Our approach generalizes to vision-based domains, providing consistent improvements over existing approaches.

We see in Figure 5[2] that RLPD provides consistent improvements over purely online approaches, and in many cases greatly improves over a BC baseline. We see through the difference in RLPD (dashed black and purple lines) and the Online baseline that RLPD *effectively* utilizes the offline data to bootstrap learning. This conclusion holds as we compare to the SoTA vision-based RL method (blue).

Lastly, we test increasing UTD to 10 on one of the tasks—Cheetah Run with Expert offline data. Figure 6 shows a remarkable improvement in performance when learning with the offline dataset. To our knowledge, this is the first demonstration of a high UTD approach improving model-free pixel-based continuous control.

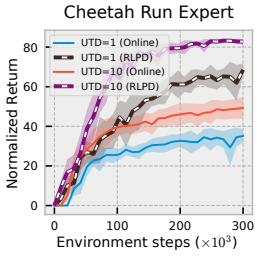


Figure 6. Increasing UTD with RLPD greatly improves sample efficiency from pixels.

5.1. RLPD Analysis and Ablation Study

Here, we address (3) and (4) by quantifying the effect of LayerNorm, and demonstrating the reliability of our proposed workflow (see Subsection 4.5).

Does LayerNorm mitigate value divergence? We now illustrate the importance of LayerNorm in mitigating catastrophic value divergence, and focus on tasks where overestimation is particularly prone. In Figure 7, we see LayerNorm is crucial for strong performance in the Adroit domain; excluding LayerNorm results in significantly higher variance across seeds and reduces mean performance.

To more clearly illustrate this effect, we construct a dataset of *only the expert human demonstration data* from the Adroit Sparse tasks (see "Expert Adroit Sparse Tasks" in Figure 7). This subset comprises just **22** of the 500 trajectories in the original dataset and is much more narrowly distributed by nature—representing a task with sparse rewards, limited demonstrations, and narrow offline data coverage—likely to exacerbate value divergence. Here we see a remarkable result: RLPD still *exceeds* prior work, despite significant restrictions in data. Moreover, removing LayerNorm now

[2]V-D4RL normalized return is episode return divided by 10.

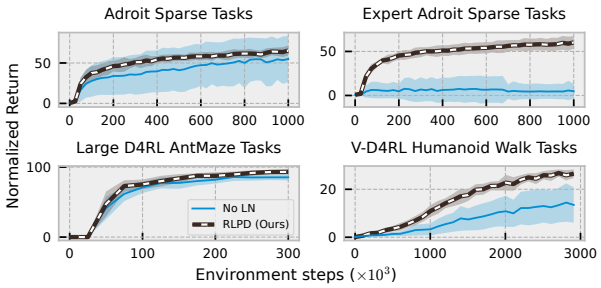


Figure 7. LayerNorm is crucial for strong performance, particularly when data are limited or narrowly distributed.

results in collapsed performance, with no progress made on any task. We further observe improvements in sample efficiency through the inclusion of LayerNorm in AntMaze and Humanoid Walk through reducing excessive extrapolation. Additional experiments and results are in Appendix A.

Design choice workflow. We now motivate our workflow in Section 4.5, demonstrating the importance of these design choices. We focus on the hardest tasks, namely 'Relocate' in Adroit Sparse, 'Large Diverse' in AntMaze, and 'Humanoid Expert' in V-D4RL, as we found these to be most sensitive. We provide full results on all tasks in Appendix A, noting that optimal decisions in the harder domains also produce optimal results in the easier domains.

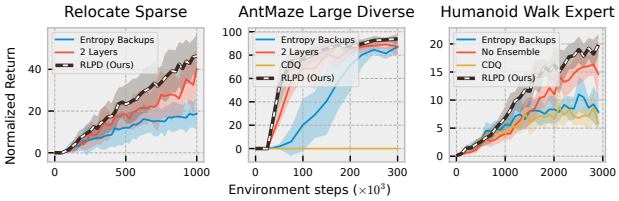


Figure 8. Our recommended starting design choices and workflow leads to strong performance on all tasks.

We see in Figure 8 that with the recommended environment-specific design choices provide strong performance; we see using entropy backups and smaller networks always results in worse performance. In 'AntMaze Large Diverse' however we see that with CDQ, performance deteriorates. Following our workflow, and ablating this by subsetting 1 critic is crucial to recovering strong performance. The same applies to 'Humanoid Walk Expert', whereby we surprisingly see that CDQ is detrimental to performance, despite its popularity in recent implementations. We also show the surprising positive effect of larger ensembles in pixel-based tasks, with the standard 2 member critic ensemble performing worse than the 10 member ensemble we use by default in RLPD.

Lastly, we conduct additional ablations to understand the importance of the design choices that we propose for our method, showing that, though the individual design decisions are simple, they are vital for good performance.

Critic regularization. Here we examine the effects of critic regularization on performance. We compare 3 approaches: weight-decay[3] (Večerík et al., 2017), Dropout (Hiraoka et al., 2022) and ensembling (Chen et al., 2021). We choose a subset of the prior experiments to evaluate on from the AntMaze, Adroit, and Locomotion sections.

