清華大學
Tsinghua University

Adaptive Neighborhood-Constrained Q Learning for Offline Reinforcement Learning

Yixiu Mao, Yun Qu, Qi Wang, Xiangyang Ji

Tsinghua University

2025.11

Constraints in Offline RL

- To mitigate extrapolation error from OOD actions, offline RL algorithms typically impose constraints on action selection, which can be systematically categorized into **density**, **support**, and **sample constraints**.

Table 1: A brief summary of constraint types in offline RL research.

Constraint type	Description	Algorithms	Key characteristics
Density	Enforce density proximity between the trained and behavior policies	BRAC [83], TD3BC [21], CQL [42]	Straightforward but heavily limited by the overall quality of behavior policy
Sample	Restrict action selection to dataset actions	IQL[40], XQL[25], SQL[88]	Avoid extrapolation error but lack action generalization beyond the dataset
Support	Restrict action selection to behavior policy's support	BCQ[23], BEAR[41], SPOT [82]	Least restrictive but require accurate behavior policy modeling
Neighborhood	Restrict action selection to certain neighborhoods of dataset actions	ANQ (Ours)	Flexible and approximate support constraint without behavior modeling

Constraints in Offline RL

- **Density Constraint**

  Definition 1 (Density constraint). *The trained policy satisfies the density constraint* $\mathrm{D}(\pi, \pi_\beta) \leq \epsilon$, *where* D *represents a divergence measure between the trained policy* π *and the behavior policy* π_β, *e.g., KL, total variation (TV), or Fisher divergence.*

- Description: explicitly or implicitly align the probability densities of trained and behavior policies
- Advantage: straightforward, easy to implement
- Limitation: performance is heavily limited by the overall quality of behavior policy
  - performance upper bound:

    Lemma 1 (Performance bound under density constraints). *If any of the conditions* $\mathrm{D}_{\mathrm{KL}}(\pi \| \pi_\beta) \leq 2\epsilon$, $\mathrm{D}_{\mathrm{KL}}(\pi_\beta \| \pi) \leq 2\epsilon$, *or* $\mathrm{D}_{\mathrm{TV}}(\pi, \pi_\beta) \leq \sqrt{\epsilon}$ *holds, then the policy performance* η *is bounded as follows:*

$$\eta(\pi) \leq \eta(\pi_\beta) + \frac{2R_{\max}}{(1-\gamma)^2}\sqrt{\epsilon}. \tag{3}$$

- Algorithms: BRAC, TD3BC, CQL, ReBRAC

Constraints in Offline RL

- **Support Constraint**

> **Definition 2** (Support constraint). *The selected action in the Bellman target is restricted to the support of the behavior policy, which is defined as* $\mathcal{C}_{\text{Supp}}(s) := \{a \in \mathcal{A} \mid \pi_\beta(a|s) > \epsilon\}$*, where* π_β *is the behavior policy and* ϵ *is a threshold that determines the support.*

- Description: restrict action selection in the Bellman target to the support of behavior policy
- Advantage: least restrictive
- Limitation: require accurate estimation of behavior policy, which is challenging
  - modeling high-dimensional and multi-modal real-world data
  - e.g., using CVAEs, autoregressive models, flow-GANs, and diffusion models
- Algorithms: BCQ, BEAR, SPOT, STR, CPED, SVR

Constraints in Offline RL

- **Sample Constraint**

Definition 3 (Sample constraint). *The selected action in the Bellman target is restricted to the sample set* $\mathcal{C}_{\text{Samp}}(s) := \{a \in \mathcal{A} \mid (s, a) \in \mathcal{D}\}$, *consisting of actions in the dataset for a given state* $s \in \mathcal{D}$.

