Online Fine-Tuning with Uncertainty Quantification for Offline Pre-Trained Agents

Ingook Jang, Seonghyun Kim, Samyeul Noh
Electronics and Telecommunications Research Institute
ingook@etri.re.kr

Abstract

This paper proposes an online fine-tuning with uncertainty quantification for offline pre-trained agents in deep reinforcement learning (RL). Offline RL allows agents to learn from pre-collected datasets without additional environment interactions, but faces challenges like distributional shifts and uncertainty during online fine-tuning. Our method incorporates uncertainty quantification into an ensemble of pessimistic Q-functions. The uncertainty-based penalization mitigates the effects of distributional shift during online fine-tuning, resulting in more stable and sample-efficient learning. Through experiments on D4RL locomotion tasks with various datasets, we demonstrate that the proposed method outperforms existing baseline methods, achieving superior performance with fewer environment interactions. The results highlight the effectiveness of uncertainty quantification in managing distributional shift and improving the robustness of online fine-tuning from offline pre-trained agents.

1 Introduction

Deep offline reinforcement learning (RL) trains neural networks using previously collected datasets, enabling agents to learn without additional interactions with the environment, as shown by Levine et al. [2020]. Recent advances in offline RL, such as the works of Fujimoto et al. [2019], Kumar et al. [2019], Agarwal et al. [2020], Yu et al. [2020], Kumar et al. [2020], have demonstrated that offline RL can outperform the behavior policies used to generate the offline datasets. However, offline RL still faces several key challenges: (1) the datasets may stem from suboptimal behavior policies; (2) the behavior policies may lack sufficient exploration; or (3) the datasets may be too small to fully capture the environment's dynamics. These limitations often necessitate further online fine-tuning through additional interactions with the environment.

The goal of online fine-tuning is to leverage offline datasets to improve the sample efficiency and the asymptotic performance of the agent during online learning. This process involves using the offline dataset to estimate an initial policy, which ideally provides a strong starting point for the agent. The challenge arises when transitioning to online learning, where the distributional shift between the offline dataset and the new online interactions leads to large initial temporal difference (TD) errors. These errors can cause significant performance degradation and result in the agent "forgetting" valuable information learned during offline training, thus reducing sample efficiency. Fine-tuning an offline pre-trained agent can be particularly difficult when using traditional off-policy RL methods due to this distributional shift.

To address these issues, the offline-to-online RL framework (Off2OnRL) proposed by Lee et al. [2022] introduces a pessimistic Q-ensemble scheme, which trains multiple Q-functions to mitigate the bootstrapping errors caused by the distributional shift. By constraining the learning policy to remain close to the distribution of the behavior policy, this method stabilizes the fine-tuning process. Off2OnRL also utilizes a prioritized buffer with balanced replay, combining offline and online

38th Workshop on Fine-Tuning in Machine Learning (NeurIPS 2024).

samples to ensure that the Q-function is trained with a mixture of data, which helps maintain stable value estimates.

However, despite the use of balanced replay with near-on-policy samples, the method is not completely free from uncertainty issues. This is because both samples generated by the learning agent and those generated by the behavior agent are used together in the training process. As a result, uncertainty can still arise, potentially destabilizing learning performance. To further enhance learning stability and performance, it is essential to quantify uncertainty and incorporate it as a penalization factor during learning. By effectively managing this uncertainty, we can improve the performance of online fine-tuning from offline pre-trained agents.

In this paper, we introduce an Uncertainty-driven Pessimistic Q-ensemble (UPQ) for online fine-tuning for offline pre-trained agents to address these challenges. Our approach builds on the Off2OnRL framework by adopting an ensemble of pessimistic actor-critic agents, which are trained with uncertainty quantification to guide the learning policy more effectively. By incorporating uncertainty-based penalization into the replay buffer, we aim to mitigate the impact of distributional shift during online fine-tuning. This helps to stabilize Q-function estimates and improves sample efficiency.

Through our experiments, we demonstrate that UPQ achieves superior performance with fewer online interactions compared to state-of-the-art methods. Our method not only improves sample efficiency but also delivers robust policies that better leverage the initial offline dataset, addressing the key challenges in online fine-tuning for offline pre-trained agents.

2 Methodology

2.1 Offline Pre-Training

We use Conservative Q-Learning (CQL) proposed in Kumar et al. [2020], which is a method designed to address the Q-value overestimation for out-of-distribution (OOD) actions in offline RL. An agent learns a policy from a fixed dataset, and this can lead to the overestimation of Q-values for actions that are either not represented or poorly represented in the dataset. CQL introduces conservative regularization to mitigate this problem by penalizing Q-values for OOD actions, effectively biasing the learned policy toward actions observed in the dataset. This makes CQL well-suited for learning robust and reliable policies from fixed offline data.

