Offline Reinforcement Learning with Implicit Q-Learning

Ilya Kostrikov, Ashvin Nair, Sergey Levine

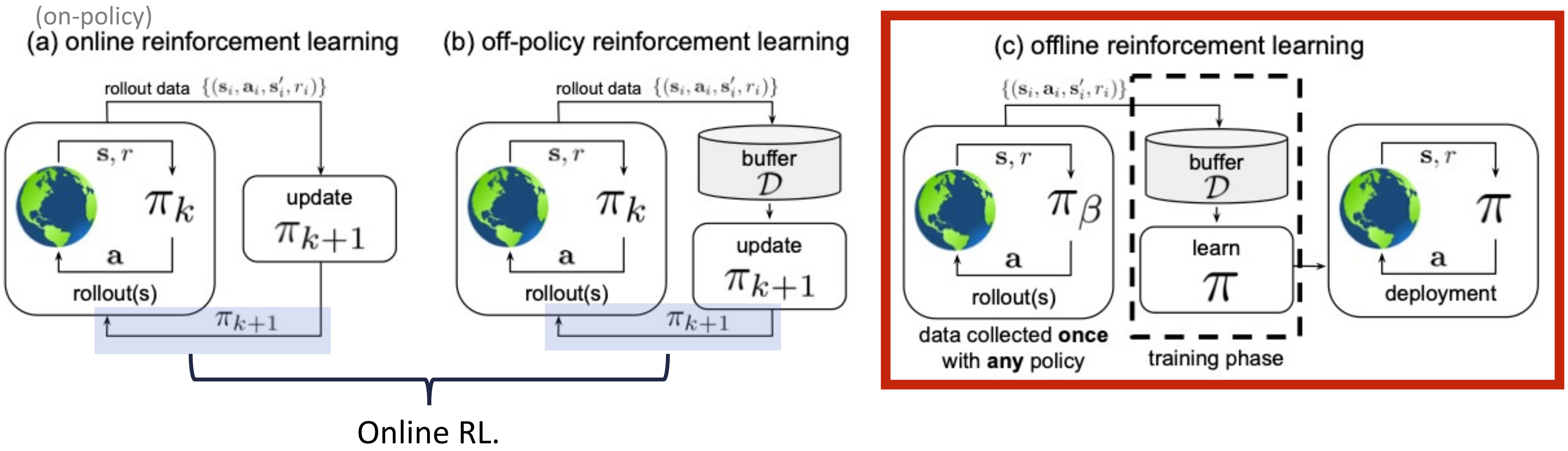
ICLR 2022 Poster

AI611 Paper Presentation

Presenter: Hanseul Cho (KAIST AI)

Offline RL

- RL **w/o online interaction**.



Offline RL

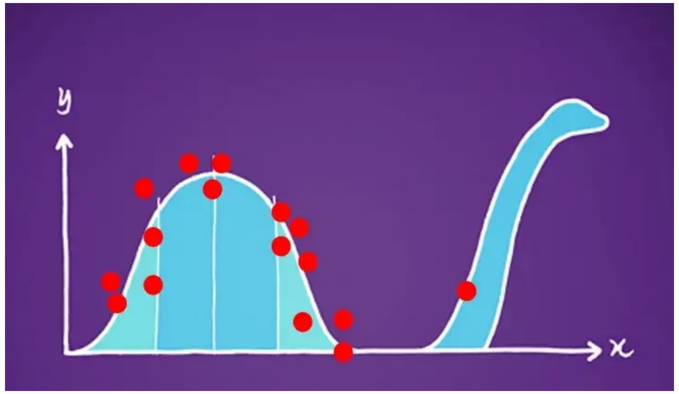
- How?

$$L_{TD}(\theta) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}}[(r(s,a) + \gamma \max_{a'} Q_{\hat{\theta}}(s',a') - Q_\theta(s,a))^2]$$

e.g.)

