Enhancing Robustness of Offline Reinforcement Learning Under Data Corruption via Sharpness-Aware Minimization (Student Abstract)

Le Xu[1], **Jiayu Chen**[2]
[1]Tsinghua University
[2]The University of Hong Kong
xu-l22@mails.tsinghua.edu.cn, jiayuc@hku.hk

Abstract

Offline reinforcement learning (RL) is vulnerable to real-world data corruption, with even robust algorithms failing under challenging observation and mixture corruptions. We posit this failure stems from data corruption creating sharp minima in the loss landscape, leading to poor generalization. To address this, we are the first to apply Sharpness-Aware Minimization (SAM) as a general-purpose, plug-and-play optimizer for offline RL. SAM seeks flatter minima, guiding models to more robust parameter regions. We integrate SAM into strong baselines for data corruption: IQL, a top-performing offline RL algorithm in this setting, and RIQL, an algorithm designed specifically for data-corruption robustness. We evaluate them on D4RL benchmarks with both random and adversarial corruption. Our SAM-enhanced methods consistently and significantly outperform the original baselines. Visualizations of the reward surface confirm that SAM finds smoother solutions, providing strong evidence for its effectiveness in improving the robustness of offline RL agents.

1 Introduction

Offline reinforcement learning (RL) learns policies from static datasets (Levine et al. 2020), a vital paradigm for real-world applications where online interaction is infeasible. We frame this problem within a Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where the goal is to learn a policy $\pi(a|s)$ from a static, and potentially corrupted, dataset $\mathcal{D}$. A major challenge is the prevalence of data corruption, which can severely degrade performance.

Our work builds upon Implicit Q-Learning (IQL) (Kostrikov, Nair, and Levine 2021), a powerful offline algorithm that learns a Q-function and a value function V via expectile regression, and then extracts a policy through advantage-weighted behavioral cloning. Due to this design, IQL has shown notable resilience to some forms of data corruption. However, even IQL and its robust variant, RIQL (Yang et al. 2024), exhibit significant performance drops under observation and mixture corruptions.

We hypothesize this vulnerability arises because data corruption creates sharp, unreliable minima in the loss landscape. To this end, we are the first to apply Sharpness-Aware Minimization (SAM) (Foret et al. 2020) as a general-purpose optimizer for offline RL. SAM seeks flat minima, which are known to improve generalization and robustness. By integrating SAM, we guide training towards more stable solutions without altering the core logic of the base algorithm. Our experiments demonstrate that this approach consistently enhances the performance of IQL and RIQL on challenging D4RL benchmarks under both random and adversarial corruption, with reward surface visualizations confirming that SAM finds smoother, more robust solutions.

2 Methodology

Our core proposal is to integrate Sharpness-Aware Minimization (SAM) as a plug-and-play optimization module for offline RL agents. This approach is motivated by the hypothesis that data corruption induces sharp minima in the value function's loss landscape. Models converging to these sharp minima tend to be less robust, as small perturbations in the input data can lead to large errors in value estimation, ultimately degrading policy performance. Instead of altering the algorithm's loss functions, we replace the standard optimizer (e.g., Adam) with a SAM wrapper to explicitly seek out flatter, more generalizable solution regions.

SAM Optimization Process. The SAM optimizer seeks parameters θ that lie in neighborhoods with uniformly low loss. It achieves this via a two-step minimax procedure for a given loss function $L(\theta)$:

1. **First step (ascent):** It computes the gradient $\nabla_\theta L(\theta)$ to find an adversarial weight perturbation $\hat{\epsilon}(\theta)$ that locally maximizes the loss, effectively probing the sharpness of the landscape. It then ascends to the perturbed weights $\theta' = \theta + \hat{\epsilon}(\theta)$.

$$\hat{\epsilon}(\theta) = \rho \frac{\nabla_\theta L(\theta)}{\|\nabla_\theta L(\theta)\|_2} \tag{1}$$

2. **Second step (descent):** It computes the gradient $\nabla_\theta L(\theta')$ at this "worst-case" perturbed point and uses this gradient to update the original parameters θ.

This process penalizes sharpness by minimizing the highest loss value within the local neighborhood, forcing the optimizer to find flat regions where the loss remains low even after perturbation.

Integration with Value Function Learning. We implement SAM as a custom PyTorch 'Optimizer' class that wraps a base optimizer like Adam, allowing for seamless integration. Based on ablation studies (see Appendix), we found that applying SAM exclusively to the value function network yields the most significant and stable improvements. Therefore, in our main experiments, the standard update step for the value function's parameters is replaced by SAM's two-step procedure, ensuring that the learned value function is robust to perturbations in its parameter space.

3 Experiments

We conduct experiments to answer: Can applying SAM to SOTA offline RL algorithms improve their robustness against observation and mixture corruption?

Experimental Setup. We evaluate on three continuous control environments from the D4RL benchmark (Fu et al. 2021): 'halfcheetah-v2', 'walker2d-v2', and 'hopper-v2', using the 'medium-replay' datasets for each. We test on two challenging corruption types: **Observation** and **Mixture**, under both **random** and **adversarial** settings. Our base algorithms are IQL and RIQL. All corruptions are applied to 30% of the dataset. Detailed corruption methods are described in the Appendix. All experiments are conducted over three random seeds, and we report the mean and standard deviation of the normalized scores.(Fu et al. 2021)

Main Results. Tables 1 and 2 show our main results. Under both random and adversarial corruption, applying SAM consistently improves the performance of both IQL and RIQL across all environments and corruption settings. The average performance gains are significant, demonstrating that SAM is an effective technique for enhancing robustness. The improvements are particularly pronounced in the more challenging Mixture corruption setting, validating SAM's ability to handle complex noise distributions.

