Improving Offline-to-Online Reinforcement Learning with Q-Ensembles

Kai Zhao [*1] **Yi Ma** [*1] **Jinyi Liu** [1] **Jianye Hao** [1] **Yan Zheng** [1] **Zhaopeng Meng** [1]

Abstract

Offline reinforcement learning (RL) is a learning paradigm where an agent learns from a fixed dataset of experience. However, learning solely from a static dataset can limit the performance due to the lack of exploration. To overcome it, offline-to-online RL combines offline pre-training with online fine-tuning, which enables the agent to further refine its policy by interacting with the environment in real-time. Despite its benefits, existing offline-to-online RL methods suffer from performance degradation and slow improvement during the online phase. To tackle these challenges, we propose a novel framework called **E**nsemble-based **O**ffline-to-**O**nline (E2O) RL. By increasing the number of Q-networks, we seamlessly bridge offline pre-training and online fine-tuning without degrading performance. Moreover, to expedite online performance enhancement, we appropriately loosen the pessimism of Q-value estimation and incorporate ensemble-based exploration mechanisms into our framework. Experimental results demonstrate that E2O can substantially improve the training stability, learning efficiency, and final performance of existing offline RL methods during online fine-tuning on a range of locomotion and navigation tasks, significantly outperforming existing offline-to-online RL methods.

1. Introduction

Reinforcement learning (RL) has shown remarkable success in solving complex decision-making problems, from playing virtual games (Vinyals et al., 2019; Silver et al., 2017; 2018) to controlling tangible robots (Mnih et al., 2015; Tsividis et al., 2021; Schrittwieser et al., 2020). In RL, an agent learns to maximize the cumulative return from large amount of experience data obtained by interacting with an environment. However, in many real-world applications, collecting experience data can be expensive, time-consuming, or even dangerous. This challenge has motivated the development of offline RL, where an agent learns from a fixed dataset of experience collected prior to learning (Fujimoto et al., 2019; Wu et al., 2019).

Offline RL has several advantages over online RL, including the ability to reuse existing data, the potential for faster learning, and the possibility of learning from experiences that are too risky or costly to collect online (Silver et al., 2018). However, offline RL also poses significant challenges, such as the potential for overfiting to the training data and the difficulty of ensuring that the learned policy is safe and optimal in the real-world environment. To address these challenges, offline-to-online RL has emerged as an attractive research direction. This approach combines offline pre-training with online fine-tuning using RL, with the goal of learning from a fixed dataset of offline experience and then continuing to learn online in the real-world environment (Nair et al., 2020; Lee et al., 2022). Offline-to-online RL has the potential to address the limitations of offline RL, such as the sub-optimality of the learned policy. Furthermore, starting with an offline RL policy enables offline-to-online RL to achieve strong performance using fewer online environment samples, compared to collecting large amounts of training data by rolling out policies from scratch.

Prior researches have shown that directly initializing an agent with an offline RL method for online fine-tuning can impede efficient policy improvement due to pessimistic learning (Nair et al., 2020; Zhao et al., 2022). A naive solution to this problem is directly removing the pessimistic term during online training. However, this approach can lead to unstable learning or degraded performance in that the distributional shift between offline datasets and online interactions creates large initial temporal difference errors, causing the oblivion of information learned from offline RL (Lee et al., 2022; Mark et al., 2022). Existing offline-to-online RL methods have attempted to address these challenges through implicit policy constraints (Nair et al., 2020), filtering offline data used for online fine-tuning (Lee et al., 2022; Mao et al., 2022; Mark et al., 2022), adjusting policy constraint weights carefully (Zhao et al., 2022), or training more online policies (Zhang et al., 2023). Nevertheless,

[*]Equal contribution [1]College of Intelligence and Computing, Tianjin University, Tianjin, China. Correspondence to: Jianye Hao <jianye.hao@tju.edu.cn>.

Workshop on New Frontiers in Learning, Control, and Dynamical Systems at the International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA, 2023. Copyright 2023 by the author(s).

these methods still face performance degradation in some tasks and settings, and their performance improvement in the online phase is limited.

Taking inspiration from leveraging Q-ensembles in offline RL (An et al., 2021), we propose a novel approach to address the challenges of offline-to-online RL. Specifically, we conduct comprehensive experiments by discarding the pessimistic term in existing offline RL algorithms and increasing the number of Q-networks in both offline and online phases. We find that Q-ensembles help to alleviate unstable training and performance degradation, and can serve as a more flexible pessimistic term by encompassing various target computation and exploration methods during the online fine-tuning phase. Based on this discovery, we propose an **E**nsemble-based framework for **O**ffline-to-**O**nline RL (E2O) that bridges offline pre-training and online fine-tuning. We demonstrate the effectiveness of E2O framework by instantiating it on existing offline RL algorithms (Kumar et al., 2020; Chen et al., 2022) across diverse benchmark tasks. The main contributions of this work are summarized as follows:

- We demonstrate the effectiveness of Q-ensembles in bridging the gap between offline pre-training and online fine-tuning, providing a solution for mitigating the common problem of unstable training and performance drop.
- We propose a unified framework E2O for offline-to-online RL, which enables a wide range of offline RL algorithms to transition from pessimistic offline pre-training to optimistic online fine-tuning, leading to stable and efficient performance improvement.
- We empirically validate the effectiveness of E2O on various benchmark tasks, including locomotion and navigation tasks, and verify that E2O achieves state-of-the-art performance in comparison to all baseline methods.

2. Related Works

Offline RL Offline RL algorithms focus on training RL agents with pre-collected datasets. However, these algorithms face the challenge of distribution shift between the behavior policy and the policy being learned, which can cause issues due to out-of-distribution (OOD) actions sampled from the learned policy and passed into the learned critic. To mitigate this problem, prior methods constrain the learned policy to stay close to the behavior policy via explicit policy regularization (Fujimoto et al., 2019; Kumar et al., 2019; Wu et al., 2019), via implicit policy constraints (Peng et al., 2019; Siegel et al., 2020; Kostrikov et al., 2021; Zhou et al., 2021; Chen et al., 2022), by leveraging auxiliary behavioral cloning losses (Fujimoto & Gu, 2021), by penalizing the Q-value of OOD actions to prevent selecting them (Kumar et al., 2020; Lyu et al., 2022; An et al., 2021; Bai et al., 2022; Yang et al., 2022), or through model-based training with conservative penalties (Yu et al., 2020; Kidambi et al., 2020; Argenson & Dulac-Arnold, 2020; Yu et al., 2021). Among the above methods, we choose several representative algorithms such as CQL (Kumar et al., 2020) and LAPO (Chen et al., 2022) to be the base methods in the offline component of our framework due to their wide applicability and superior performance.

Offline-to-Online RL Offline-to-online RL refers to the process of improving the well-trained offline policy by incorporating online interactions. Directly applying pre-trained offline policy to the online fine-tuning stage may lead to poor performance due to excess conservatism (Nair et al., 2020; Lee et al., 2022; Zhao et al., 2022). To adapt offline RL algorithms to the online environment, several modifications are required. AWAC (Nair et al., 2020) is the first algorithm proposed to perform well in the offline-to-online RL setting, which forces the policy to imitate actions with high advantage estimates. AW-Opt (Lu et al., 2021) improves upon AWAC by incorporating positive sample filtering and hybrid actor-critic exploration during the online stage. Balanced Replay (Lee et al., 2022) trains an additional neural network to prioritize samples in order to effectively use new data as well as near-on-policy samples from the offline dataset. PEX (Zhang et al., 2023) proposes a policy expansion approach for offline-to-online RL, which trains more policies from scratch in the online phase and combine them with the pre-trained offline policy to make decisions jointly. A concurrent work with us is O3F (Mark et al., 2022), which also aims to eliminate pessimism in online learning. However, our research takes a more holistic perspective by connecting the pessimistic offline learning and optimistic online phases through ensemble modeling, proposing a more applicable framework that encapsulates the implementation of O3F as a special case.

Q-Ensembles in RL Q-Ensemble methods have been widely utilized to enhance the performance of RL (Van Hasselt et al., 2016; Osband et al., 2016; Anschel et al., 2017; Fujimoto et al., 2018; Ciosek et al., 2019; Lan et al., 2020; Chen et al., 2021; Lee et al., 2021). TD3 (Fujimoto et al., 2018) leverages an ensemble of two value functions and uses their minimum for computing the target value during Bellman error minimization. REDQ (Chen et al., 2021) minimizes over a random subset of Q-functions in the target to reduce over-estimation bias. For offline RL, a number of works have extended this to propose backing up minimums or lower confidence bound estimates over larger ensembles (Fujimoto et al., 2019; Kumar et al., 2019; Wu et al., 2019; Agarwal et al., 2020; An et al., 2021; Bai et al., 2022; Yang et al., 2022). In particular, EDAC (An et al., 2021) achieves

impressive performance by simply increasing the number of Q-networks along with the clipped Q-learning and further proposes to reduce the required number of ensemble networks through ensemble gradient diversification for the purpose of reducing computational cost. PBRL (Bai et al., 2022) and RORL (Yang et al., 2022) both employ an ensemble of bootstrapped Q-functions for uncertainty quantification and perform pessimistic updates to penalize Q functions with high uncertainties. Recent work (Ghasemipour et al., 2022) advocates for using independently learned ensembles without sharing target values and optimizing a policy based on the lower confidence bound of predicted action values. In our work, we discover that Q-ensembles are highly effective in addressing the performance degradation that occurs in the offline-to-online setting, and we advocate for using Q-ensembles for both offline and online algorithms to achieve steady and sample-efficient offline-to-online RL.

Exploration Mechanisms Various exploration approaches have been proposed to accelerate the efficiency of online training in recent years. These methods can typically be divided into two main categories following (Hao et al., 2023): uncertainty-oriented exploration and intrinsic motivation-oriented exploration. The former employs heuristic design to formulate various intrinsic motivations for exploration, based on factors such as visitation count (Bellemare et al., 2016; Ostrovski et al., 2017; Tang et al., 2017), curiosity (Savinov et al., 2018), information gain (Houthooft et al., 2016), etc. Uncertainty-oriented exploration methods adopt the principle of optimism in the face of uncertainty and use Q-ensembles to encourage agents to explore areas with higher uncertainty (Badia et al., 2020; Ecoffet et al., 2021; Lee et al., 2021). In this paper, we investigate uncertainty-based exploration mechanisms under the unified framework of Q-ensembles to enhance the performance during the online fine-tuning phase.