An actor-critic agent $\{Q_\theta, \pi_\phi\}$ learns its Q-function and policy from a replay buffer $\mathcal{B}$ using CQL with KL-divergence against a prior distribution over actions for policy evaluation, which minimize the following:

$$\alpha \mathbb{E}_{s\sim\mathcal{B}}\left[\log \sum_a \exp(Q(s,a)) - \mathbb{E}_{a\sim\hat{\pi}_\beta(a|s)}\left[Q(s,a)\right]\right] + \frac{1}{2}\mathbb{E}_{(s,a,s')\sim\mathcal{B}}\left[\left(Q_\theta - B^{\pi_\phi}Q_{\bar{\theta}}\right)^2\right], \quad (1)$$

where α is the tradeoff factor, $\bar{\theta}$ is the parameters of the target network, $\hat{\pi}_\beta$ is the empirical behaviral policy, and B^π is the Bellman operator. This variant maintains a soft-maximum of Q-values, ensuring a conservative estimate while minimizing the risks of overestimation for OOD actions.

2.2 Online Fine-Tuning

In our proposed method, we use multiple pre-trained pessimistic Q-functions and stochastic policies (i.e. N CQL agents $\{Q_{\theta_i}, \pi_{\phi_i}\}_{i\in[N]}$) to alleviate distribution shift effectively, we leverage multiple pessimistically trained Q-functions, where θ_i and ϕ_i are defined as the parameters of the i agent's critic and actor, respectively. We define the Q-function and the policy of the ensemble as follows:

$$Q^E_\theta(s,a) := \frac{1}{N}\sum_{i=1}^{N} Q_{\theta_i}(s,a), \quad (2)$$

$$\pi^E_\phi(\cdot|s) = \mathcal{N}\left(\frac{1}{N}\sum_{i=1}^{N}\mu_{\phi_i}(s), \quad \frac{1}{N}\sum_{i=1}^{N}\left(\sigma^2_{\phi_i}(s) + \mu^2_{\phi_i}(s)\right) - \mu^2_\phi(s)\right), \quad (3)$$

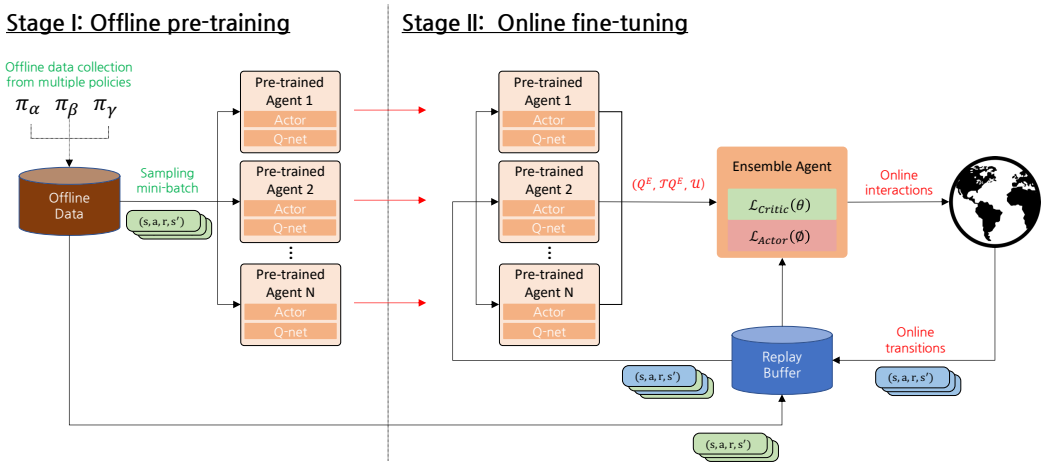

Figure 1: Illustration of the overall architecture of the proposed online fine-tuning with uncertainty quantification. Our method calculates the Q-target $\mathcal{T}Q^E$ and the corresponding uncertainty $\mathcal{U}$ by using the ensemble of the N offline agents pre-trained from the offline dataset to update the gradients.

where the parameters are defined as $\theta := \{\theta_i\}_{i\in[N]}$ and $\phi := \{\phi_i\}_{i\in[N]}$, respectively. The defined π^E_ϕ follows a normal distribution with mean and variance of the Gaussian mixture $\frac{1}{N}\sum_{i=1}^N \pi_{\phi_i}$ for parameterization. The modeled policy is the same as Off2OnRL.