$$Q(s,a) \leftarrow r(s,a) + \gamma \cdot \mathbb{E}_{(s,a,s')\sim\mathcal{D},\, a'\sim\pi_{new}(\cdot|s')}[Q(s',a')]$$

$$\pi_{new}(a|s) = \arg\max_{\pi} \mathbb{E}_{a\sim\pi(a|s)}[Q(s,a)]$$



Offline RL

- How? (Prior works)
  - Directly constraint the policies, (Kumar et al. '19; Wu et al. '19; Levin et al. '20...)
  - Regularization on Q, ... (Kumar et al. '20; Kostrikov et al. '21; Fakoor et al. '21...)

e.g.)

$$\boxed{L_{TD}(\theta) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}}[(r(s,a) + \gamma \max_{a'} Q_{\hat{\theta}}(s',a') - Q_\theta(s,a))^2]}$$

$$Q(s,a) \leftarrow r(s,a) + \gamma \cdot \mathbb{E}_{(s,a,s')\sim\mathcal{D},\, a'\sim\pi_{new}(\cdot|s')}[Q(s',a')]$$

$$\pi_{new}(a|s) = \arg\max_{\pi} \mathbb{E}_{a\sim\pi(a|s)}[Q(s,a)] \;\; \text{s.t. } D_{KL}(\pi\|\pi_\beta) \le \epsilon$$

- A couple of conflicting aims:
  1. **Improve over** the behavior policy π_β (that collected the dataset)
  2. **Not too far** from behavior policy π_β (**⏬ distribution shift)**

- Anyway, <u>Requires **unseen actions.**</u> → Or does it?

Overview: Implicit Q-Learning (IQL)

- **No need to evaluate unseen actions.**
- Still improves over the best behavior.
- Aims to (but not directly) learn

$$L(\theta) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\left[\left(r(s,a) + \gamma \max_{\substack{a' \in \mathcal{A} \\ \text{s.t. } \boxed{\pi_\beta(a'|s')>0}}} Q_{\hat{\theta}}(s',a') - Q_\theta(s,a)\right)^2\right].$$

'max' is taken over the
Support of data distribution.

- Main ingredients:
  - SARSA
  - Expectile Regression
  - "Lucky sample" problem

SARSA : a starting point

- Unlike simple TD loss, the SARSA-like objective

$$L(\theta) = \mathbb{E}_{(s,a,s',a')\sim\mathcal{D}}[(r(s,a) + \gamma Q_{\hat{\theta}}(s',a') - Q_\theta(s,a))^2].$$

does not require out-of-sample actions.

- "It uses mean squared error (MSE) that fits $Q_\theta(s,a)$ to predict the mean statistics of the TD targets."

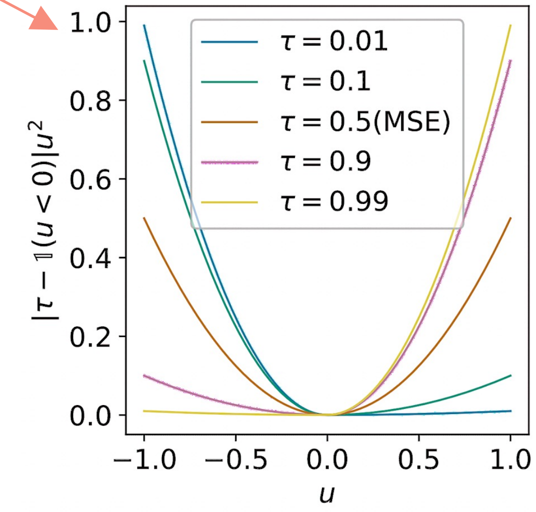
- Problem: poor performance on more <u>complex tasks</u>
  - *e.g.,* multi-step dynamic programming.