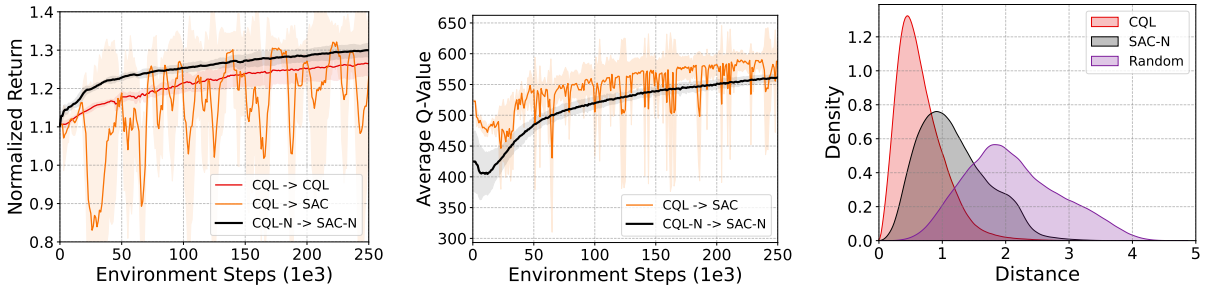
3. Why Can Q-Ensembles Help Offline-to-Online RL?

To get a better understanding of our ensemble-based framework, we begin with examples that highlight the advantages of Q-ensembles for offline-to-online RL. A natural starting point for offline-to-online RL is to simply initialize the agent with the one trained by an existing offline RL method and then directly perform online fine-tuning without using the offline dataset. However, this approach can hinder efficient online performance improvement due to the inherent pessimism of the offline learning paradigm (Lee et al., 2022; Mark et al., 2022). To support this claim, we present CQL (Kumar et al., 2020) as a representative and conduct preliminary experiments on the D4RL Walker2d-medium-expert-v2 dataset. The learning curve of CQL during online fine-tuning in Figure 1(a) shows that CQL can maintain the offline performance at the initial stage of online fine-tuning and steadily improve during the training process. This can be attributed to the use of pessimistic Q-functions, which enables the agent to visit states resembling those in the offline dataset and maintain pessimistic towards unseen actions during the initial stage of online fine-tuning. However, the pessimistic objective impedes proper exploration in the online stage and restrict the agent from efficiently improving its performance (Lee et al., 2022; Mark et al., 2022).

To tackle the aforementioned issue of limited exploration, one might be tempted to remove the conservative estimation component in order to reduce the conservatism of the learning process. However, as shown in Figure 1(a), this naive solution leads to unstable training or performance degradation when switching from CQL to Soft Actor-Critic (SAC) (Haarnoja et al., 2018) during online fine-tuning, which has also been reported in previous offline-to-online RL works (Lu et al., 2021; Nair et al., 2020; Lee et al., 2022; Mark et al., 2022). The reason is that SAC lacks accurate estimation of Q-values for unknown state-action pairs. Without the conservative constraints of CQL, the Q-values tend to be overestimated, leading to policy misguidance.

So is it possible to find a method that retains suitable pessimistic constraints to mitigate performance degradation, while also tailoring these constraints to be more conducive to exploration during the online phase, rather than being as conservative as traditional offline RL algorithms such as CQL? Inspired by increasing the number of Q-networks in (An et al., 2021), we introduce Q-ensembles and set the number of Q functions in CQL and SAC to N. Specifically, the target Q value is estimated by selecting the minimum value from all the Q-ensembles. We refer to these intermediate methods as CQL-N and SAC-N. Figure 1(a) shows the effectiveness of using SAC-N for online fine-tuning of an offline policy pre-trained with CQL-N. Surprisingly, after incorporating Q-ensembles, we observe that the training becomes more stable and performance drop is no longer observed when switching to online fine-tuning. Moreover, this constraint method not only enhances the final performance of the offline stage, but also improves the efficiency of online learning.

To comprehend the reason behind how Q-ensembles help alleviate unstable training and performance drop, we examine the averaged Q-values over the dataset of different algorithms in Figure 1(b). We observe that if we directly remove the pessimistic constraints during the online fine-tuning stage (i.e. CQL→SAC), the estimation of the Q-value will fluctuate violently, resulting in unstable training and performance drop, as depicted in Figure 1(a). However, with our integration of Q-ensembles, SAC-N still has the ability to conservatively estimate, and the variation range of Q-value in CQL-N→SAC-N is much smaller than that of



(a) Normalized Return (b) Average Q-value (c) Action Distance

Figure 1. (a) Normalized return curves of some motivated examples while performing online fine-tuning with offline policy trained on Walker2d-medium-expert-v2 dataset. (b) Comparison of the average Q-values of SAC and SAC-N. (c) Histograms of the distances between the actions from each method (CQL, SAC-N, and a random policy) and the actions from the dataset.

CQL→SAC. This phenomenon indicates that appropriately retaining the conservative capabilities is crucial in avoiding unstable training and performance drop.

We have seen that both SAC-N and CQL can prevent performance drop during online fine-tuning, but why does SAC-N exhibit better performance compared to CQL? To answer this question, we analyze the distance between the actions selected by each method and the actions in the dataset, as shown in Figure 1(c). Specifically, we measure $\mathbb{E}_{(\mathbf{s},\mathbf{a})\sim\mathcal{D},\hat{\mathbf{a}}\sim\pi_\theta(\cdot|\mathbf{s})}\left[\|\hat{\mathbf{a}}-\mathbf{a}\|_2^2\right]$ for SAC-N, CQL and a random policy by performing online fine-tuning on the Walker2d-medium-expert-v2 dataset. Our findings reveal that SAC-N has a wider range of action choices compared to CQL, and a more diverse set of actions can lead to improved performance, as stated in previous exploration methods (Badia et al., 2020; Ecoffet et al., 2021; Lee et al., 2021). Therefore, we can incorporate Q-ensembles into existing offline RL algorithms like CQL, and discard the original conservative term designed for offline algorithms during the online phase to improve the online learning efficiency.

To summarize, our primary empirical analysis indicates the following observation:

> **Q**-ensembles can maintain certain conservative capabilities to mitigate unstable training and performance drop, functioning as a more versatile constraint method for exploring more diverse actions during online fine-tuning compared to offline RL algorithms such as CQL.

With Q-ensembles in hand, we can further improve online learning efficiency by flexibly leveraging various approaches based on this mechanism, which will be presented in our proposed framework in the following section.

4. Ensemble-based Offline-to-Online Reinforcement Learning

Based on the empirical observations discussed earlier, we propose our **E**nsemble-based **O**ffline-to-**O**nline (E2O) RL Framework. In this section, we first present merits of Q-ensemble using additional empirical results and then progressively introduce more ensemble-based mechanisms into our framework. Although each individual design decision in E2O may seem relatively simple, their specific combination outperforms baselines in terms of training stability, learning efficiency and final performance.

4.1. Q-Ensembles

As discussed in Section 3, Q-ensembles can bridge offline and online phases to help pre-trained offline agents perform stable online fine-tuning. In this section, we present comprehensive empirical results to further verify its advantages.

Given an offline RL algorithm named *OfflineRL*, we introduce Q-ensembles to get *OfflineRL-N*, indicating that the algorithm uses N Q-networks and takes the minimum value of all the Q-networks in the ensemble to compute the target. With the pre-trained *OfflineRL-N* agent, we load it as the initialization of the online agent and remove the originally designed pessimistic term (if possible) to obtain *OnlineRL-N*. Then *OnlineRL-N* is trained online. In all methodology sections, we instantiate *OfflineRL* as CQL, and thus *OfflineRL-N* refers to CQL-N, and *OnlineRL-N* refers to SAC-N, as described in Section 3. To comprehensively verify the effectiveness of Q-ensembles in stabilizing training process and mitigating performance drop, we consider three MuJoCo locomotion tasks (Todorov et al., 2012): HalfCheetah, Hopper, and Walker2d from the D4RL benchmark suite (Fu et al., 2020). Specifically, we consider the medium, medium-replay and medium-expert datasets, as in typical real-world scenarios, we rarely use a random policy or have an expert policy for system control.

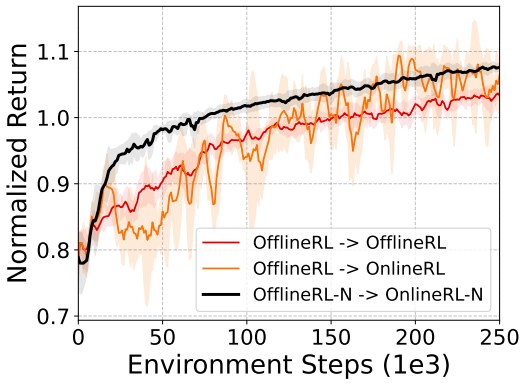


Figure 2. Aggregated learning curves of different offline-to-online RL approaches on all considered MuJoCo datasets.

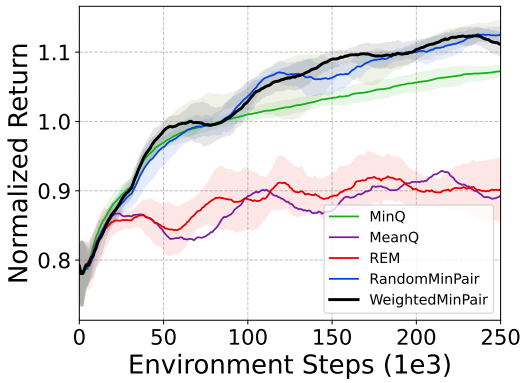


Figure 3. Aggregated learning curves of *OnlineRL-N* using different Q-target computation methods on all considered MuJoCo datasets.