Since the ensemble estimates the posterior distribution of its Q-functions, we use the standard deviation-based uncertainty quantification technique borrowed from Pessimistic Bootstrapping for offline RL (PBRL) proposed by Bai et al. [2022]). The uncertainty quantification at (s', a') of the target Q-functions is defined as follows:

$$\mathcal{U}_{\bar{\theta}}(s', a') := \sigma(Q_{\bar{\theta}_i}(s', a')) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(Q_{\bar{\theta}_i}(s', a') - Q^E_{\bar{\theta}}(s', a')\right)^2}, \tag{4}$$

where $\bar{\theta}_i$ is the parameters of the i agent's target Q-network and we denote the mean over the ensemble of the target Q-functions by $Q^E_{\bar{\theta}}$. In policy evaluation, we use such uncertainty quantification as a penalization term to the next Q-value for a mixture of online and offline samples from the replay buffer. The Q^E of the ensemble agent is updated through pessimistic Q-function updates by fitting the following target for state-action pairs sampled from $\mathcal{B}$:

$$\mathcal{T}Q^E_\theta(s, a) := r(s, a) + \gamma\mathbb{E}_{a'\sim\pi^E_\phi}\left[Q^E_{\bar{\theta}}(s', a') - \alpha\log\pi^E_\phi(a'|s') - \beta\mathcal{U}_{\bar{\theta}}(s', a')\right], \tag{5}$$

where β is the penalization parameter for the uncertainty quantification. To this end, the parameters of the Q-network and the policy of the ensemble agent, θ and ϕ, are updated by minimizing the following objectives, respectively:

$$\mathcal{L}_{Critic}(\theta) = \mathbb{E}_{(s,a,s')\sim\mathcal{B}}\left[\left(Q^E_\theta(s, a) - \mathcal{T}Q^E_\theta(s, a)\right)^2\right], \tag{6}$$

$$\mathcal{L}_{Actor}(\phi) = \mathbb{E}_{s\sim\mathcal{B}, a\sim\pi^E_\phi}\left[\alpha\log\pi^E_\phi(a|s) - Q^E_\theta(s, a)\right], \tag{7}$$

where α is the parameter for temperature.

Figure 1 illustrates the overall workflow of the online fine-tuning with uncertainty quantification. The learning Q-function and policy are updated via Eq. (6) and (7) by utilizing multiple offline pre-trained agents. Pseudocode is represented in Algorithm 1, with differences from conventional online fine-tuning algorithms in red.

Algorithm 1 Online fine-tuning with uncertainty quantification

Require: Ensemble agent $\{\pi_\phi, Q_\theta\}$, offline dataset $\mathcal{D}$
Initialize replay buffer $\mathcal{B} \leftarrow \emptyset$
for $j = 1, \ldots, |\mathcal{D}|$ **do**
    $\mathcal{B} \leftarrow \mathcal{B} \cup \{\tau_j^{\text{off}}\}, \tau_j^{\text{off}} = (s, a, r, s')_j$ from $\mathcal{D}$
end for
for each iteration **do**
    // COLLECT TRAINING SAMPLES
    Collect a transition $\tau^{\text{on}} = (s, a, r, s')$ via online interaction with π_ϕ
    Update $\mathcal{B} \leftarrow \mathcal{B} \cup \{\tau^{\text{on}}\}$
    // QUANTIFY UNCERTAINTY
    Sample a random minibatch $\{\tau_i\}_{i=1}^N \sim \mathcal{B}$
    Calculate the uncertainty $\mathcal{U}_{\bar{\theta}}(s', a')$ in Eq.(4) through the target-networks
    Calculate the Q-target in Eq.(5)
    // UPDATE AGENT
    Calculate $\mathcal{L}_{Critic}(\theta)$ in Eq.(6) and update θ
    Calculate $\mathcal{L}_{Actor}(\phi)$ in Eq.(7) and update ϕ
end for

3 Experiments

3.1 Experimental Setup

We evaluate the proposed method using the D4RL benchmark proposed in Fu et al. [2020], which provides a diverse set of datasets designed for data-driven deep RL. Our experiments focus on three widely used Mujoco locomotion tasks: *HalfCheetah*, *Walker2d*, and *Hopper*. For comparative analysis, we use three types of datasets: *medium*, *medium-replay*, and *medium-expert*. The *medium* dataset consists of samples from a medium-level policy trained using the Soft Actor-Critic (SAC) algorithm by Haarnoja et al. [2018]. The *medium-replay* dataset includes all samples encountered during the training of the medium-level policy, and the *medium-expert* dataset combines data from both the medium-level agent and expert demonstrations. All experiments are conducted using 9 task setups with the *v2* version of these datasets to ensure a standardized evaluation.