Expectile Regression (1)

- Given a random variable X.
- τ^{th} expectile ($\tau \in (0,1)$) of X: the solution m_τ of

$$\arg\min_{m_\tau} \mathbb{E}_{x\sim X}[L_2^\tau(x - m_\tau)], \text{ where } L_2^\tau(u) = \underbrace{|\tau - \mathbb{1}(u<0)|}_{\begin{cases}\tau & (u \geq 0),\\ 1-\tau & (u<0).\end{cases}} u^2.$$

- τ : How much we weight the cost for $x \geq m_\tau$?
  - $\tau = 0.5$: vanilla least square.
  - Larger τ ➡ Larger m_τ.
  - ***Lemma 1***. $\lim_{\tau \to 1-} m_\tau =$ (**supremum** of bdd r.v. X).

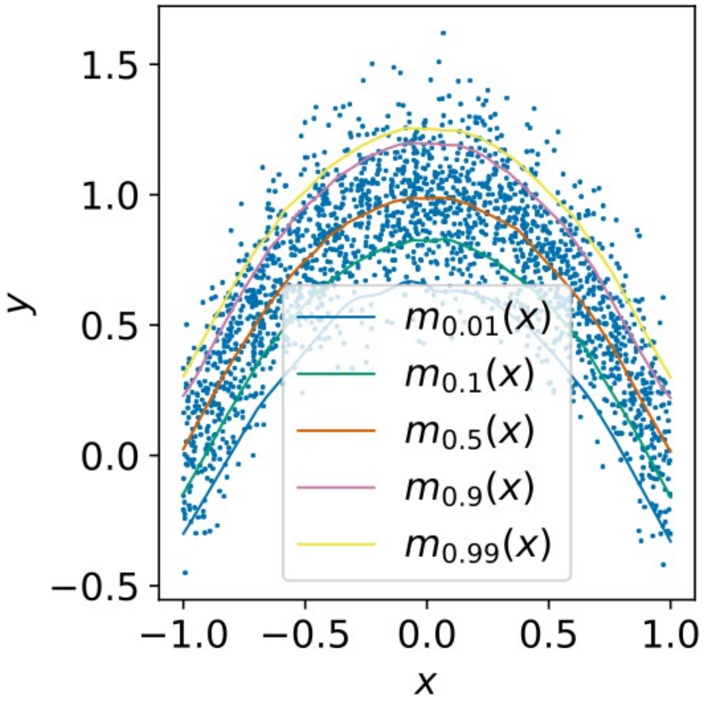


Expectile Regression (2)

- Expectile regression: to obtain expectile of $Y|X = x$.

$$\underset{m_\tau(x)}{\arg\min}\, \mathbb{E}_{(x,y)\sim\mathcal{D}}[L_2^\tau(y - m_\tau(x))].$$

  - $\tau = 0.5$: conditional mean statistics
  - $\tau \approx 1$: approximates maximum operator
    over in-support values of y.



Learning Value Functions with expectile regression

- First trial: $L(\theta) = \mathbb{E}_{(s,a,s',a') \sim \mathcal{D}}[L_2^{\tau}(r(s,a) + \gamma Q_{\hat{\theta}}(s',a') - Q_{\theta}(s,a))].$

SARSA Expectile Regression

👍 SARSA + Expectile regression

- ➢ No need for out-of-sample actions
- ➢ Approximate max$(r + \gamma Q_{\hat{\theta}})$ over data <u>distribution.</u>

👎 Problem? : "**Lucky samples**"

- ➢ $L(\theta)$ incorporates stochasticity from transitions $p(s'|s,a)$
- ➢ Large $r + \gamma Q_{\hat{\theta}}$ may caused by a lucky transition into a good state.