- Description: restrict action selection in the Bellman target to dataset actions
- Advantage: avoid extrapolation errors, easy to implement
- Limitation: performance is inherently limited by a lack of action generalization beyond the offline dataset
  - overly conservative when the dataset lacks coverage of near-optimal actions
- Algorithms: Onestep RL, IQL, $\mathcal{X}$QL, IAC, SQL

Neighborhood Constraint

- This work aims to address the over-conservatism of the density and sample constraints, while avoiding complex behavior policy modeling required by the support constraint.
- Propose: **Neighborhood Constraint**

> **Definition 4** (Neighborhood constraint)**.** *The selected action in the Bellman target is restricted to the neighborhood set* $\mathcal{C}_{\mathrm{N}}(s) := \{\tilde{a} \in \mathcal{A} \mid \|\tilde{a} - a\| \leq \epsilon, (s, a) \in \mathcal{D}\}$*, which comprises actions located within the* ϵ*-neighborhoods of all dataset actions on a given state* $s \in \mathcal{D}$*.*

- Description: restrict action selection in the Bellman target to the union of neighborhoods of dataset actions.

Neighborhood Constraint

- Property: approximate the support constraint without behavior policy modeling

Theorem 1 (Support approximation via neighborhoods). *Let $S \subseteq \mathbb{R}^d$ be the compact support of a distribution ν, and let $X_1, \ldots, X_n$ be independent and identically distributed samples from ν. Define $U_{n,\epsilon} = \bigcup_{i=1}^{n} B(X_i, \epsilon)$ as the union of closed balls of radius ϵ centered at the samples. Let $\mathcal{N}(S, \epsilon/2)$ denote the covering number of S, i.e., the minimal number of $\epsilon/2$-balls required to cover S. Under the standardness Assumption 1 with constants $r_0, C_0 > 0$, for any $\delta \in (0,1)$ and $\epsilon \leq 2r_0$, if*

$$n \geq \frac{1}{C_0(\epsilon/2)^d} \left(\log \mathcal{N}(S, \epsilon/2) + \log(1/\delta)\right), \tag{6}$$

then with probability at least $1-\delta$, the Hausdorff distance between S and $U_{n,\epsilon}$ satisfies

$$d_H(S, U_{n,\epsilon}) := \max\left(\sup_{x \in S} \inf_{u \in U_{n,\epsilon}} d(x,u), \sup_{u \in U_{n,\epsilon}} \inf_{x \in S} d(x,u)\right) \leq \epsilon. \tag{7}$$

Neighborhood Constraint

- Property: extrapolation error

Lemma 2 (Extrapolation behavior). *Under the neural tangent kernel (NTK) regime [30], for any in-sample state-action pair $(s, a) \in \mathcal{D}$ and in-neighborhood state-action pair $(s, \tilde{a})$ such that $\|\tilde{a} - a\| \leq \epsilon$, the value difference of the deep Q function can be bounded as:*

$$\|Q_\theta(s, \tilde{a}) - Q_\theta(s, a)\| \leq C(\sqrt{\min(\|s \oplus a\|, \|s \oplus \tilde{a}\|)}\sqrt{\epsilon} + 2\epsilon), \tag{8}$$

where $\oplus$ denotes the vector concatenation operation, and C is a finite constant.

- Property: distribution shift

Proposition 1 (Distribution shift). *Let π_1 be a deterministic policy that satisfies the neighborhood constraint with threshold ϵ. Assume that the transition dynamics P is K_P-Lipschitz continuous: $\forall s \in \mathcal{S}, \forall a_1, a_2 \in \mathcal{A}, \|P(s'|s, a_1) - P(s'|s, a_2)\| \leq K_P\|a_1 - a_2\|$. Then, there exists a policy π_2 satisfying the sample constraint such that:*

$$\mathrm{D_{TV}}\left(d^{\pi_1}(\cdot), d^{\pi_2}(\cdot)\right) \leq \frac{\gamma K_P \epsilon}{2(1-\gamma)}, \tag{9}$$

where $d^\pi(s) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \mathbb{E}_\pi\left[\mathbb{I}\left[s_t = s\right]\right]$ is the state occupancy induced by policy π.

