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

Kai Zhao[1,2] , **Jianye Hao**[1,*] , **Yi Ma**[1] , **Jinyi Liu**[1] , **Yan Zheng**[1] and **Zhaopeng Meng**[1]
[1]College of Intelligence and Computing, Tianjin University
[2]Bilibili
{kaizhao, jianye.hao, mayi, jyliu, yanzheng, mengzp}@tju.edu.cn

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 **EN**semble-based **O**ffline-**T**o-**O**nline (ENOTO) 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 ENOTO 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

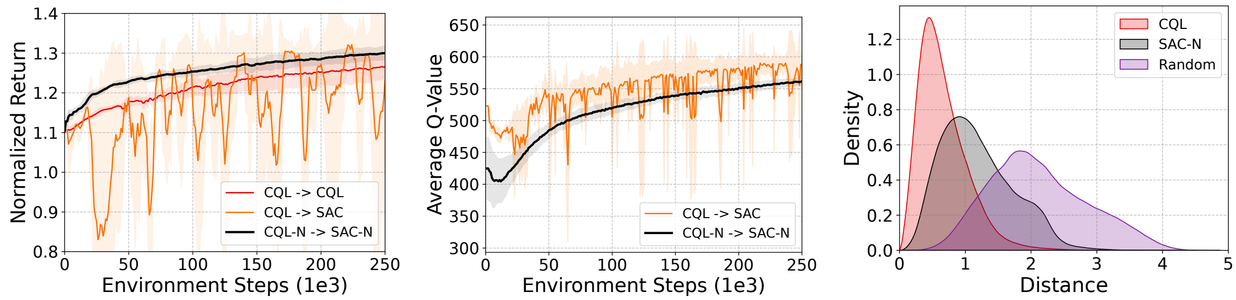
Reinforcement learning (RL) has shown remarkable success in solving complex decision-making problems, from playing virtual games [Silver *et al.*, 2017; Vinyals *et al.*, 2019] to controlling tangible robots [Mnih *et al.*, 2015; Tsividis *et al.*, 2021; Schrittwieser *et al.*, 2020]. In RL, an agent 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; Bai *et al.*, 2022; Liu *et al.*, 2023; Yu *et al.*, 2020; Kidambi *et al.*, 2020].

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 learned policy. Furthermore, starting with an offline RL policy can achieve strong performance with 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, 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

*Corresponding author.



(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.

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 **EN**semble-based **O**ffline-**T**o-**O**nline (ENOTO) RL framework that bridges offline pre-training and online fine-tuning. We demonstrate the effectiveness of ENOTO 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 ENOTO 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 ENOTO on various benchmark tasks, including locomotion and navigation tasks, and verify that ENOTO achieves state-of-the-art performance in comparison to all baseline methods.

2 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 Fig. 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; Hao *et al.*, 2023; Ghasemipour *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 Fig. 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. Fig. 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 Fig. 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 Fig. 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 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 Fig. 1(c). Specifically, we measure 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 [Ecoffet *et al.*, 2021; Lee *et al.*, 2021; Liu *et al.*, 2024; Savinov *et al.*, 2018; Houthooft *et al.*, 2016]. 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.

3 Ensemble-based Offline-to-Online Reinforcement Learning

Based on the empirical observations discussed earlier, we propose our **EN**nsemble-based **O**ffline-**T**o-**O**nline (ENOTO) 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 ENOTO may seem relatively simple, their specific combination outperforms baselines in terms of training stability, learning efficiency and final performance.

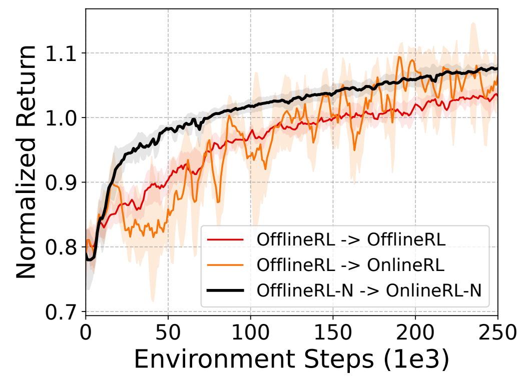


Figure 2: Aggregated learning curves of different offline-to-online RL approaches on all considered MuJoCo datasets.

