DON'T CHANGE THE ALGORITHM, CHANGE THE DATA: EXPLORATORY DATA FOR OFFLINE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

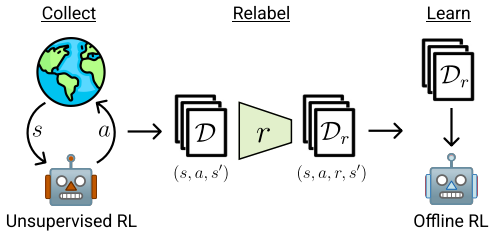
ABSTRACT

Recent progress in deep learning has relied on access to large and diverse datasets. Such data-driven progress has been less evident in offline reinforcement learning (RL), because offline RL data is usually collected to optimize specific target tasks limiting the data's diversity. In this work, we propose Exploratory data for Offline RL (ExORL), a data-centric approach to offline RL. ExORL first generates data with unsupervised reward-free exploration, then relabels this data with a downstream reward before training a policy with offline RL. We find that exploratory data allows vanilla off-policy RL algorithms, without any offline-specific modifications, to outperform or match state-of-the-art offline RL algorithms on downstream tasks. Our findings suggest that data generation is as important as algorithmic advances for offline RL and hence requires careful consideration from the community.

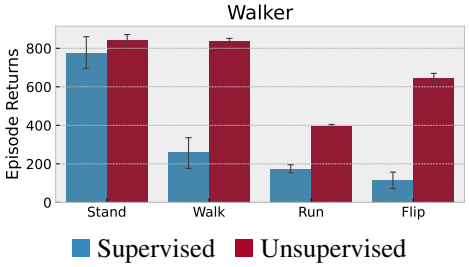
1 INTRODUCTION

Large and diverse datasets have been the cornerstones of many impressive successes at the frontier of machine learning, such as image recognition (Deng et al., 2009; Krizhevsky et al., 2012; Radford et al., 2021), natural language processing (Radford et al., 2019; Raffel et al., 2019; Wang et al., 2019; Brown et al., 2020), and protein folding (Consortium, 2020; Jumper et al., 2021). Training on diverse datasets often dramatically improves performance and enables impressive generalization capabilities.

In reinforcement learning (RL), however, the potential of large and diverse datasets has not yet been fully realized. The historically dominant approach in RL has been to train online from scratch by interacting with a task-specific environment (Sutton & Barto, 2018; Mnih et al., 2013). More recently there has been increased interest in learning offline from fixed datasets (Levine et al., 2020; Fujimoto et al., 2019). This line of research has been accelerated by benchmark datasets such as D4RL (Fu et al., 2020) and RL Unplugged (Gulcehre et al., 2020).



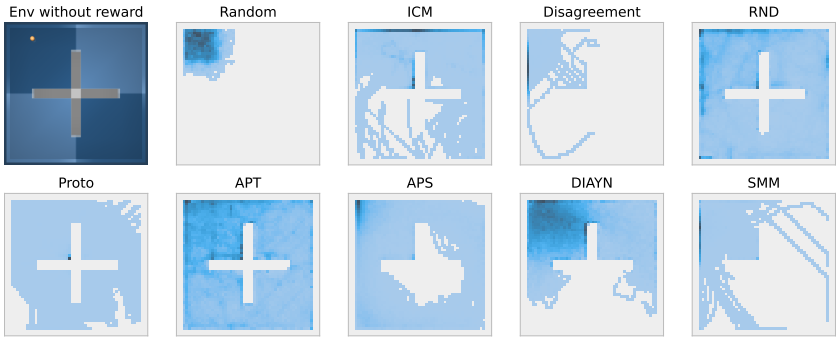
(a) The three phases of ExORL: data collection with unsupervised RL, data relabeling with the downstream reward, and learning on the labeled data with offline RL.



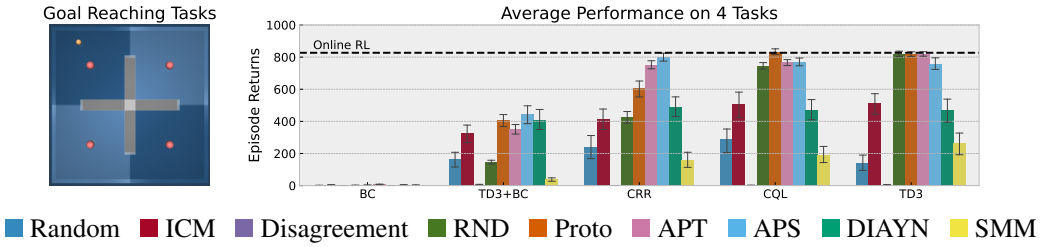
(b) Offline RL evaluation using vanilla TD3 of supervised data (the replay buffer of a TD3 agent trained on Walker Stand) versus unsupervised exploratory data (collected reward-free on Walker with Proto). Unsupervised data demonstrates an average performance improvement of 106% across the four evaluation tasks.

Figure 1: Summary of the ExORL framework.

Currently, offline RL datasets are collected by training task-specific RL algorithms and saving the data visited during the training process. Offline RL algorithms are then evaluated on the same task that the dataset was created with. As a result, very strong performance on these benchmarks can be achieved with simple methods like be-



(a) **Collect**: an unsupervised RL agent interacts with the environment in a reward-free manner and stores the acquired transitions into a dataset $\mathcal{D}$ of (s, a, s') tuples. We visualize state distributions of the pointmass for each collected dataset $\mathcal{D}$.



(b) **Relabel** and **Learn**: Each tuple in $\mathcal{D}$ is relabeled by a task-specific reward function to yield a new dataset $\mathcal{D}_r$ of (s, a, r, s') tuples (there are 4 different rewards in this example, one for each goal). Finally, we run an offline RL algorithm on each dataset $\mathcal{D}_r$ in order to learn a policy to maximize the specified reward. Here, offline RL is performed by various state of the art offline RL algorithms as well as standard supervised learning (BC) and off-policy RL (TD3).

Figure 2: Our ExORL framework exploratory data collection for offline RL consists of three phases: *collect* **(a)**, *relabel* and *learn* **(b)**. Here we showcase our method on a pointmass maze environment to demonstrate that it is possible to collect exploratory data in a completely unsupervised manner that enables effective multi-task offline RL.

havior cloning (Pomerleau, 1988; Chen et al., 2021b) or one step of policy iteration (Brandfonbrener et al., 2021; Gulcehre et al., 2021). This raises a fundamental question for offline RL – What happens when your data does not come from the same task?

In this work, we propose a new framework for offline RL that can test the ability of offline RL algorithms to generalize to tasks that are not observed during the creation of the dataset. This framework, which we call Exploratory data for Offline RL (ExORL), has three steps: collect, relabel, and learn. To instantiate ExORL, we *collect* datasets using nine different unsupervised exploration strategies (Laskin et al., 2021) on several environments from the DeepMind control suite Tassa et al. (2018). Then, we *relabel* the data with different reward functions. Finally, we *learn* a policy by running several different offline RL algorithms for each relabeled dataset. A schematic of the framework can be found in Figure 1a and an example in pointmass maze environment in Figure 2.

We provide a thorough analysis of current offline RL methods by running large-scale experiments with ExORL. The main findings of these experiments are:

1. With sufficiently diverse exploratory data, vanilla off-policy RL agents can effectively learn offline, and even outperform carefully designed offline RL algorithms. This suggests that advances in data collection are as important as algorithmic advances for offline RL.

2. Compared to data collected in a supervised manner (i.e. for a task-specific reward function), the unsupervised exploratory data allows offline RL algorithms to learn multiple behaviors from a single ExORL dataset. For example, unsupervised exploration yields a 106% performance increase over supervised data on multi-task learning in the Walker environment (Figure 1b).