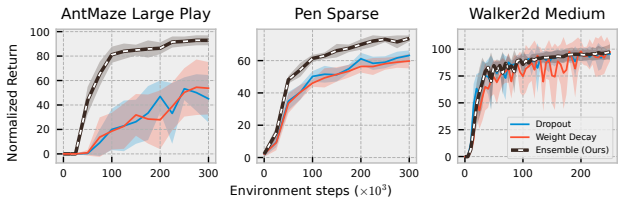


Figure 9. In general, critic ensembling provides the best performance. Dropout performs worse in sparse reward tasks.

In Figure 9, we see that ensembling is the strongest form of regularization. Notably, while Dropout performs well in the Locomotion domain 'walker2d-medium-v0', as affirmed by Hiraoka et al. (2022), it does not generalize to challenging sparse reward environments. We also see that weight-decay regularization is less performant in all domains.

Buffer initialization. In this section, we compare symmetric sampling with that of one which relies on *initializing* a replay buffer with offline data.

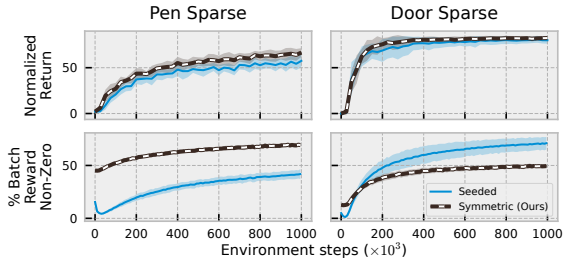


Figure 10. Symmetric sampling improves sample efficiency and reduces variance across seeds, and does not work by simply increasing the reward density in a batch.

First returning to the challenging Human Expert Demonstrations setting in Adroit, in Figure 10 we show two contrasting examples that demonstrate symmetric sampling effectively trades-off between replay and offline data. We observe that in the Pen environment, symmetric sampling clearly improves exploration by ensuring increasing reward density within mini-batches. In contrast, we see that although the buffer initialization approach explores just as well in the Door environment, symmetric sampling has the important effect of improving stability and decreases variance due to relying less on the higher-variance data generated by the online policy.

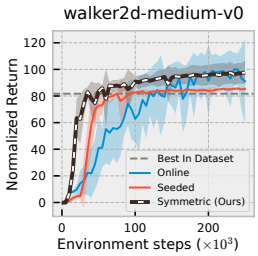


Figure 11. Initializing the buffer with large amounts of data limits improvement.

We now consider a situation whereby we have abundant suboptimal offline data, and see that again, our balanced sampling approach improves sample-efficiency. In Figure 11, initializing the buffer with high volumes of medium quality locomotion data gives initial improvement over online performance, but struggles asymptotically, likely due to a lack of on-policy data sampling, vital for online improvement. On the other hand, our symmetric sampling approach improves sample efficiency and matches asymptotic performance while reducing variance.

Sampling proportion sensitivity. We assess the sensitivity of our sampling approach away away from the "symmetric" ratio of 50% online/offline.

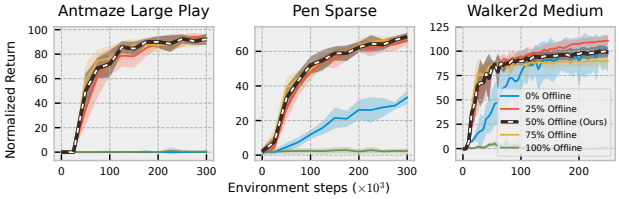


Figure 12. RLPD is not sensitive to replay proportion; 50% offers the best compromise between variance, speed of convergence, and asymptotic performance.

As we see in Figure 12, RLPD is not very sensitive to sampling proportion. While sampling 25% offline can marginally help asymptotic performance in 'walker2d-medium-v0', this comes at the expense of variance and sample efficiency in sparse reward tasks. We also affirm with our 100% offline results that RLPD is not an offline method, and key to its success is how it controls for divergence *without restricting exploration or behavior learning*.

6. Conclusion

In this work, we show that off-policy approaches can be adapted to leverage offline data when training online. We see that with careful application of key design choices, RLPD can attain remarkably strong performance, and demonstrate this on a total of **30 different tasks**. Concretely, we show that the unique combination of symmetric sampling, LayerNorm as a value extrapolation regularizer, and sample efficient learning is key to its success, resulting in our outperforming prior work by up to $\mathbf{2.5}\times$ on a large variety of competitive benchmarks. Moreover, our recommendations have negligible impact on computational efficiency compared to pure-online approaches, and are simple, allowing practi-

[3]We found a weight decay value of 0.01 worked best across a number of settings.

tioners to easily incorporate the insights of this work into existing approaches. To further improve adoptability, we recommend and demonstrate a workflow for practitioners that improves performance on a wide variety of tasks, and thus demonstrate that certain canonical design choices should be reconsidered when applying off-policy methods. Finally, to facilitate future research, we have released the RLPD codebase here: github.com/ikostrikov/rlpd, which features highly optimized off-policy algorithms for proprioceptive and pixel-based tasks in a single codebase written in JAX (Bradbury et al., 2018).

