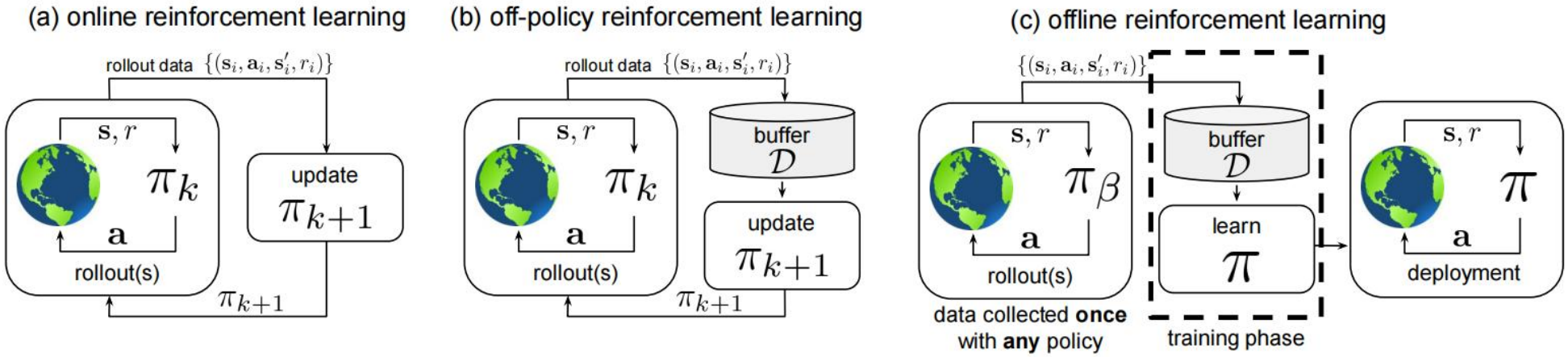
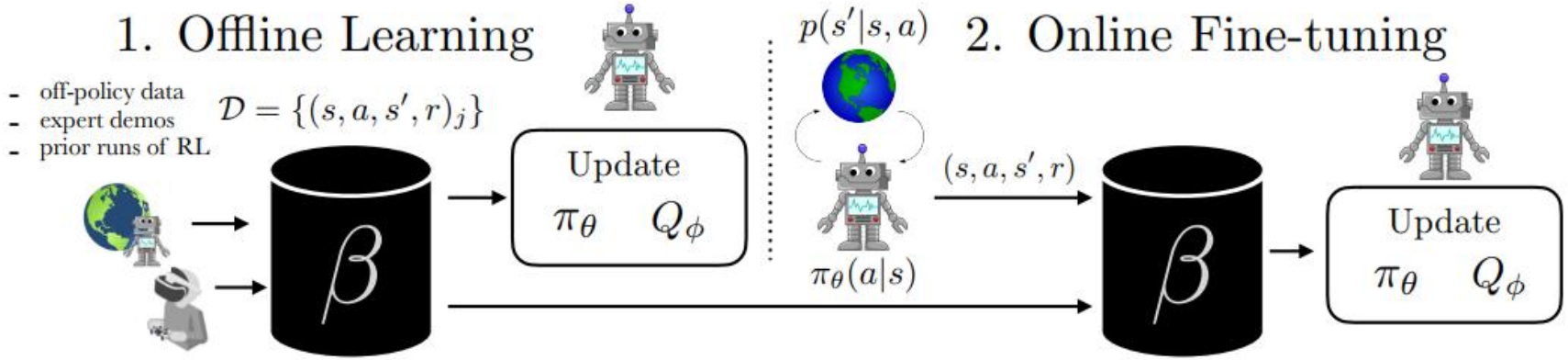
Offline-to-Online Reinforcement Learning

Background



MDP=$(S, A, R, \gamma, P, \rho_0)$ objective: $J(\pi) = \mathbb{E}_{s_0 \sim \rho_0, a \sim \pi(\cdot|s), s' \sim P(\cdot|s,a)}[\sum_{t=0}^{T} \gamma^t r(s_t, a_t)]$

policy: $\pi(a|s)$ state value function: $V(s)$ action value function: $Q(s, a)$



Background

main ideas in offline RL

- Policy constraint

$$\pi_{k+1} = \arg\max_{\pi \in \Pi} \mathbb{E}_{\mathbf{a} \sim \pi(\cdot|\mathbf{s})}[A^{\pi_k}(\mathbf{s}, \mathbf{a})]$$
$$\text{s.t. } D_{\mathrm{KL}}(\pi(\cdot|\mathbf{s})||\pi_\beta(\cdot|\mathbf{s})) \leq \epsilon$$

- Value Regularization

$$\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t) - \alpha \cdot f\left(\frac{\pi(a_t|s_t)}{\mu(a_t|s_t)}\right)\right)\right]$$

- Supervised Learning

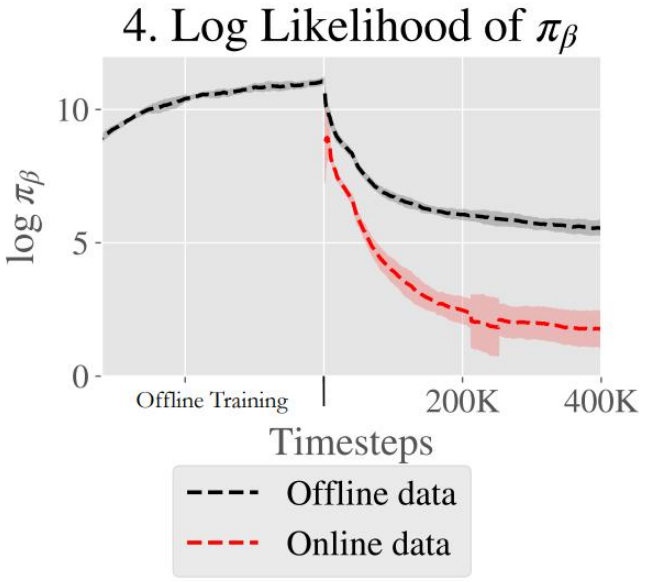
$$\text{Decision transformer } (R_1, s_1, a_1, R_2, s_2, a_2 \ldots)$$

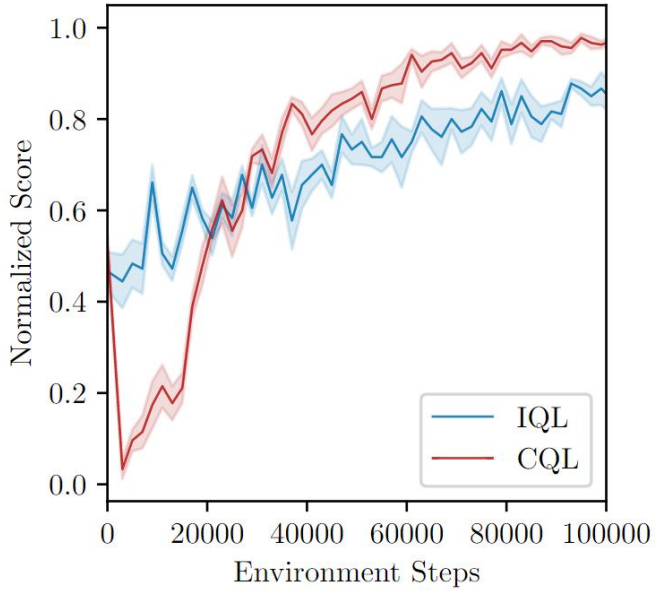
When offline-online task is aligned, supervised learning cannot be applied to online reinforcement learning, and the other two methods are difficult to applied to online fune-tuning directly.

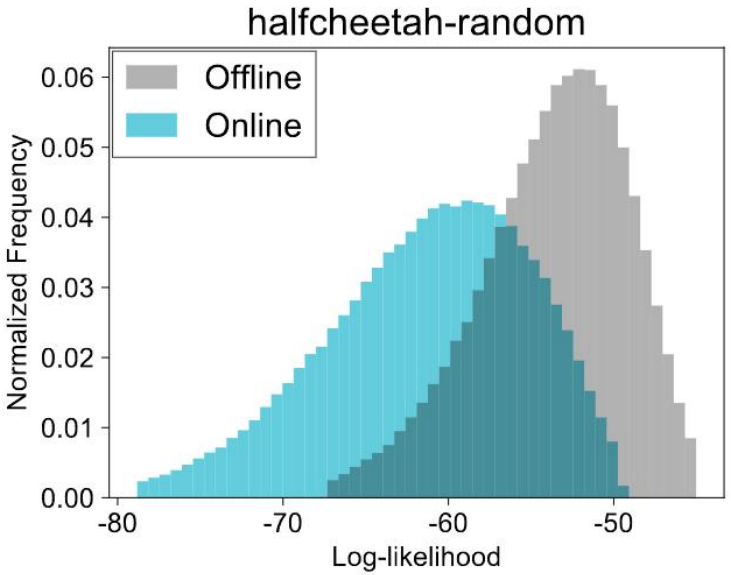
Background

difficulties in online fine-tuning

- Policy constraint — too slow to achieve best performance
- Value Regularization — performance drop at beginning because of conservatism
- Environment factors — State-action distribution shift, especially in narrow datasets







