
A IMPLEMENTATION

We applied both SO2 and SUF to CQL and IQL, respectively. Below are the implementation details:

Implementation of SO2. Excluding the Q-ensemble, SO2 relies on two key techniques: (1) Perturbed Value Update, and (2) increased Frequency of Q-value Update.

In CQL, the loss function is defined as:

$$\mathcal{L}_{\text{CQL}}(Q) = \alpha \cdot \mathbb{E}_{s \sim \mathcal{D}} [\log \sum_{a'} \exp(Q(s, a')) - Q(s, a)] + \frac{1}{2} \cdot \mathbb{E}_{(s, a, s') \sim \mathcal{D}} \left[\left(Q(s, a) - \hat{B}^\pi \hat{Q}_{\text{target}}(s, a) \right)^2 \right]. \quad (1)$$

The Bellman backup for the target Q-function is expressed as:

$$\hat{B}^\pi \hat{Q}_{\text{target}}(s, a) = r + \gamma \cdot \left(\hat{Q}_{\text{target}}(s', a' + \epsilon) - \beta \log \pi(a'|s') \right). \quad (2)$$

In IQL, we apply noise to the value loss as follows:

$$\mathcal{L}_{\text{IQL}}(V) = \mathbb{E}_{(s, a) \sim \mathcal{D}} \left[L_\tau^2(Q(s, a + \epsilon) - V(s)) \right]. \quad (3)$$

The value network then affects the Q-value loss through the following equation:

$$\mathcal{L}_{\text{IQL}}(Q) = \mathbb{E}_{(s, a, s') \sim \mathcal{D}} \left[\left(r(s, a) + \gamma V(s') - Q(s, a) \right)^2 \right]. \quad (4)$$

The noise term, ϵ , is defined as:

$$\epsilon \sim \text{clip}(\mathcal{N}(0, \sigma), -c, c). \quad (5)$$

Following the instructions in the SO2 (Zhang et al., 2024), we set $\sigma = 0.3$, $c = 0.6$, and the update frequency $N_{\text{upc}} = 10$.

Implementation of SUF. This method increases the Critic UTD to expedite the fitting of the value network, addressing the issue of value network underfitting on out-of-distribution (OOD) data and reducing estimation bias. Simultaneously, it decreases the Actor UTD to improve the accuracy of policy updates and mitigate misguidance caused by value bias. In the case of IQL, the value network is updated together with the critic, ensuring consistent learning dynamics between the two components. Following the instructions in the SUF (Feng et al., 2024), we set $G_c = 20$ and $G_a = 1/4$.

B EFFECT ON DATASET SIZE

We evaluate BAQ on MuJoCo tasks from the D4RL-v2 dataset¹, which includes three environments: HalfCheetah, Walker2d, and Hopper. Each environment contains datasets collected by policies of varying quality, categorized as Medium, Medium-Replay, and Medium-Expert. To effectively model the behavior of offline data, we first train a Behavior Cloning (BC) model on the offline datasets. The size of these datasets significantly impacts the performance of our algorithm, as shown in Table 1.

Dataset	Size
halfcheetah-medium-expert-v2	1,998,000
hopper-medium-expert-v2	1,998,966
walker2d-medium-expert-v2	1,998,318
halfcheetah-medium-replay-v2	201,798
hopper-medium-replay-v2	401,598
walker2d-medium-replay-v2	301,698
halfcheetah-medium-v2	999,000
hopper-medium-v2	999,998
walker2d-medium-v2	999,322

Table 1: Dataset sizes for different environments.

¹<https://github.com/Farama-Foundation/D4RL>

As shown in Table 1, the Medium-Expert datasets are approximately 2×10^6 in size, whereas the Medium-Replay datasets are considerably smaller. The performance scores for Medium-Expert and Medium-Replay are illustrated in Fig. 1 for CQL + Ours and Fig. 2 for IQL + Ours. Notably, both approaches exhibit the same optimal parameters: $(k_q = 1, k_\rho = 2)$ for the larger datasets (Medium-Expert) and $(k_q = 2, k_\rho = 1)$ for the smaller datasets (Medium-Replay).

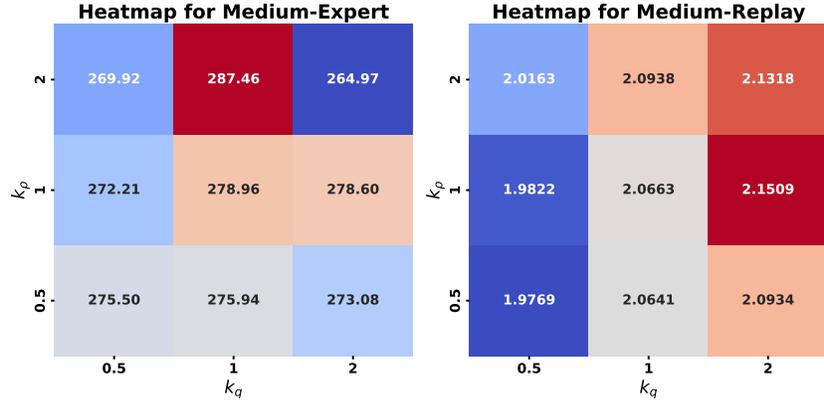


Figure 1: Heatmaps showing the relationship between k_ρ and k_q in CQL + Ours settings.

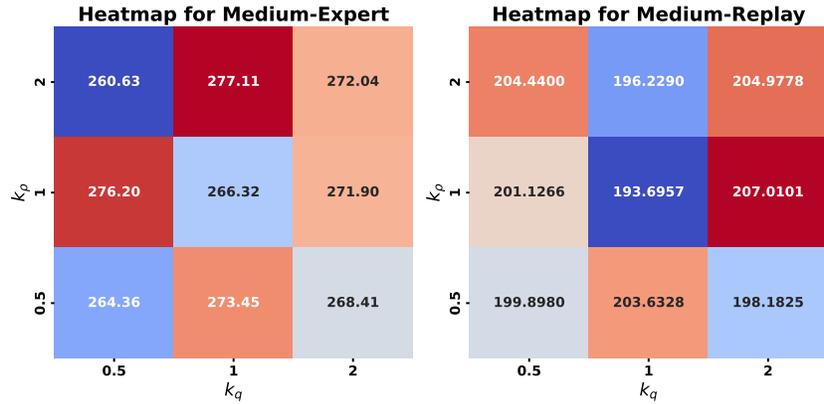


Figure 2: Heatmaps showing the relationship between k_ρ and k_q in IQL + Ours settings.

REFERENCES

- Jiaheng Feng, Mingxiao Feng, Haolin Song, Wengang Zhou, and Houqiang Li. Suf: Stabilized unconstrained fine-tuning for offline-to-online reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11961–11969, 2024.
- Yinmin Zhang, Jie Liu, Chuming Li, Yazhe Niu, Yaodong Yang, Yu Liu, and Wanli Ouyang. A perspective of q-value estimation on offline-to-online reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 16908–16916, 2024.