Acknowledgements

PJB is funded through the Willowgrove Foundation and the Les Woods Memorial Fund. This research was partly supported by the DARPA RACER program, ARO W911NF-21-1-0097, the Office of Naval Research under N00014-21-1-2838 and N00014-19-12042. This research used the Savio computational cluster resource provided by the Berkeley Research Computing program at the University of California, Berkeley (supported by the UC Berkeley Chancellor, Vice Chancellor for Research, and Chief Information Officer).

References

Andrychowicz, M., Raichuk, A., Stańczyk, P., Orsini, M., Girgin, S., Marinier, R., Hussenot, L., Geist, M., Pietquin, O., Michalski, M., Gelly, S., and Bachem, O. What matters for on-policy deep actor-critic methods? a large-scale study. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=nIAxjsniDzg.

Asada, H. and Hanafusa, H. Playback control of force teachable robots. *Transactions of the Society of Instrument and Control Engineers*, 15(3):410–411, 1979. doi: 10.9746/sicetr1965.15.410.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

Ball, P. J. and Roberts, S. J. Offcon^3: What is state of the art anyway?, 2021. URL https://arxiv.org/abs/2101.11331.

Bellman, R. A markovian decision process. *Indiana Univ. Math. J.*, 6:679–684, 1957. ISSN 0022-2518.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Cabi, S., Colmenarejo, S. G., Novikov, A., Konyushkova, K., Reed, S. E., Jeong, R., Zolna, K., Aytar, Y., Budden, D., Vecerík, M., Sushkov, O. O., Barker, D., Scholz, J., Denil, M., de Freitas, N., and Wang, Z. Scaling data-driven robotics with reward sketching and batch reinforcement learning. *Robotics: Science and Systems XVI*, 2019.

Cetin, E., Ball, P. J., Roberts, S., and Celiktutan, O. Stabilizing off-policy deep reinforcement learning from pixels. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 2784–2810. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/cetin22a.html.

Chen, X., Wang, C., Zhou, Z., and Ross, K. W. Randomized ensembled double q-learning: Learning fast without a model. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=AY8zfZm0tDd.

Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., and Madry, A. Implementation matters in deep rl: A case study on ppo and trpo. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1etN1rtPB.

Ernst, D., Geurts, P., and Wehenkel, L. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6(18):503–556, 2005. URL http://jmlr.org/papers/v6/ernst05a.html.

Fox, R., Pakman, A., and Tishby, N. Taming the noise in reinforcement learning via soft updates. In *32nd Conference on Uncertainty in Artificial Intelligence (UAI)*, 2016.

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

Fujimoto, S., van Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. *ArXiv*, abs/1802.09477, 2018.

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*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2052–2062. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/fujimoto19a.html.

Furuta, H., Kozuno, T., Matsushima, T., Matsuo, Y., and Gu, S. Co-adaptation of algorithmic and implementational innovations in inference-based deep reinforcement learning. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=vLyI__SoeAe.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *International Conference on Machine Learning (ICML)*, 2018a.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., and Levine, S. Soft actor-critic algorithms and applications, 2018b. URL https://arxiv.org/abs/1812.05905.

Hansen, N., Lin, Y., Su, H., Wang, X., Kumar, V., and Rajeswaran, A. Modem: Accelerating visual model-based reinforcement learning with demonstrations. *arXiv preprint*, 2022.

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., and Meger, D. Deep reinforcement learning that matters. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. doi: 10.1609/aaai.v32i1.11694. URL https://ojs.aaai.org/index.php/AAAI/article/view/11694.

Hester, T., Vecerik, M., Pietquin, O., Lanctot, M., Schaul, T., Piot, B., Horgan, D., Quan, J., Sendonaris, A., Osband, I., Dulac-Arnold, G., Agapiou, J., Leibo, J., and Gruslys, A. Deep q-learning from demonstrations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. doi: 10.1609/aaai.v32i1.11757. URL https://ojs.aaai.org/index.php/AAAI/article/view/11757.

Hiraoka, T., Imagawa, T., Hashimoto, T., Onishi, T., and Tsuruoka, Y. Dropout q-functions for doubly efficient reinforcement learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=xCVJMsPv3RT.

Kalashnikov, D., Irpan, A., Pastor, P., Ibarz, J., Herzog, A., Jang, E., Quillen, D., Holly, E., Kalakrishnan, M., Vanhoucke, V., and Levine, S. Scalable deep reinforcement learning for vision-based robotic manipulation. In Billard, A., Dragan, A., Peters, J., and Morimoto, J. (eds.), *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pp. 651–673. PMLR, 29–31 Oct 2018. URL https://proceedings.mlr.press/v87/kalashnikov18a.html.

Kostrikov, I., Yarats, D., and Fergus, R. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=GY6-6sTvGaf.