Background

An offline-to-online algorithm should:

(1) perform well after trainning in offline datasets → offline algorithm

(2) have stable performance when starting online training

(3) achieve best performance in few steps → online algorithm with good initialization

→ Present solutions
- Policy adjustment
- Remodel Q function
- Data rebalance

Policy adjustment

Adaptive Behavior Cloning Regularization for Stable Offline-to-Online Reinforcement Learning

TD3+BC:

$$\pi = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[\lambda\, Q(s, \pi(s)) - (\pi(s) - a)^2 \right]$$

$$\lambda = \frac{\alpha}{\frac{1}{N} \sum_{(s_i, a_i)} |Q(s_i, a_i)|}$$

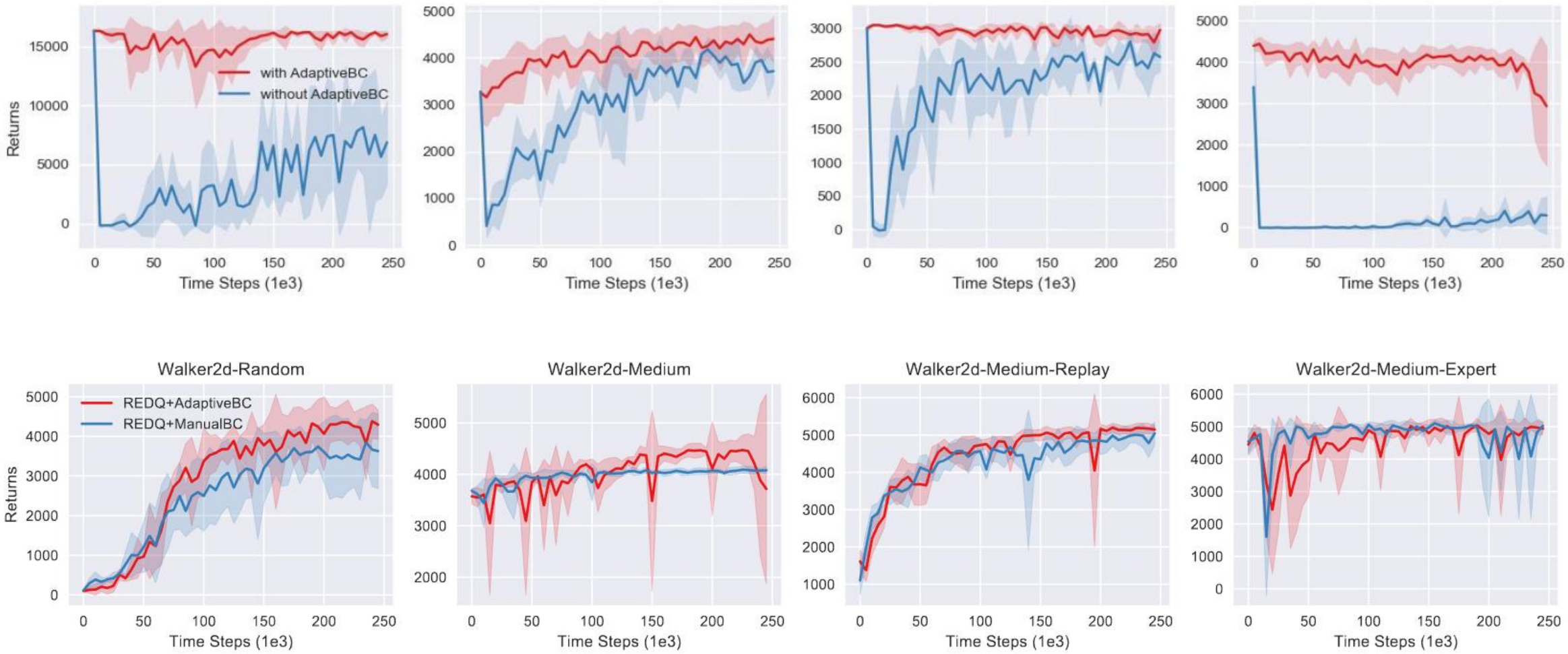
AdaptiveBC:

$$\pi_\theta = \arg\max_\theta \mathbb{E}_{(\mathbf{s},\mathbf{a})\sim\mathcal{D}} \left[\bar{Q}(\mathbf{s}, \pi_\theta(\mathbf{s})) - \alpha(\pi_\theta(\mathbf{s}) - \mathbf{a})^2 \right]$$

$$\Delta(\alpha_{\text{online}}) = K_P \cdot (R_{\text{avg}} - R_{\text{target}}) + K_D \cdot \max(0, R_{\text{avg}} - R_{\text{current}})$$

When current return is lower than averge return, strengthen the constraint, otherwise relieve constraint.

Experiment


with AdaptiveBC
without AdaptiveBC
Returns
Time Steps (1e3)
Walker2d-Random
Walker2d-Medium
Walker2d-Medium-Replay
Walker2d-Medium-Expert
REDQ+AdaptiveBC
REDQ+ManualBC

Policy adjustment

Improving TD3-BC: Relaxed Policy Constraint for Offline Learning and Stable Online Fine-Tuning

TD3+BC: $$\pi = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[\lambda\, Q(s, \pi(s)) - (\pi(s) - a)^2 \right] \qquad \lambda = \frac{\alpha}{\frac{1}{N}\sum_{(s_i,a_i)} |Q(s_i, a_i)|}$$

$$\pi = \arg\max_{\pi} \mathbb{E}_{(s,a)\sim D} \left[Q(s, \pi(s)) - \alpha (\pi(s) - a)^2 \right]$$

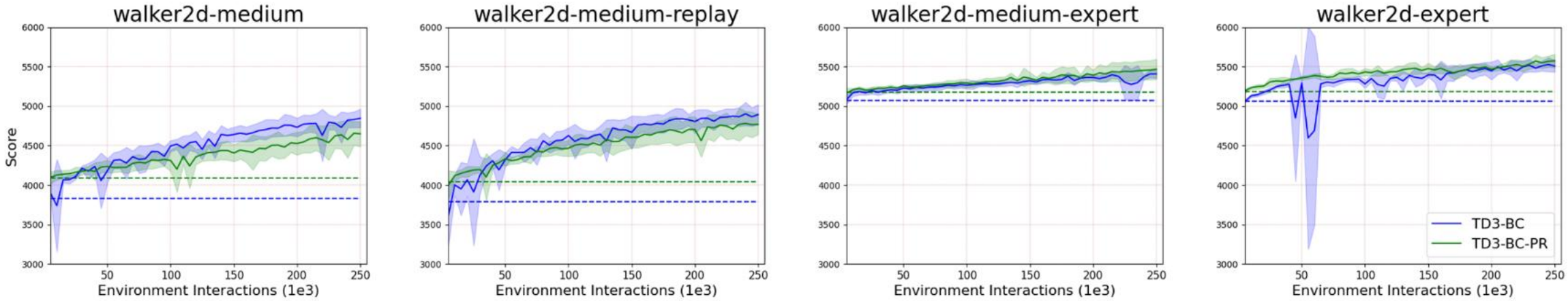
Refine: $$\alpha' = \frac{\alpha}{\lambda}$$

Online: $$\kappa_\alpha = \exp\left[\frac{1}{N} \log\left(\frac{\alpha_{end}}{\alpha_{start}} \right) \right]$$