We compare the performance of our proposed method against two baseline algorithms, CQL and Off2OnRL. CQL serves as a strong baseline for offline RL, and its output is also used to initialize both Off2OnRL and our approach. To ensure a fair comparison, we utilize the official implementations of CQL[1] and Off2OnRL[2]. In terms of hyperparameters, we keep most settings consistent with the official Off2OnRL implementation to minimize confounding factors. We use $N = 4$ CQL agents as offline pre-trained models for $1,000$ epochs with different random seeds across all tasks. These pre-trained agents are then used to initialize both Off2OnRL and our method, which are trained for an additional 200 epochs (equivalent to $200,000$ environment steps).

Our method uses an uncertainty quantification parameter β to account for the uncertainty in action-value estimation. It helps guide the policy to be more conservative in uncertain regions of the state-action space. For experiments, we vary β over the values $[0.1, 0.01, 0.001, 0.0001]$, systematically exploring the effects of different uncertainty penalization levels on the policy performance.

3.2 Empirical Results

Table 1 compares the normalized average scores of four methods across three environments. UPQ consistently performs the best with an average score of 103.8. It shows its ability to handle uncertainty during online fine-tuning. This method achieves top scores across different environments and dataset types. We find that UPQ especially has advantages in the tasks with non-optimal datasets marked as *medium*, *medium-replay*, and *medium-expert*. Uncertainty quantification of UPQ provides a reliable mechanism for fine-tuning policies in an online manner. It highlights the benefits of combining Uncertainty quantification with online fine-tuning from offline pre-trained models.

[1] available at `https://github.com/aviral-kumar2907/CQL`
[2] available at `https://github.com/shlee94/Off2OnRL`

Table 1: Normalized average returns on Mujoco locomotion tasks. Results of CQL and Off2OnRL are obtained by reproduction with the 'v2' dataset of D4RL. The top score for each task is highlighted.

		CQL	Off2onRL	UPQ
Medium	HalfCheetah	43.9	83.2	**89.6**
	Hopper	61.3	101.4	**106.0**
	Walker2d	71.6	101.0	**108.3**
Medium Replay	HalfCheetah	44.3	86.1	**90.0**
	Hopper	52.8	**108.8**	106.3
	Walker2d	68.4	103.7	**111.1**
Medium Expert	HalfCheetah	11.6	92.1	**93.7**
	Hopper	39.7	78.4	**105.8**
	Walker2d	80.0	112.8	**123.8**
	Average	52.6	96.4	**103.8**

