Q-SAM: Unlocking Sharpness-Aware Minimization for Generalization in Offline Reinforcement Learning

Da Wang[1], Yi Ma[1]*, Ting Guo[2], Lin Li[1], Wei Wei[1], Jiye Liang[1]

[1] School of Computer and Information Technology, Shanxi University.
[2] Data Science and Technology, North University of China.

Motivation: Why SAM in Offline RL?

- **Generalization is the bottleneck.** Offline RL policies are trained solely from static datasets and must perform reliably under distribution shift.
- **Optimizer choice is overlooked.** Existing methods focus on minimizing training loss (e.g., with Adam) but largely ignore the geometry of the loss landscape-specifically, sharpness.
- **SAM works in supervised learning** by seeking flatter minima, but **direct application to offline RL is non-trivial**:
    - No ground-truth labels; training relies on **bootstrapped targets**.
    - Bootstrapping creates a **non-stationary loss landscape**, making sharpness estimation noisy and optimization unstable.
- **The core conflict:** SAM's stationary objective clashes with the non-stationary nature of bootstrapped value estimation. Blindly pursuing flatness exacerbates **extrapolation errors** and **Bellman error accumulation.**

Proposed Method

We present **Q bound weighted Sharpness-Aware Minimization (Q-SAM)**, a robust framework tailored for offfine RL. Q-SAM leverages two mechanisms - **Weighted Sharpness and Weighted Samples** - to address the optimization challenges in offfine settings.

A. Weighted Sharpness (Taming the Sharpness Term)

Adopt the WSAM with a tunable hyperparameter

λ ∈ [0, 1): $\mathcal{L}_{WSAM}(\phi) := \frac{\lambda}{1-\lambda} g(\phi) + \mathcal{L}(\pi_\phi)$

B. Weighted Samples (Selecting Reliable Data for SAM)

Leverage Q-value bounds (DisCor) to quantify accumulated Bellman error: $\Delta_k = |Q_k - \mathcal{T}Q_{k-1}| + \gamma P^{\pi_{k-1}} \Delta_{k-1}$

Assign per-sample weights:

$$w_k(s,a) \propto \exp\left(-\frac{\gamma[P^{\pi_{k-1}}\Delta_{k-1}](s,a)}{\tau}\right)$$

Algorithm 1 Q-SAM: A Concise Version

Input: Offline dataset $\mathcal{D}$, Q-network Q_θ, error model Δ_φ, policy π_ϕ, parameters λ, ρ, α.

for iteration k in $\{1, \cdots, N\}$ **do**

Sample batch $\mathcal{B} = \{(s_i, a_i)\}_{i=1}^b \sim \mathcal{D}$;

Stage 1: Weighted Samples (for Critic)

Compute $w_k(s,a)$ using Equation (7);

Update Critic:

$\theta_{k+1} \leftarrow \operatorname{argmin}_\theta \frac{1}{N} \sum_i^N w_k(s_i,a_i)(Q_\theta(s_i,a_i) - y_i)^2$;

Stage 2: Weighted Sharpness (for Actor)

Compute the gradient g_t^{Adam} and the perturbed policy gradient g^{SAM} using the same weighted batch;

Update Actor:

$\phi_{k+1} = \phi_k - \alpha\left(\frac{\lambda}{1-\lambda} g_k^{SAM} + \frac{1-2\lambda}{1-\lambda} g_k^{Adam}\right)$;

end for

Experiments on the D4RL Benchmark

Main results (MuJoCo-v2 & AntMaze-v2)

Table 2. Average normalized scores over five seeds. We **bold** the highest mean.