$$\alpha \leftarrow \kappa_\alpha \alpha$$

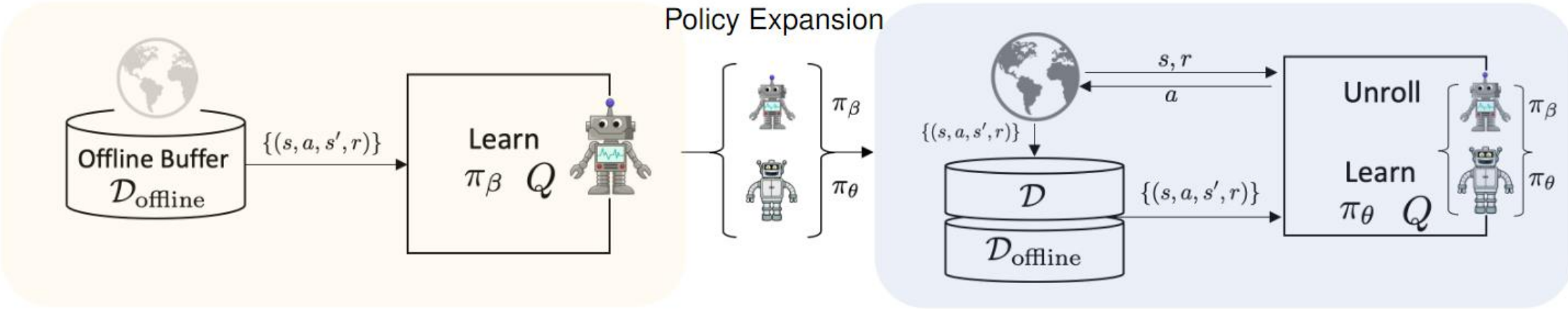
Experiment

Dataset	BC	CQL	IQL	TD3-BC	TD3-BC PR	TD3-BC FT	TD3-BC PR-FT
halfcheetah-med	42.6	44.0	47.4	48.5 ±0.7	**55.3** ±0.8	74.5 ±2.1	74.4 ±2.4
hopper-med	52.9	58.5	66.3	56.6 ±9.4	**100.1** ±2.8	93.7 ±20.6	102.8 ±0.9
walker2d-med	75.3	72.5	78.3	83.3 ±7.0	89.1 ±1.7	105.5 ±2.6	101.2 ±3.6
halfcheetah-med-rep	36.6	45.5	44.2	44.5 ±0.8	**48.7** ±1.2	68.0 ±2.6	69.8 ±1.7
hopper-med-rep	18.1	95.0	94.7	55.2 ±24.6	**100.5** ±1.2	102.8 ±1.8	103.4 ±1.7
walker2d-med-rep	26.0	77.2	73.9	82.5 ±13.6	87.9 ±2.8	106.6 ±2.9	103.9 ±8.1
halfcheetah-med-exp	55.2	91.6	86.7	91.5 ±15.8	91.9 ±11.1	96.2 ±1.3	96.6 ±1.1
hopper-med-exp	52.5	105.4	91.5	101.6 ±23.2	103.9 ±15.9	110.8 ±6.4	111.2 ±5.6
walker2d-med-exp	107.5	108.8	109.6	110.4 ±0.4	**112.7** ±0.3	117.8 ±1.2	119.1 ±2.8
halfcheetah-exp	-	-	-	97.1 ±1.4	97.5 ±1.1	96.7 ±2.1	97.7 ±1.8
hopper-exp	-	-	-	112 ±1.0	112.4 ±0.8	111.8 ±0.8	111.9 ±0.7
walker2d-exp	-	-	-	110.2 ±0.2	**113.0** ±0.5	120.0 ±1.7	121.4 ±1.9



Policy adjustment

Policy Expansion for Bridging Offline-to-Online Reinforcement Learning(ICLR 2023)



$$P_{\mathbf{w}}[i] = \frac{\exp(Q_\phi(s,a_i)/\alpha)}{\sum_j \exp(Q_\phi(s,a_j)/\alpha)}, \quad \forall i \in [1,\cdots K] \qquad \tilde{\pi}(a|s) = [\delta_{a\sim\pi_\beta(s)}, \delta_{a\sim\pi_\theta(s)}]\mathbf{w}, \quad \mathbf{w} \sim P_{\mathbf{w}}$$

Policy Expansion: $\tilde{\pi} = [\pi_\beta, \pi_\theta]$; transfer Q_ϕ π_θ: randomly initialized policy

while in *online training phase* **do**

  for each environment step **do**

    $a_t \sim \tilde{\pi}(a_t|s_t)$ according to (6), $s_{t+1} \sim T(s_{t+1}|s_t,a_t)$, $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t,a_t,r(s_t,a_t),s_{t+1})\}$

  end for

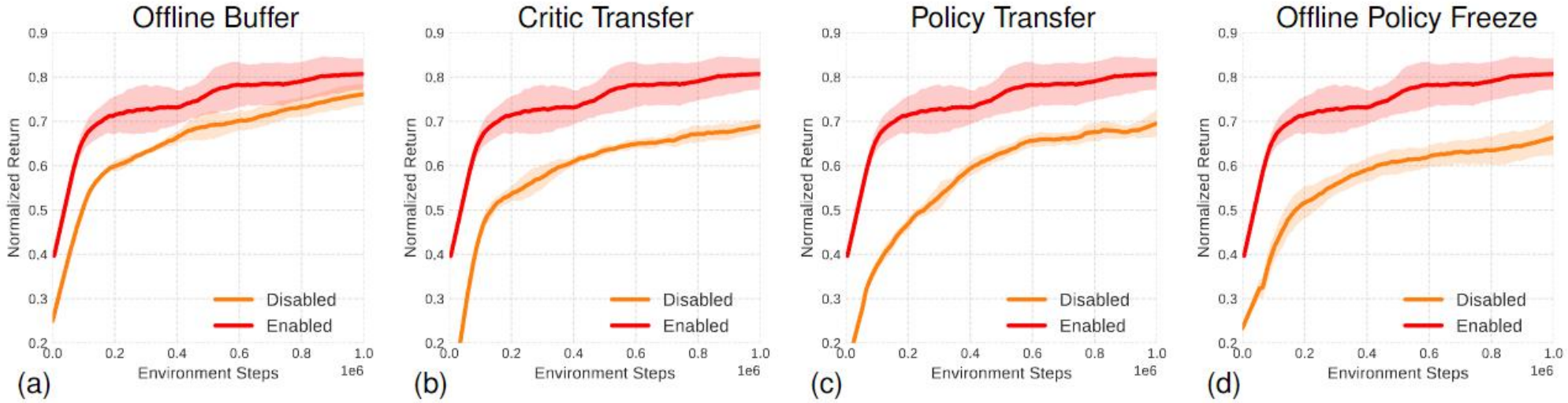
  for each gradient step **do**

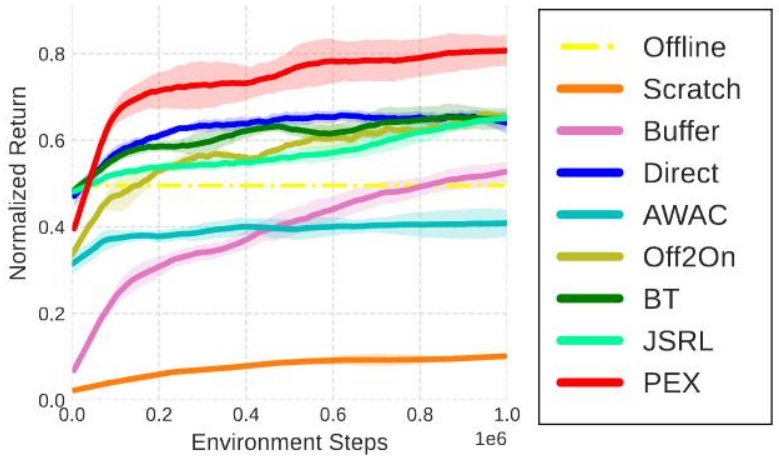
    % online training using batches from both $\mathcal{D}_{\text{offline}}$ and $\mathcal{D}$

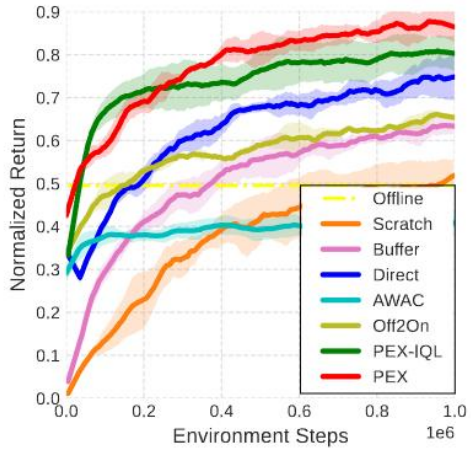
    $\phi \leftarrow \phi - \lambda_Q \nabla_\phi L^Q_{\text{online}}(\phi)$, $\theta \leftarrow \theta - \lambda_\pi \nabla_\theta L^{\pi_\theta}_{\text{online}}(\theta)$

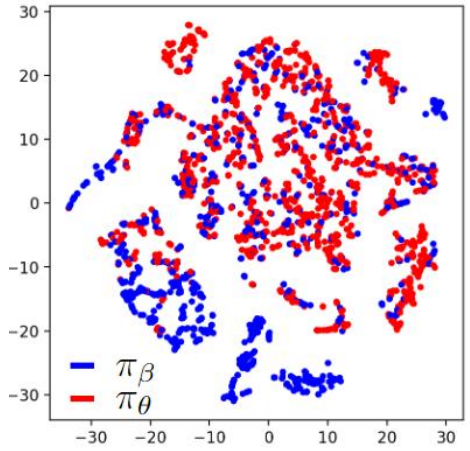
  end for

Experiment


Offline Buffer
Critic Transfer
Policy Transfer
Offline Policy Freeze
Normalized Return
Environment Steps
1e6
Disabled
Enabled
(a)
(b)
(c)
(d)


Normalized Return
Environment Steps
1e6
Offline
Scratch
Buffer
Direct
AWAC
Off2On
BT
JSRL
PEX


Normalized Return
Environment Steps
1e6
Offline
Scratch
Buffer
Direct
AWAC
Off2On
PEX-IQL
PEX