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=68n2s9ZJWF8.

Lee, S., Seo, Y., Lee, K., Abbeel, P., and Shin, J. Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble. In *5th Annual Conference on Robot Learning*, 2021. URL https://openreview.net/forum?id=AlJXhEI6J5W.

Levine, S. and Koltun, V. Guided policy search. In Dasgupta, S. and McAllester, D. (eds.), *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 1–9, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/levine13.html.

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

Li, Q., Kumar, A., Kostrikov, I., and Levine, S. Efficient deep reinforcement learning requires regulating statistical overfitting. In *Deep Reinforcement Learning Workshop NeurIPS 2022*, 2022. URL https://openreview.net/forum?id=Jwfa-oyQduy.

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. In *Workshop on Learning from Diverse, Offline Data*, 2022. URL https://openreview.net/forum?id=bPOBIKaqLba.

Lu, Y., Hausman, K., Chebotar, Y., Yan, M., Jang, E., Herzog, A., Xiao, T., Irpan, A., Khansari, M., Kalashnikov, D., and Levine, S. Aw-opt: Learning robotic skills with imitation andreinforcement at scale. In *5th Annual Conference on Robot Learning*, 2021.

Mirhoseini, A., Goldie, A., Yazgan, M., Jiang, J. W., Songhori, E. M., Wang, S., Lee, Y.-J., Johnson, E., Pathak, O., Nazi, A., Pak, J., Tong, A., Srinivasa, K., Hang, W., Tuncer, E., Le, Q. V., Laudon, J., Ho, R., Carpenter, R., and Dean, J. A graph placement methodology for fast chip design. *Nature*, 594 7862:207–212, 2021.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. Human-level control through deep reinforcement learning. *Nature*, 518 (7540):529–533, February 2015. ISSN 00280836. URL http://dx.doi.org/10.1038/nature14236.

Moskovitz, T., Parker-Holder, J., Pacchiano, A., Arbel, M., and Jordan, M. Tactical optimism and pessimism for deep reinforcement learning. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 12849–12863. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/6abcc8f24321d1eb8c95855eab78ee95-Paper.pdf.

Nair, A., McGrew, B., Andrychowicz, M., Zaremba, W., and Abbeel, P. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6292–6299. IEEE Press, 2018a. doi: 10.1109/ICRA.2018.8463162. URL https://doi.org/10.1109/ICRA.2018.8463162.

Nair, A., McGrew, B., Andrychowicz, M., Zaremba, W., and Abbeel, P. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 6292–6299. IEEE, 2018b.

Nair, A., Gupta, A., Dalal, M., and Levine, S. AWAC: Accelerating online reinforcement learning with offline datasets. *arXiv*, June 2020.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Gray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=TG8KACxEON.

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 *Proceedings of Robotics: Science and Systems (RSS)*, 2018.

Rezaeifar, S., Dadashi, R., Vieillard, N., Hussenot, L., Bachem, O., Pietquin, O., and Geist, M. Offline reinforcement learning as anti-exploration. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36 (7):8106–8114, Jun. 2022. doi: 10.1609/aaai.v36i7.20783. URL https://ojs.aaai.org/index.php/AAAI/article/view/20783.

Ross, S. and Bagnell, J. A. Agnostic system identification for model-based reinforcement learning. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pp. 1905–1912, Madison, WI, USA, 2012. Omnipress. ISBN 9781450312851.

Rudner, T. G. J., Lu, C., Osborne, M., Gal, Y., and Teh, Y. W. On pathologies in KL-regularized reinforcement learning from expert demonstrations. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=sS8rRmgAatA.

Schaal, S. Learning from demonstration. In Mozer, M., Jordan, M., and Petsche, T. (eds.), *Advances in Neural Information Processing Systems*, volume 9. MIT Press, 1996. URL https://proceedings.neurips.cc/paper/1996/file/68d13cf26c4b4f4f932e3eff990093ba-Paper.pdf.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016. doi: 10.1038/nature16961.

Song, Y., Zhou, Y., Sekhari, A., Bagnell, D., Krishnamurthy, A., and Sun, W. Hybrid RL: Using both offline and online data can make RL efficient. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=yyBis80iUuU.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

Thrun, S. and Schwartz, A. Issues in using function approximation for reinforcement learning. In *Proceedings of 4th Connectionist Models Summer School*. Erlbaum Associates, June 1993.

van Hasselt, H., Guez, A., and Silver, D. Deep reinforcement learning with double q-learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1), Mar. 2016. doi: 10.1609/aaai.v30i1.

10295. URL https://ojs.aaai.org/index.php/AAAI/article/view/10295.

Večerík, M., Hester, T., Scholz, J., Wang, F., Pietquin, O., Piot, B., Heess, N., Rothörl, T., Lampe, T., and Riedmiller, M. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv*, July 2017.

Wagenmaker, A. and Pacchiano, A. Leveraging offline data in online reinforcement learning. *arXiv preprint arXiv:2211.04974*, 2022.