Neighborhood Constraint

- Summary: mitigate the over-conservatism of density and sample constraints, while approximating the least restrictive support constraint without behavior policy modeling

Table 1: A brief summary of constraint types in offline RL research.

Constraint type	Description	Algorithms	Key characteristics
Density	Enforce density proximity between the trained and behavior policies	BRAC [83], TD3BC [21], CQL [42]	Straightforward but heavily limited by the overall quality of behavior policy
Sample	Restrict action selection to dataset actions	IQL[40], XQL[25], SQL[88]	Avoid extrapolation error but lack action generalization beyond the dataset
Support	Restrict action selection to behavior policy's support	BCQ[23], BEAR[41], SPOT [82]	Least restrictive but require accurate behavior policy modeling
Neighborhood	Restrict action selection to certain neighborhoods of dataset actions	ANQ (Ours)	Flexible and approximate support constraint without behavior modeling

Adaptive Neighborhood Constraint

- The neighborhood constraint is highly flexible and can achieve pointwise conservatism by adapting neighborhood radius for each data point → **Adaptive Neighborhood Constraint**
- Advantage-based instantiation

> **Definition 5** (Adaptive neighborhood constraint). *The selected action in the Bellman target is restricted to the adaptive neighborhood set* $\mathcal{C}_{\mathrm{AN}}(s) := \{\tilde{a} \in \mathcal{A} \mid \|\tilde{a} - a\| \leq \epsilon \exp(-\alpha A(s,a)), (s,a) \in \mathcal{D}\}$, *where* A *denotes the advantage function and* α *is an inverse temperature parameter that modulates the sensitivity of the neighborhood radius to advantage values.*

  - larger neighborhood radii to dataset actions with low advantage, promoting a broader search
  - smaller neighborhood radii to dataset actions with high advantage, reducing extrapolation error
- Generic adaptive neighborhood
  - replace $\epsilon \exp(-\alpha A(s,a))$ with $\epsilon f(s,a)$ to define a generic per-sample neighborhood radius, where $f: S \times A \to \mathbb{R}^+$ is an arbitrary function.

Algorithm: Adaptive Neighborhood-Constrained Q Learning (ANQ)

- Goal: Q learning under the adaptive neighborhood constraint → minimize the ANQ loss:

$$L_{\mathrm{ANQ}}(\theta) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\left[\left(Q_\theta(s,a) - R(s,a) - \gamma \max_{a'\in\mathcal{C}_{\mathrm{AN}}(s')} Q_{\theta'}(s',a')\right)^2\right]$$

$$\mathcal{C}_{\mathrm{AN}}(s) := \{\tilde{a}\in\mathcal{A} \mid \|\tilde{a}-a\| \le \epsilon \exp(-\alpha A(s,a)), (s,a)\in\mathcal{D}\}$$

- Decompose the objective into bilevel optimization

$$\max_{a\in\mathcal{C}_{\mathrm{AN}}(s)} Q(s,a),\ \forall s\in\mathcal{D} \iff \begin{array}{l} \max\limits_{a\in\mathcal{D}(s)} Q(s,a+\delta_{sa}),\ \forall s\in\mathcal{D} \\ \text{s.t. } \delta_{sa} = \operatorname{argmax}\limits_{\|\delta\|\le\epsilon\exp(-\alpha A(s,a))} Q(s,a+\delta),\ \forall (s,a)\in\mathcal{D} \end{array}$$

Algorithm: Adaptive Neighborhood-Constrained Q Learning (ANQ)

- Bilevel objective:

$$\max_{a \in \mathcal{D}(s)} Q(s, a + \delta_{sa}),\ \forall s \in \mathcal{D}$$
$$\text{s.t. } \delta_{sa} = \operatorname*{argmax}_{\|\delta\| \le \epsilon \exp(-\alpha A(s,a))} Q(s, a + \delta),\ \forall (s, a) \in \mathcal{D}$$