π_β
π_θ

Data rebalance

Offline-to-Online Reinforcement Learning via Balanced Replay and Pessimistic Q-Ensemble

Balanced Experience Replay

$$w(s,a) := d^{\text{on}}(s,a)/d^{\text{off}}(s,a)$$ measure the online-ness

Make more use of the offline samples of near-on-policy to update safely

$$f(y) := y \log \frac{2y}{y+1} + \log \frac{2}{y+1}$$

$$D_{JS}(P||Q) = \int_{\mathcal{X}} f(dP(x)/dQ(x))dQ(x)$$

$$\mathcal{L}^{\text{DR}}(\psi) = \mathbb{E}_{x\sim P}[f'(w_\psi(x))] - \mathbb{E}_{x\sim Q}[f^*(f'(w_\psi(x)))]$$

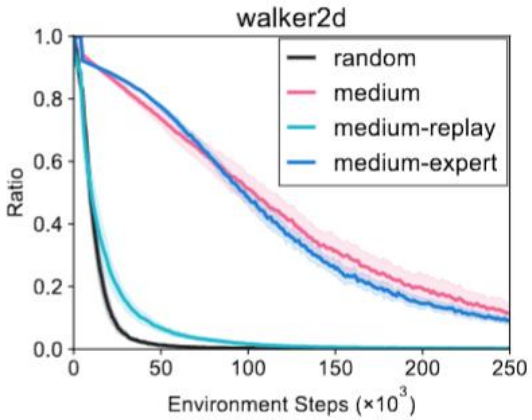
$$\widetilde{w}_\psi(x) = \frac{w_\psi(x)^{1/T}}{\mathbb{E}_{x\sim P}[w_\psi(x)^{1/T}]}$$

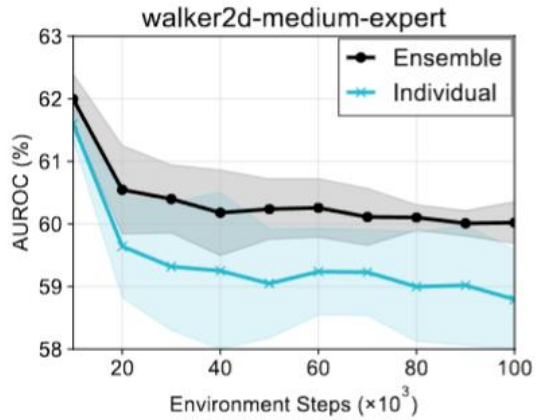
P:offline datasets
Q:online datasets

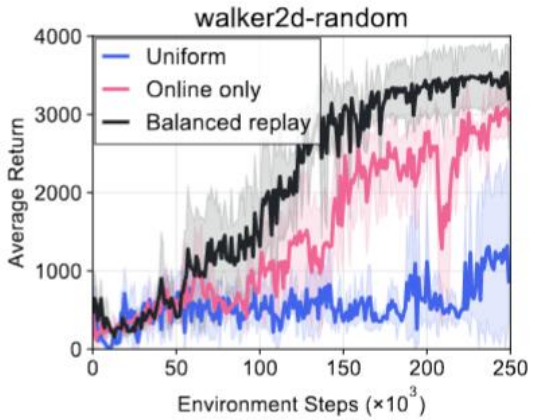
Pessimistic Q-Ensemble

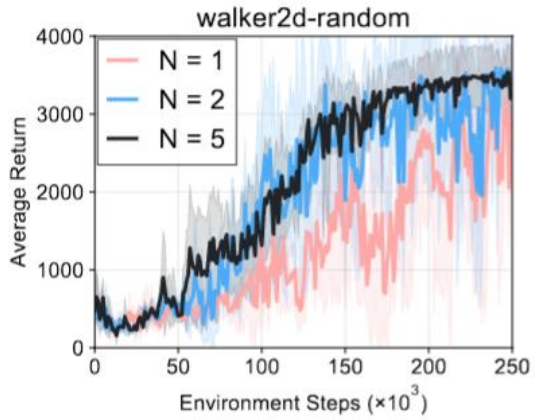
$$Q_\theta := \frac{1}{N}\sum_{i=1}^{N} Q_{\theta_i}, \quad \pi_\phi(\cdot|s) = \mathcal{N}\left(\frac{1}{N}\sum_{i=1}^{N}\mu_{\phi_i}(s), \quad \frac{1}{N}\sum_{i=1}^{N}(\sigma^2_{\phi_i}(s) + \mu^2_{\phi_i}(s)) - \mu^2_{\phi}(s)\right)$$

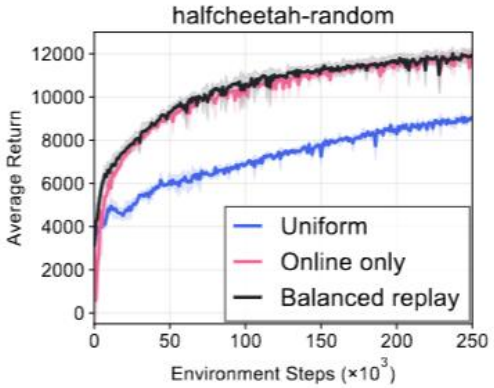
Experiment

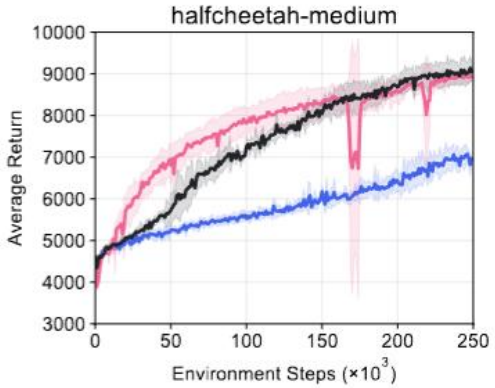

walker2d
random
medium
medium-replay
medium-expert
Ratio
Environment Steps (×10^3)

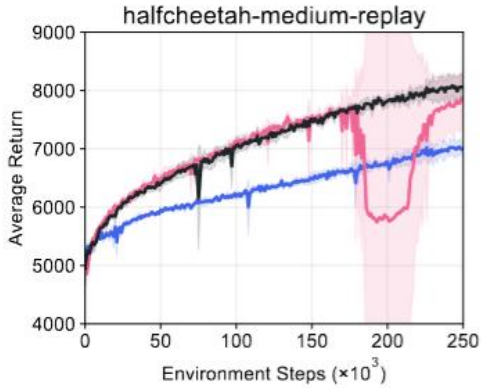

walker2d-medium-expert
Ensemble
Individual
AUROC (%)
Environment Steps (×10^3)

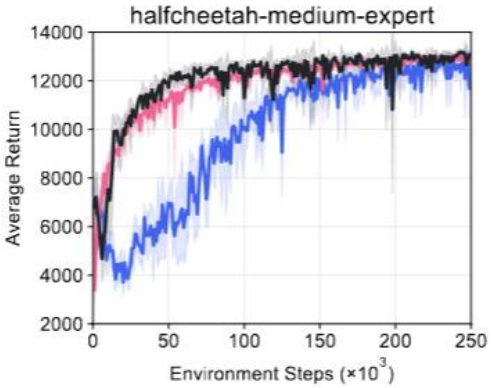

walker2d-random
Uniform
Online only
Balanced replay
Average Return
Environment Steps (×10^3)

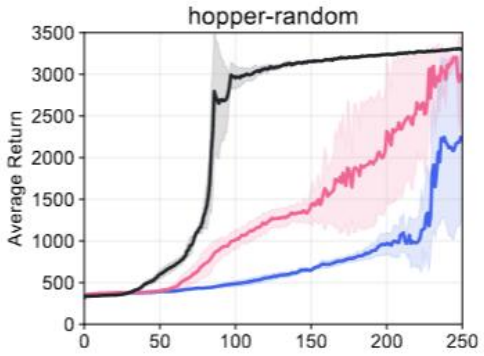

walker2d-random
N = 1
N = 2
N = 5
Average Return
Environment Steps (×10^3)

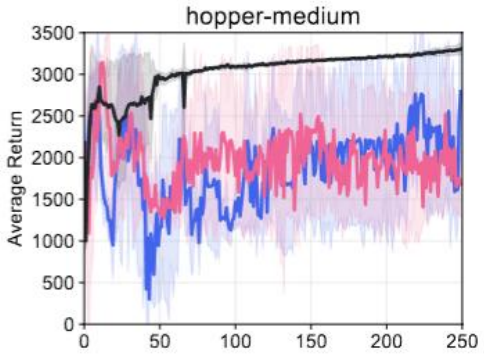

halfcheetah-random
Uniform
Online only
Balanced replay
Average Return
Environment Steps (×10^3)