Yarats, D., Fergus, R., Lazaric, A., and Pinto, L. Mastering visual continuous control: Improved data-augmented reinforcement learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=_SJ-_yyes8.

Zhang, H., Xu, W., and Yu, H. Policy expansion for bridging offline-to-online reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=-Y34L45JR6z.

A. Detailed Experiments

A.1. Sparse Adroit

Full Results Here we present the full results for Sparse Adroit, including LayerNorm and environment-specific design choice ablations.

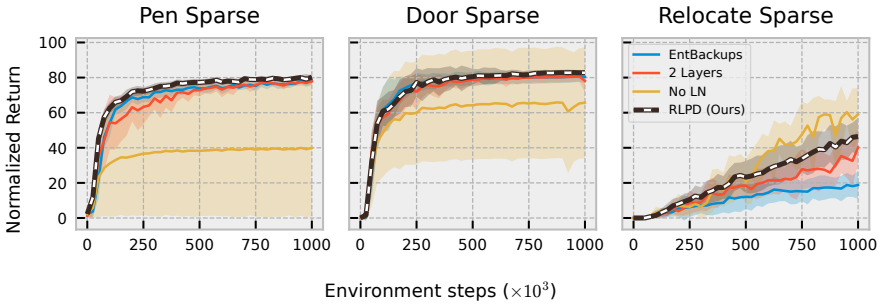


Figure 13. Full Adroit Results.

We see that LayerNorm greatly reduces variance on the Pen and Door environments, however slightly harms mean performance on Relocate. Taken together however, LayerNorm is still a vital ingredient for reliable performance on this domain.

A.2. AntMaze

Full Results Here we present the full results for AntMaze, including LayerNorm and environment-specific design choice ablations.

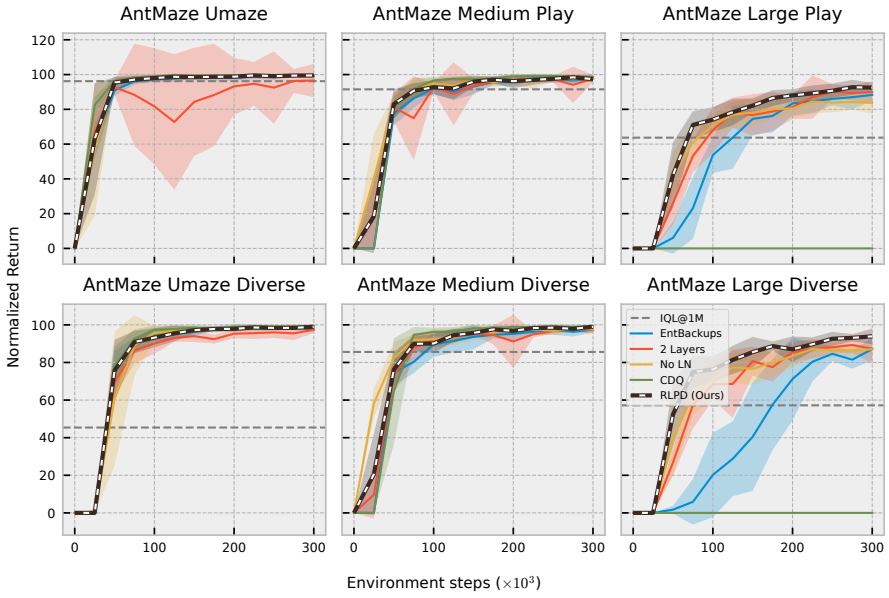


Figure 14. Full AntMaze results

We see that our recommended design choices are vital for strong performance on the harder Large tasks, and furthermore see that using deeper 3-layer networks seems to help stability across all tasks. As we see, the best design choices on the Large environments are also optimal for the Umaze and Medium environments.

Gradient Steps per Environment Step Ablations We see LayerNorm is incredibly effective in a setting whereby we perform a single gradient update per time-step in Figure 15. This may be required for learning on real robots, where computational efficiency is important.

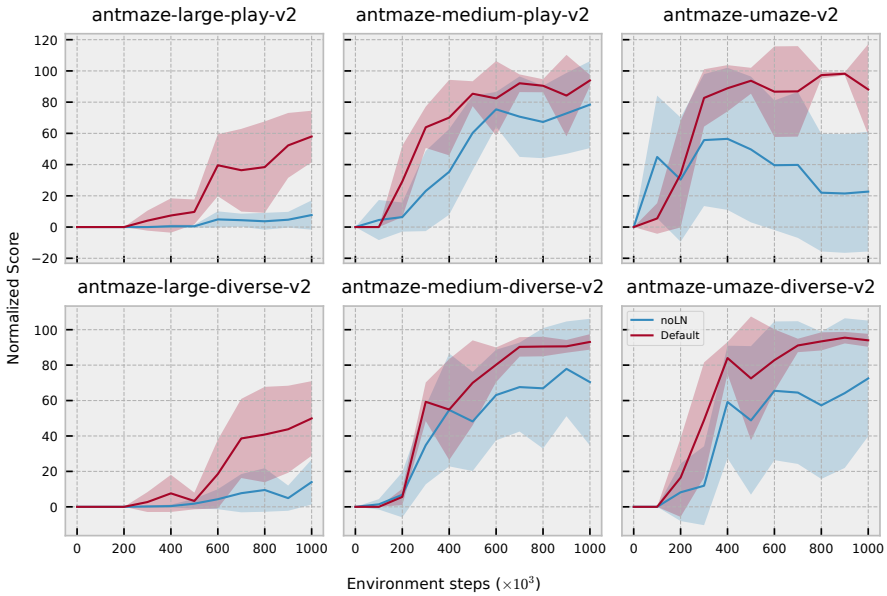


Figure 15. SAC with and without Layer Normalization.