Finally, ExORL will be released as a benchmark suite of exploratory datasets to test offline RL algorithms in a substantially different regime than prior benchmark suites. While prior work focuses on datasets with limited coverage, our unsupervised datasets are collected to maximize various notions of coverage or diversity. We should emphasize that we are not saying that these new datasets are better offline RL benchmarks, just that they provide a significantly different and important challenge on the road to useful offline RL.

2 Related Work

Datasets for offline RL. The most closely related line of work is that of data-collection and benchmarking in offline RL. Both the D4RL (Fu et al., 2020) and RL Unplugged (Gulcehre et al., 2020; Mathieu et al., 2021) benchmark suites consist of datasets collected by policies attempting to optimize task-specific rewards. These datasets are either replay buffers of the training run or trajectories collected by a snapshot from somewhere in training. D4RL also includes some demonstration and goal-directed data. Importantly, the datasets are then tested on the same reward they were trained on (or the demonstrations they were targeting).

In contrast, our contribution is to use unsupervised exploratoy data collection to test offline RL algorithms. This creates datasets that facilitate multi-task offline RL from a single dataset and poses a very different challenge for offline RL than prior benchmarking work.

Unsupervised pre-training for RL. Recently, URLB (Laskin et al., 2021) proposed a benchmarking protocol for unsupervised pre-training algorithms in RL. In that work, each pre-training algorithm is allowed a fixed budget of unsupervised interactions with the environment and then oututs a pre-trained policy that is used to initialize an online RL algorithm at test time. In our work, instead of using a pre-trained policy as the output of the pre-training phase and then training online, we use the *dataset* as the output of the pre-training phase and then train entirely *offline*. To perform this data collection, we consider the same eight pre-training algorithms as in URLB (Pathak et al., 2017; Burda et al., 2019; Eysenbach et al., 2019; Pathak et al., 2019; Lee et al., 2019; Liu & Abbeel, 2021b;a; Yarats et al., 2021) as well as a uniformly random data collector.

There is also a line of purely theoretical work that studies the "reward-free RL" problem in the tabular and linear cases Jin et al. (2020); Wang et al. (2020a). Much like the goal of our protocol, these papers focus on collecting a dataset that allows for offline learning of a near optimal policy for any reward function. Our work can be viewed as an empirical evaluation of several different ways to scale up these reward-free RL algorithms to large domains with neural networks.

Multi-task reinforcement learning. Another important related line of work focuses on sharing data across tasks. Here, data collection is generally directed at multiple different goals with some combination of demonstrations and standard RL with task-specific rewards Riedmiller et al. (2018); Cabi et al. (2019); Kalashnikov et al. (2021); Yu et al. (2021); Chen et al. (2021a). In contrast, we focus on studying the capabilities of purely unsupervised data collection. We view these two research directions as complementary. For example, they could potentially be combined in future work, where unsupervised data could be added to multi-task data to facilitate transfer.

Another line of work generates goal-based multi-task data using different goals as the different tasks where the goals can be generated in an unsupervised manner (Andrychowicz et al., 2017; Ecoffet et al., 2019; Dendorfer et al., 2020; Mendonca et al., 2021). Endrawis et al. (2021) in particular uses goal-based task generation to collect exploratory data to use with offline RL on robotic tasks. In contrast, we do not use any goal-based and focus our efforts on understanding how offline RL methods perform in the exploratory setting.

We should also note that our relabeling procedure is not novel and similar ideas have been used widely in the multi-task and goal-based RL literature (Andrychowicz et al., 2017; Riedmiller et al., 2018; Sekar et al., 2020; Eysenbach et al., 2020; Li et al., 2020; Kalashnikov et al., 2021).

Decoupling exploration and exploitation. A few recent position papers argue for separating the exploration and exploitation problems to facilitate scalable reinforcement learning (Levine, 2021; Riedmiller et al., 2021). Ideally, collecting large datasets in an unsupervised manner and re-using the data across many downstream tasks can dramatically improve data-efficiency on a per-task basis.

Moreover, by splitting the problem into two, we can more effectively isolate the effects of each component of the algorithm. Our paper can be seen as taking a first step towards instantiating this vision of effectively decoupling exploration and exploitation.

Concurrent work. Concurrently to our paper, Lambert et al. (2022) proposes a similar framework for unsupervised exploration followed by offline RL. In contrast to our work, they focus on proposing a novel MPC-based exploration technique and do not compare multiple offline RL algorithms in the exploratory setting, instead they only use CRR.

3 Our framework: ExORL

In this section we explicitly define our framework for exploratory data collection for offline RL (ExORL). The framework consists of three steps: collect, relabel, and learn. The rest of this section goes through each step in detail and the protocol is summarized in Algorithm 1.

Algorithm 1 ExORL: Collect-Relabel-Learn

Input: trajectory budget B, environment $\mathcal{E}$, data collection algo $\mathcal{C}$, reward function r, offline RL algo $\mathcal{O}$
Collect: Run $\mathcal{C}$ on $\mathcal{E}$ to collect B trajectories into a dataset $\mathcal{D}$ of (s, a, s') tuples.
Relabel: Update each tuple in $\mathcal{D}$ by using r to get $(s, a, r(s, a), s')$ tuples. Call the relabeled dataset $\mathcal{D}_r$.
Learn: Run $\mathcal{O}$ on the labeled data in $\mathcal{D}_r$.

We assume that we have access to a reward-free episodic MDP environment $\mathcal{E}$ with state space $\mathcal{S}$, action space $\mathcal{A}$, and stochastic transition dynamics determined by $P(\cdot|s, a)$. In our problem setup we are given an interaction budget B that limits the number of trajectories that we can collect in the collection phase. Rewards for relabeling can be defined by any function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$.

Collect. The first phase of ExORL is to collect the unlabeled exploratory data. To do this, we can use any algorithm $\mathcal{C}$ that at each episode k outputs a policy π_k conditioned on the history of prior interactions. Each episode collected by $\mathcal{C}$ is stored in the dataset $\mathcal{D}$ as a series of (s, a, s') tuples. We run the collection algorithm $\mathcal{C}$ until we have collected B episodes into the dataset $\mathcal{D}$.

In practice, we evaluate nine different unsupervised data collection algorithms as $\mathcal{C}$ in this phase. The first is a simple baseline that always outputs the uniformly random policy. The remaining eight are taken from URLB (Laskin et al., 2021) and can be sorted into three groups:

- Knowledge-based methods that maximize error of a predictive model: ICM (Pathak et al., 2017), Disagreement (Pathak et al., 2019), RND (Burda et al., 2019).
- Data-based algorithms that maximize some estimate of coverage of the state space: APT (Liu & Abbeel, 2021b) and ProtoRL Yarats et al. (2021).
- Competence-based algorithms that learn a diverse set of skills: DIAYN Eysenbach et al. (2019), SMM Lee et al. (2019), and APS (Liu & Abbeel, 2021a).

Relabel. Having collected a dataset $\mathcal{D}$ of (s, a, s') tuples, the next phase of our protocol relabels the data using the given reward function r. This simply requires evaluating $r(s, a)$ at each tuple in the dataset and then adding $(s, a, r(s, a), s')$ to the relabeled dataset $\mathcal{D}_r$.

In practice, we use either standard or hand-designed reward functions in each environment (Tassa et al., 2018; Laskin et al., 2021), and are detailed in Appendix H. We note, however, that our framework also allows for learning reward function (i.e. inverse RL or successor features).

Learn. Finally, we can learn a policy by running an offline RL algorithm $\mathcal{O}$ on our labeled dataset $\mathcal{D}_r$. The offline RL algorithm learns entirely offline by sampling tuples from the dataset and then outputs the final policy to be evaluated online in the environment $\mathcal{E}$ by calculating the return with respect to the reward function r.