Figure 2 shows the aggregated normalized return across all nine datasets. Consistent with the results of the previous illustrative experiment, online training of *OfflineRL* is stable but leads to slower asymptotic performance. Directly switching to *OnlineRL* causes unstable training process and performance drop. In contrast, *OfflineRL-N* $\rightarrow$ *OnlineRL-N* no longer experiences performance collapse after switching to online fine-tuning, and the training process is relatively stable. Additionally, *OfflineRL-N* $\rightarrow$ *OnlineRL-N* achieves better fine-tuned performance than *OfflineRL* $\rightarrow$ *OfflineRL*.

Although the ensemble-based method *OfflineRL-N* $\rightarrow$ *OnlineRL-N* has made certain improvements compared to existing method *OfflineRL* $\rightarrow$ *OfflineRL*, it still fails to be improved rapidly in the online stage compared with standard online RL algorithms. Therefore, we shift our focus to analyzing whether we can appropriately loosen the pessimistic estimation of Q-values in the online phase to further improve learning efficiency while ensuring stable training.

4.2. Loosing Pessimism

In the previous section, we employ *OnlineRL-N* as our primary method for the online phase. This method selects the minimum value of N parallel Q-networks as the Bellman target to enforce their Q-value estimates to be conservative. While *OfflineRL-N* $\rightarrow$ *OnlineRL-N* has achieved satisfactory performance, selecting the minimum of N Q-networks in the ensemble to compute the Q-target is still too conservative for online training, compared with standard online RL algorithms without pessimistic constraint. Consequently, while ensuring that the online training process is stable, we consider to appropriately loosen the pessimistic estimation of Q-values by modifying the Q-target computation method in *OnlineRL-N* to efficiently improve online performance.

Specifically, we compare several Q-target computation methods. **(a) MinQ** is what we use in *OnlineRL-N*, where the minimum value of all the Q-networks in the ensemble is taken to compute the target. **(b) MeanQ** leverages the average of all the Q-values to compute the target. **(c) REM** is a method originally proposed to boost performance of DQN in the discrete-action setting, which uses the random convex combination of Q-values to compute the target (Agarwal et al., 2020). It is similar to ensemble average (MeanQ), but with more randomization. **(d) RandomMinPair** uses a minimization over a random subset 2 of the N Q-functions, which is proposed in prior methods (Chen et al., 2021). **(e) WeightedMinPair** computes the target as the expectation of all the RandomMinPair targets, where the expectation is taken over all N-choose-2 pairs of Q-functions. RandomMinPair can be considered as a uniform-sampled version of WeightedMinPair.

Figure 3 presents the results of using different Q-target computation methods in the online phase based on *OnlineRL-N*. With MinQ, which is originally used in *OnlineRL-N*, as the bound, both MeanQ and REM exhibit poor performance, while RandomMinPair and WeightedMinPair outperform the other candidates with their efficient and stable online learning process. As the WeightedMinPair method is more stable on many datasets than the RandomMinPair method, we adopt the WeightedMinPair. Proceeding here, we refer to this intermediate algorithm as *OnlineRL-N + WeightedMinPair*. Despite the superior online fine-tuning performance of this approach, we continue to explore ways to further improve the online learning efficiency by leveraging the ensemble characteristics.

4.3. Optimistic Exploration

In the previous sections, we use pessimistic learning to obtain a satisfactory start point for online learning and gradually loosen the pessimistic constraint to improve online learning. In this section, we investigate the use of ensemble-

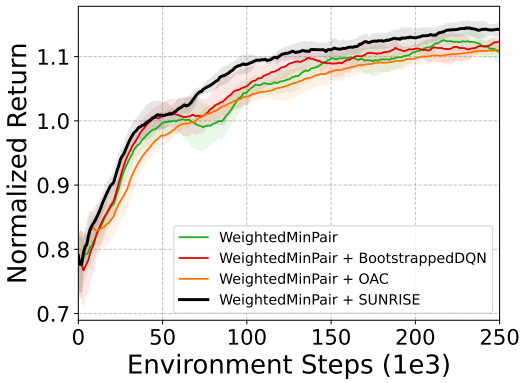


Figure 4. Aggregated learning curves of *OnlineRL-N + WeightedMinPair* using different exploration methods on all considered MuJoCo datasets.

based exploration methods to further improve performance and learning efficiency.

Specifically, we compare three ensemble-based exploration methods. **(a) Bootstrapped DQN** (Osband et al., 2016) uses ensembles to address some shortcomings of alternative posterior approximation schemes, whose network consists of a shared architecture with N bootstrapped "heads" branching off independently. **(b) OAC** (Ciosek et al., 2019) proposes an off-policy exploration strategy that adjusts to maximize an upper confidence bound to the critic, obtained from an epistemic uncertainty estimate on the Q-function computed with the bootstrap through Q-ensembles. **(c) SUNRISE** (Lee et al., 2021) presents ensemble-based weighted Bellman backups that improve the learning process by reweighting target Q-values based on uncertainty estimates.

The results of different exploration methods is presented in Figure 4. Among them, *OnlineRL-N + WeightedMinPair + SUNRISE* achieves the highest aggregated return. Consequently, we turn *OnlineRL-N + WeightedMinPair + SUNRISE* into our final ensemble-based framework E2O. Algorithm 1 summarizes the offline and online procedures of E2O. Note that as many offline RL algorithms can integrate ensemble technique in Q-functions, E2O can thus serve as a common plugin. We will further show the plug-and-play character of E2O by applying ***OfflineRL-N* → *OnlineRL-N + WeightedMinPair + SUNRISE*** on different offline RL algorithms in the experiments. For a comprehensive view of the detailed results of this section, appending the combination of RandomMinPair and different exploration methods, please refer to Appendix D.4.

5. Experiments

In this section, we present the empirical evaluations of our E2O framework. We begin with locomotion tasks from D4RL (Fu et al., 2020) to measure the training stability, learning efficiency and final performance of E2O by comparing it with several state-of-the-art offline-to-online RL methods. Additionally, we evaluate E2O on more challenging navigation tasks to verify its versatility.

5.1. Locomotion Tasks

We first evaluate our E2O framework on MuJoCo (Todorov et al., 2012) locomotion tasks, i.e., HalfCheetah, Walker2d, and Hopper from the D4RL benchmark suite (Fu et al., 2020). To demonstrate the applicability of E2O on various suboptimal datasets, we use three dataset types: medium, medium-replay, and medium-expert. Specifically, medium datasets contain samples collected by a medium-level policy, medium-replay datasets include all samples encountered while training a medium-level agent from scratch, and medium-expert datasets consist of samples collected by both medium-level and expert-level policies. We pre-train the agent for 1M training steps in the offline phase and perform online fine-tuning for 250K environmental steps. Additional experimental details can be found in Appendix C.

Comparative Evaluation We consider the following methods as baselines.

- **AWAC** (Nair et al., 2020): an offline-to-online RL method that forces the policy to imitate actions with high advantage estimates in the dataset.
- **Balanced Replay** (Lee et al., 2022): an offline-to-online RL method that trains an additional network to prioritize samples in order to effectively use new data as well as near-on-policy samples in the offline dataset.
- **PEX** (Zhang et al., 2023): a recent offline-to-online RL method utilizing an offline policy within a policy set, expanding it with additional policies, and constructing a categorical distribution based on their values at the current state to select the final action.
- **IQL** (Kostrikov et al., 2021): a representative RL algorithm demonstrating superior offline performance and enabling seamless online fine-tuning through direct parameter transfer.
- **SAC** (Haarnoja et al., 2018): a SAC agent trained from scratch. This baseline highlights the benefit of offline-to-online RL, as opposed to fully online RL, in terms of learning efficiency.
- **Scratch**: training SAC-N + WeightedMinPair + SUNRISE online from scratch without offline pre-training, as opposed to our E2O framework.

Figure 5 shows the performance of the E2O-CQL method (E2O instantiated on CQL) and baseline methods during