- Inner maximization
  - maximize Q function within each dataset action's neighborhood separately with an auxiliary policy

$$\max_{\mu_\omega} Q_\theta(s, a + \mu_\omega(s, a)) \ \text{ s.t. } \ \exp(\alpha A(s, a)) \|\mu_\omega(s, a)\| \le \epsilon,\ \forall (s, a) \in \mathcal{D}$$

$$\Longrightarrow \max_{\mu_\omega} \mathbb{E}_{(s,a) \sim \mathcal{D}} \left[Q_\theta(s, a + \mu_\omega(s, a)) - \lambda \exp(\alpha (Q_{\theta'}(s, a) - V_\psi(s))) \|\mu_\omega(s, a)\| \right]$$

- Outer optimization
  - implicitly maximize the Q function over all available neighborhoods via expectile regression

$$\min_{V_\psi} \mathop{\mathbb{E}}_{(s,a) \sim \mathcal{D}} \left[L_2^\tau \left(Q_{\theta'}(s, a + \mu_{\omega'}(s, a)) - V_\psi(s) \right) \right]$$

$$\text{where } L_2^\tau(x) = |\tau - \mathbb{1}(x < 0)| x^2,\ \tau \in (0, 1)$$

Algorithm: Adaptive Neighborhood-Constrained Q Learning (ANQ)

- ANQ loss
  - For $\tau \approx 1$, $V_\psi(s)$ captures the maximum Q value within $C_{\text{AN}}(s) \rightarrow$ the ANQ loss:

$$\min_{Q_\theta} \mathbb{E}_{(s,a,s')\sim\mathcal{D}} \left[\left(Q_\theta(s,a) - R(s,a) - \gamma V_\psi(s') \right)^2 \right]$$

- Final policy extraction
  - Independently extract final policy π_ϕ via weighted regression toward optimized actions within $C_{\text{AN}}(s)$

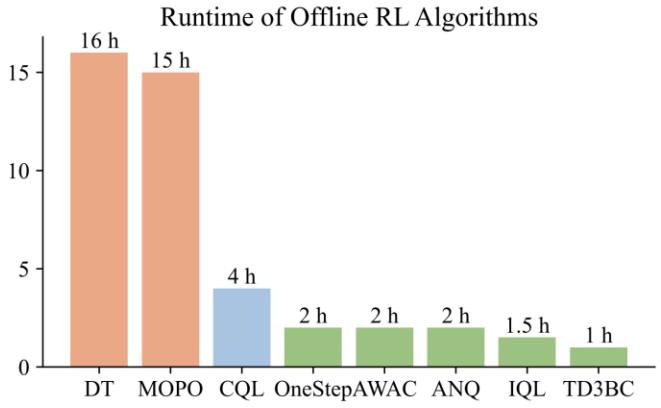
$$\min_{\pi_\phi} \mathbb{E}_{(s,a)\sim\mathcal{D}} \exp(\beta(Q_{\theta'}(s, a + \mu_\omega(s,a)) - V_\psi(s))) \| a + \mu_\omega(s,a) - \pi_\phi(s) \|_2^2$$

Experiment: Results on D4RL Benchmark

Table 2: Averaged normalized scores on Gym locomotion and Antmaze tasks over five random seeds. m = medium, m-r = medium-replay, m-e = medium-expert, e = expert, r = random; u = umaze, u-d = umaze-diverse, m-p = medium-play, m-d = medium-diverse, l-p= large-play, l-d = large-diverse.