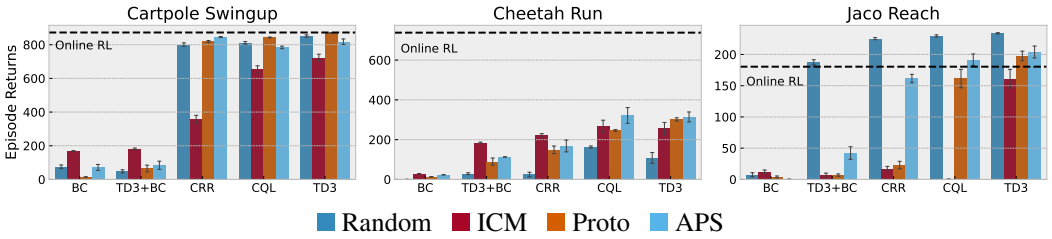


Figure 3: Offline evaluation of unsupervised datasets on one task for each of three different domains. Here we choose four representative unsupervised exploration algorithms, for full results see Appendix D. Vanilla TD3 usually outperforms all three offline RL algorithms.

In practice, we run five different offline RL algorithms for each reward function on each dataset. As a baseline we run simple behavior cloning. Then we run three different state of the art offline RL algorithms that each use a different mechanism to prevent extrapolation beyond the actions in the data. These algorithms are: CRR (Wang et al., 2020b) which uses filtered behavior cloning, CQL (Kumar et al., 2020) which uses pessimistic Q estimates, and TD3+BC (Fujimoto & Gu, 2021) which regularizes toward the behavior. Finally, we run standard TD3 (Fujimoto et al., 2018) as a baseline to test what happens when we run an off-policy RL algorithm that was originally designed for the online setting and has no mechanism explicitly designed to prevent extrapolation in the offline setting.

4 Experiments

In this section we attempt to answer these questions: **Q1**: Can the diversity of unsupervised data collected by ExORL enable vanilla off-policy RL agents to work in the offline setting? **Q2**: Is it possible to relabel this unsupervised data to facilitate multi-task offline RL? **Q3**: Is exploratory data necessary for multi-task offline RL? **Q4**: Is it useful to mix exploratory data with task-specific data?

4.1 Setup

We evaluate our framework on a set of challenging environments from the DeepMind control suite (Tassa et al., 2018). Each environment has several different rewards that we use to evaluate multi-task generalization. See Appendix H for detailed overview of the environments. We use URLB (Laskin et al., 2021) for unsupervised data collection with a budget of 1M reward-free environment interactions (1000 episodes) unless stated otherwise. The transitions in these datasets are then relabeled under a specific reward and used for offline RL. For offline RL we use five algorithms: BC, TD3+BC, CRR, CQL, and TD3. We adhere closely to the original hyper-parameter settings for each algorithm, but in several cases we perform hyper-parameter tuning to achieve best possible performance. See Appendix B for more details. We train offline RL algorithms for 500k gradient updates and then evaluate by rolling out 10 episodes in the environment. We report mean and standard error across 10 seeds. A description of computational requirements is in Appendix G.

4.2 Comparing offline RL algorithms on ExORL

Q1: Can the diversity unsupervised data collected by ExORL enable vanilla off-policy RL agents to work in the offline setting? **A1**: *Yes, vanilla off-policy RL agents can perform well from* ExORL *data, in particular vanilla TD3 outperforms the other offline RL algorithms.*.

To verify this, we took three environments (Cartpole, Cheetah, and Jaco) and one reward for each environment (Swingup, Run, and Reach respectively). For each environment we run all nine exploration algorithm, but for clarity only present four in the main text that represent each of the main families of algrithms (ICM: knowledge-based, Proto: data-based, APS: competence-based, and Random as a baseline). We then evaluate with five different offline RL algorithms. Results are in Figure 3 and Appendix D.

Specifically, comparing the results on these datasets we see that vanilla TD3 consistently outperforms more sophisticated offline RL algorithms (TD3+BC, CRR, and CQL). This is a somewhat surprising

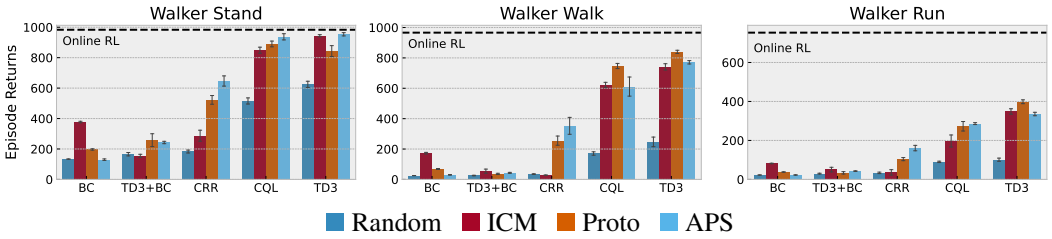


Figure 4: Offline evaluation of datasets for the Walker environment under three different rewards (Stand, Walk, and Run). We observe that ExORL allows for data relabeling to enable multi-task offline RL. See Appendix E for full results across all exploration algorithms.

result because the offline RL algorithms were designed specifically for the offline setting while TD3 was not. This emphasizes how the unsupervised data collected by ExORL provides a substantially different setting than the traditional offline RL problems that the algorithms were designed for. We hypothesize that this difference occurs because ExORL produces diverse datasets that allow for more aggressive trajectory stitching and less severe extrapolation than prior datasets.

While the relative strength of TD3 is consistent, the results are somewhat more varied across the different data collection algorithms. For example, in Jaco, the random data collection surprisingly outperforms all of the unsupervised exploration algorithms. This is due to over-exploration by the unsupervised algorithms which causes the arm to explore low-reward positions outside of the workspace, while the random agent keeps the hand closer to initialization. Also, while ExORL achieves performance comparable to or exceeding online RL in Cartpole Swingup and Jaco Reach, the performance on Cheetah Run is still not competitive with the optimal policy. This is likely due to insufficient exploration of the higher-dimensional state space in regions of high reward, suggesting there is still room to improve unsupervised exploration algorithms.

4.3 Multi-task learning with ExORL

Q2: Is it possible to relabel this unsupervised data to facilitate multi-task offline RL? **A2:** *Yes, multi-task offline RL can perform well from* ExORL *data*.

To confirm this we evaluate on one environment (Walker) across three different tasks (Stand, Walk, and Run). This means that each data collection algorithm only collects one dataset and we relabel it with three different rewards. Results are in Figure 4.

In particular, we are able to nearly solve both the Stand and Walk tasks from a single dataset collected with no knowledge of the rewards. Performance on Run is further away from optimal, but still strong enough to suggest meaningful progress towards the desired behavior. So, while there is room for improvement of the unsupervised exploration, the ExORL framework has clear promise to learn multiple tasks from a single task-agnostic dataset.

Again, we observe consistently strong performance of vanilla TD3 on exploratory data, confirming the result from the prior experiment. And comparing the different exploration algorithms, we see similar performance across each of the three representative algorithms (ICM, Proto, and APS) with each of them being much stronger than random.

4.4 Comparing ExORL to supervised data collection

Q3: Is exploratory data necessary for multi-task offline RL? **A3:** *Yes, the unsupervised exploratory data facilitates transfer where the supervised data does not*.