In Figure 16, we see the impact of increasing the number of gradient steps per time-step. As we see, both sample efficiency stability improves greatly.

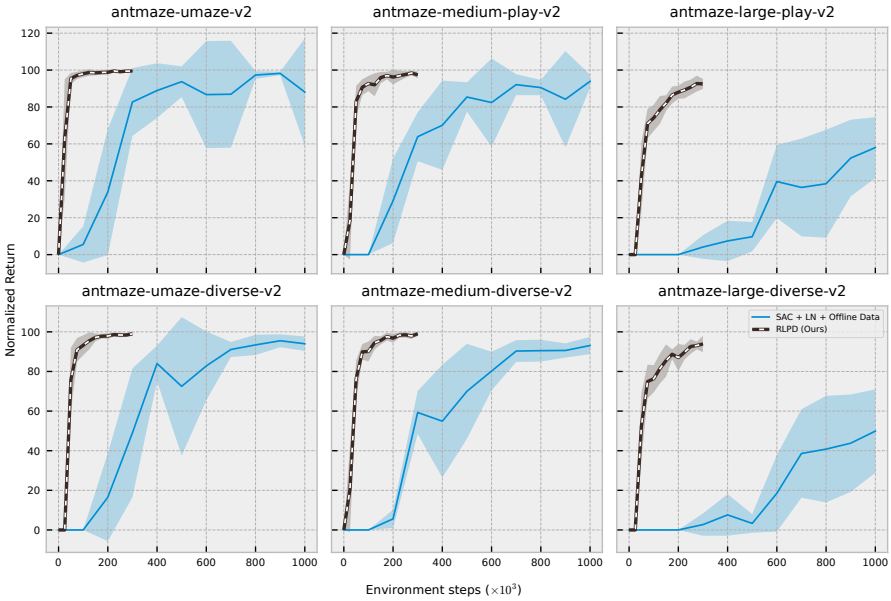


Figure 16. RLPD v.s. SAC + LN + Offline Data.

Full IQL comparison Here we compare to IQL + Finetuning. We show pre-training gradient steps on IQL as negative indices in the x-axis. As we see in Figure 17, despite IQL learning a strong initialization, it struggles to improve beyond this. In comparison, RLPD is able to quickly match, then greatly exceed IQL.

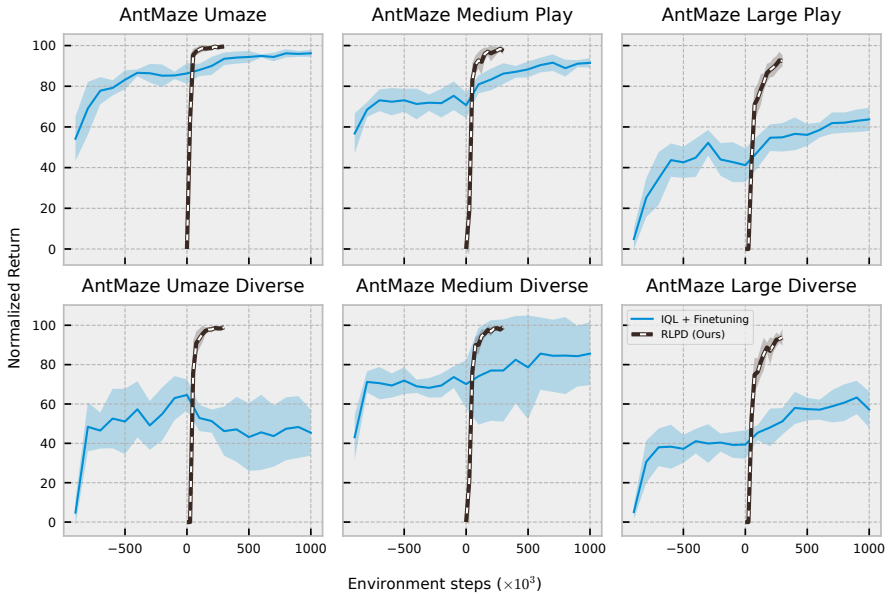


Figure 17. RLPD v.s. IQL + Finetuning. We show offline pre-training steps for IQL using negative indices.

A.3. D4RL Locomotion

Full Results Here we present the full results for D4RL Locomotion, including LayerNorm and baselines. We emphasize that these tasks are not necessarily a good use case for online learning with offline datasets, as the tasks can be solved relatively quickly with purely online approaches, and there is no inherent difficulty regarding exploration.