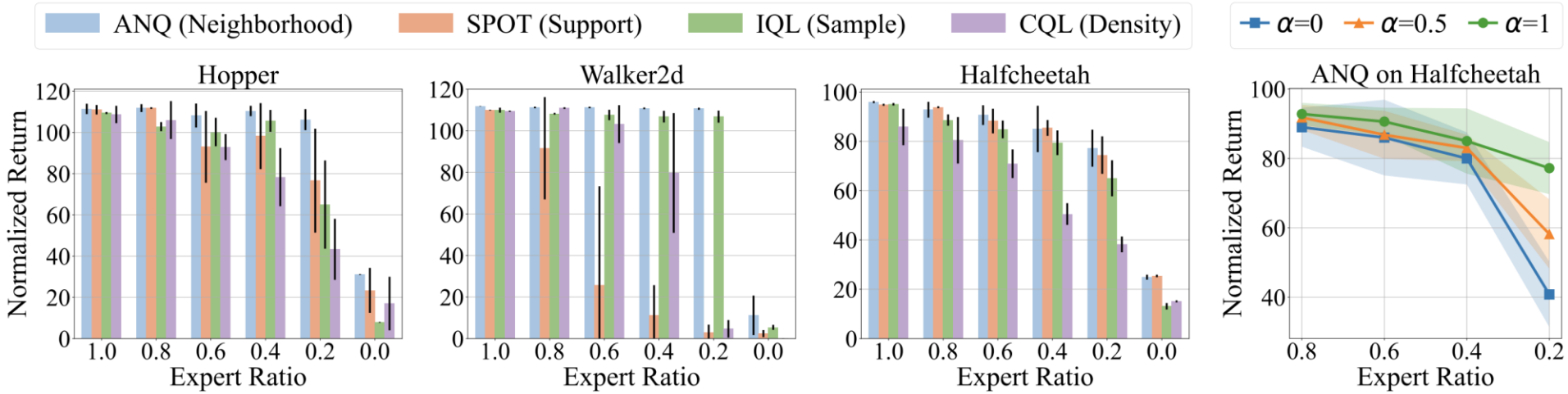
Dataset-v2	BCQ	BEAR	DT	AWAC	OneStep	TD3BC	CQL	IQL	SPOT	ANQ (Ours)
halfcheetah-m	46.6	43.0	42.6	47.9	50.4	48.3	47.0	47.4	58.4	**61.8±1.4**
hopper-m	59.4	51.8	67.6	59.8	87.5	59.3	53.0	66.2	86.0	**100.9±0.6**
walker2d-m	71.8	-0.2	74.0	83.1	84.8	83.7	73.3	78.3	**86.4**	82.9±1.5
halfcheetah-m-r	42.2	36.3	36.6	44.8	42.7	44.6	45.5	44.2	52.2	**55.5±1.4**
hopper-m-r	60.9	52.2	82.7	69.8	98.5	60.9	88.7	94.7	**100.2**	**101.5±2.7**
walker2d-m-r	57.0	7.0	66.6	78.1	61.7	81.8	81.8	73.8	**91.6**	**92.7±3.8**
halfcheetah-m-e	**95.4**	46.0	86.8	64.9	75.1	90.7	75.6	86.7	86.9	**94.2±0.8**
hopper-m-e	**106.9**	50.6	**107.6**	100.1	**108.6**	98.0	105.6	91.5	99.3	**107.0±4.9**
walker2d-m-e	107.7	22.1	108.1	110.0	**111.3**	110.1	107.9	109.6	**112.0**	**111.7±0.2**
halfcheetah-e	89.9	92.7	87.7	81.7	88.2	**96.7**	**96.3**	**95.0**	94.8	**95.9±0.4**
hopper-e	109.0	54.6	94.2	109.5	106.9	107.8	96.5	109.4	**111.0**	**111.4±2.5**
walker2d-e	106.3	106.6	108.3	**110.1**	**110.7**	**110.2**	108.5	109.9	109.9	**111.8±0.1**
halfcheetah-r	2.2	2.3	2.2	6.1	2.3	11.0	17.5	13.1	**25.4**	**24.9±1.0**
hopper-r	7.8	3.9	5.4	9.2	5.6	8.5	7.9	7.9	23.4	**31.1±0.2**
walker2d-r	4.9	**12.8**	2.2	0.2	6.9	1.6	5.1	5.4	2.4	11.2±9.5
locomotion total	968.0	581.7	972.6	975.3	1041.2	1013.2	1010.2	1033.1	1139.9	**1194.5**
antmaze-u	78.9	73.0	54.2	80.0	54.0	73.0	82.6	89.6	**93.5**	**96.0±1.6**
antmaze-u-d	55.0	61.0	41.2	52.0	57.8	47.0	10.2	65.6	40.7	**80.2±1.8**
antmaze-m-p	0.0	0.0	0.0	0.0	0.0	0.0	59.0	**76.4**	**74.7**	**76.2±3.3**
antmaze-m-d	0.0	8.0	0.0	0.2	0.6	0.2	46.6	72.8	**79.1**	**77.2±6.1**
antmaze-l-p	6.7	0.0	0.0	0.0	0.0	0.0	16.4	42.0	35.3	**56.2±4.9**
antmaze-l-d	2.2	0.0	0.0	0.0	0.2	0.0	3.2	46.0	36.3	**55.8±4.0**
antmaze total	142.8	142.0	95.4	132.2	112.6	120.2	218.0	392.4	359.6	**441.6**