3.1 Q-Ensembles

As discussed in the previous section, 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. 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.

Fig. 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* → *OnlineRL-N* no longer experiences performance collapse after switching to online fine-tuning, and the training process is relatively stable. Additionally, *OfflineRL-N* → *OnlineRL-N* achieves better fine-tuned performance than *OfflineRL* → *OfflineRL*.

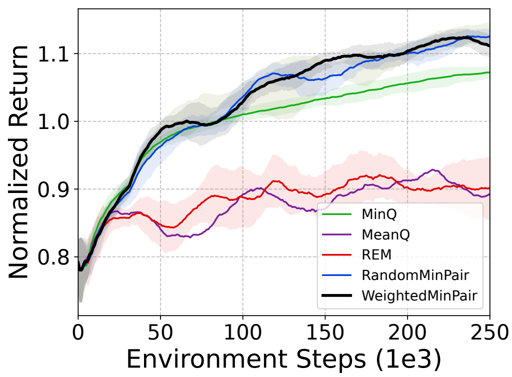


Figure 3: Aggregated learning curves of *OnlineRL-N* using different Q-target computation methods on all considered MuJoCo datasets.

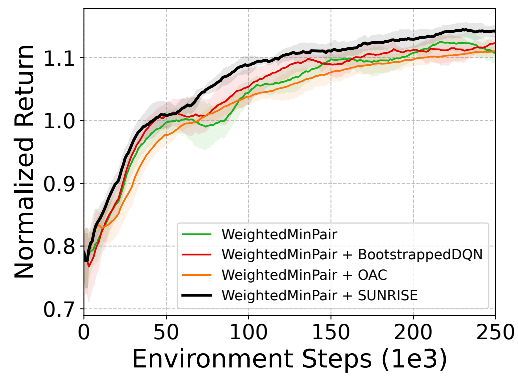


Figure 4: Aggregated learning curves of *OnlineRL-N + WeightedMinPair* using different exploration methods on all considered MuJoCo datasets.

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.

3.2 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.

Fig. 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.

3.3 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-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 re-weighting target Q-values based on uncertainty estimates.

The results of different exploration methods is presented in Fig. 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 ENOTO. Algorithm 1 summarizes the offline and online procedures of ENOTO. Note that as many offline RL algorithms can integrate ensemble technique in Q-functions, ENOTO can thus

Algorithm 1 ENOTO: **EN**semble-based **O**ffline-**T**o-**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*

serve as a common plugin. We will further show the plug-and-play character of ENOTO by applying ***OfflineRL-N*** $\rightarrow$ ***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.

4 Experiments

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

4.1 Locomotion Tasks

We first evaluate our ENOTO 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 ENOTO 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 the appendix.

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.
- **BR** [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.
- **Cal-QL** [Nakamoto *et al.*, 2023]: a recent offline-to-online RL method learning a conservative value function initialization that underestimates the value of the learned policy from offline data, while also being calibrated, in the sense that the learned Q-values are at a reasonable scale.
- **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 ENOTO framework.

Fig. 5 shows the performance of the ENOTO-CQL method (ENOTO instantiated on CQL) and baseline methods during the online fine-tuning phase. Compared with pure online RL methods such as SAC and Scratch, ENOTO-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 ENOTO-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 BR can attain performance second only to ENOTO-CQL on some datasets, it also suffers from unstable training. PEX exhibits a notable decline in performance during the initial stages of online fine-tuning across various datasets, attributed to the randomness of newly trained policies in the early phase, which negatively affects training stability. Although the original PEX paper does not explicitly address this phenomenon, a meticulous examination of its experimental section reveals that performance drop indeed affects PEX. We contend that the phenomenon of performance drop is a pivotal concern in the domain of offline-to-online RL, warranting significant attention. Turning to the Cal-QL algorithm, while its efficacy is prominently showcased in intricate tasks such as Antmaze, Adroit, and Kitchen, as emphasized in the paper, we note a more subdued performance in traditional MuJoCo tasks. The enhancements during the online phase appear relatively constrained. However, its most salient attribute lies in its exceptional stability, effectively circumventing the issue of per-