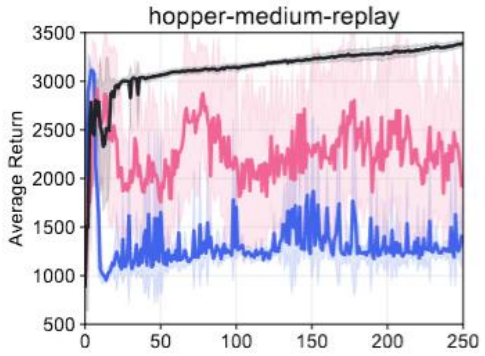

halfcheetah-medium
Average Return
Environment Steps (×10^3)

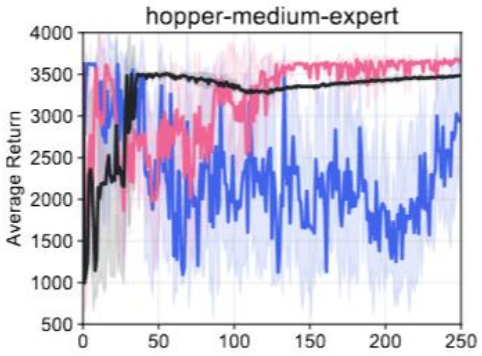

halfcheetah-medium-replay
Average Return
Environment Steps (×10^3)


halfcheetah-medium-expert
Average Return
Environment Steps (×10^3)


hopper-random
Average Return


hopper-medium
Average Return


hopper-medium-replay
Average Return


hopper-medium-expert
Average Return

Data rebalance

MOORe: Model-based Offline-to-Online Reinforcement Learning

Prioritized Sample $Pr(d,t) = \begin{cases} f(d,t), & d \in \mathcal{D}_{off} \\ 1.0, & d \in \mathcal{D}_{on} \end{cases}$ $f(d,t) = \frac{1}{\alpha t}$

Reward Penalty $\tilde{r}(s,a) = \hat{r}(s,a) - \lambda u(s,a)$

Algorithm 1 Model-based **O**ffline-to-**O**nline **Re**inforcement learning (MOORe)

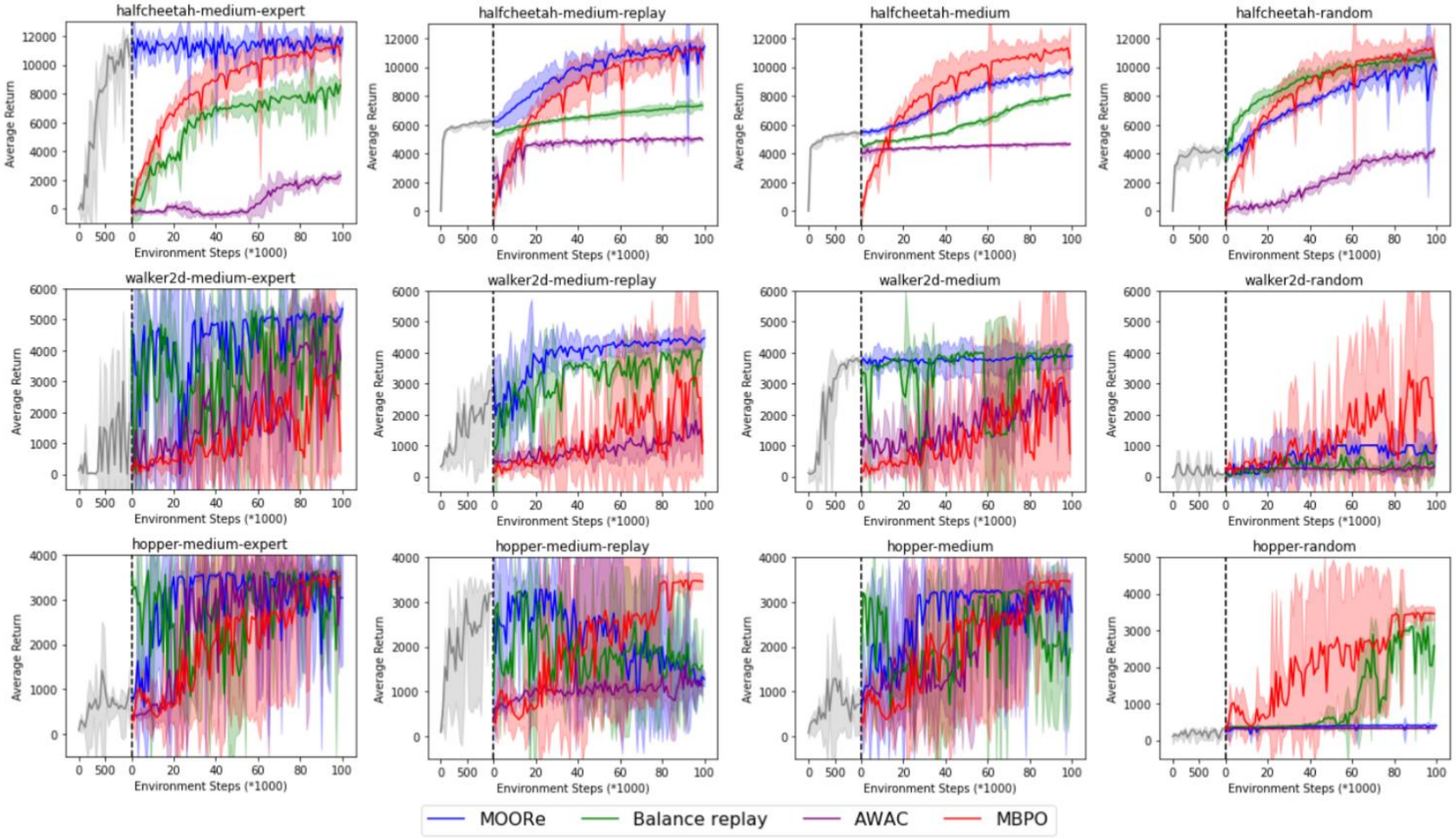
 # Offline learning stage
Run MOPO on $\mathcal{D}_{off}$ to obtain π_{off}, an ensemble of K estimated models $\{\hat{M}_{off}\}_{k=1}^{K}$, and the action-value functions $\hat{Q}_{off}$

 # Online learning stage
Initialize $\pi_{on} = \pi_{off}$, $\{\hat{M}_{on} = \hat{M}_{off}\}_{k=1}^{K}$, the action-value function $\hat{Q}_{on} = \hat{Q}_{off}$
Initialize $\mathcal{D}_{on} = \emptyset$, $\mathcal{D}_{model} = \emptyset$ and the number of gradient updates N
Initialize the online episode length S, the estimated model's update frequency ϕ, rollout batch size R and length H
for Epoch $t = 1...T$ **do**
  Set the priority value of each transition $d \in D_{off}$ as $Pr(d,t)$
  for Step $\tau = 1...S$ **do**
    Interact with the real environment for one step using π_{on} to obtain the transition $d_\tau = (s_\tau, a_\tau, s_{\tau+1}, r_\tau)$
    Compute the priority $Pr(d_\tau, t)$ and add the tuple $((s_\tau, a_\tau, s_{\tau+1}, r_\tau), Pr(d_\tau, t))$ to $\mathcal{D}_{on}$
    if $\tau \mod \phi == 0$ **then**
      Train $\hat{M}$ until convergence on $\mathcal{D}_{off} \cup \mathcal{D}_{on}$ by prioritized sampling [Schaul *et al.*, 2016]
      Sample a batch of R initial states $\{s_i\}_{i=1}^{R}$ from $\mathcal{D}_{off} \cup \mathcal{D}_{on}$ by prioritized sampling [Schaul *et al.*, 2016]
      Generate $\{d_i\}_{i=1}^{R \times H}$ transitions by rollouting $\hat{M}$ using π_{on} for H steps starting from $\{s_i\}_{i=1}^{R}$
      Obtain the penalized rewards using equation (2) (as done in MOPO) and add the updated transitions to $\mathcal{D}_{model}$
    end if
    Perform N gradient updates on π_{on} and $\hat{Q}_{on}$ using data uniformly sampled from $\mathcal{D}_{model}$
  end for
end for