Solution: Separate Value functions

1. V: approximates expectile only w.r.t. action distrib.

$$L_V(\psi) = \mathbb{E}_{(s,a) \sim \mathcal{D}}[L_2^\tau(Q_{\hat{\theta}}(s,a) - V_\psi(s))].$$

- Still approximates **max $\boldsymbol{Q_{\hat{\theta}}}$** (in-support; if τ is large)
- ≈ **Implicit policy/value improvement**

2. Q: update with MSE loss & V

$$L_Q(\theta) = \mathbb{E}_{(s,a,s') \sim \mathcal{D}}[(r(s,a) + \gamma V_\psi(s') - Q_\theta(s,a))^2].$$

- Average out the stochasticity due to transitions
- ≈ **Policy/value evaluation**

Full Algorithm : Two-Stage (IQL → AWR)

Algorithm 1 Implicit Q-learning

Initialize parameters $\psi, \theta, \hat{\theta}, \phi$.
TD learning (IQL):
for each gradient step **do**
  $\psi \leftarrow \psi - \lambda_V \nabla_\psi L_V(\psi)$
  $\theta \leftarrow \theta - \lambda_Q \nabla_\theta L_Q(\theta)$
  $\hat{\theta} \leftarrow (1-\alpha)\hat{\theta} + \alpha\theta$
end for
Policy extraction (AWR):
for each gradient step **do**
  $\phi \leftarrow \phi - \lambda_\pi \nabla_\phi L_\pi(\phi)$
end for

---

1. TD Learning with Implicit Q-Learning
   1) Update V (expectile approximation/improvement)
   2) Update Q (state-action value evaluation)
   3) Update target net $\hat{\theta}$ (Polyak averaging)

---

2. Policy extraction by **AWR**
   - Advantage-Weighted Regression
     (Peters & Schaal, 2007; Peng et al., 2019)
   - Also doesn't need external actions.
   - "learns a policy that maximizes the Q-values subject to a distribution constraint."

$$L_\pi(\phi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}[\exp(\beta(Q_{\hat{\theta}}(s,a) - V_\psi(s))) \log \pi_\phi(a|s)],$$

Theoretical results

Suppose IQL converged to $V_\psi \to V_\tau$.

Let Q^* be optimial state-action value under behavior policy constraint.

- Lemma 2. $\tau_1 < \tau_2 \implies V_{\tau_1}(s) \le V_{\tau_2}(s) \quad (\forall s)$

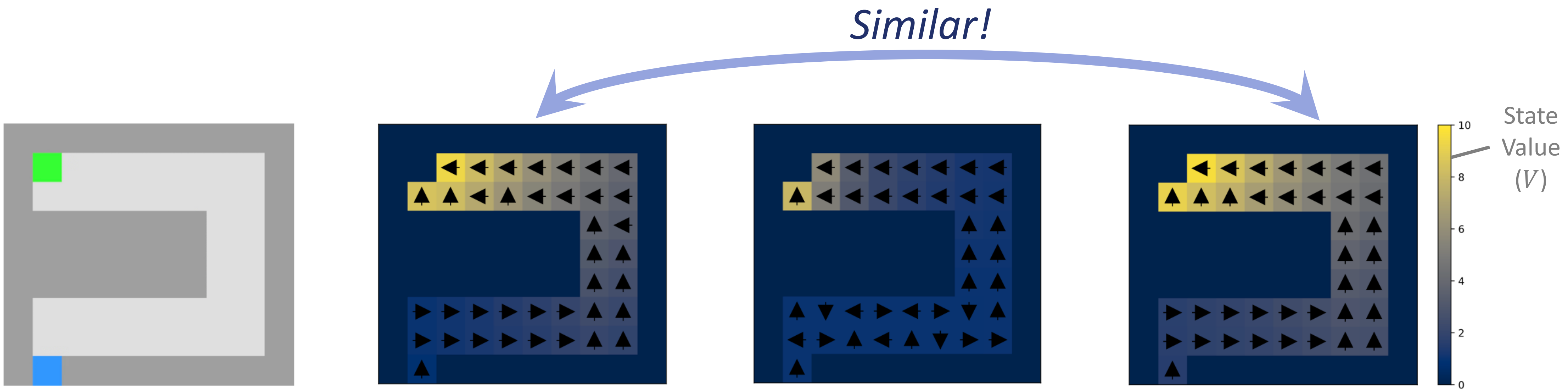
- Theorem 3. $\lim_{\tau \to 1} V_\tau(s) = \max_{\substack{a \in \mathcal{A} \\ s.t.\ \pi_\beta(a|s) > 0}} Q^*(s, a).$