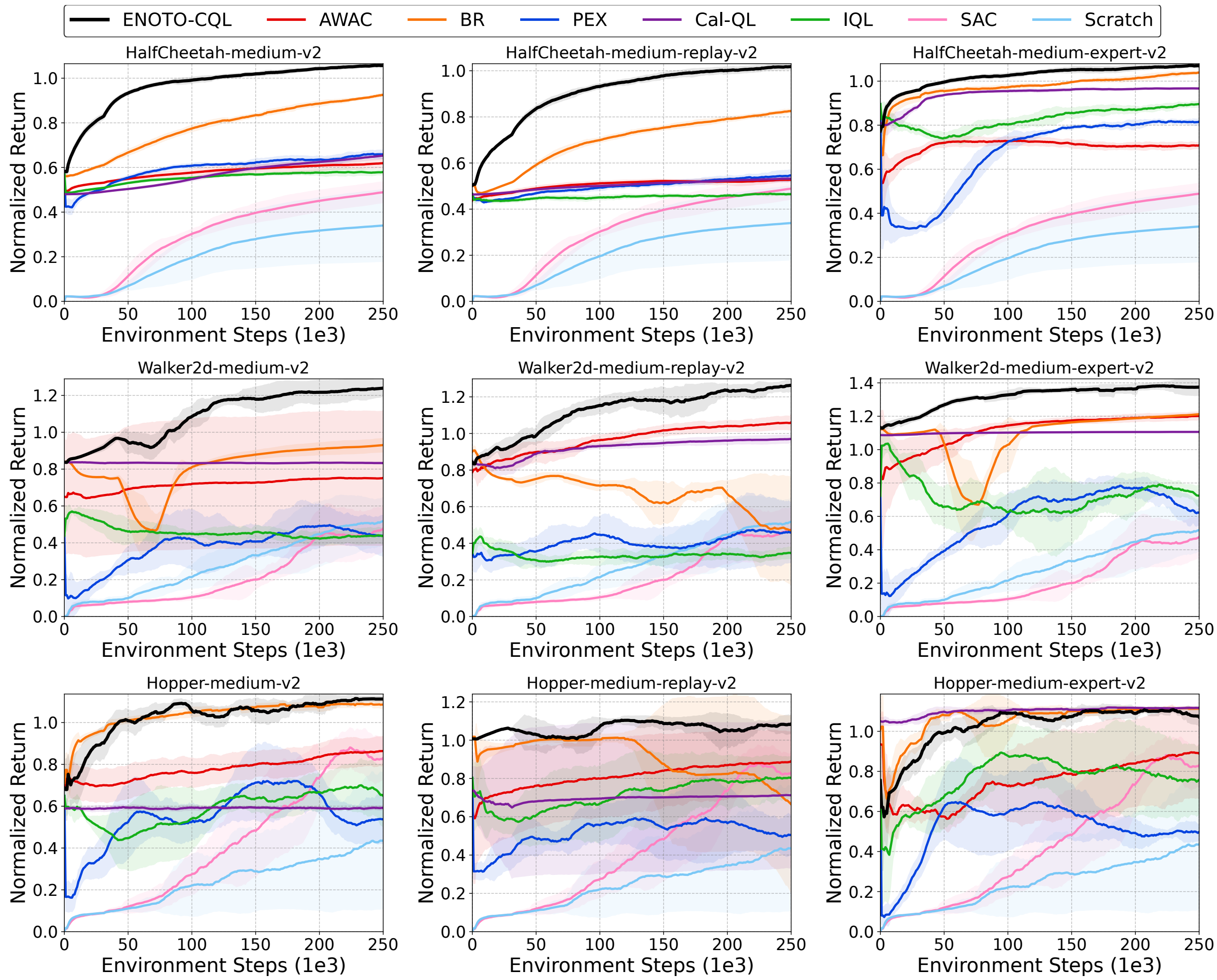


Figure 5: Online learning curves of different methods across five seeds on MuJoCo locomotion tasks. The solid lines and shaded regions represent mean and standard deviation, respectively.

formance drop. It is worth noting that the Hopper-medium-expert-v2 dataset represents a special case where most considered offline-to-online RL methods exhibit varying degrees of performance drop, except for Cal-QL, which maintains its offline-stage performance while remaining stable.