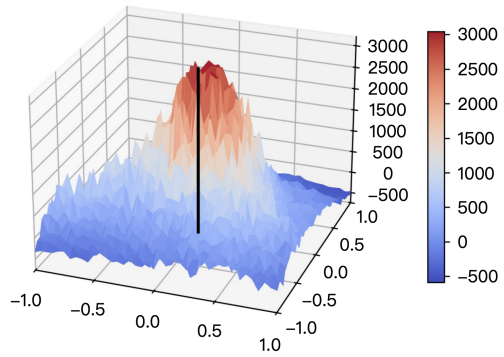
Reward Surface Visualization. To provide intuition, we visualize the reward surface, following the methodology of (Sullivan et al. 2022). Figure 1 shows surfaces for IQL and IQL+SAM models trained on 'halfcheetah' under random observation corruption. The standard IQL model converges to a region with sharp peaks and valleys, making it sensitive to perturbations. In contrast, IQL with SAM learns a smoother and flatter reward surface. This visually confirms that SAM guides the agent to a more robust solution.

Env.(Corruption)	IQL		RIQL	
	Naïve	SAM (Ours)	Naïve	SAM (Ours)
Halfcheetah (Obs)	21.01(3.01)	**33.33(1.49)**	26.03(3.61)	**33.74(2.48)**
Halfcheetah (Mix)	20.93(4.21)	**33.02(2.81)**	22.08(2.47)	**32.06(2.17)**
Walker2d (Obs)	24.74(7.10)	**31.75(8.87)**	30.48(11.82)	**30.93(9.56)**
Walker2d (Mix)	25.90(6.25)	**31.68(2.93)**	**26.92(10.90)**	19.55(17.04)
Hopper (Obs)	58.42(1.49)	**73.21(11.72)**	44.09(8.35)	**53.19(2.01)**
Hopper (Mix)	55.86(18.37)	**63.42(12.29)**	54.20(13.80)	**67.36(10.61)**
Average score ↑	34.47	**44.40**	33.97	**39.47**

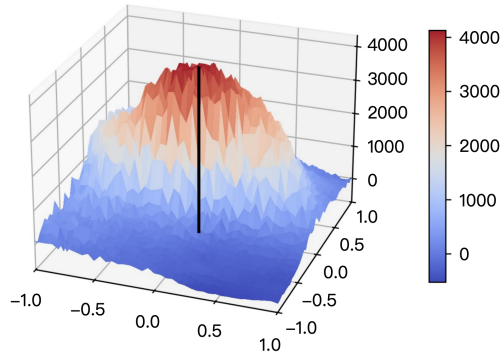
Table 1: Average Performance under random corruption

Env.(Corruption)	IQL		RIQL	
	Naïve	SAM (Ours)	Naïve	SAM (Ours)
Halfcheetah (Obs)	30.32(4.29)	**32.42(2.26)**	**38.46(0.71)**	35.92(3.10)
Halfcheetah (Mix)	12.68(0.48)	**16.72(1.28)**	**19.65(0.22)**	18.19(3.50)
Walker2d (Obs)	21.19(12.58)	**31.86(3.96)**	41.93(8.43)	**50.07(10.76)**
Walker2d (Mix)	12.31(1.25)	**12.61(4.27)**	19.61(3.33)	**21.22(8.85)**
Hopper (Obs)	41.88(1.29)	**67.22(10.41)**	49.12(2.99)	**65.34(15.09)**
Hopper (Mix)	16.33(3.40)	**55.34(27.87)**	**60.40(11.67)**	49.81(15.96)
Average score ↑	22.45	**36.03**	38.20	**40.09**

Table 2: Average Performance under adversarial corruption



(a) IQL



(b) IQL+SAM

Figure 1: Reward surface visualization on HalfCheetah with random observation corruption. (a) The landscape of standard IQL. (b) The smoother landscape produced by IQL+SAM.

4 Conclusion

We address the vulnerability of offline RL to data corruption by introducing Sharpness-Aware Minimization (SAM) as a plug-and-play optimizer. Applying SAM to strong baselines (IQL, RIQL) consistently and significantly improves performance on D4RL benchmarks under both random and adversarial corruption. Visualizations confirm SAM finds smoother reward surfaces, validating that optimizing for flatter minima is a promising direction for creating robust offline RL agents.

Acknowledgments

We would like to extend our special thanks to Professor Jeff Schneider from Carnegie Mellon University for generously providing the essential computational resources for this research.

References

Foret, P.; Kleiner, A.; Mobahi, H.; and Neyshabur, B. 2020. Sharpness-Aware Minimization for Efficiently Improving Generalization. *CoRR*, abs/2010.01412.

Fu, J.; Kumar, A.; Nachum, O.; Tucker, G.; and Levine, S. 2021. D4RL: Datasets for Deep Data-Driven Reinforcement Learning. arXiv:2004.07219.

Kostrikov, I.; Nair, A.; and Levine, S. 2021. Offline Reinforcement Learning with Implicit Q-Learning. arXiv:2110.06169.

Levine, S.; Kumar, A.; Tucker, G.; and Fu, J. 2020. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. arXiv:2005.01643.

Sullivan, R.; Terry, J. K.; Black, B.; and Dickerson, J. P. 2022. Cliff Diving: Exploring Reward Surfaces in Reinforcement Learning Environments. arXiv:2205.07015.

Yang, R.; Zhong, H.; Xu, J.; Zhang, A.; Zhang, C.; Han, L.; and Zhang, T. 2024. Towards Robust Offline Reinforcement Learning under Diverse Data Corruption. arXiv:2310.12955.