To demonstrate this, we consider different data-collection strategies ranging from unsupervised to supervised. *Unsupervised* corresponds to our standard ExORL approach with the Proto exploration algorithm. *Supervised* is similar to the approach in D4RL (Fu et al., 2020) and RL Unplugged (Gulcehre et al., 2020) of using the replay buffer of a policy trained for a task-specific reward. We use TD3 as the data collection agent and refer to the reward during training as the "data task". We then evaluate three offline RL agents (BC, TD3+BC, and TD3) on both the data task and a different

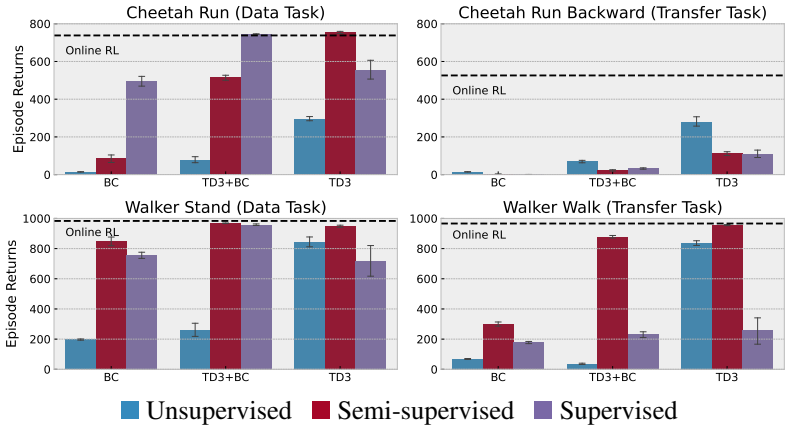


Figure 5: A comparison of three different data-collection strategies: unsupervised (intrinsic reward only), semi-supervised (intrinsic reward plus data task reward), and supervised (data task reward only). Data diversity is key to enable more aggressive trajectory stitching and task transfer.

"transfer task" that was not seen during training. Results in the Walker environment with Stand as the data task and Walk, Run, and Flip as the transfer tasks, see Figure 1b.

For a second experiment we also add a *semi-supervised* strategy as a baseline. This strategy uses a TD3 agent to maximize the sum of the extrinsic reward from the data task as well as the task-agnostic intrinsic reward from Proto. We run this procedure on Cheetah and Walker using Run and Stand respectively as the data tasks and Run Backward and Walk as the transfer tasks, see results in Figure 5.

As we hypothesized, unsupervised data allows for strong performance on both data and transfers tasks while supervised data only facilitates strong performance on the data task while failing to transfer. Semi-supervised data was capable of facilitating transfer on the easier Stand-Walk pair, but failed more clearly on the Run-Run Backward task. This is likely because Stand is a pre-requisite to Walk, while successful Run and Run Backward behaviors cover different parts of the state space.

Examining the offline RL algorithms, we find that TD3+BC is the strongest on the supervised data, but struggles to transfer from unsupervised data. This makes sense because the TD3+BC algorithm was designed for the D4RL benchamrk that consists of supervised-style datasets. In contrast, vanilla TD3 performs stronger on the unsupervised data. Unsurprisingly, BC can work fairly well on supervised data, but fails on the unsupervised data and the transfer tasks.

Interestingly, vanilla TD3 outperforms TD3+BC on the semi-supervised data, suggesting that data diversity can facilitate standard TD learning without offline modifications. We will build on this insight in the next subsection.

4.5 Combining ExORL with supervised data

Q4: Is it useful to mix exploratory data with task-specific data? **A4:** *Yes, it is useful to mix even a small amount of unsupervised data to facilitate transfer.*

To support this, we consider mixing datasets rather than modifying the data collection algorithms. We consider the same pairs of data and transfer tasks as in the prior subsection, but this time we mix the supervised and unsupervised datasets themselves. To do this, we randomly sample trajectories from the supervised and unsupervised datasets with varying proportions of unsupervised data while keeping the size of the resulting mixed dataset constant at 1M transitions (1000 episodes). We can then see the performance of BC, TD3+BC, and TD3 on datasets with varying proportions of unsupervised data in Figure 6.

Specifically, on the transfer tasks, TD3 achieves nearly the same performance with 25% unsupervised data as with 100%. Similarly, on the data task adding a relatively small amount of unsupervised data allows TD3 to match the performance of TD3+BC on the fully supervised data. This sends a very

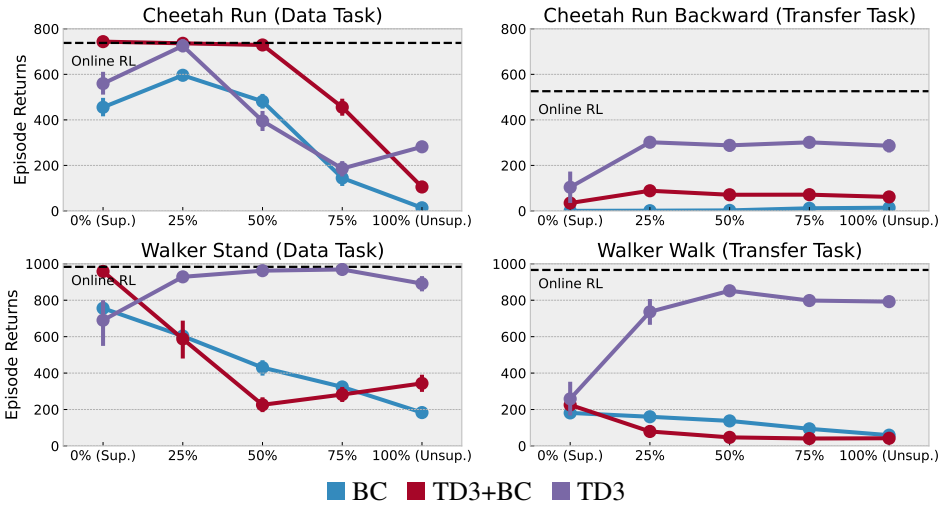


Figure 6: The effect of mixing supervised data (collected by TD3 on the data tasks) with unsupervised data (collected reward-free by Proto). Unsupervised data allows easier transfer and more efficient generalization to unseen tasks.

hopeful message that even adding a small amount of unsupervised exploratory data can both facilitate transfer to novel tasks and prevention of extrapolation error in the offline setting.

Comparing the offline RL algorithms more directly, we see that TD3 effectively takes advantage of the mixed datasets while TD3+BC and BC do not. By attempting to constrain to be near the behavior policy, algorithms like TD3+BC and BC can struggle to deal with data that is generated by a mixture of different behaviors, especially when these behaviors are not optimizing the test-time task.

We provide one extra experiment in Appendix A that tests how ExORL scales with an increased interaction budget.

5 Discussion

In this paper we propose ExORL – a novel framework for exploratory data collection for offline RL. Through an extensive empirical evaluation we conclude that using diverse data can greatly simplify offline RL algorithms by removing the need to fight the extrapolation problem. We then demonstrate that exploratory data is more suitable for efficient multi-task offline RL than standard task-directed offline RL data. Finally, we release ExORL's datasets as well as the experimentation code to encourage future research in this area. Now we will discuss a few of the possible directions for future research with ExORL.

Designing adaptive offline RL algorithms. Our dataset release provides a new testbed for improving offline RL algorithms. Our results suggest that previously designed offline RL algorithms perform well on supervised data, but are beaten by TD3 on unsupervised ExORL data. Ideally, an offline RL algorithm would be able to automatically adapt to the given dataset to recover the best of both worlds. There is a growing literature on offline hyperparameter tuning for offline RL that may be useful for approaching this challenge (Paine et al., 2020; Kumar et al., 2021; Zhang & Jiang, 2021; Fu et al., 2021; Lee et al., 2021).

Improving unsupervised data collection. As a side benefit, our approach yields a novel evaluation protocol for unsupervised RL by evaluating the generated dataset directly via offline RL. This differs from prior work on unsupervised RL Laskin et al. (2021) by emphasizing the dataset rather than the pre-trained policy. This allows for comparisons across different unsupervised RL algorithms without having to fine-tune online. In future work we hope to explore how to optimize the unsupervised exploration algorithms to improve performance of the downstream offline RL. ExORL provides a useful starting point and framework for this problem.

REFERENCES

Andrychowicz, M., Crow, D., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, P., and Zaremba, W. Hindsight experience replay. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5048–5058, 2017.

Brandfonbrener, D., Whitney, W. F., Ranganath, R., and Bruna, J. Offline rl without off-policy evaluation. *arXiv preprint arXiv:2106.08909*, 2021.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Burda, Y., Edwards, H., Storkey, A. J., and Klimov, O. Exploration by random network distillation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Cabi, S., Colmenarejo, S. G., Novikov, A., Konyushkova, K., Reed, S., Jeong, R., Zolna, K., Aytar, Y., Budden, D., Vecerik, M., et al. Scaling data-driven robotics with reward sketching and batch reinforcement learning. *arXiv preprint arXiv:1909.12200*, 2019.