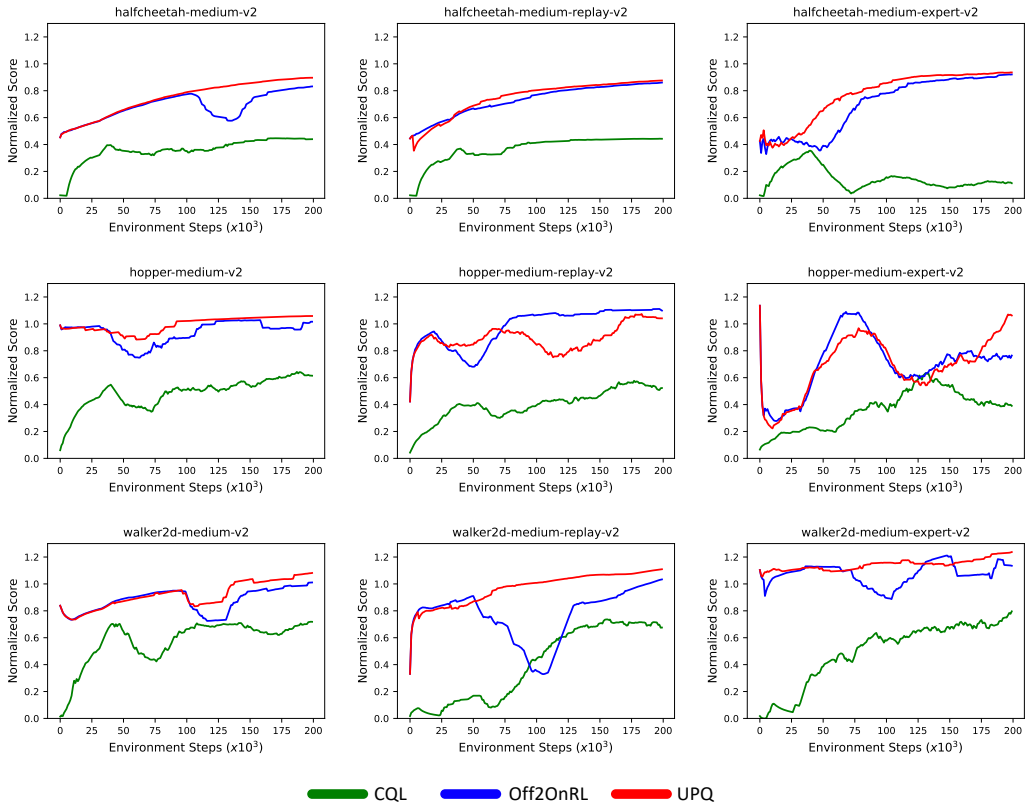

Figure 2: Normalized average score on Mujoco locomotion tasks with CQL, Off2OnRL, and UPQ.

The training curves in Figure 2 show the performance of CQL, Off2onRL, and UPQ across different tasks and datasets. It is clear from the results that UPQ consistently achieves the highest performance across almost all tasks, while Off2onRL also performs well but generally falls slightly behind UPQ. The results reveal that UPQ shows more stability across most tasks during the early stage of fine-tuning, while Off2onRL shows some performance drop within the first $100,000$ environment steps.

Overall, UPQ not only achieves superior final performance but also demonstrates more stable and consistent training across various datasets. The stability and sample efficiency, especially in the critical early phase of fine-tuning, highlight the effectiveness of UPQ's uncertainty quantification

Table 2: Normalized average returns on Mujoco locomotion tasks with UPQ variants with different values of $\beta = [0.1, 0.01, 0.001, 0.0001]$

		UPQ $\beta = 0.1$	UPQ $\beta = 0.01$	UPQ $\beta = 0.001$	UPQ $\beta = 0.0001$
Medium	HalfCheetah	85.7	**89.6**	88.2	77.5
	Hopper	105.6	103.9	**105.8**	**106.0**
	Walker2d	94.9	103.1	**108.3**	95.1
Medium Replay	HalfCheetah	83.8	87.6	85.3	**90.0**
	Hopper	98.8	99.2	**106.3**	105.3
	Walker2d	98.1	103.5	**111.1**	96.1
Medium Expert	HalfCheetah	93.0	93.1	**93.7**	92.5
	Hopper	68.3	80.4	**105.8**	99.9
	Walker2d	120.3	**123.8**	118.6	122.3

approach in ensuring more reliable learning during online fine-tuning for offline pre-trained policies. The early convergence observed in UPQ suggests that it is better suited for environments where rapid online fine-tuning is crucial for optimal policy performance.

Table 2 shows the performance of the UPQ algorithm across different environments under various datasets as the uncertainty penalization parameter β is varied. As we observe, changing β has a noticeable impact on performance. The results indicate that the UPQ algorithm generally achieves its best performance when the uncertainty penalization parameter β is around 0.001. This suggests that a β value near 0.001 strikes a good balance between conservatism and exploration, allowing the algorithm to manage uncertainty effectively while maintaining high performance across different environments.

4 Conclusion

In this work, we introduced a method for improving the stability and performance of online fine-tuning for offline pre-trained agents through uncertainty quantification. Our approach leverages an ensemble of pessimistic Q-functions, incorporating uncertainty estimates to guide the learning process more effectively during online fine-tuning. Our experiments on several Mujoco locomotion tasks demonstrate that our method consistently outperforms existing methods in terms of sample efficiency and final performance. By addressing the limitations of existing fine-tuning methods, UPQ enhances both the robustness and efficiency of online learning, making it a promising solution for real-world applications where safe and efficient policy fine-tuning is critical. Future work could explore adaptive uncertainty penalization for further improvements in generalization over different tasks and applying UPQ to more complex, real-world tasks beyond simulation environments.

Acknowledgments and Disclosure of Funding

This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government. [24ZR1100, A Study of Hyper Connected Thinking Internet Technology by autonomous connecting, controlling and evolving ways].

References

Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, 2020.

Chenjia Bai, Lingxiao Wang, Zhuoran Yang, Zhi-Hong Deng, Animesh Garg, Peng Liu, and Zhaoran Wang. Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning. In *International Conference on Learning Representations*, 2022.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning, 2020.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2052–2062, 2019.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018. URL `https://arxiv.org/abs/1801.01290`.

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. In *Advances in Neural Information Processing Systems*, 2019.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 1179–1191, 2020.

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, 2022.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 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. In *Advances in Neural Information Processing Systems*, pages 14129–14142, 2020.