Dataset	TD3BC	TD3BC + SAM	TD3BC + **Q-SAM**	CQL	CQL + SAM	CQL + **Q-SAM**	IQL	IQL + SAM	IQL + **Q-SAM**
halfcheetah-m	48.1	**48.5**	48.4±0.4	47.0	47.1	**47.3**±0.2	48.3	48.4	**49.2**±0.1
halfcheetah-mr	44.8	44.6	**45.3**±0.4	45.0	45.5	**45.6**±0.2	44.5	44.6	**45.1**±0.2
halfcheetah-me	90.8	90.3	**93.9**±4.6	95.6	94.1	95.6±0.5	94.7	95.0	**95.9**±0.4
hopper-m	60.4	60.9	**65.7**±2.6	59.1	59.7	**70.3**±2.5	67.5	68.5	**75.3**±2.2
hopper-mr	64.4	69.4	**87.3**±5.0	95.1	94.9	**95.5**±5.6	97.4	100.5	**103.0**±5.5
hopper-me	101.2	101.5	**105.1**±8.4	99.3	108.7	**109.8**±1.7	107.4	109.4	**113.4**±5.1
walker2d-m	82.7	82.9	**85.4**±1.0	80.8	81.7	**84.2**±1.2	80.9	86.9	**87.6**±3.0
walker2d-mr	85.6	86.7	**90.1**±1.0	73.1	74.9	**84.4**±4.2	82.2	87.2	**90.9**±4.7
walker2d-me	110.0	109.0	**111.5**±0.2	109.5	109.6	**112.3**±0.3	111.7	111.3	**111.7**±1.0
Total	688.0	693.8	**732.7**	704.5	716.2	**745.0**	734.6	751.8	**772.1**
antmaze-u	70.8	80.3	**92.8**±2.5	92.8	92.3	**93.3**±3.3	77.0	77.5	**81.3**±6.8
antmaze-ud	44.8	52.8	**66.3**±15.5	37.3	36.3	**38.3**±4.1	54.3	56.7	**65.9**±6.3
antmaze-mp	0.3	0.3	**3.8**±0.9	65.8	68.5	**71.8**±7.6	65.8	68.5	**80.8**±8.5
antmaze-md	0.3	0.3	**0.5**±0.3	67.3	68.1	**68.5**±2.5	73.8	75.9	**81.2**±5.5
antmaze-lp	0.0	0.0	0.0±0.0	20.8	28.0	**33.8**±6.2	42.0	48.5	**63.3**±7.6
antmaze-ld	0.0	0.0	0.0±0.0	20.5	23.3	**29.3**±6.4	30.3	30.5	**30.8**±3.9
Total	116.2	133.7	**163.4**	304.5	316.5	**335.0**	343.2	357.6	**403.3**

v.s SOTA generalization methods

Table 3. Comparison with several generalization methods.

Dataset	POR	DOGE	TSRL	IQL + Q-SAM
halfcheetah-m	48.8	45.3	48.2	**49.2 ± 0.1**
halfcheetah-mr	43.5	42.8	42.2	**45.1 ± 0.2**
halfcheetah-me	94.7	78.7	92.0	**95.9 ± 0.4**
hopper-m	78.6	**98.6**	86.7	75.3 ± 2.2
hopper-mr	98.9	76.2	78.7	**103.0 ± 5.5**
hopper-me	90.0	102.7	95.9	**113.4 ± 5.1**
walker2d-m	81.1	86.8	77.5	**87.6 ± 3.0**
walker2d-mr	76.6	87.3	66.1	**90.9 ± 4.7**
walker2d-me	109.1	110.4	109.8	**111.7 ± 1.0**
Total	721.3	728.8	697.1	**772.1**

- **Q-SAM consistently improves the performance of three strong baselines** (TD3BC, CQL, and IQL) and **achieves competitive results against several representative generalization methods**.

Validation of Q-SAM in Small-Sample Generalization

Small-Sample Generalization (5% Data)

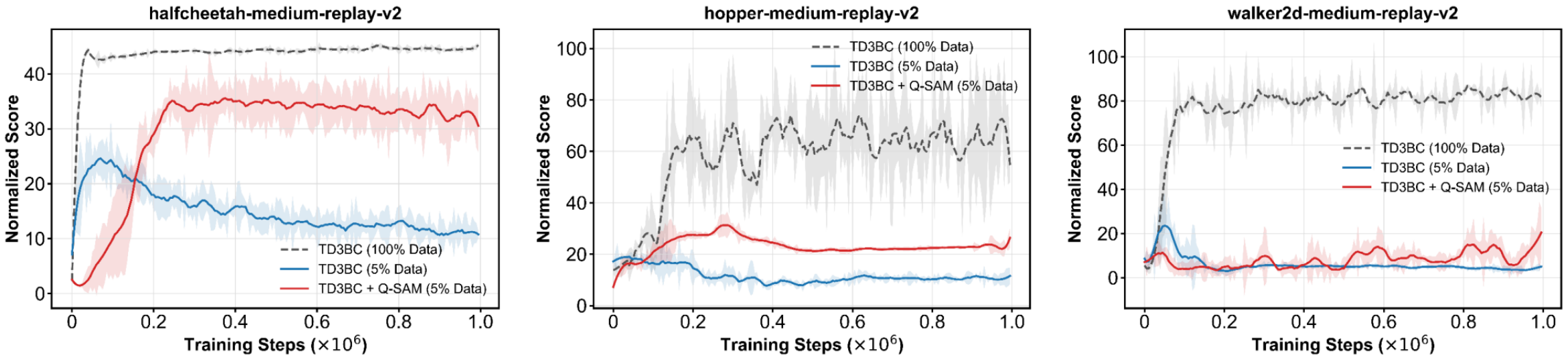

Figure 4. Performance comparison on full (100%) versus small-sample (5%) datasets.

- In extreme data-scarce settings, Q-SAM maintains competent performance where baselines collapse (e.g., halfcheetah-mr, hopper-mr), **demonstrating superior out-of-distribution generalization**.