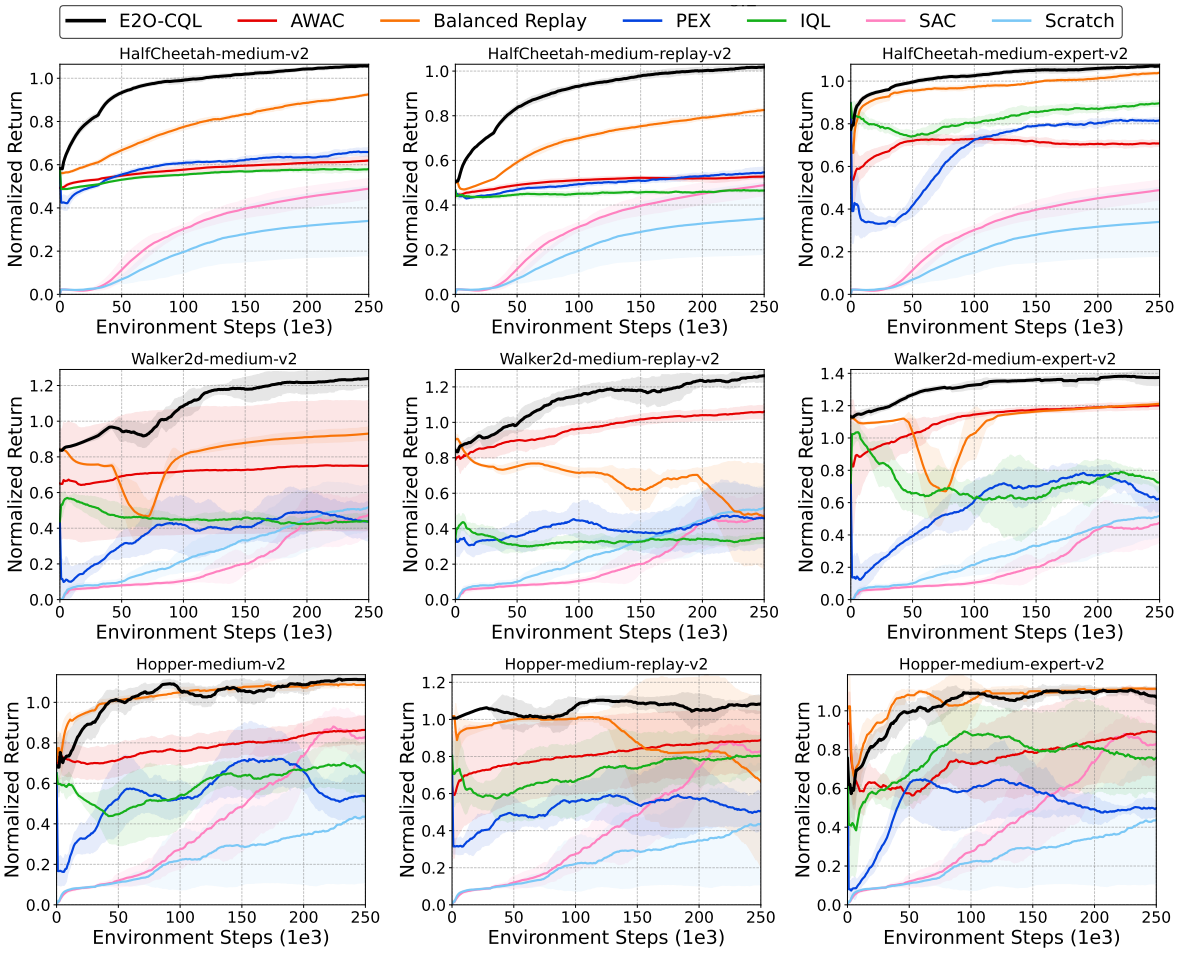


Figure 5. Online learning curves of different methods across five seeds on MuJoCo locomotion tasks. The solid lines and shaded regions represent mean and standard deviation, respectively.

the online fine-tuning phase. Compared with pure online RL methods such as SAC and Scratch, E2O-CQL starts with a well-performed policy and learns quickly and stably, proving the benefits of offline pre-training. For offline RL methods, IQL shows limited improvement as complete pessimistic training is no longer suitable for online fine-tuning, while E2O-CQL displays fast fine-tuning. Among other offline-to-online RL methods, the performance of AWAC is limited by the quality of the dataset due to the operation of training its policy to imitate actions with high advantage estimates, resulting in slow improvement during the online phase. While Balanced Replay can attain performance second only to E2O-CQL on some datasets, it also suffers from unstable training. PEX experiences severe performance drop during the initial stage of online fine-tuning on several datasets as the new policies trained in the online phase are random at the beginning and have a negative impact on the training stability. Overall, our E2O-CQL method outperforms the baseline methods in terms of training stability, learning efficiency and final performance on most tasks.

5.2. Navigation Tasks

We further verify the effectiveness of E2O on D4RL navigation task Antmaze (Fu et al., 2020) by integrating another offline RL algorithm LAPO (Chen et al., 2022). In detail, we specialize E2O as LAPO-N + WeightedMinPair + SUNRISE, i.e., E2O-LAPO. For the Antmaze task, we consider three types of mazes: umaze, medium and large mazes, and two data compositions: play and diverse. The data compositions vary in their action coverage of different regions of the state space and the sub-optimality of the behavior policy.

Comparative Evaluation Since Antmaze is a more challenging task, most offline RL methods struggle to achieve satisfactory results in the offline phase, we only compare our E2O-LAPO method on this task with two effective baseline methods, IQL and PEX.

Figure 6 presents the performance of E2O-LAPO and baseline methods during the online fine-tuning phase. First, LAPO demonstrates better offline performance than IQL,

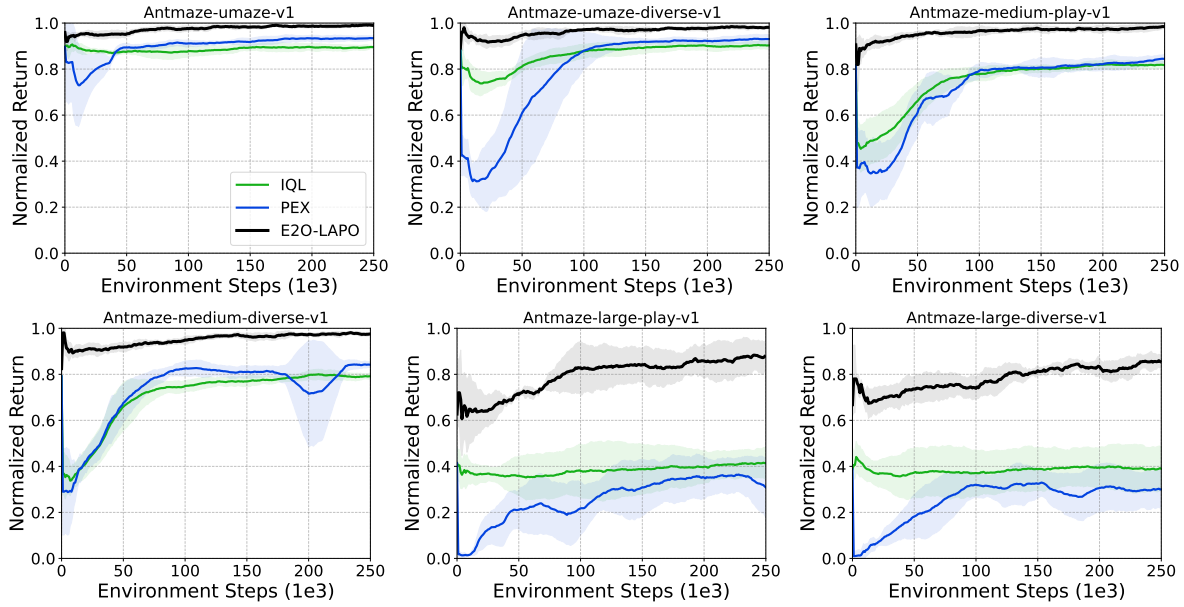


Figure 6. Online learning curves of different methods across five seeds on Antmaze navigation tasks. The solid lines and shaded regions represent mean and standard deviation, respectively.

providing a higher starting point for the online phase, especially in the umaze and medium maze environments where it almost reaches the performance ceiling. In the online stage, IQL shows slower asymptotic performance due to offline policy constraints. Building upon IQL, PEX enhances the degree of exploration by incorporating additional new policies trained from scratch, but the strong randomness of these policies in the early online stage causes performance drop. Note that although both IQL and PEX share the same starting point, PEX exhibits more severe performance drop on most tasks. By employing our proposed E2O framework, we demonstrate that E2O-LAPO can not only enhance the offline performance, but also facilitate stable and rapid performance improvement while maintaining the offline performance without degradation. This approach enables the offline agent to quickly adapt to the real-world environment, providing efficient and effective online fine-tuning. Additionally, we directly leverage LAPO with two Q networks for offline-to-online training and use the comparison with our E2O-LAPO method to further verify the effectiveness of our E2O framework. The results including some ablation studies can be found in Appendix D.

6. Conclusions and Limitations

In this work, we have demonstrated that Q-ensembles can be efficiently leveraged to alleviate unstable training and performance drop, and serve as a more flexible constraint method for online fine-tuning in various settings. Based on this observation, we propose Ensemble-based Offline-to-Online (E2O) RL Framework, which enables many pessimistic offline RL algorithms to perform optimistic online fine-tuning and improve their performance efficiently while maintaining stable training process. The proposed framework is straightforward and can be combined with many existing offline RL algorithms. We instantiate E2O with different combinations and conducted experiments on a wide range of tasks to demonstrate its effectiveness.

Despite the promising results, there are some limitations to our work that should be acknowledged. First, although E2O is designed to be a flexible plugin for various offline RL algorithms, it may require further modifications or fine-tuning to achieve optimal performance in different contexts. For instance, adjusting the weight coefficient of the BC item may result in better online fine-tuning performance for TD3+BC (Fujimoto & Gu, 2021). Second, the computational cost of using ensembles and uncertainty estimates may limit the scalability of E2O to large-scale problems. Future work could investigate ways to reduce the computational overhead by using deep ensembles (Fort et al., 2019) or ensemble distillation (Hinton et al., 2015) while maintaining the performance benefits of ensembles and uncertainty estimation by using Bayesian compression (Louizos et al., 2017) or variational approximations (Kingma & Welling, 2013). These methods could make E2O more scalable and practical for large-scale problems and real-world applications, enabling the development of more efficient and reliable offline-to-online RL systems.

References

Agarwal, R., Schuurmans, D., and Norouzi, M. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, pp. 104–114. PMLR, 2020.

An, G., Moon, S., Kim, J.-H., and Song, H. O. Uncertainty-based offline reinforcement learning with diversified q-ensemble. *Advances in neural information processing systems*, 34:7436–7447, 2021.

Anschel, O., Baram, N., and Shimkin, N. Averaged-dqn: Variance reduction and stabilization for deep reinforcement learning. In *International conference on machine learning*, pp. 176–185. PMLR, 2017.

Argenson, A. and Dulac-Arnold, G. Model-based offline planning. *arXiv preprint arXiv:2008.05556*, 2020.

Badia, A. P., Sprechmann, P., Vitvitskyi, A., Guo, D., Piot, B., Kapturowski, S., Tieleman, O., Arjovsky, M., Pritzel, A., Bolt, A., et al. Never give up: Learning directed exploration strategies. *arXiv preprint arXiv:2002.06038*, 2020.

Bai, C., Wang, L., Yang, Z., Deng, Z., Garg, A., Liu, P., and Wang, Z. Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning. *arXiv preprint arXiv:2202.11566*, 2022.

Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

Chen, X., Wang, C., Zhou, Z., and Ross, K. Randomized ensembled double q-learning: Learning fast without a model. *arXiv preprint arXiv:2101.05982*, 2021.

Chen, X., Ghadirzadeh, A., Yu, T., Gao, Y., Wang, J., Li, W., Liang, B., Finn, C., and Zhang, C. Latent-variable advantage-weighted policy optimization for offline rl. *arXiv preprint arXiv:2203.08949*, 2022.

Ciosek, K., Vuong, Q., Loftin, R., and Hofmann, K. Better exploration with optimistic actor critic. *Advances in Neural Information Processing Systems*, 32, 2019.