Dataset-v2	IAC	SQL	EQL	STR	CPI	CPED	SVR	ANQ (Ours)
halfcheetah-m	51.6±0.3	48.3±0.2	47.2±0.3	51.8±0.3	**64.4±1.3**	61.8±1.6	60.5±1.2	61.8±1.4
hopper-m	74.6±11.5	75.5±3.4	70.6±2.6	**101.3±0.4**	98.5±3.0	**100.1±2.8**	**103.5±0.4**	**100.9±0.6**
walker2d-m	85.2±0.4	84.2±4.6	83.2±4.4	85.9±1.1	85.8±0.8	90.2±1.7	**92.4±1.2**	82.9±1.5
halfcheetah-m-r	47.2±0.3	44.8±0.7	44.5±0.5	47.5±0.2	54.6±1.3	**55.8±2.9**	52.5±3.0	**55.5±1.4**
hopper-m-r	**103.2±1.0**	**101.7±3.3**	98.1±3.6	100.0±1.2	**101.7±1.6**	98.1±2.1	**103.7±1.3**	**101.5±2.7**
walker2d-m-r	**93.2±1.8**	77.2±3.8	81.6±4.2	85.7±2.2	91.8±2.9	91.9±0.9	**95.6±2.5**	92.7±3.8
halfcheetah-m-e	92.9±0.7	**94.0±0.4**	**94.6±0.5**	**94.9±1.6**	**94.7±1.1**	85.4±10.9	**94.2±2.2**	**94.2±0.8**
hopper-m-e	109.3±4.0	**111.8±2.2**	**111.5±2.1**	**111.9±0.6**	106.4±4.3	95.3±13.5	**111.2±0.9**	107.0±4.9
walker2d-m-e	110.1±0.1	110.0±0.8	110.2±0.8	110.2±0.1	110.9±0.4	**113.0±1.4**	109.3±0.2	**111.7±0.2**
halfcheetah-e	94.5±0.5	-	-	95.2±0.3	**96.5±0.2**	-	**96.1±0.7**	**95.9±0.4**
hopper-e	110.6±1.9	-	-	**111.2±0.3**	**112.2±0.5**	-	**111.1±0.4**	**111.4±2.5**
walker2d-e	**114.8±1.2**	-	-	110.1±0.1	110.6±0.1	-	110.0±0.2	111.8±0.1
halfcheetah-r	20.9±1.2	-	-	20.6±1.1	**29.7±1.1**	-	27.2±1.2	24.9±1.0
hopper-r	**31.3±0.3**	-	-	**31.3±0.3**	29.5±3.7	-	**31.0±0.3**	**31.1±0.2**
walker2d-r	3.0±1.3	-	-	4.7±3.8	5.9±1.7	-	2.2±1.5	**11.2±9.5**
locomotion total	1142.4	-	-	1162.2	1193.2	-	**1200.5**	1194.5
antmaze-u	77.6±3.8	92.2±1.4	93.2±2.2	93.6±4.0	**98.8±1.1**	**96.8±2.6**	-	**96.0±1.6**
antmaze-u-d	71.2±8.6	74.0±2.3	70.4±2.7	77.4±7.2	**88.6±5.7**	55.6±2.2	-	80.2±1.8
antmaze-m-p	72.0±7.6	80.2±3.7	77.5±4.3	82.6±5.4	82.4±5.8	**85.1±3.4**	-	76.2±3.3
antmaze-m-d	74.2±4.1	75.1±4.2	74.0±3.7	**87.0±4.2**	80.4±8.9	72.1±2.9	-	77.2±6.1
antmaze-l-p	**57.0±7.4**	50.2±4.8	45.6±4.2	42.8±8.7	20.6±16.3	34.9±5.3	-	**56.2±4.9**
antmaze-l-d	47.2±9.4	**52.3±5.2**	49.5±4.7	46.8±7.6	45.2±6.9	32.3±7.4	-	**55.8±4.0**
antmaze total	399.2	424.0	410.2	430.2	416.0	376.8	-	**441.6**