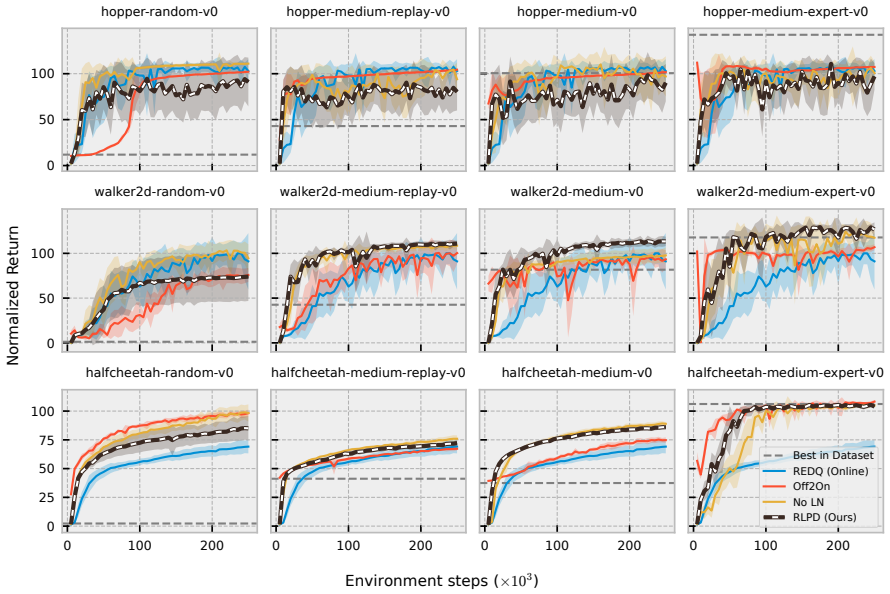


Figure 18. D4RL Ablations.

The impact of LayerNorm is not so clear cut in Figure 18; this is to be expected as online approaches already achieve strong results in this domain. Notably, we see strong performance compared to baselines in the medium-expert domains.

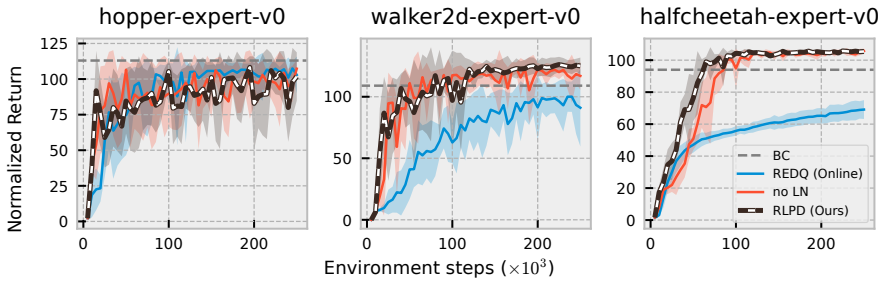


Figure 19. D4RL Ablation on Expert Data.

We also test on the narrow Expert dataset, not tested by (Lee et al., 2021). In Figure 19 we see LayerNorm can marginally help both sample efficiency, and asymptotic performance.

A.4. V-D4RL Locomotion

Full Results Here we present the full results for V-D4RL Locomotion, including LayerNorm and environment-specific design choice ablations.

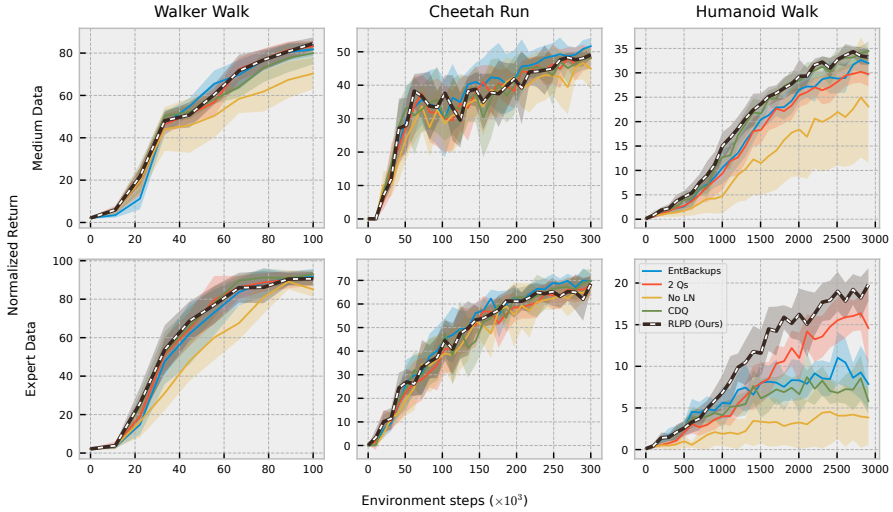
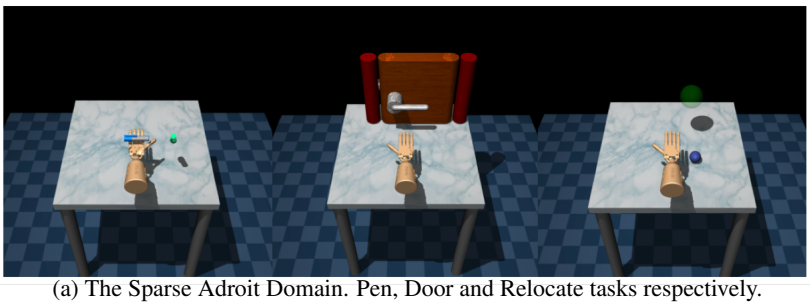