Ecoffet, A., Huizinga, J., Lehman, J., Stanley, K. O., and Clune, J. First return, then explore. *Nature*, 590(7847): 580–586, 2021.

Fort, S., Hu, H., and Lakshminarayanan, B. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019.

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

Fujimoto, S. and Gu, S. S. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

Fujimoto, S., Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Fujimoto, S., Meger, D., and Precup, D. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pp. 2052–2062. PMLR, 2019.

Ghasemipour, S. K. S., Gu, S. S., and Nachum, O. Why so pessimistic? estimating uncertainties for offline rl through ensembles, and why their independence matters. *arXiv preprint arXiv:2205.13703*, 2022.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Hao, J., Yang, T., Tang, H., Bai, C., Liu, J., Meng, Z., Liu, P., and Wang, Z. Exploration in deep reinforcement learning: From single-agent to multiagent domain. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2023. doi: 10.1109/TNNLS.2023.3236361.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Hinton, G. E. and Roweis, S. Stochastic neighbor embedding. *Advances in neural information processing systems*, 15, 2002.

Houthooft, R., Chen, X., Duan, Y., Schulman, J., De Turck, F., and Abbeel, P. Vime: Variational information maximizing exploration. *Advances in neural information processing systems*, 29, 2016.

Kidambi, R., Rajeswaran, A., Netrapalli, P., and Joachims, T. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33: 21810–21823, 2020.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

Kumar, A., Fu, J., Soh, M., Tucker, G., and Levine, S. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 1179–1191, 2020.

Lan, Q., Pan, Y., Fyshe, A., and White, M. Maxmin q-learning: Controlling the estimation bias of q-learning. *arXiv preprint arXiv:2002.06487*, 2020.

Lee, K., Laskin, M., Srinivas, A., and Abbeel, P. Sunrise: A simple unified framework for ensemble learning in deep reinforcement learning. In *International Conference on Machine Learning*, pp. 6131–6141. PMLR, 2021.

Lee, S., Seo, Y., Lee, K., Abbeel, P., and Shin, J. Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble. In *Conference on Robot Learning*, pp. 1702–1712. PMLR, 2022.

Louizos, C., Ullrich, K., and Welling, M. Bayesian compression for deep learning. *Advances in neural information processing systems*, 30, 2017.

Lu, Y., Hausman, K., Chebotar, Y., Yan, M., Jang, E., Herzog, A., Xiao, T., Irpan, A., Khansari, M., Kalashnikov, D., et al. Aw-opt: Learning robotic skills with imitation and reinforcement at scale. *arXiv preprint arXiv:2111.05424*, 2021.

Lyu, J., Ma, X., Li, X., and Lu, Z. Mildly conservative q-learning for offline reinforcement learning. *arXiv preprint arXiv:2206.04745*, 2022.

Mao, Y., Wang, C., Wang, B., and Zhang, C. Moore: Model-based offline-to-online reinforcement learning. *arXiv preprint arXiv:2201.10070*, 2022.

Mark, M. S., Ghadirzadeh, A., Chen, X., and Finn, C. Fine-tuning offline policies with optimistic action selection. In *Deep Reinforcement Learning Workshop NeurIPS 2022*, 2022.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533, 2015.

Nair, A., Gupta, A., Dalal, M., and Levine, S. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29, 2016.

Ostrovski, G., Bellemare, M. G., Oord, A., and Munos, R. Count-based exploration with neural density models. In *International conference on machine learning*, pp. 2721–2730. PMLR, 2017.

Peng, X. B., Kumar, A., Zhang, G., and Levine, S. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Savinov, N., Raichuk, A., Marinier, R., Vincent, D., Pollefeys, M., Lillicrap, T., and Gelly, S. Episodic curiosity through reachability. *arXiv preprint arXiv:1810.02274*, 2018.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839): 604–609, 2020.

Seno, T. and Imai, M. d3rlpy: An offline deep reinforcement learning library. *The Journal of Machine Learning Research*, 23(1):14205–14224, 2022.

Siegel, N. Y., Springenberg, J. T., Berkenkamp, F., Abdolmaleki, A., Neunert, M., Lampe, T., Hafner, R., Heess, N., and Riedmiller, M. Keep doing what worked: Behavioral modelling priors for offline reinforcement learning. *arXiv preprint arXiv:2002.08396*, 2020.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

Tang, H., Houthooft, R., Foote, D., Stooke, A., Xi Chen, O., Duan, Y., Schulman, J., DeTurck, F., and Abbeel, P. # exploration: A study of count-based exploration for deep reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Tsividis, P. A., Loula, J., Burga, J., Foss, N., Campero, A., Pouncy, T., Gershman, S. J., and Tenenbaum, J. B. Human-level reinforcement learning through theory-based modeling, exploration, and planning. *arXiv preprint arXiv:2107.12544*, 2021.

Van Hasselt, H., Guez, A., and Silver, D. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Vinyals, O., Babuschkin, I., Chung, J., Mathieu, M., Jaderberg, M., Czarnecki, W. M., Dudzik, A., Huang, A., Georgiev, P., Powell, R., et al. Alphastar: Mastering the real-time strategy game starcraft ii. *DeepMind blog*, 2:20, 2019.

Wu, Y., Tucker, G., and Nachum, O. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

Yang, R., Bai, C., Ma, X., Wang, Z., Zhang, C., and Han, L. Rorl: Robust offline reinforcement learning via conservative smoothing. *arXiv preprint arXiv:2206.02829*, 2022.

Yu, T., Thomas, G., Yu, L., Ermon, S., Zou, J. Y., Levine, S., Finn, C., and Ma, T. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020.

Yu, T., Kumar, A., Rafailov, R., Rajeswaran, A., Levine, S., and Finn, C. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34:28954–28967, 2021.

Zhang, H., Xu, W., and Yu, H. Policy expansion for bridging offline-to-online reinforcement learning. *arXiv preprint arXiv:2302.00935*, 2023.

Zhao, Y., Boney, R., Ilin, A., Kannala, J., and Pajarinen, J. Adaptive behavior cloning regularization for stable offline-to-online reinforcement learning. *arXiv preprint arXiv:2210.13846*, 2022.

Zhou, W., Bajracharya, S., and Held, D. Plas: Latent action space for offline reinforcement learning. In *Conference on Robot Learning*, pp. 1719–1735. PMLR, 2021.

A. Pseudocode

Algorithm 1 E2O: **E**nsemble-based **O**ffline-to-**O**nline RL Framework

Input: Offline dataset $D_{offline}$, offline RL algorithm *OfflineRL*
Output: Offline to online learning algorithm
// **Offline Phase**
Turning offline RL algorithm *OfflineRL* into *OfflineRL-N* with integration of Q-ensembles.
Training *OfflineRL-N* using $D_{offline}$
// **Online Phase**
Removing original pessimistic term in *OfflineRL* (if possible) and thus turn *OfflineRL-N* to *OnlineRL-N*
Setting the Q-target computation method to *WeightedMinPair* and obtain *OnlineRL-N* + *WeightedMinPair*
Introducing *SUNRISE* to encourage exploration and obtain *OnlineRL-N* + *WeightedMinPair* + *SUNRISE*
return *OfflineRL-N* $\rightarrow$ *OnlineRL-N* + *WeightedMinPair* + *SUNRISE*

B. Environment Settings

MuJoCo Gym We investigate three MuJoCo locomotion tasks, namely HalfCheetah, Walker2d, and Hopper (Todorov et al., 2012). The goal of each task is to move forward as fast as possible, while keeping the control cost minimal. For each task, we consider four types of datasets. The random datasets consist of policy rollouts generated by random policies. The medium datasets contain rollouts from medium-level policies. The medium-replay datasets encompass all samples collected during the training of a medium-level agent from scratch. In the case of the medium-expert datasets, half of the data comprises rollouts from medium-level policies, while the other half consists of rollouts from expert-level policies. In this study, we exclude the random and the expert datasets, as in typical real-world scenarios, we rarely use a random policy or have an expert policy for system control. We utilize the v2 version of each dataset.

Antmaze We investigate the Antmaze navigation tasks that involve controlling an 8-DoF ant quadruped robot to navigate through mazes and reach a desired goal. The agent receives sparse rewards of +1/0 based on whether it successfully reaches the goal or not. We study each method using the following datasets from D4RL (Fu et al., 2020): large-diverse, large-play, medium-diverse, medium-play, umaze-diverse, and umaze. The difference between diverse and play datasets is the optimality of the trajectories they contain. The diverse datasets consist of trajectories directed towards random goals from random starting points, whereas the play datasets comprise trajectories directed towards specific locations that may not necessarily correspond to the goal. We use the v1 version of each dataset.

C. Experiment Details

Baselines For CQL, SAC and AWAC, we use the implementation provided by (Seno & Imai, 2022): `https://github.com/takuseno/d3rlpy` with default hyperparameters. For CQL-N and SAC-N, we keep the default setting from the CQL and SAC experiments other than the ensemble size N. For Balanced Replay (BR), as the official implementation provided by the author of (Lee et al., 2022) does not contain the offline pre-training part, we implement Balanced Replay based on d3rlpy. For LAPO-N, we extend the official implementation provided by the author of (Chen et al., 2022): `https://github.com/pcchenxi/LAPO-offlineRL` to easily adjust the size of ensemble. For PEX and IQL, we use the original implementation provided by the author of (Zhang et al., 2023): `https://github.com/Haichao-Zhang/PEX`. While we do not utilize the code provided by the original paper for certain methods, some of the our results obtained using the employed code demonstrate superior performance compared to those reported in the original paper. Moreover, when compared to the results provided in the original paper, our proposed E2O framework consistently outperforms them. We list the hyperparameters for these methods in Table 1.