update model

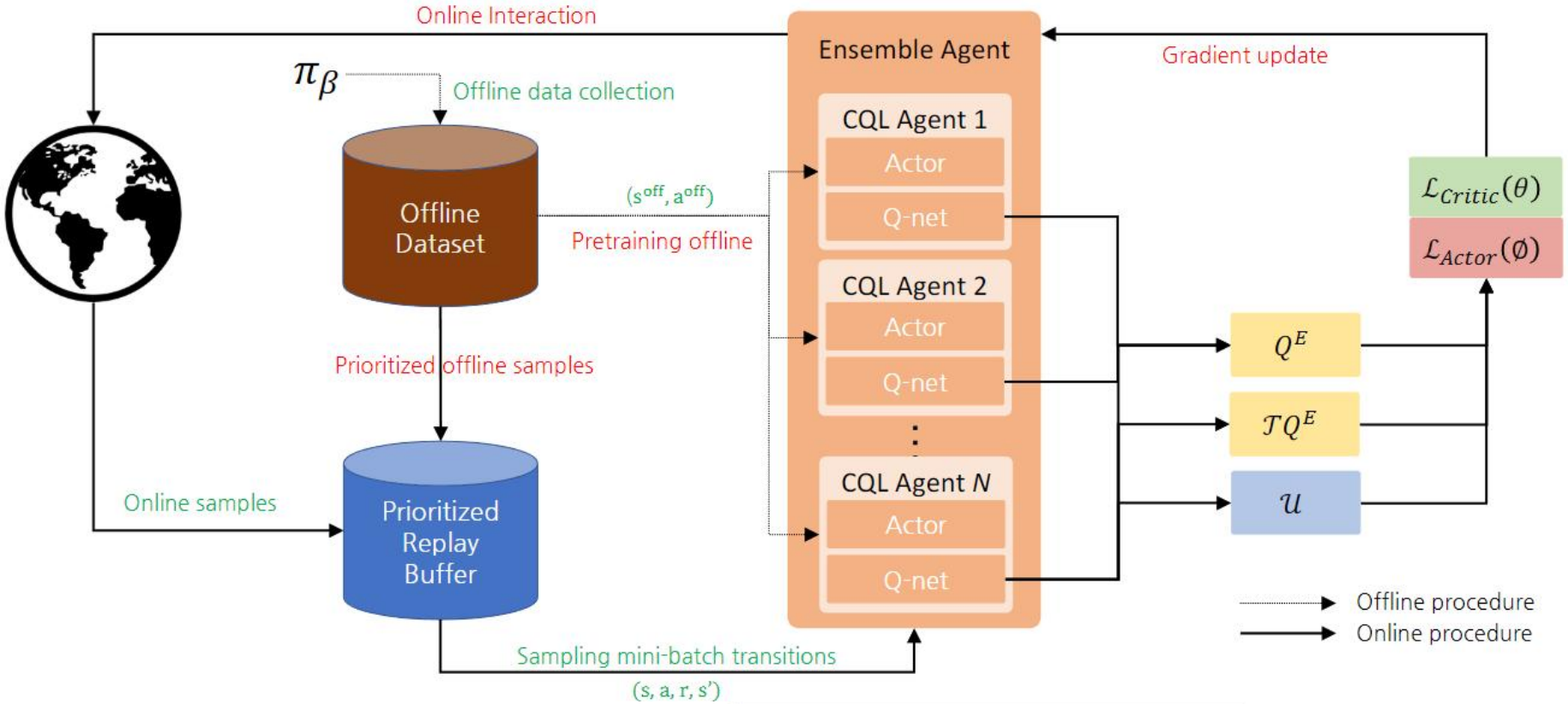
update π and Q

Experiment


halfcheetah-medium-expert
halfcheetah-medium-replay
halfcheetah-medium
halfcheetah-random
walker2d-medium-expert
walker2d-medium-replay
walker2d-medium
walker2d-random
hopper-medium-expert
hopper-medium-replay
hopper-medium
hopper-random
Average Return
Environment Steps (*1000)
MOORe
Balance replay
AWAC
MBPO

Remodel Q function

Uncertainty-Driven Pessimistic Q-Ensemble for Offline-to-Online Reinforcement Learning



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

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

Experiment

		CQL	Off2onRL	UPQ $\beta = 0.1$	UPQ $\beta = 0.01$	UPQ $\beta = 0.001$
Random	HalfCheetah	2455.2	11474.5	11068.6	11172.4	11280.5
	Hopper	323.5	3213	945.3	2622.6	2967.8
	Walker2d	372.2	2706.1	2224.1	3190.8	2759.2
Medium	HalfCheetah	5171.1	10049.7	10364.9	10847.9	10668.2
	Hopper	1973.4	3279.7	3416.8	3361.3	3424.1
	Walker2d	3288.4	4638.5	4356.5	4736.9	4971.2
Medium Replay	HalfCheetah	5214	10413.4	10118.3	10600.4	10313.2
	Hopper	1698.8	3521.6	3195.2	3206.7	3440.4
	Walker2d	3142.1	4760.9	4505.2	4750.9	5100.2
Medium Expert	HalfCheetah	1158.8	11159.0	11261.2	11277.9	11353.4
	Hopper	1272.1	2531.3	2201.1	2594.8	3423.4
	Walker2d	3675	5181.8	5525.7	5682.6	5446

Remodel Q function

Adaptive Policy Learning for Offline-to-Online Reinforcement Learning

$$C^{k+1} \leftarrow \mathcal{F}\left(\mathbb{A}(C^k) + \mathcal{W}(\mathbf{s},\mathbf{a})\mathbb{B}(C^k)\right)$$

Online-Offline Replay Buffer

- online buffer: recent online data
  Set the $\mathcal{W}(\mathbf{s},\mathbf{a})$ to 0
- offline buffer: offline datasets and previous online data
  Set the $\mathcal{W}(\mathbf{s},\mathbf{a})$ to 1

$$\mathbb{A}(Q_i^k) = \mathbb{E}_{\mathbf{s},\mathbf{a},\mathbf{s}'\sim\text{OORB},\mathbf{a}'\sim\pi^k(\cdot|s')}\left[\left(Q_i^k(\mathbf{s},\mathbf{a}) - \mathcal{B}^\pi\hat{Q}^k(\mathbf{s}',\mathbf{a}')\right)^2\right]$$

$$\mathbb{B}(Q_i^k) = \alpha\mathbb{E}_{\mathbf{s}\sim\text{OORB}}\left[\log\sum_{\mathbf{a}'}\exp(Q_i^k(\mathbf{s},\mathbf{a}')) - \mathbb{E}_{\mathbf{a}\sim\text{OORB}}[Q_i^k(\mathbf{s},\mathbf{a})]\right]$$