Chen, A. S., Nam, H., Nair, S., and Finn, C. Batch exploration with examples for scalable robotic reinforcement learning. *IEEE Robotics and Automation Letters*, 6(3):4401–4408, 2021a.

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. *CoRR*, abs/2106.01345, 2021b. URL `https://arxiv.org/abs/2106.01345`.

Consortium, T. U. UniProt: the universal protein knowledgebase in 2021. *Nucleic Acids Research*, 49(D1):D480–D489, 11 2020. ISSN 0305-1048. doi: 10.1093/nar/gkaa1100. URL `https://doi.org/10.1093/nar/gkaa1100`.

Dendorfer, P., Osep, A., and Leal-Taixé, L. Goal-gan: Multimodal trajectory prediction based on goal position estimation. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Li, F. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pp. 248–255. IEEE Computer Society, 2009. doi: 10.1109/CVPR.2009.5206848.

Ecoffet, A., Huizinga, J., Lehman, J., Stanley, K. O., and Clune, J. Go-explore: a new approach for hard-exploration problems. *arXiv preprint arXiv:1901.10995*, 2019.

Endrawis, S., Leibovich, G., Jacob, G., Novik, G., and Tamar, A. Efficient self-supervised data collection for offline robot learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4650–4656. IEEE, 2021.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. Diversity is all you need: Learning skills without a reward function. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Eysenbach, B., Geng, X., Levine, S., and Salakhutdinov, R. R. Rewriting history with inverse RL: hindsight inference for policy improvement. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

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.

Fu, J., Norouzi, M., Nachum, O., Tucker, G., Wang, Z., Novikov, A., Yang, M., Zhang, M. R., Chen, Y., Kumar, A., et al. Benchmarks for deep off-policy evaluation. *arXiv preprint arXiv:2103.16596*, 2021.

Fujimoto, S. and Gu, S. S. A minimalist approach to offline reinforcement learning. *arXiv preprint arXiv:2106.06860*, 2021.

Fujimoto, S., van Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1582–1591. PMLR, 2018.

Fujimoto, S., Meger, D., and Precup, D. Off-policy deep reinforcement learning without exploration. In *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.

Gulcehre, C., Wang, Z., Novikov, A., Paine, T. L., Colmenarejo, S. G., Zolna, K., Agarwal, R., Merel, J., Mankowitz, D., Paduraru, C., et al. Rl unplugged: A suite of benchmarks for offline reinforcement learning. *arXiv preprint arXiv:2006.13888*, 2020.

Gulcehre, C., Colmenarejo, S. G., Wang, Z., Sygnowski, J., Paine, T., Zolna, K., Chen, Y., Hoffman, M., Pascanu, R., and de Freitas, N. Regularized behavior value estimation. *arXiv preprint arXiv:2103.09575*, 2021.

Jin, C., Krishnamurthy, A., Simchowitz, M., and Yu, T. Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*, pp. 4870–4879. PMLR, 2020.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Kalashnikov, D., Varley, J., Chebotar, Y., Swanson, B., Jonschkowski, R., Finn, C., Levine, S., and Hausman, K. Mt-opt: Continuous multi-task robotic reinforcement learning at scale. *arXiv preprint arXiv:2104.08212*, 2021.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pp. 1106–1114, 2012.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. *arXiv preprint arXiv:2006.04779*, 2020.

Kumar, A., Singh, A., Tian, S., Finn, C., and Levine, S. A workflow for offline model-free robotic reinforcement learning. *arXiv preprint arXiv:2109.10813*, 2021.

Lambert, N., Wulfmeier, M., Whitney, W., Byravan, A., Bloesch, M., Dasagi, V., Hertweck, T., and Riedmiller, M. The challenges of exploration for offline reinforcement learning. *arXiv preprint arXiv:2201.11861*, 2022.

Laskin, M., Yarats, D., Liu, H., Lee, K., Zhan, A., Lu, K., Cang, C., Pinto, L., and Abbeel, P. Urlb: Unsupervised reinforcement learning benchmark. *arXiv preprint arXiv:2110.15191*, 2021.

Lee, J. N., Tucker, G., Nachum, O., and Dai, B. Model selection in batch policy optimization. *arXiv preprint arXiv:2112.12320*, 2021.

Lee, L., Eysenbach, B., Parisotto, E., Xing, E., Levine, S., and Salakhutdinov, R. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*, 2019.

Levine, S. Understanding the world through action. *arXiv preprint arXiv:2110.12543*, 2021.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Li, A. C., Pinto, L., and Abbeel, P. Generalized hindsight for reinforcement learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Liu, H. and Abbeel, P. Aps: Active pretraining with successor features. In *International Conference on Machine Learning*, pp. 6736–6747. PMLR, 2021a.

Liu, H. and Abbeel, P. Behavior from the void: Unsupervised active pre-training. *arXiv preprint arXiv:2103.04551*, 2021b.

Mathieu, M., Ozair, S., Srinivasan, S., Gulcehre, C., Zhang, S., Jiang, R., Le Paine, T., Zolna, K., Powell, R., Schrittwieser, J., et al. Starcraft ii unplugged: Large scale offline reinforcement learning. In *Deep RL Workshop NeurIPS 2021*, 2021.

Mendonca, R., Rybkin, O., Daniilidis, K., Hafner, D., and Pathak, D. Discovering and achieving goals via world models. *Advances in Neural Information Processing Systems*, 34, 2021.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Paine, T. L., Paduraru, C., Michi, A., Gulcehre, C., Zolna, K., Novikov, A., Wang, Z., and de Freitas, N. Hyperparameter selection for offline reinforcement learning. *arXiv preprint arXiv:2007.09055*, 2020.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *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.

Pathak, D., Gandhi, D., and Gupta, A. Self-supervised exploration via disagreement. In *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. 5062–5071. PMLR, 2019.

Pomerleau, D. A. Alvinn: An autonomous land vehicle in a neural network. In *NIPS*, 1988.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, volume 139, pp. 8748–8763, 2021.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

Riedmiller, M., Springenberg, J. T., Hafner, R., and Heess, N. Collect & infer–a fresh look at data-efficient reinforcement learning. *arXiv preprint arXiv:2108.10273*, 2021.

Riedmiller, M. A., Hafner, R., Lampe, T., Neunert, M., Degrave, J., de Wiele, T. V., Mnih, V., Heess, N., and Springenberg, J. T. Learning by playing solving sparse reward tasks from scratch. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4341–4350. PMLR, 2018.

Sekar, R., Rybkin, O., Daniilidis, K., Abbeel, P., Hafner, D., and Pathak, D. Planning to explore via self-supervised world models. 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. 8583–8592. PMLR, 2020.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., de Las Casas, D., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A., Lillicrap, T., and Riedmiller, M. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Wang, R., Du, S. S., Yang, L. F., and Salakhutdinov, R. On reward-free reinforcement learning with linear function approximation. *arXiv preprint arXiv:2006.11274*, 2020a.

Wang, Z., Novikov, A., Zolna, K., Springenberg, J. T., Reed, S., Shahriari, B., Siegel, N., Merel, J., Gulcehre, C., Heess, N., et al. Critic regularized regression. *arXiv preprint arXiv:2006.15134*, 2020b.