Offline Pre-training For all experiments, we conduct each algorithm for 1M training steps with 5 different seeds, following the common practice in offline RL works. Specifically, for the D4RL Antmaze tasks, IQL and PEX apply a reward modification by subtracting 1 from all rewards, as described in `https://github.com/tinkoff-ai/CORL/issues/14`. This modification effectively introduces a survival penalty that encourages the agent to complete the maze as quickly as possible. Additionally, LAPO multiplies all rewards by 100, which also enhances the distinction between the rewards for completing tasks and the rewards for unfinished tasks. These reward transformation techniques prove to be

Table 1. Hyperparameters used in the D4RL MuJoCo experiments

Hyperparameters	CQL-N	SAC-N	LAPO-N	AWAC	BR	PEX	IQL
policy learning rate	3e-5	3e-5	2e-4	3e-4	3e-5	3e-4	3e-4
critic learning rate	3e-4	3e-4	2e-4	3e-4	3e-4	3e-4	3e-4
alpha learning rate	1e-4	1e-4	-	-	3e-4	-	-
VAE learning rate	-	-	2e-4	-	-	-	-
value learning rate	-	-	-	-	-	3e-4	3e-4
ensemble size	10	10	10	2	5	2	2
batch size	256	256	512	1024	256	256	256

crucial for achieving desirable performance on the Antmaze tasks.

Online Fine-tuning For all experiments, we report the online fine-tuning performance over 250K timesteps with 5 seeds. Specifically, our framework loads all pre-trained networks, including the policy network, ensemble Q network and ensemble target Q network, while appending the necessary temperature hyperparameter for SAC to facilitate further fine-tuning. In the Antmaze environment, we maintain the same reward modification as the offline phase during training but keep the rewards unchanged during evaluation. To ensure a fair comparison with IQL and PEX, which utilize offline data, we also load LAPO and E2O-LAPO with offline data for online fine-tuning.

D. Additional Results

In this section, we provide more experiments and detailed results to help understand our proposed E2O framework more comprehensively.

D.1. Ablation on Offline Data

We conduct an ablation study to investigate the impact of using offline data during the online fine-tuning phase for all MuJoCo locomotion tasks, as shown in Figure 7. Our results show that E2O-CQL_buffer, which initializes the online buffer with offline data, exhibits slow performance improvement, while discarding the offline data allows it to achieve higher sample efficiency. This suggests that although many offline-to-online RL methods utilize offline data to alleviate performance degradation, it can adversely affect their online sample efficiency. In contrast, E2O-CQL successfully avoids significant performance drop even without using offline data, thereby enhancing learning efficiency during the online stage.

D.2. Comparison of LAPO and E2O-LAPO

On the Antmaze tasks, we have conducted a comparative analysis between our E2O-LAPO method and several offline-to-online RL methods. To further validate the effectiveness of our E2O framework, we directly utilize LAPO with two Q networks for offline-to-online training and compare it with our E2O-LAPO method. The results are shown in Figure 8. As original LAPO can achieve near-optimal performance in simple environments such as umaze and medium mazes during the offline stage, the online fine-tuning performance of both LAPO and E2O-LAPO is comparable. However, in the more challenging large maze environment, directly using the offline pre-trained LAPO agent for online fine-tuning leads to slow performance improvement. By employing our proposed E2O framework, we demonstrate that E2O-LAPO can not only enhance the offline performance of LAPO, but also facilitate more rapid performance improvement while maintaining the offline performance without degradation. This approach enables the offline agent to quickly adapt to the real-world environment, providing efficient and effective online fine-tuning.

D.3. Visualization and Analysis

To better understand the training efficiency of E2O in comparison to traditional pessimistic offline RL algorithms, we compare the distribution of states generated by CQL, E2O-CQL in the online phase, and the distribution of states from the offline dataset. To visualize the results clearly, we plot the distribution with t-Distributed Stochastic Neighbor Embedding (t-SNE) (Hinton & Roweis, 2002). The results are shown in Figure 9, it can be found that both the distribution of E2O-CQL

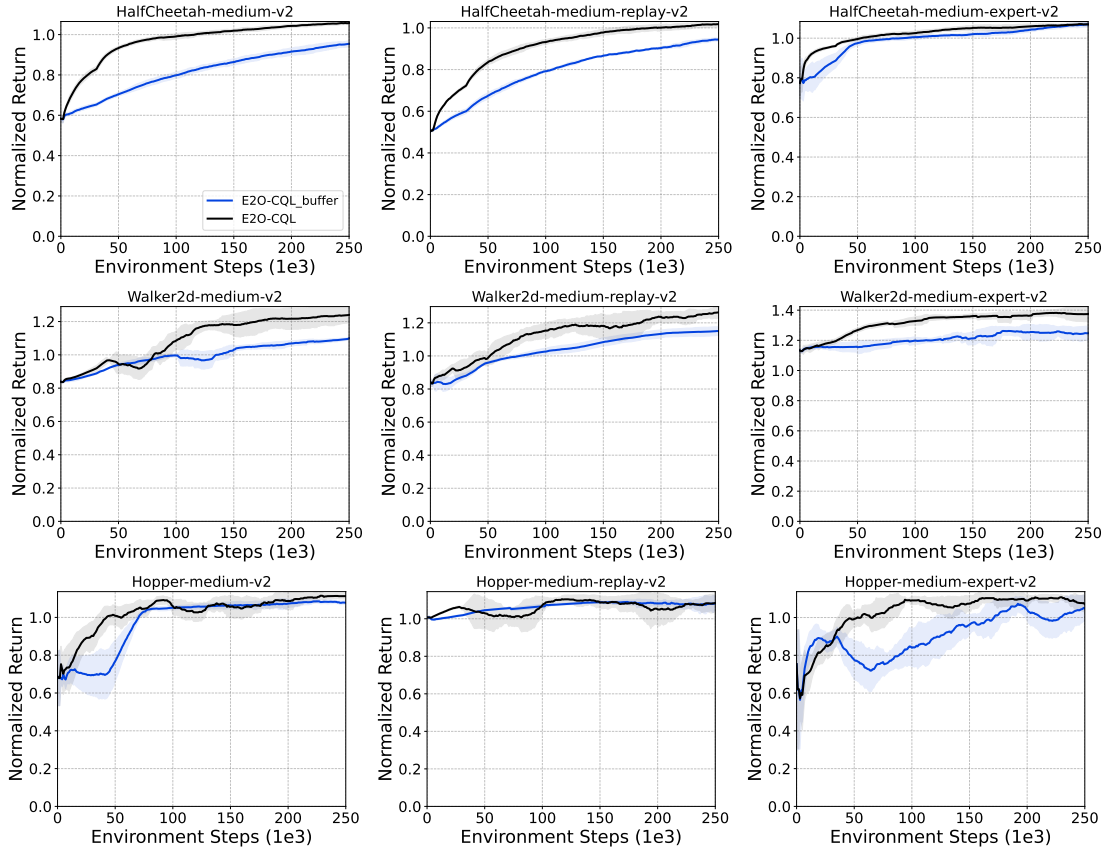


Figure 7. Ablation on offline data. The solid lines and shaded regions represent mean and standard deviation, respectively, across five runs.

states and CQL states bear some similarity to the distribution of offline states. However, the online states accessed by CQL are located on the edge of the offline area, but most still overlap with the offline states. On the other hand, the online states accessed by E2O-CQL deviate further from the offline states. With our ensemble-based design for optimistic exploration, E2O empowers the offline agent to explore more states beyond those contained in the offline dataset. This capability allows for swift adaptation to online environments and facilitates rapid performance improvement.

D.4. Detailed Results of E2O Components

In this section, we provide all learning curves of E2O components that are restricted by the length of the text in the main paper.

Q-Ensembles Figure 10 illustrates the performance of various offline-to-online RL approaches on MuJoCo locomotion tasks. It is evident that the *OfflineRL* $\rightarrow$ *OnlineRL* method exhibits the best performance in the HalfCheetah environment. However, it demonstrates unstable learning in the more complex environments of Walker2d and Hopper. On the other hand, the *OfflineRL* $\rightarrow$ *OfflineRL* approach remains stable but shows slower asymptotic performance. In contrast, the *OfflineRL-N* $\rightarrow$ *OnlineRL-N* method no longer experiences performance collapse after transitioning to online fine-tuning, and its training process is relatively stable across all tasks. Additionally, *OfflineRL-N* $\rightarrow$ *OnlineRL-N* achieves superior fine-tuned performance compared to *OfflineRL* $\rightarrow$ *OfflineRL*. It is worth noting that the Hopper-medium-expert-v2 dataset represents a special case where all considered offline-to-online methods exhibit varying degrees of performance drop, as depicted in this figure and subsequent figures. Nevertheless, our E2O framework consistently achieves state-of-the-art performance in comparison to all baseline methods across most tasks.

Loosing Pessimism Figure 11 displays the performance of *OnlineRL-N* utilizing different Q-target computation methods on MuJoCo locomotion tasks. It is evident that MinQ exhibits remarkable stability across all tasks, albeit with slower

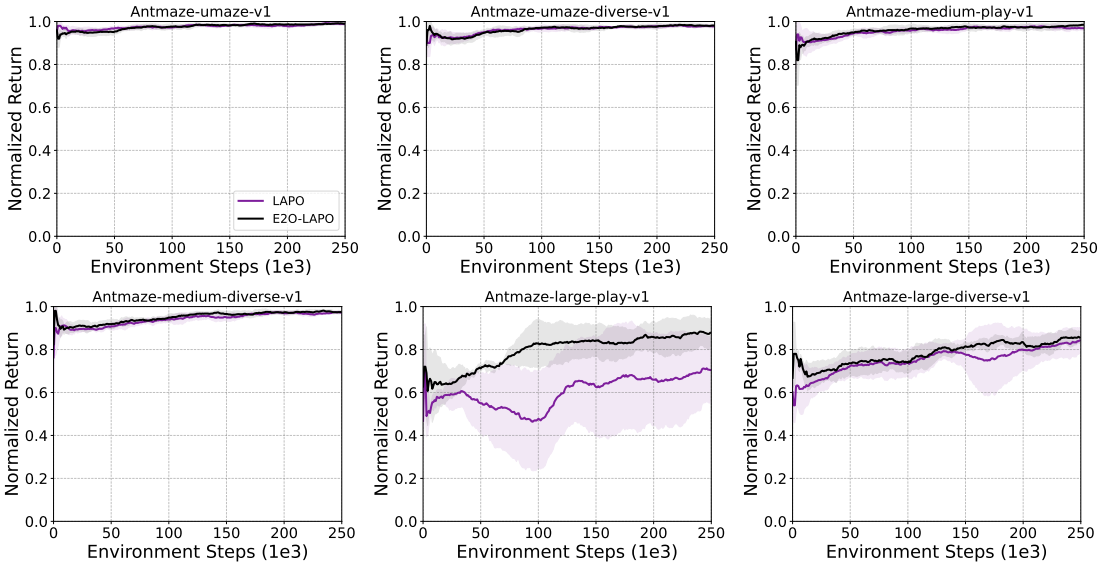


Figure 8. Online learning curves of LAPO and E2O-LAPO across five seeds on Antmaze tasks. The solid lines and shaded regions represent mean and standard deviation, respectively.