It is important to underscore that due to the partial incompleteness of code provided by certain baseline algorithms, our experiments partially rely on publicly available and widely accepted code repositories from GitHub [Seno and Imai, 2022; Tarasov *et al.*, 2022]. Consequently, the experimental results may exhibit slight deviations from the reported outcomes in the original papers, which will be comprehensively detailed in the appendix. Nevertheless, through rigorous comparisons encompassing both the baseline papers' original performance metrics and the results obtained from our code implementation, our ENOTO method consistently surpasses the baseline approaches in terms of training stability, learning efficiency, and final performance across most tasks. Unfortunately, due to constraints within this text, we can only present the results attained from executing the code, as graphical representations from the source papers cannot be seamlessly incorporated.

4.2 Navigation Tasks

We further verify the effectiveness of ENOTO on D4RL navigation task Antmaze [Fu *et al.*, 2020] by integrating another offline RL algorithm LAPO [Chen *et al.*, 2022]. In detail, we specialize ENOTO as LAPO-N + WeightedMinPair + SUNRISE, i.e., ENOTO-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 ENOTO-LAPO method on this task with three effective baseline methods, IQL, PEX and Cal-QL. Specifically, for the D4RL Antmaze tasks, these methods apply a reward modification following previous works. This modification effectively introduces a survival penalty that encourages the agent to complete the maze as quickly as possible. In the online

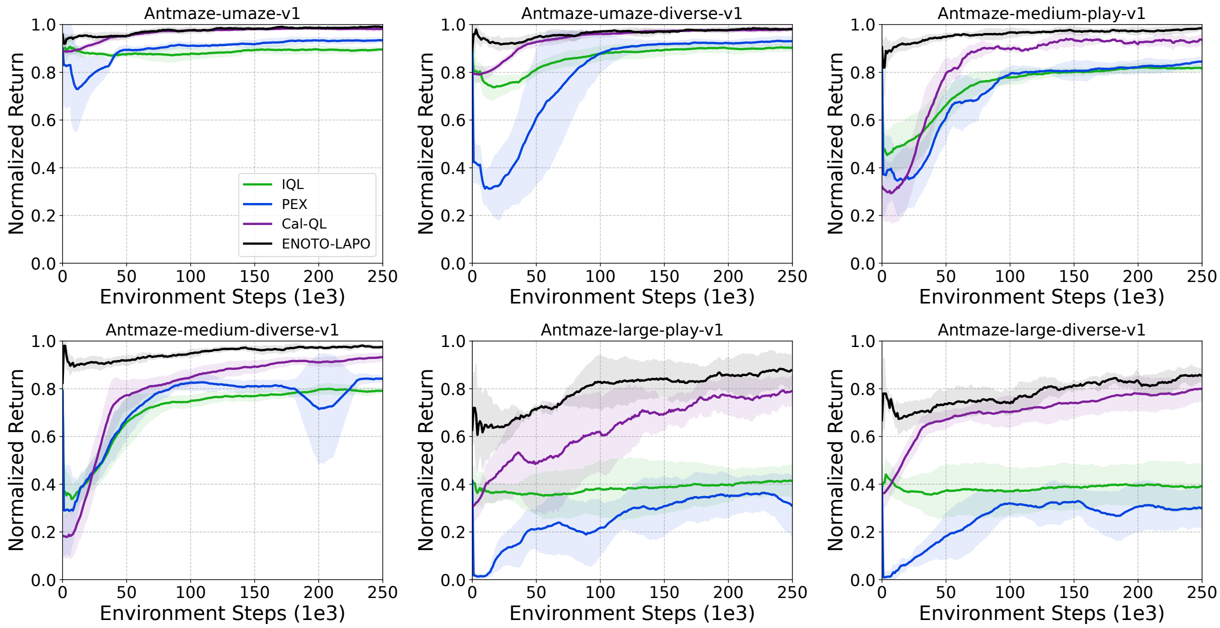


Figure 6: Online learning curves of different methods across five seeds on Antmaze navigation tasks. The solid lines and shaded regions represent mean and standard deviation, respectively.

phase, we maintain the same reward modification as the offline phase during training but keep the rewards unchanged during evaluation.