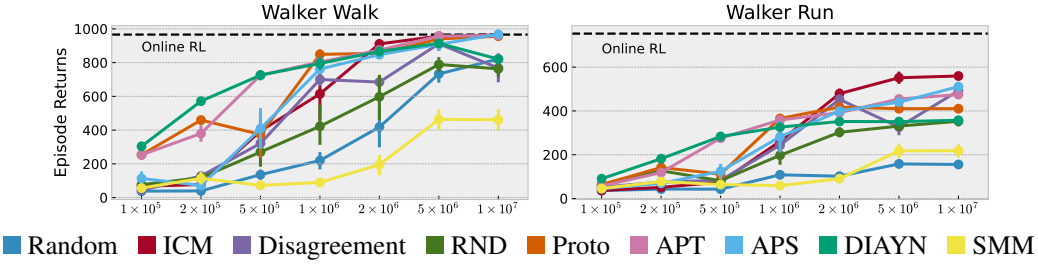
Yarats, D., Fergus, R., Lazaric, A., and Pinto, L. Reinforcement learning with prototypical representations. *arXiv preprint arXiv:2102.11271*, 2021.

Yu, T., Kumar, A., Chebotar, Y., Hausman, K., Levine, S., and Finn, C. Conservative data sharing for multi-task offline reinforcement learning. *Advances in Neural Information Processing Systems*, 34, 2021.

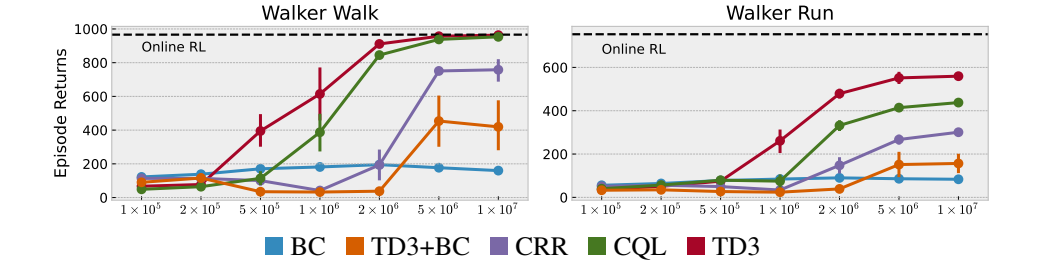
Zhang, S. and Jiang, N. Towards hyperparameter-free policy selection for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 34, 2021.

APPENDIX

A SCALING OF EXORL



(a) Correlation between TD3 performance and the unsupervised data collection transition budget.



(b) Correlation between performance of various offline RL algorithms and unsupervised data collection transition budget for ICM.

Figure 7: Testing the impact of the size of unsupervised data collection budget on downstream offline RL performance. Figure 7a provides a breakdown by datasets, while Figure 7b provides a breakdown by offline RL algorithms. Generally, larger datasets demonstrate better performance across both exploration and offline RL algorithms.

Q5: How effective is ExORL as we scale the data collection budget? **A1:** *In general, all algorithms see improved performance with increased data collection budget*.

To confirm, this we collect datasets in the Walker domain with data collection budgets ranging from 100k to 10M transitions (100 to 10k episodes) using all nine unsupervised data collection algorithms. We then run offline RL on these datasets for two different rewards: Walk and Run. To compare the different datasets we run offline TD3 on each (since TD3 was the strongest performer in prior experiments). Then to compare the offline RL algorithms, we run each of our five offline RL algorithms on the datasets generated by ICM (since ICM was the strongest performing dataset with TD3). Results are in Figure 7.

First, we evaluate the results across the different exploration algorithms. All algorithms see returns to scale, but we do observe gains beginning to saturate by 10M transitions on the Run task before reaching optimal performance suggesting there is still room for improved algorithms. In low-data settings we see the best performance from DIAYN while in the high-data regime ICM begins to dominate. We note that the random agent also sees return to scale, suggesting that the Walk task is not a difficult exploration problem.

We can also compare across the different offline RL algorithms on the ICM data. Again we see clear returns to scale as performance improves with dataset size. As in previous experiments we see that TD3 is the strongest algorithm and this holds true across dataset sizes.

B HYPER-PARAMETERS

In this section we comprehensively describe configuration of both unsupervised RL and offline RL algorithms.

B.1 UNSUPERVISED RL HYPER-PARAMETERS

We adhere closely to the parameter settings from URLB (Laskin et al., 2021). We list the common hyper-parameters in Table 1 and per-algorithm hyper-parameters in Table 2.

Table 1: Common hyper-parameters for unsupervised RL algorithms.

Common hyper-parameter	Value
Replay buffer capacity	10^6
Seed frames	4000
Mini-batch size	1024
Discount (γ)	0.99
Optimizer	Adam
Learning rate	10^{-4}
Agent update frequency	2
Critic target EMA rate (τ_Q)	0.01
Hidden dim.	1024
Exploration stddev clip	0.3
Exploration stddev value	0.2

Table 2: Individual hyper-parameters for unsupervised RL algorithms.

Random hyper-parameter	Value
Policy distribution	uniform
ICM hyper-parameter	**Value**
Reward transformation	$\log(r + 1.0)$
Forward net arch.	$(\lvert\mathcal{O}\rvert + \lvert\mathcal{A}\rvert) \rightarrow 1024 \rightarrow 1024 \rightarrow \lvert\mathcal{O}\rvert$ ReLU MLP
Inverse net arch.	$(2 \times \lvert\mathcal{O}\rvert) \rightarrow 1024 \rightarrow 1024 \rightarrow \lvert\mathcal{A}\rvert$ ReLU MLP
Disagreement hyper-parameter	**Value**
Ensemble size	5
Forward net arch:	$(\lvert\mathcal{O}\rvert + \lvert\mathcal{A}\rvert) \rightarrow 1024 \rightarrow 1024 \rightarrow \lvert\mathcal{O}\rvert$ ReLU MLP
RND hyper-parameter	**Value**
Predictor & target net arch.	$\lvert\mathcal{O}\rvert \rightarrow 1024 \rightarrow 1024 \rightarrow 512$ ReLU MLP
Normalized observation clipping	5
APT hyper-parameter	**Value**
Reward transformation	$\log(r + 1.0)$
Forward net arch.	$(512 + \lvert\mathcal{A}\rvert) \rightarrow 1024 \rightarrow 512$ ReLU MLP
Inverse net arch.	$(2 \times 512) \rightarrow 1024 \rightarrow \lvert\mathcal{A}\rvert$ ReLU MLP
k in NN	12
Avg top k in NN	True
Proto hyper-parameter	**Value**
Predictor dim.	128
Projector dim.	512
Number of prototypes	512
Softmax temperature	0.1
k in NN	3
Number of candidates per prototype	4
Encoder target EMA rate (τ_{enc})	0.05
SMM hyper-parameter	**Value**
Skill dim.	4
Skill discriminator learning rate	10^{-3}
VAE lr	10^{-2}
DIAYN hyper-parameter	**Value**
Skill dim	16
Skill sampling frequency (steps)	50
Discriminator net arch.	$512 \rightarrow 1024 \rightarrow 1024 \rightarrow 16$ ReLU MLP
APS hyper-parameter	**Value**
Reward transformation	$\log(r + 1.0)$
Successor feature dim.	10
Successor feature net arch.	$\lvert\mathcal{O}\rvert \rightarrow 1024 \rightarrow 1024 \rightarrow 10$ ReLU MLP
k in NN	12
Avg top k in NN	True
Least square batch size	4096

B.2 OFFLINE RL HYPER-PARAMETERS

In Table 3 we present a common set of hyper-parameters used in our experiments, while in Table 4 we list individual hyper-parameters for each method. To select hyper-parameters for offline RL algorithms we started with from the original hyper-parameters presented in the corresponding papers. We then performed a grid search over the most important parameters (i.e. learning rate, α, mini-batch size, etc.) on a supervised dataset collected by TD3 (Fujimoto et al., 2018) on Cheetah Run.

TD3+BC We swept over learning rate $[10^{-3}, 3 \cdot 10^{-4}, 10^{-4}]$, mini-batch size $[256, 512, 1024]$ and α $[0.01, 0.1, 1.0, 2.5, 5.0, 10.0, 50.0]$. We also found it useful to increase the dimension of hidden layers to 1024.