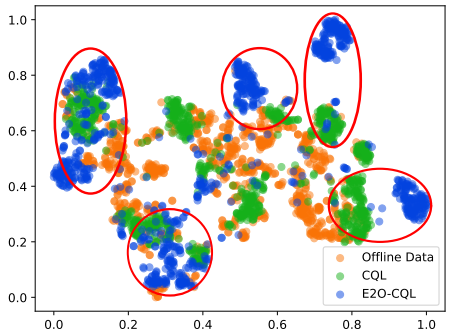


Figure 9. Visualization of the distribution of states generated by E2O-CQL, CQL in online phase and states in offline dataset.

performance improvement in the HalfCheetah and Walker2d environments. MeanQ and REM demonstrate excellent performance in the HalfCheetah environment, but struggle to improve in the more challenging environments of Walker2d and Hopper, and their learning process is characterized by instability. In contrast, RandomMinPair and WeightedMinPair showcase superior performance across most tasks, with the exception of the Hopper-medium-replay-v2 dataset where they exhibit slight instability in learning. Among these two methods, WeightedMinPair demonstrates slightly better stability and performance, thus we select it as the component of our final E2O framework and present the experiments related to RandomMinPair in the appendix.

Optimistic Exploration Figure 12 and Figure 13 present the performance of *OnlineRL-N + WeightedMinPair* and *OnlineRL-N + RandomMinPair* using different exploration methods on MuJoCo locomotion tasks, respectively. These two figures exhibit similar observations. In the Hopper environment, OAC achieves the best performance, but its performance improvement in HalfCheetah and Walker2d is relatively slow. The use of Bootstrapped DQN leads to minimal improvement in performance, while SUNRISE enhances the learning efficiency of *OnlineRL-N + WeightedMinPair* across most tasks, with the exception of the Hopper-medium-replay-v2 dataset where they exhibit slight instability in learning.

D.5. Offline Performance

In this section, we provide the offline learning curves of different methods on locomotion and navigation tasks.

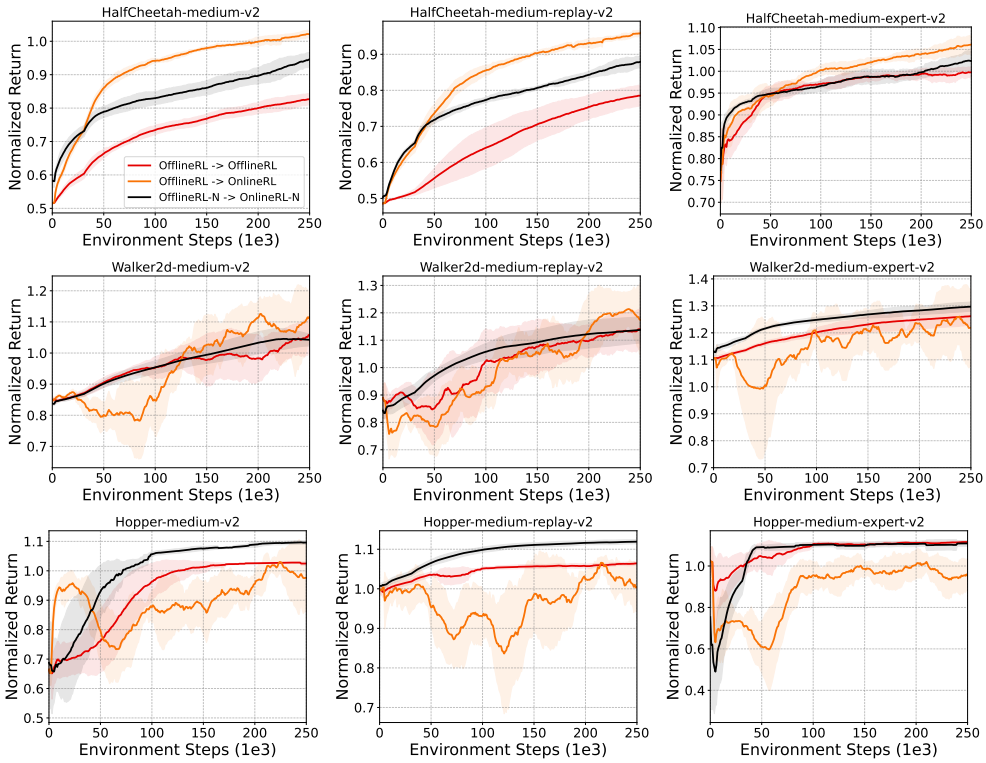


Figure 10. Online learning curves of different offline-to-online approaches across five seeds on MuJoCo locomotion tasks. The solid lines and shaded regions represent mean and standard deviation, respectively.

Locomotion Tasks Figure 14 shows the offline performance of different methods on MuJoCo locomotion tasks. We observe that on certain datasets such as HalfCheetah-medium-v2 and HalfCheetah-medium-replay-v2, E2O-CQL exhibits a slight performance improvement compared to CQL. This indicates that the introduction of Q-ensembles indeed has some benefits for the performance in the offline stage. However, it is important to note that the use of Q-ensembles can impact the convergence speed. For instance, on the Hopper-medium-expert-v2 dataset, E2O-CQL demonstrates a noticeably slower convergence speed compared to CQL. Nevertheless, if both algorithms are allowed to continue training, for example, for 3M training steps, we believe that the performance of E2O-CQL can still surpass that of CQL.

Although different offline RL algorithms may have varying final performance in the offline stage, with some methods potentially performing worse than E2O-CQL, our comparisons in the online stage remain fair. On one hand, the introduction of Q-ensembles can enhance the performance of existing algorithms in the offline stage, which is a inherent advantage of Q-ensembles. On the other hand, higher performance in the offline stage can actually lead to performance drop in the online stage, while lower performance in the offline stage is less prone to such drop. Our proposed E2O framework ensures that the offline policy maintains high performance from the offline stage, and achieves rapid performance improvement in the online stage without encountering performance drop.

Navigation Tasks Figure 15 displays the offline performance of various methods on Antmaze navigation tasks. Firstly, we observe that LAPO outperforms IQL in terms of offline performance, providing a higher starting point for the online phase. This is particularly evident in the umaze and medium maze environments, where LAPO nearly reaches the performance ceiling. Regarding LAPO and E2O-LAPO, since LAPO achieves near-optimal performance in simple environments such as umaze and medium mazes, their offline performance is comparable. However, in the more challenging large maze environment, the inclusion of Q-ensembles enables E2O-LAPO to surpass LAPO in terms of performance.

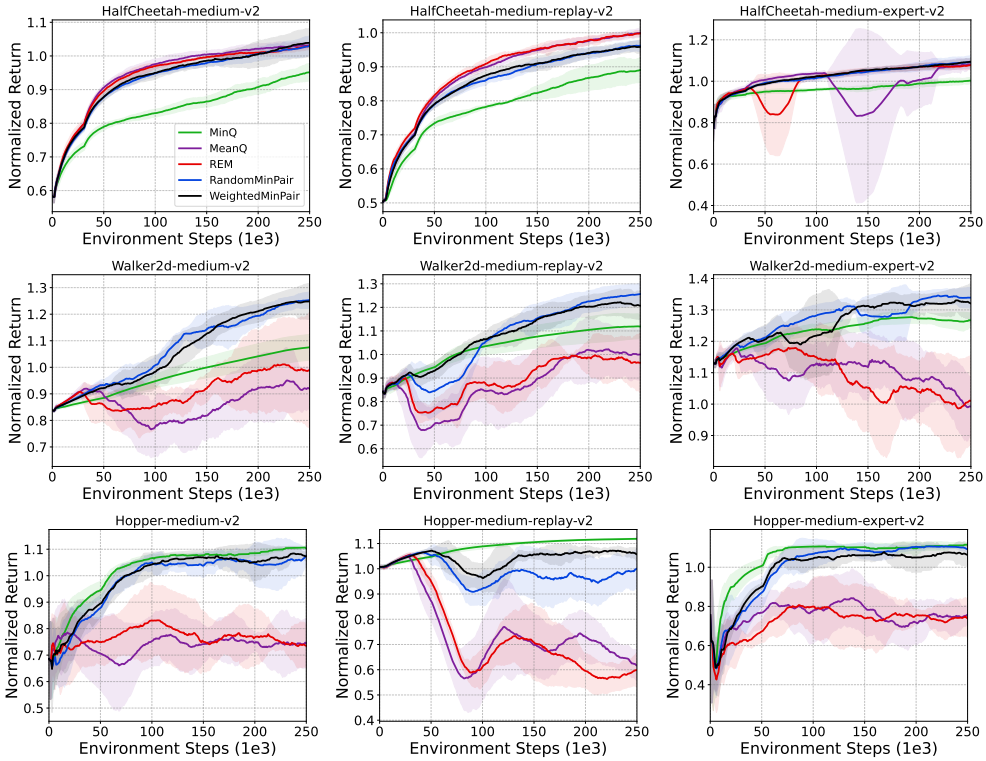


Figure 11. Online learning curves of *OnlineRL-N* using different Q-target computation methods across five seeds on MuJoCo locomotion tasks. The solid lines and shaded regions represent mean and standard deviation, respectively.