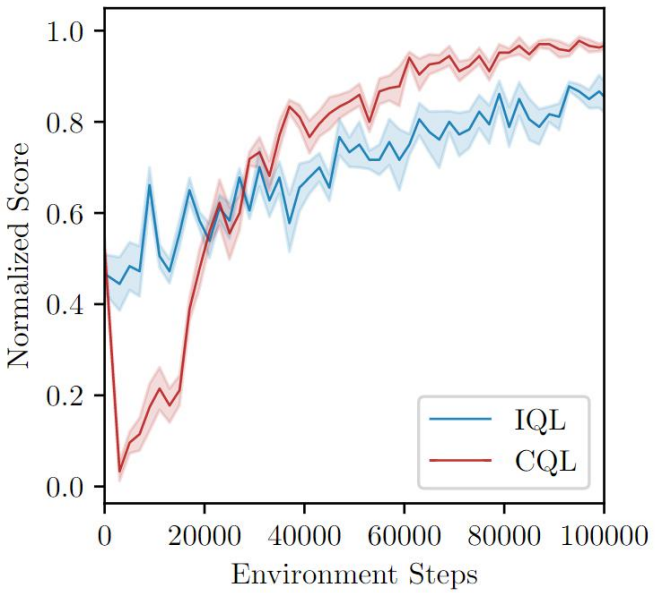
Experiment

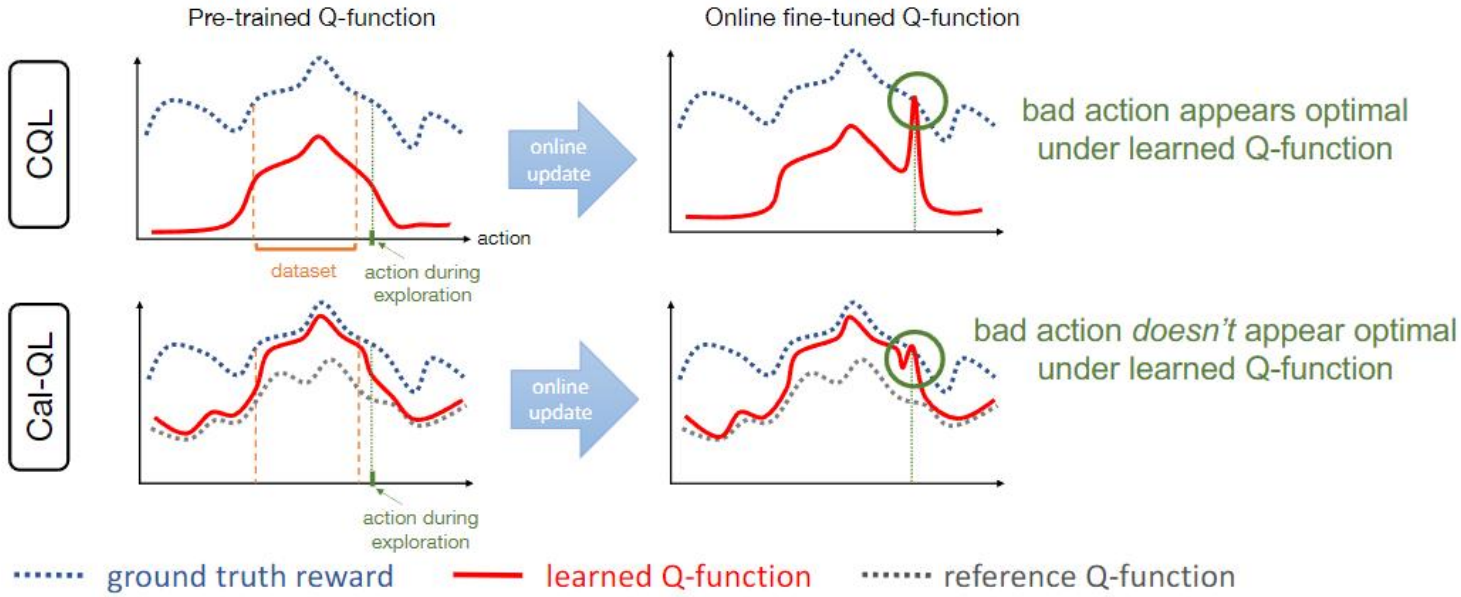
Environment	GCQL	GCTD3BC	CQL	REDQ_ON	REDQ	TD3_ON	TD3BC	AWAC	OFF2ON	IQL
walker2d-r	31±27	5±3	7±9	**71±11**	5±3	7±2	6±3	12	20±13	7±3
hopper-r	58±30	35±22	10±1	78±37	2±1	10±2	11±0	63	**81±21**	10±2
halfcheetah-r	**101±2**	69±8	46±4	59±2	32±1	39±1	35±3	53	85 ±3	28±7
walker2d-m	**94±6**	90±7	83±1	71±11	2±3	7±2	79±2	80	89±2	51±13
hopper-m	83±11	**99±3**	70±23	78±37	3±1	10±2	80±13	91	59 ±9	42±9
halfcheetah-m	**66±3**	62±2	25±8	59±2	46±1	39±1	43±1	41	58±2	40±0
walker2d-me	93±12	102±2	105±1	71±11	12±3	7±2	**110±3**	78	101±24	58±22
hopper-me	**110±1**	**110±2**	109±5	78±37	40±15	10±2	**110±0**	**112**	82 ±21	72±16
halfcheetah-me	**102±1**	**103±2**	92±2	59±2	9±3	39±1	98±2	41	100±1	38±17
walker2d-mr	**97±16**	90±9	57±5	71±11	13±2	7±2	60±5	-	71±32	30±13
hopper-mr	72±20	**87±11**	37±4	78±37	0±1	10±2	38±1	-	**60±23**	31±10
halfcheetah-mr	**62±2**	53±1	50±0	**59±2**	28±25	39±1	47±0	-	57±1	42±2
Total	**969±131**	905±71	691±63	624±200	192±59	224±20	717±33	-	863±152	449±114

Remodel Q function

Cal-QL: Calibrated Offline RL Pre-Training for Efficient Online Fine-Tuning

The main idea: Conservative Q-values learned by the conservative offline RL method must be lower-bounded by the ground-truth Q-value of a sub-optimal reference policy



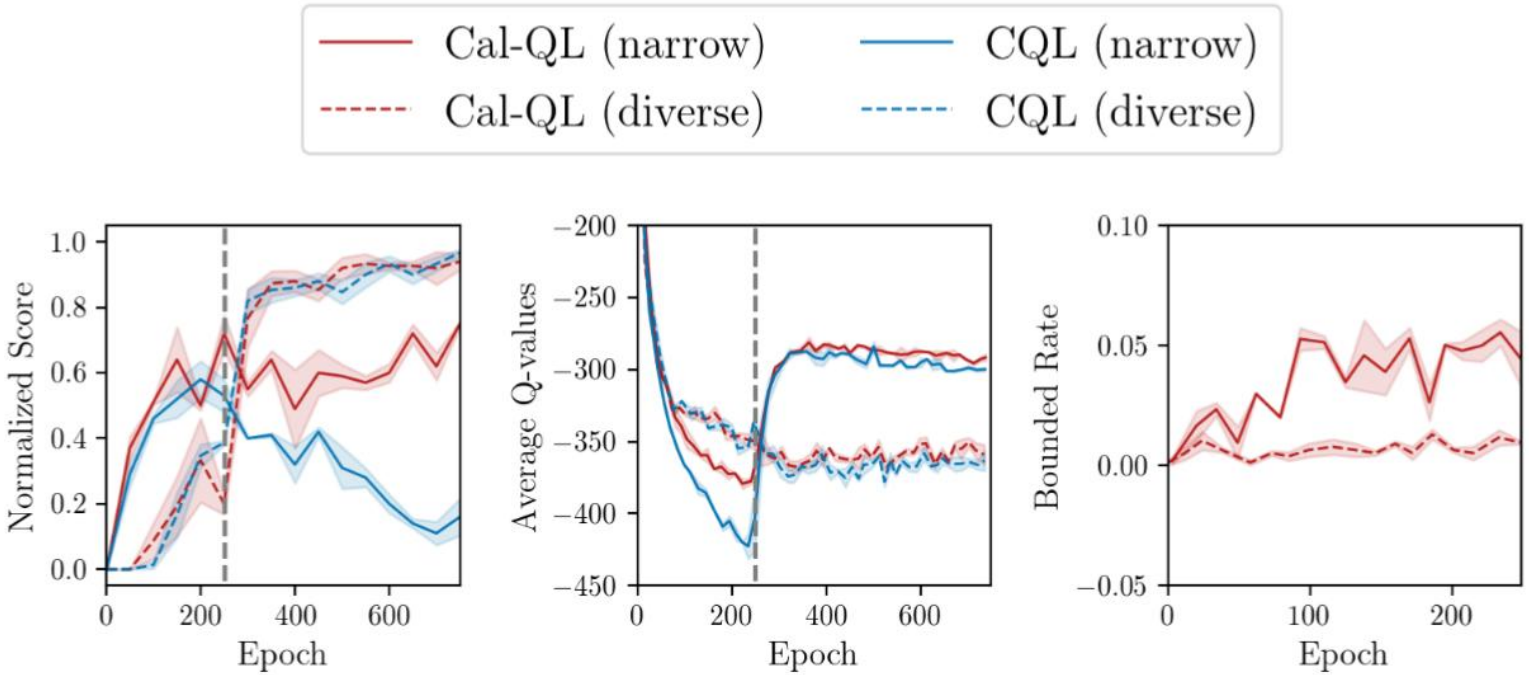


$$\min_\theta \alpha \underbrace{\left(\mathbb{E}_{s\sim\mathcal{D},a\sim\pi}\left[Q_\theta(s,a)\right] - \mathbb{E}_{s,a\sim\mathcal{D}}\left[Q_\theta(s,a)\right]\right)}_{\text{Conservative regularizer } \mathcal{R}(\theta)} + \frac{1}{2}\mathbb{E}_{s,a,s'\sim\mathcal{D}}\left[\left(Q_\theta(s,a) - \mathcal{B}^\pi\bar{Q}(s,a)\right)^2\right]$$

$$\mathbb{E}_{s\sim\mathcal{D},a\sim\pi}\left[\max\left(Q_\theta(s,a), Q^\mu(s,a)\right)\right] - \mathbb{E}_{s,a\sim\mathcal{D}}\left[Q_\theta(s,a)\right]$$

Experiment

Domain	Task	IQL	CQL	SAC + offline data	SAC	O3F	Cal-QL (Ours)
antmaze	large-diverse	0.40 → 0.59	0.27 → 0.84	0.00 → 0.00	0.00 → 0.00	0.74 → 0.04	0.32 → **0.94**
	large-play	0.41 → 0.51	0.29 → 0.65	0.00 → 0.00	0.00 → 0.00	0.60 → 0.03	0.28 → **0.90**
	medium-diverse	0.70 → 0.92	0.68 → **0.98**	0.00 → 0.02	0.00 → 0.00	0.85 → 0.96	0.73 → **0.98**
	medium-play	0.72 → 0.94	0.63 → 0.99	0.00 → 0.25	0.00 → 0.00	0.89 → **0.99**	0.54 → 0.98
kitchen	partial	0.37 → 0.61	0.69 → 0.71	0.00 → 0.00	0.00 → 0.03	N/A → N/A	0.62 → **0.81**
	mixed	0.46 → 0.45	0.60 → 0.50	0.00 → 0.00	0.00 → 0.02	N/A → N/A	0.37 → **0.71**
	complete	0.62 → **0.62**	0.12 → 0.34	0.00 → 0.00	0.00 → 0.05	N/A → N/A	0.20 → **0.62**
adroit	pen-binary	0.88 → 0.89	0.08 → 0.19	0.10 → 0.45	0.00 → 0.00	0.91 → 0.92	0.79 → **0.96**
	door-binary	0.29 → 0.87	0.13 → 0.62	0.00 → 0.86	0.00 → 0.01	0.00 → 0.00	0.40 → **0.90**
	relocate-binary	0.04 → 0.37	0.09 → 0.75	0.00 → **0.99**	0.01 → 0.24	0.03 → 0.35	0.03 → **0.99**
COG	manipulation	0.47 → 0.89	0.51 → **0.97**	0.00 → 0.00	0.00 → 0.04	N/A → N/A	0.61 → **0.97**
	average	0.49 → 0.70 (+ 42.9%)	0.37 → 0.69 (+ 84.4%)	N/A → 0.23	N/A → 0.04	0.57 → 0.47 (- 18.2%)	0.44 → **0.89** (+ **99.6%**)