CRR We swept over the transformation function (indicator, exponent, identity), the number of samples to estimate value function $[5, 10, 20]$ as well as learning rate $[10^{-3}, 3 \cdot 10^{-4}, 10^{-4}]$ and mini-batch size $[256, 512, 1024]$.

CQL We swept over α $[0.01, 0.1, 0.5, 1.0, 2.0, 5.0, 10.0]$, the number of sampled actions $[3, 5, 10, 20]$, learning rate $[10^{-3}, 3 \cdot 10^{-4}, 10^{-4}]$ and mini-batch size $[256, 512, 1024]$.

Table 3: Common hyper-parameters for offline RL algorithms.

Common hyper-parameter	Value
Replay buffer capacity	10^6
Mini-batch size	1024
Discount (γ)	0.99
Optimizer	Adam
Learning rate	10^{-4}
Agent update frequency	2
Training steps	5×10^5

Table 4: Individual hyper-parameters for offline RL algorithms.

BC hyper-parameter	Value
Num hidden layers	2
Hidden dim.	1024
TD3+BC hyper-parameter	**Value**
α	2.5
Critic target EMA rate (τ_Q)	0.01
Num hidden layers	2
Hidden dim.	1024
CRR hyper-parameter	**Value**
Num samples to estimate V	10
Critic target EMA rate (τ_Q)	0.01
Transformation	indicator
Num hidden layers	2
Hidden dim.	1024
CQL hyper-parameter	**Value**
α	0.01
Critic target EMA rate (τ_Q)	0.01
Lagrange	False
Num sampled actions	3
Num hidden layers	2
Hidden dim.	1024
TD3 hyper-parameter	**Value**
Stddev clip	0.3
Critic target EMA rate (τ_Q)	0.01
Num hidden layers	2
Hidden dim.	1024

C FULL RESULTS ON THE POINTMASS EXPERIMENT

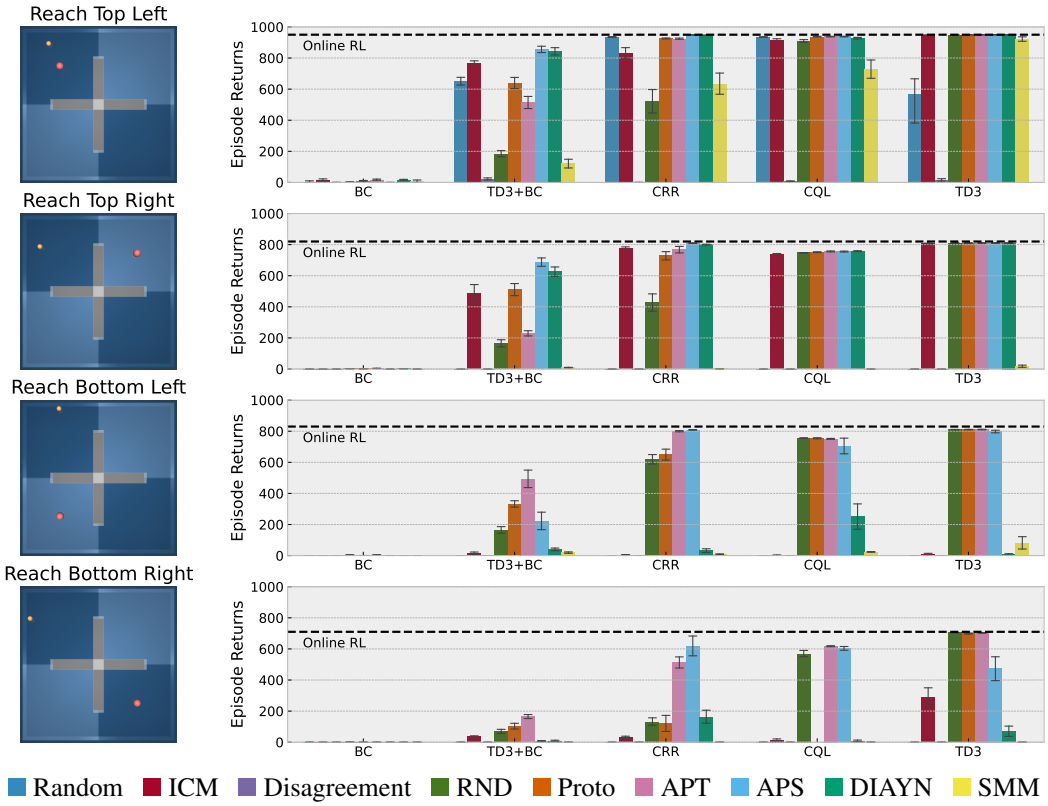


Figure 8: Extended figures from Figure 2.

D FULL RESULTS ON SINGLETASK SETTINGS

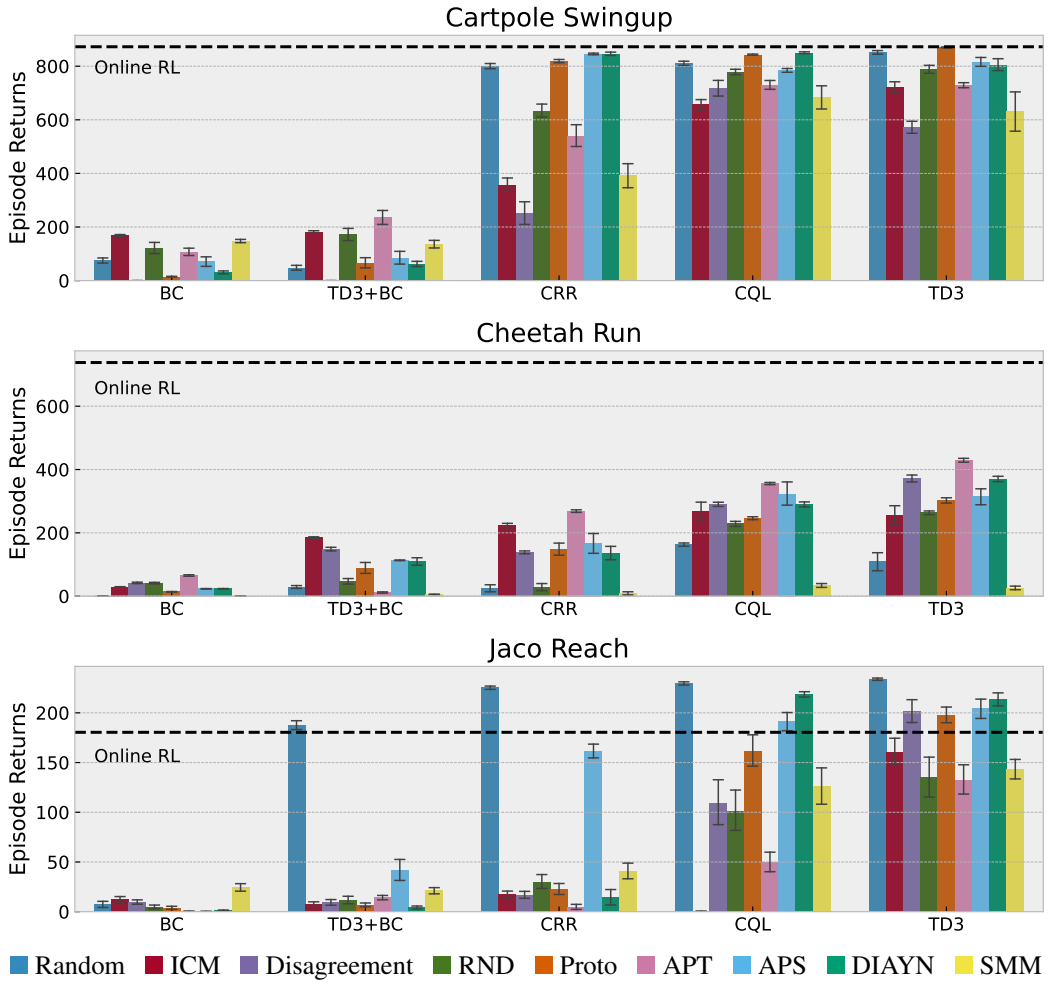


Figure 9: Offline evaluation of unsupervised datasets on one task for each of three different domains. Here we choose four representative unsupervised exploration algorithms.