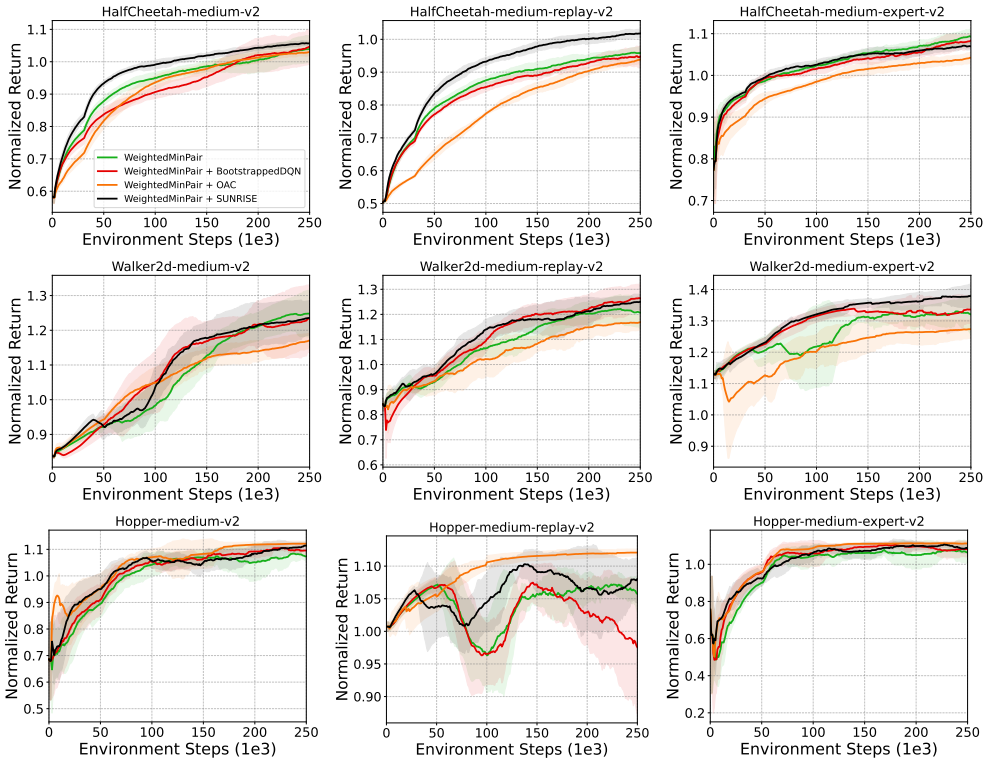


Figure 12. Online learning curves of *OnlineRL-N* + *WeightedMinPair* using different exploration methods across five seeds on MuJoCo locomotion tasks. The solid lines and shaded regions represent mean and standard deviation, respectively.

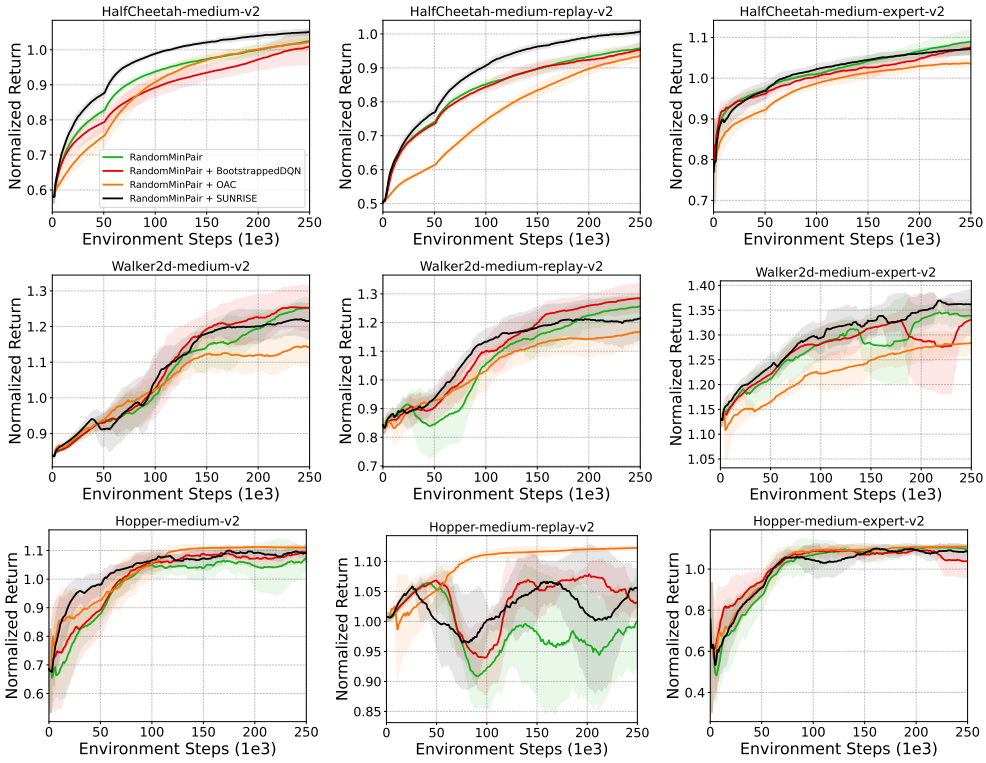


Figure 13. Online learning curves of *OnlineRL-N + RandomMinPair* using different exploration methods across five seeds on MuJoCo locomotion tasks. The solid lines and shaded regions represent mean and standard deviation, respectively.

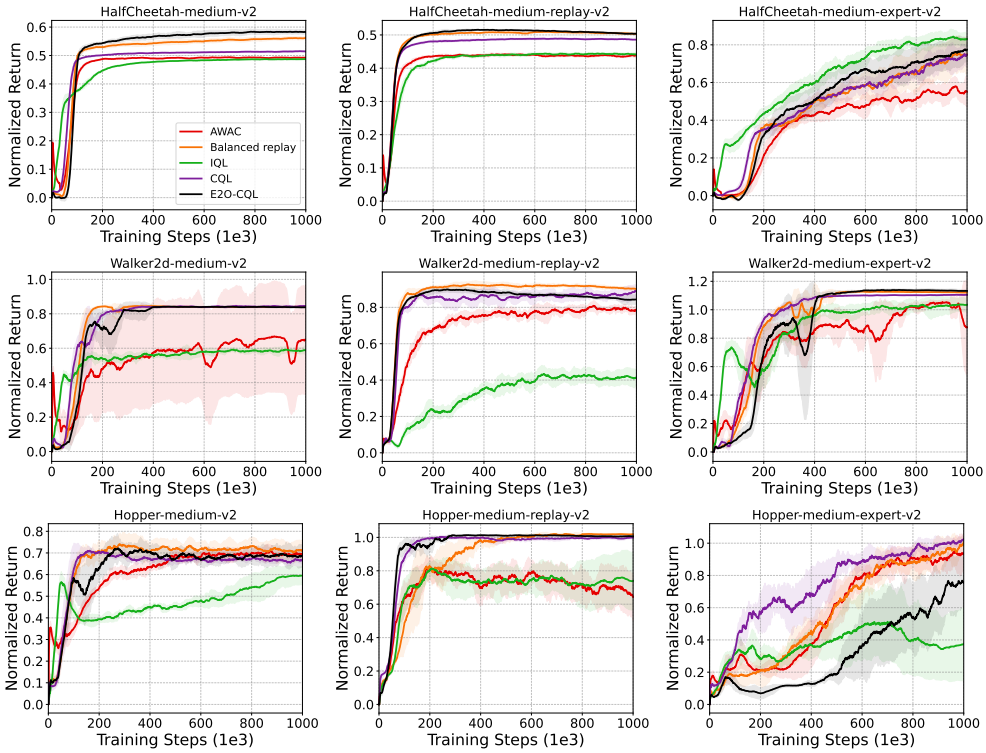


Figure 14. Offline learning curves of different methods across five seeds on MuJoCo locomotion tasks. The solid lines and shaded regions represent mean and standard deviation, respectively.

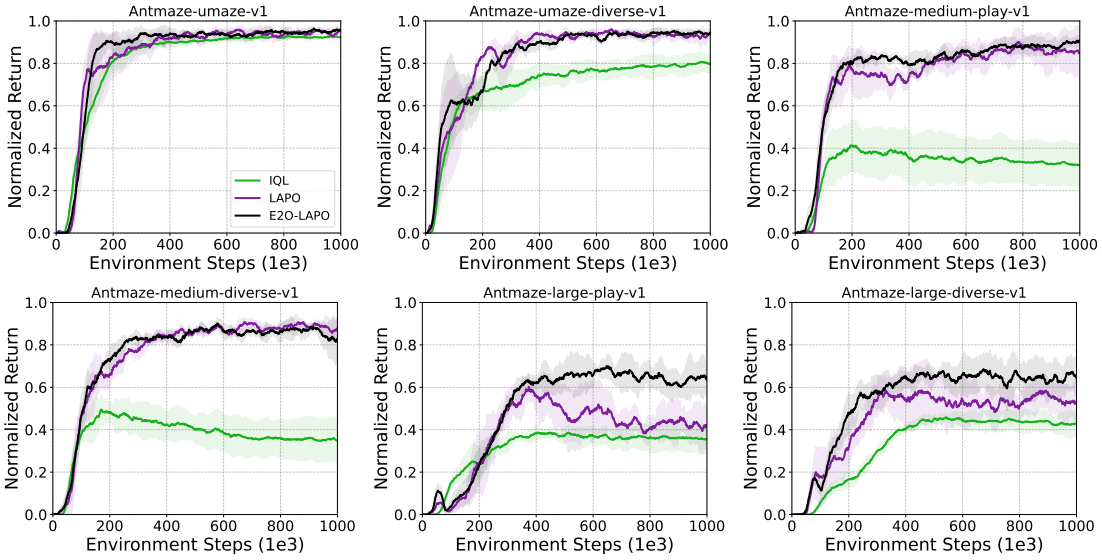


Figure 15. Offline learning curves of different methods across five seeds on Antmaze navigation tasks. The solid lines and shaded regions represent mean and standard deviation, respectively.