Figure 20. VD4RL Ablations.

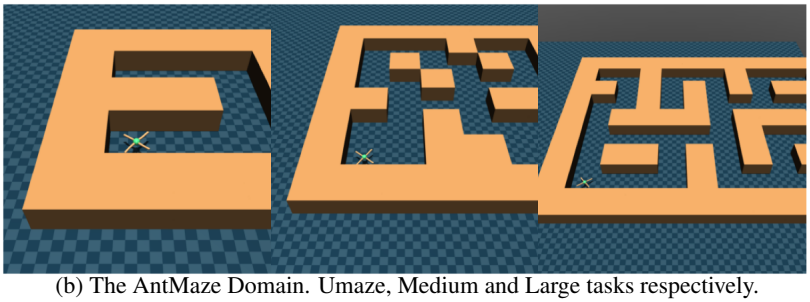
As we see, LayerNorm helps significantly in the Walker and Humanoid environments. We also see the positive impact of our recommended design choices in the complex Humanoid domain.

B. Experimental Details

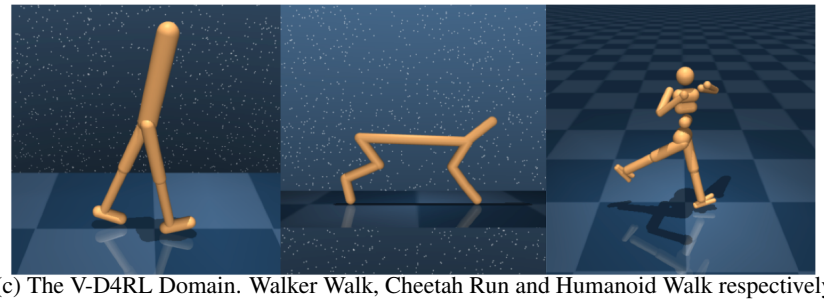
B.1. Further Environment Details



(a) The Sparse Adroit Domain. Pen, Door and Relocate tasks respectively.



(b) The AntMaze Domain. Umaze, Medium and Large tasks respectively.



(c) The V-D4RL Domain. Walker Walk, Cheetah Run and Humanoid Walk respectively.

Figure 21. Visualizations of the environments we consider.

We provide further details about the key domains we evaluate on. In Figure 21 we provide visualizations of the environments.

Sparse Adroit In these tasks, reward is a binary variable that indicates whether the task has been completed successfully or not. In prior work (Nair et al., 2020; Rudner et al., 2021), it is common to see success rate used as the metric, which simply determines if the task has been completed in any time step. However Kostrikov et al. (2022) use a more challenging metric that involves speed of completion. Concretely, return is calculated as the percentage of the total timesteps in which the task is considered solved (note that there are no early terminations). For example, in the 'Pen' task, where the horizon is 100 timesteps, if a policy achieves a Normalized Score of 80, that means in 80 timesteps, the task is considered solved. This effectively means that the policy was able to solve the task in 20 timesteps. At each evaluation, we perform 100 trials.

D4RL: AntMaze In these tasks, reward is a binary variable that indicates whether the agent has reached the goal. Upon reaching the goal, the episode terminates. The normalized return is therefore the proportion of evaluation trials that were successful. We follow prior work, and perform 100 trials, and measure Normalized Return as the percentage of successful trials.

B.2. Hyperparameters

Here we list the hyperparameters for RLPD in Table 1, and the environment-specific hyperparameters in Table 2.

Table 1. RLPD hyperparameters.

Parameter	Value
Online batch size	128
Offline batch size	128
Discount (γ)	0.99
Optimizer	Adam
Learning rate	3×10^{-4}
Ensemble size (E)	10
Critic EMA weight (ρ)	0.005
Gradient Steps (State Based) (G or UTD)	20
Network Width	256 Units
Initial Entropy Temperature (α)	1.0
Target Entropy	$-\dim(\mathcal{A})/2$
Pixel-Based Hyperparameters	
Action repeat	2
Observation size	[64, 64]
Image shift amount	4

Table 2. Environment specific hyperparameters.

Environment	CDQ	Entropy Backups	MLP Architecture
Locomotion	True	True	2 Layer
AntMaze	False	False	3 Layer
Adroit	True	False	3 Layer
DMC (Pixels)	False	False	2 Layer