E FULL RESULTS ON MULTITASK SETTINGS

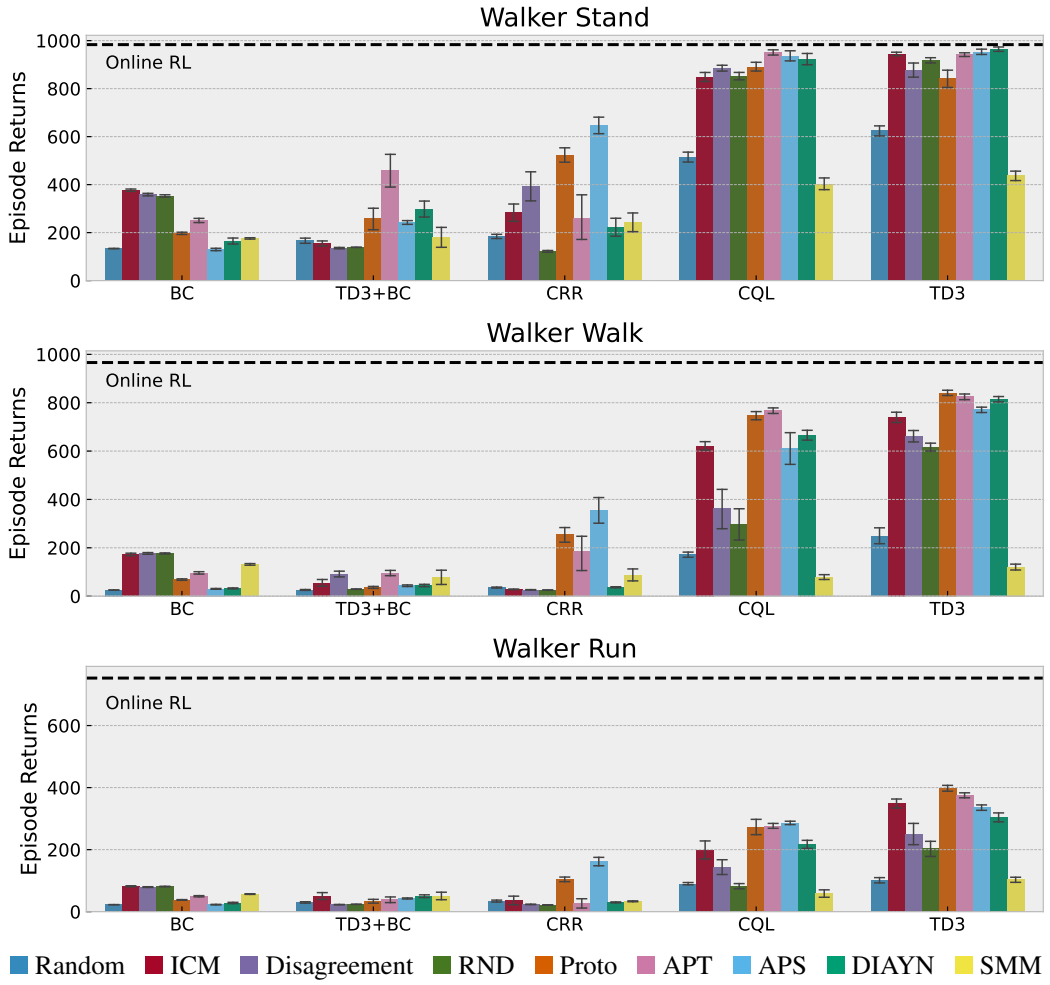


Figure 10: Offline evaluation of datasets for the Walker environment under three different rewards (Stand, Walk, and Run). We observe that ExORL allows for data relabeling to enable multi-task offline RL.

F OFFLINE RL ON SUFFIXES OF UNSUPERVISED DATASETS

In Figure 11 we demonstrate that it is important to use the entire replay buffers collected by ExORL, rather than only training on the later parts of this data.

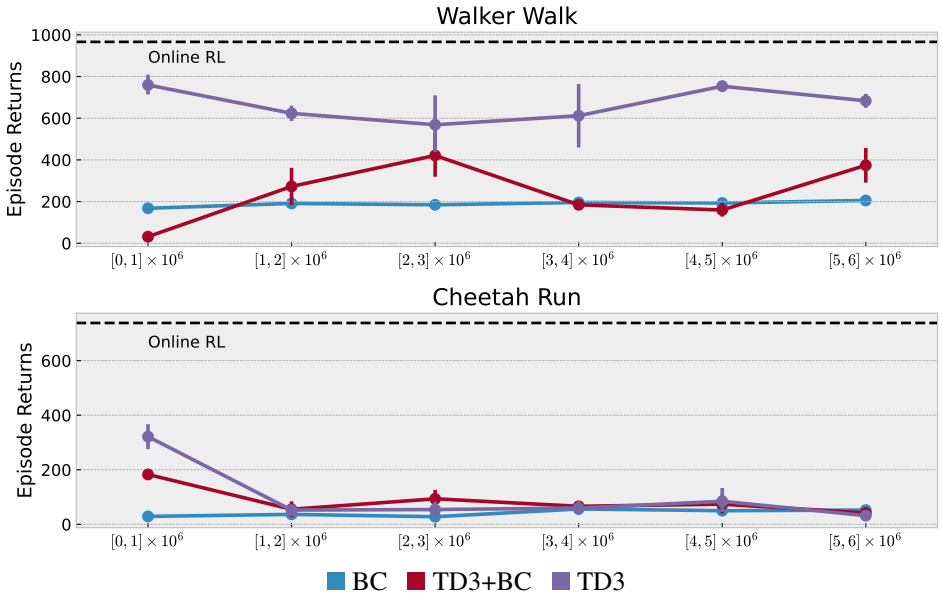


Figure 11: Instad of training on the full replay buffer collected by ICM, we perform offline RL training on various suffixes of the replay buffer. The x-axis represents segments of the replay buffuer. We observe that it is important to keep around early transitions.

G COMPUTE RESOURCES

ExORL is designed to be accessible to the RL research community. We provide an efficient implementation of our framework, including data collection, relabeling, and offline RL evaluation, that requires a single GPU. For local debugging experiments we used NVIDIA RTX GPUs. For large-scale runs used to generate all results in this manuscripts, we used NVIDIA Tesla V100 GPU instances. All experiments were run on internal clusters. Each offline RL algorithm trains in roughly 2 hours for 500k gradient steps on the collected datasets.

H THE EXORL ENVIRONMENTS AND TASKS

We provide a summary of used environments and tasks in our paper in Table 5.

Environment	Task	Traits	$\dim(\mathcal{S})$	$\dim(\mathcal{A})$
Walker	Stand	dense reward, easy exploration, locomotion	18	6
	Walk	dense reward, medium exploration, locomotion	18	6
	Run	dense reward, hard exploration locomotion	18	6
	Flip	dense reward, medium exploration, locomotion	18	6
Cheetah	Run	dense reward, hard exploration, locomotion	24	6
	Run Backward	dense reward, hard exploration, hard transfer	24	6
PointMass Maze	Reach Top Left	sparse reward, hard exploration, hard transfer	4	2
	Reach Top Right	sparse reward, hard exploration, hard transfer	4	2
	Reach Bottom Left	sparse reward, hard exploration, hard transfer	4	2
	Reach Bottom Right	sparse reward, hard exploration, hard transfer	4	2
Jaco	Reach	sparse reward, hard exploration, manipulation	55	6
Cartpole	Swingup	dense reward, easy exploration	4	1

Table 5: A detailed description of used environments and tasks from the DeepMind control suite (Tassa et al., 2018).