Experiment: Results on Noisy Data

- Noisy data: mix the random and expert datasets at varying ratios

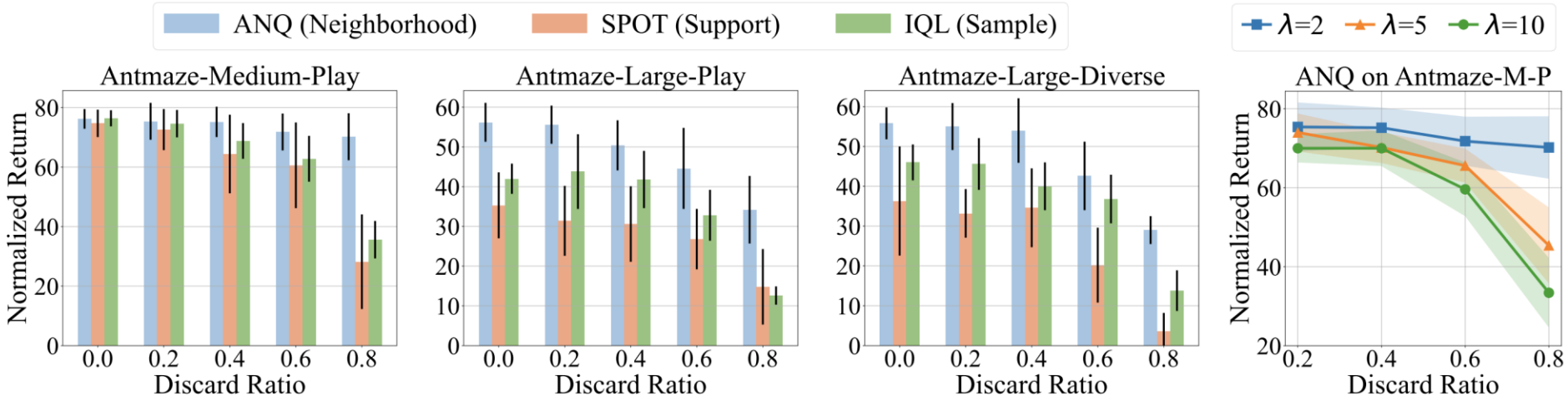


(a) Evaluation of algorithms with various constraint types on noisy datasets (b) ANQ with varying α

Figure 1: (a) Evaluation on noisy datasets over five random seeds. (b) Evaluation of ANQ on noisy datasets with varying inverse temperature α that determines the adaptiveness of neighborhood radius.

Experiment: Results on Limited Data

- Limited data: randomly discard some portion of transitions from the AntMaze datasets



(a) Evaluation of algorithms with various constraint types on reduced datasets (b) ANQ with varying λ

Figure 2: (a) Evaluation on reduced datasets over five random seeds. (b) Evaluation of ANQ on reduced datasets with varying Lagrange multiplier λ that controls the overall radius of neighborhoods.