Fig. 6 presents the performance of ENOTO-LAPO and baseline methods during the online fine-tuning phase. First, LAPO demonstrates better offline performance than IQL, 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. Regarding the Cal-QL algorithm, akin to the outcomes portrayed in the original paper, it demonstrates robust performance in the Antmaze environment, outperforming significantly its MuJoCo counterparts. Notably, it exhibits superior stability and learning efficiency compared to the two baseline methods, IQL and PEX. For our proposed ENOTO framework, we demonstrate that ENOTO-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 ENOTO-LAPO method to further verify the effectiveness of our ENOTO framework. The results including some ablation studies can be found in the appendix.

5 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 (ENOTO) 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 ENOTO 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 ENOTO is designed to be a flexible plugin for various offline RL algorithms, it may require further modifications to achieve optimal performance in different contexts. For instance, adjusting the weight coefficient of the BC item may result in better fine-tuning performance for TD3+BC [Fujimoto and Gu, 2021]. Second, the computational cost of ensembles and uncertainty estimates may limit the scalability of ENOTO 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 by using Bayesian compression [Louizos *et al.*, 2017] or variational approximations [Kingma and Welling, 2013]. These methods could make ENOTO 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.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant Nos. 92370132, 62106172), the Science and Technology on Information Systems Engineering Laboratory (Grant Nos. WDZC20235250409, 6142101220304), and the Xiaomi Young Talents Program of Xiaomi Foundation.

References

[Agarwal *et al.*, 2020] Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, pages 104–114. PMLR, 2020.

[An *et al.*, 2021] Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble. *Advances in neural information processing systems*, 34:7436–7447, 2021.

[Bai *et al.*, 2022] Chenjia Bai, Lingxiao Wang, Zhuoran Yang, Zhihong Deng, Animesh Garg, Peng Liu, and Zhaoran Wang. Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning. *arXiv preprint arXiv:2202.11566*, 2022.

[Chen *et al.*, 2021] Xinyue Chen, Che Wang, Zijian Zhou, and Keith Ross. Randomized ensembled double q-learning: Learning fast without a model. *arXiv preprint arXiv:2101.05982*, 2021.

[Chen *et al.*, 2022] Xi Chen, Ali Ghadirzadeh, Tianhe Yu, Yuan Gao, Jianhao Wang, Wenzhe Li, Bin Liang, Chelsea Finn, and Chongjie Zhang. Latent-variable advantage-weighted policy optimization for offline rl. *arXiv preprint arXiv:2203.08949*, 2022.

[Ciosek *et al.*, 2019] Kamil Ciosek, Quan Vuong, Robert Loftin, and Katja Hofmann. Better exploration with optimistic actor critic. *Advances in Neural Information Processing Systems*, 32, 2019.

[Ecoffet *et al.*, 2021] Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune. First return, then explore. *Nature*, 590(7847):580–586, 2021.

[Fort *et al.*, 2019] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019.

[Fu *et al.*, 2020] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

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

[Fujimoto *et al.*, 2019] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pages 2052–2062. PMLR, 2019.

[Ghasemipour *et al.*, 2022] Seyed Kamyar Seyed Ghasemipour, Shixiang Shane Gu, and Ofir Nachum. Why so pessimistic? estimating uncertainties for offline rl through ensembles, and why their independence matters. *arXiv preprint arXiv:2205.13703*, 2022.

[Haarnoja *et al.*, 2018] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

[Hao *et al.*, 2023] Jianye Hao, Tianpei Yang, Hongyao Tang, Chenjia Bai, Jinyi Liu, Zhaopeng Meng, Peng Liu, and Zhen Wang. Exploration in deep reinforcement learning: From single-agent to multiagent domain. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21, 2023.

[Hinton *et al.*, 2015] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[Houthooft *et al.*, 2016] Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. *Advances in neural information processing systems*, 29, 2016.

[Kidambi *et al.*, 2020] Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33:21810–21823, 2020.

[Kingma and Welling, 2013] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[Kostrikov *et al.*, 2021] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

[Kumar *et al.*, 2020] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

[Lee *et al.*, 2021] Kimin Lee, Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Sunrise: A simple unified framework for ensemble learning in deep reinforcement learning. In *International Conference on Machine Learning*, pages 6131–6141. PMLR, 2021.

[Lee *et al.*, 2022] Seunghyun Lee, Younggyo Seo, Kimin Lee, Pieter Abbeel, and Jinwoo Shin. Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble. In *Conference on Robot Learning*, pages 1702–1712. PMLR, 2022.