Remodel Q function

Actor-Critic Alignment for Offline-to-Online Reinforcement Learning

SAC: $\mathcal{L}_\pi^{\mathrm{SAC}}(\theta, \mathbf{d}) := \mathbb{E}_{s\sim\mathbf{d}}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[\alpha \log \pi_\theta(a|s) - Q_\mu(s,a)\right]$ $\pi_\theta(a|s) = \exp\left(\frac{1}{\alpha}Q_\mu(s,a)\right) \Big/ \sum_{a\in\mathcal{A}} \exp\left(\frac{1}{\alpha}Q_\mu(s,a)\right)$

1.Offline phase: $\mathcal{L}_\pi^{\mathrm{SAC+ML}}(\theta, \mathbf{d}) = \mathbb{E}_{(s,a)\sim\mathbf{d}}\mathbb{E}_{b\sim\pi_\theta(\cdot|s)}\Big[-\lambda\Big(Q_\mu(s,b) - \alpha \log \pi_\theta(b|s)\Big) - \log \pi_\theta(a|s)\Big]$

The main idea: adding a baseline function $Z(s)$ to $Q(s,a)$ does not change the optimal π_θ

$$Q_\mu(s,a) = Z(s) + \alpha \log \pi_\theta(a|s)$$

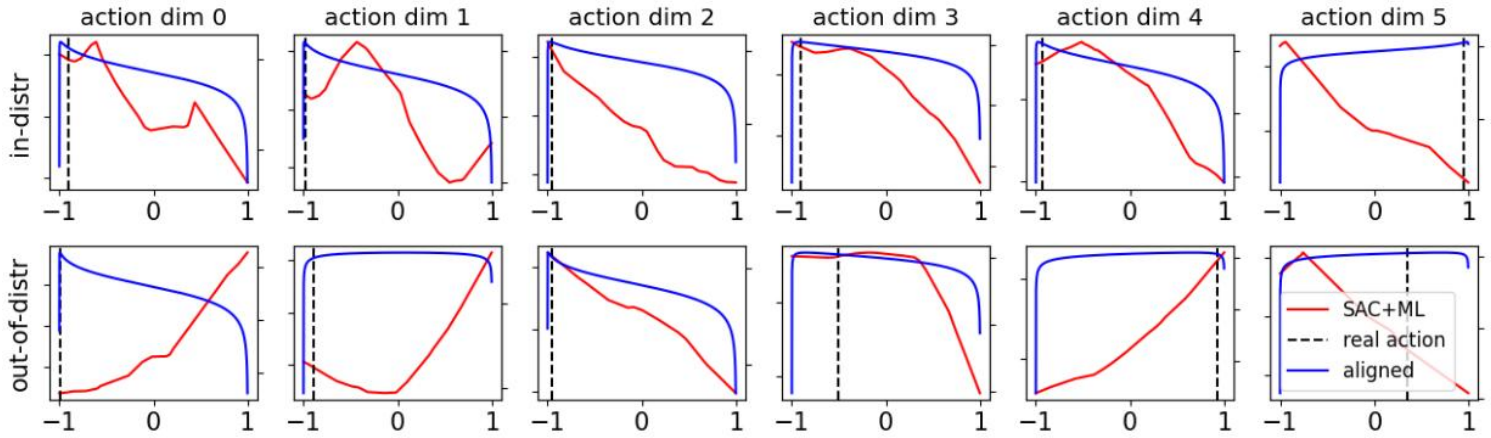
In practice $Q_i(s,a) = \log \pi_{\theta_0}(a|s) + Z_{\psi_i}(s)$ $Z(s)$ can be calibrated by minimizing the Bellman residual

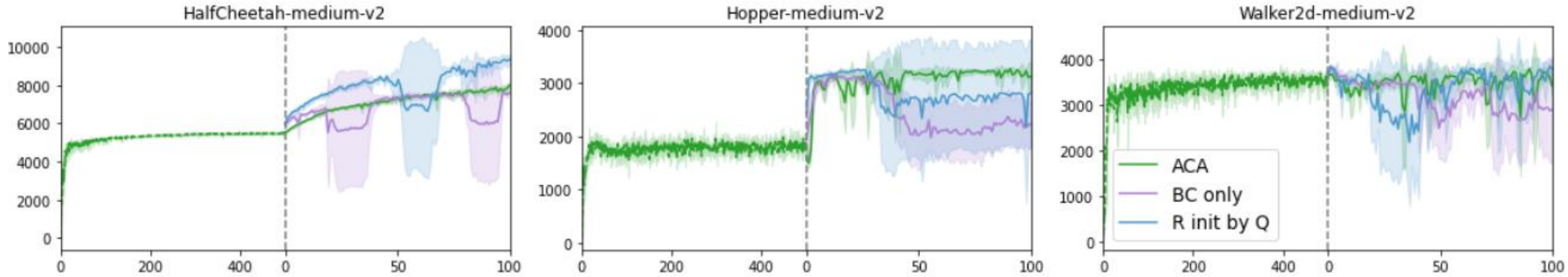
2.Online phase: $Q_{\phi_i}(s,a) := \log \pi_{\theta_0}(a|s) + R_{\phi_i}(s,a)$, where $R_{\phi_i}(s,a)$ is **initialized** with $Z_{\psi_i}(s)$

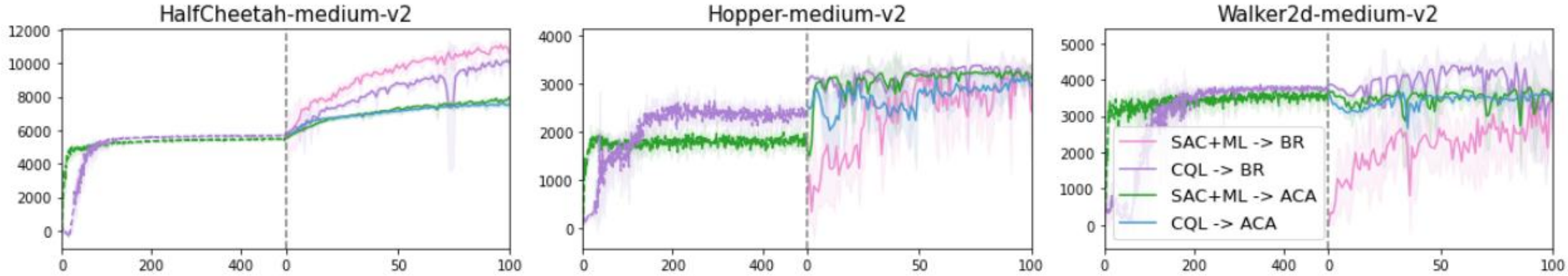
Train without offline datasets

Dataset	Env	Score(δ) w/ offline data	Score(δ) w/o offline data
Med.-Replay	HC	59.03(16.50)	59.48(16.95)
	H	85.54(36.72)	77.19(28.37)
	W	85.17(22.98)	84.27(22.08)
Med.-Expert	HC	93.74(0.21)	93.81(0.28)
	H	98.02(4.94)	105.67(12.59)
	W	110.54(2.42)	110.93(2.81)
Expert	HC	93.14(-0.46)	90.76(-2.83)
	H	110.21(-0.67)	109.22(-1.66)
	W	109.59(1.38)	110.52(2.31)
Total		844.98(84.02)	841.84(80.88)

Experiment


action dim 0
action dim 1
action dim 2
action dim 3
action dim 4
action dim 5
in-distr
out-of-distr
SAC+ML
real action
aligned


HalfCheetah-medium-v2
Hopper-medium-v2
Walker2d-medium-v2
ACA
BC only
R init by Q


HalfCheetah-medium-v2
Hopper-medium-v2
Walker2d-medium-v2
SAC+ML -> BR
CQL -> BR
SAC+ML -> ACA
CQL -> ACA

Conclusion

- Present solutions
  - Policy adjustment
    - adjust policy constraint — reducing constraint when online trainning
    - policy expansion — a general framework of bridging offline-to-online
  - Remodel Q function
    - uncertainty penalty
    - differential update
      - base on confidence
    - Q function calibration — mitigate underestimate
    - Q function alignment — align Q function
  - Data rebalance — adaptive sampling

Prospect

1. Combined with more efficient online algorithms (additional reward...)
2. More method about uncertainty (weights of update and sampling, adaptive UTD)
3. Offline-to-online on-policy method
4. Model-based method (rollout imaginary trajectories, different confidence policies)

Thanks