Validation of Q-SAM in Bellman Error Reduction

Bellman Error Reduction

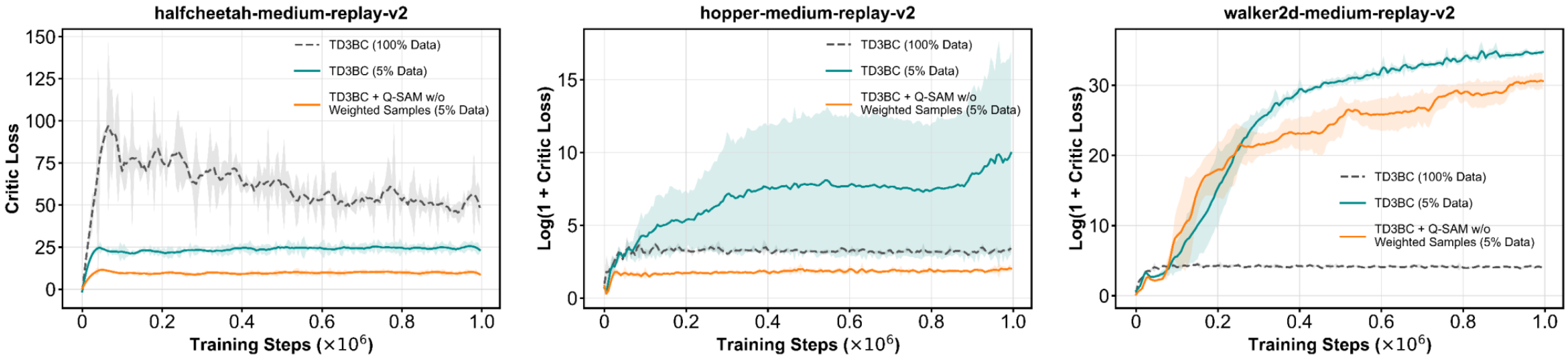

Figure 5. Critic loss values in the small-sample experiment.

- **Weighted Sharpness** significantly reduces Critic Loss (proxy for Bellman error), **validating that controlled flatness mitigates extrapolation error accumulation**.

Ablation Study and Runtime Fairness

Ablation Study

Table 4. Average normalized scores over five seeds. "Drop" means the decrease in mean values.

Dataset	CQL + Q-SAM	CQL + Q-SAM w/o Weighted Sharpness		CQL + Q-SAM w/o Weighted Samples	
		Score	Drop	Score	Drop
halfcheetah-m	47.3±0.2	47.1±0.2	0.2	47.2±0.2	0.1
halfcheetah-mr	45.6±0.2	45.4±0.2	0.2	45.7±0.2	-0.1
halfcheetah-me	95.6±0.5	95.6±0.4	0.0	95.9±0.4	-0.3
hopper-m	70.3±2.5	67.2±2.4	3.1	65.7±1.2	4.6
hopper-mr	95.5±5.6	95.4±5.4	0.1	95.5±5.3	0.0
hopper-me	109.8±1.7	107.5±2.5	2.3	109.6±1.7	0.2
walker2d-m	84.2±1.2	81.9±1.2	2.3	83.2±0.2	1.0
walker2d-mr	84.4±4.2	80.2±5.5	4.2	82.2±4.0	2.2
walker2d-me	112.3±0.3	110.5±0.3	1.8	111.5±0.3	0.8
Total	745.0±16.4	730.8±18.1	*14.2*	736.5±13.5	*8.5*

- Removing **Weighted Sharpness** causes a −14.2 total score drop (CQL); removing **Weighted Samples** causes a −8.5 drop. **Both components contribute synergistically**; sharpness weighting plays the larger role.

Runtime Fairness

Table 5. Comparison of running time and performance of different algorithms. (Time unit: minutes)

Algorithm	Steps	halfcheetah-mr		hopper-mr		walker2d-mr	
		Runtime	Score	Runtime	Score	Runtime	Score
TD3BC	1M	142.1	44.8±0.6	137.3	64.4±21.5	135.7	85.6±4.0
	2M	290.7	45.0±0.4	271.5	65.4±21.7	261.4	68.5±11.4
TD3BC + Q-SAM	1M	276.9	**45.3±0.4**	296.1	**87.3±5.0**	298.1	**90.1±1.0**

- Even with double training steps (2M), baseline TD3BC fails to match Q-SAM's 1M-step performance. The extra computation cost is justified by **a fundamental optimization geometry breakthrough**, not merely more training.

Conclusion

- **Conflict & Solution.** We reveal a fundamental conflict between sharpness minimization and non-stationary bootstrapping in offline RL, and propose Q-SAM to resolve it via Q-value-bound dynamic weighting of sharpness and samples, inducing policy smoothness without exacerbating extrapolation errors.
- **Performance.** Q-SAM achieves superior generalization and stability across diverse benchmarks, consistently improving TD3BC, CQL, and IQL.
- **Outlook.** Extending Q-SAM to value networks (requiring specialized regularization against Bellman error propagation) and dynamic λ scheduling based on value convergence rates or trajectory-level uncertainty are promising future directions.

Thanks