[Liu *et al.*, 2023] Jinyi Liu, Yi Ma, Jianye Hao, Yujing Hu, Yan Zheng, Tangjie Lv, and Changjie Fan. Prioritized trajectory replay: A replay memory for data-driven reinforcement learning. *CoRR*, abs/2306.15503, 2023.

[Liu *et al.*, 2024] Jinyi Liu, Zhi Wang, Yan Zheng, Jianye Hao, Chenjia Bai, Junjie Ye, Zhen Wang, Haiyin Piao, and

Yang Sun. Ovd-explorer: Optimism should not be the sole pursuit of exploration in noisy environments. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 13954–13962. AAAI Press, 2024.

[Louizos *et al.*, 2017] Christos Louizos, Karen Ullrich, and Max Welling. Bayesian compression for deep learning. *Advances in neural information processing systems*, 30, 2017.

[Lu *et al.*, 2021] Yao Lu, Karol Hausman, Yevgen Chebotar, Mengyuan Yan, Eric Jang, Alexander Herzog, Ted Xiao, Alex Irpan, Mohi Khansari, Dmitry Kalashnikov, et al. Aw-opt: Learning robotic skills with imitation and reinforcement at scale. *arXiv preprint arXiv:2111.05424*, 2021.

[Mao *et al.*, 2022] Yihuan Mao, Chao Wang, Bin Wang, and Chongjie Zhang. Moore: Model-based offline-to-online reinforcement learning. *arXiv preprint arXiv:2201.10070*, 2022.

[Mark *et al.*, 2022] Max Sobol Mark, Ali Ghadirzadeh, Xi Chen, and Chelsea Finn. Fine-tuning offline policies with optimistic action selection. In *Deep Reinforcement Learning Workshop NeurIPS 2022*, 2022.

[Mnih *et al.*, 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[Nair *et al.*, 2020] Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

[Nakamoto *et al.*, 2023] Mitsuhiko Nakamoto, Yuexiang Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning. *arXiv preprint arXiv:2303.05479*, 2023.

[Osband *et al.*, 2016] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29, 2016.

[Savinov *et al.*, 2018] Nikolay Savinov, Anton Raichuk, Raphaël Marinier, Damien Vincent, Marc Pollefeys, Timothy Lillicrap, and Sylvain Gelly. Episodic curiosity through reachability. *arXiv preprint arXiv:1810.02274*, 2018.

[Schrittwieser *et al.*, 2020] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

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

[Silver *et al.*, 2017] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[Silver *et al.*, 2018] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

[Tarasov *et al.*, 2022] Denis Tarasov, Alexander Nikulin, Dmitry Akimov, Vladislav Kurenkov, and Sergey Kolesnikov. CORL: Research-oriented deep offline reinforcement learning library. In *3rd Offline RL Workshop: Offline RL as a "Launchpad"*, 2022.

[Todorov *et al.*, 2012] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[Tsividis *et al.*, 2021] Pedro A Tsividis, Joao Loula, Jake Burga, Nathan Foss, Andres Campero, Thomas Pouncy, Samuel J Gershman, and Joshua B Tenenbaum. Human-level reinforcement learning through theory-based modeling, exploration, and planning. *arXiv preprint arXiv:2107.12544*, 2021.

[Vinyals *et al.*, 2019] Oriol Vinyals, Igor Babuschkin, Junyoung Chung, Michael Mathieu, Max Jaderberg, Wojciech M Czarnecki, Andrew Dudzik, Aja Huang, Petko Georgiev, Richard Powell, et al. Alphastar: Mastering the real-time strategy game starcraft ii. *DeepMind blog*, 2:20, 2019.

[Wu *et al.*, 2019] Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

[Yu *et al.*, 2020] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020.

[Zhang *et al.*, 2023] Haichao Zhang, We Xu, and Haonan Yu. Policy expansion for bridging offline-to-online reinforcement learning. *arXiv preprint arXiv:2302.00935*, 2023.

[Zhao *et al.*, 2022] Yi Zhao, Rinu Boney, Alexander Ilin, Juho Kannala, and Joni Pajarinen. Adaptive behavior cloning regularization for stable offline-to-online reinforcement learning. *arXiv preprint arXiv:2210.13846*, 2022.