Experiment: Ablation Study

- Lagrange multiplier λ: controls the overall neighborhood radius

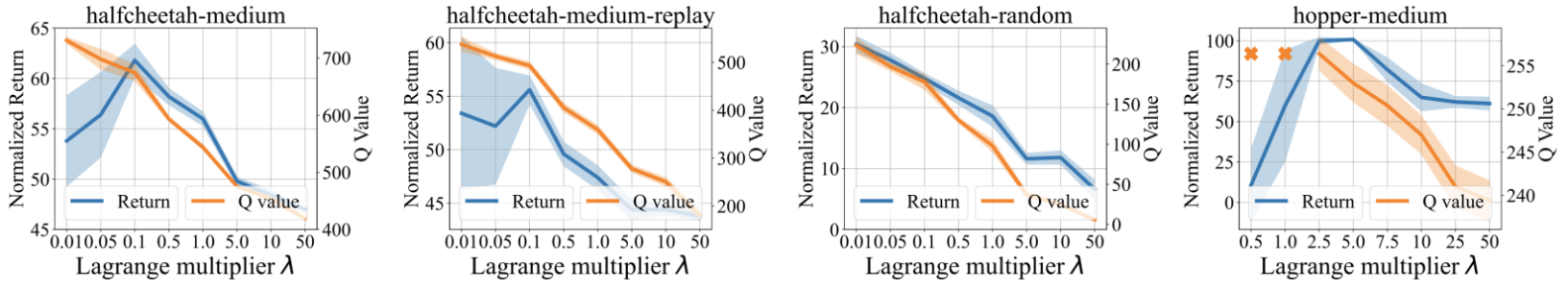


Figure 3: Performance and Q values of ANQ with varying Lagrange multiplier λ over five random seeds. The crosses $\times$ mean that the value functions diverge in some seeds. As λ decreases, ANQ enables larger overall neighborhood radii, resulting in higher and probably divergent learned Q values. A moderate λ (neighborhood constraint) is crucial for achieving superior performance.

- Inverse temperature α: determines how the neighborhood radius adapts to the action advantage

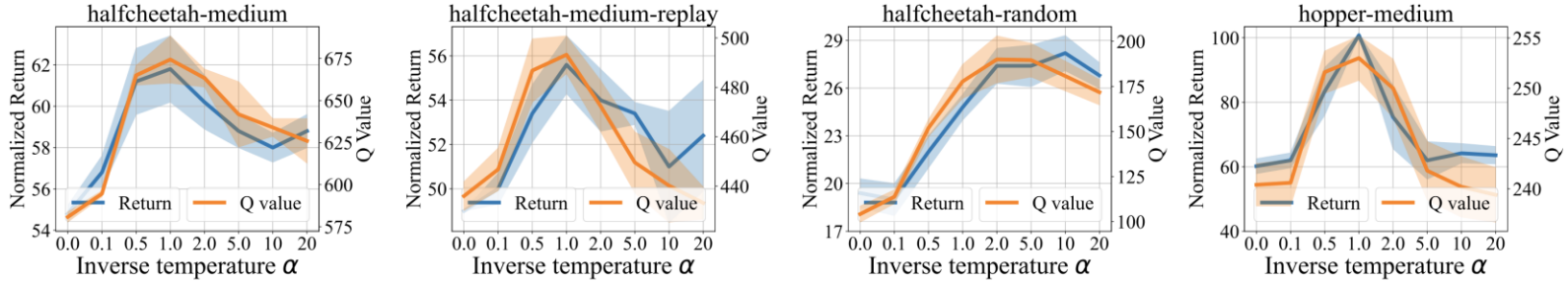


Figure 4: Performance and Q values of ANQ with varying inverse temperature α over five random seeds. An appropriately large α (adaptive neighborhoods) yields enhanced performance.

Thanks for Listening!