However, there is a trade-off:

1) Approximation : If $\tau < 1$ is large, **we approximate max Q^* better.**
2) Optimization : If $\tau < 1$ is large, **difficult to optimize.**

- **Thus, τ is regarded as a hyperparameter.**

Experiments (1) One-step DP v.s. IQL



(a) toy maze MDP (b) true optimal $V^\star$ (c) One-step Policy Eval. (d) IQL

Only SARSA-like objective:
The values decay faster.
(Brandfonbrener et al., 2021;
Wang et al., 2018;
Gulcehre et al., 2021)

$(\tau = 0.95)$

Experiments (2) IQL v.s. other offline methods

Table 1: Averaged normalized scores on MuJoCo locomotion and Ant Maze tasks. Our method outperforms prior methods on the challenging Ant Maze tasks, which require dynamic programming, and is competitive with the best prior methods on the locomotion tasks.

Dataset	BC	10%BC	BCQ	DT	ABM	AWAC	Onestep RL	TD3+BC	CQL	IQL (Ours)
halfcheetah-m-v2	42.6	42.5	**47.0**	42.6±0.1	53.6	43.5	**48.4±0.1**	**48.3±0.3**	44.0±5.4	**47.4±0.2**
hopper-m-v2	52.9	56.9	56.7	**67.6±1.0**	0.7	57.0	59.6±2.5	59.3±4.2	58.5±2.1	**66.2±5.7**
walker2d-m-v2	75.3	75.0	72.6	74.0±1.4	0.5	72.4	**81.8±2.2**	83.7±2.1	72.5±0.8	78.3± 8.7
halfcheetah-m-r-v2	36.6	40.6	40.4	36.6±0.8	**50.5**	40.5	38.1±1.3	**44.6±0.5**	**45.5±0.5**	**44.2±1.2**
hopper-m-r-v2	18.1	75.9	53.3	82.7±7.0	49.6	37.2	**97.5±0.7**	60.9±18.8	**95.0±6.4**	**94.7±8.6**
walker2d-m-r-v2	26.0	62.5	52.1	66.6±3.0	53.8	27.0	49.5±12.0	**81.8±5.5**	77.2±5.5	73.8±7.1
halfcheetah-m-e-v2	55.2	**92.9**	**89.1**	86.8±1.3	18.5	42.8	**93.4±1.6**	**90.7±4.3**	**91.6±2.8**	86.7±5.3
hopper-m-e-v2	52.5	**110.9**	81.8	**107.6±1.8**	0.7	55.8	103.3±1.9	98.0±9.4	**105.4±6.8**	91.5±14.3
walker2d-m-e-v2	**107.5**	**109.0**	**109.5**	**108.1±0.2**	3.5	74.5	**113.0±0.4**	**110.1±0.5**	**108.8±0.7**	**109.6±1.0**
locomotion-v2 total	466.7	**666.2**	602.5	**672.6±16.6**	231.4	450.7	**684.6±22.7**	**677.4±44.5**	**698.5±31.0**	**692.4±52.1**
antmaze-u-v0	54.6	62.8	**89.8**	59.2	59.9	56.7	64.3	78.6	74.0	**87.5 ± 2.6**
antmaze-u-d-v0	45.6	50.2	**83.0**	53.0	48.7	49.3	60.7	71.4	**84.0**	62.2 ± 13.8
antmaze-m-p-v0	0.0	5.4	15.0	0.0	0.0	0.0	0.3	10.6	61.2	**71.2 ± 7.3**
antmaze-m-d-v0	0.0	9.8	0.0	0.0	0.5	0.7	0.0	3.0	53.7	**70.0 ± 10.9**
antmaze-l-p-v0	0.0	0.0	0.0	0.0	0.	0.0	0.0	0.2	15.8	**39.6±5.8**
antmaze-l-d-v0	0.0	6.0	0.0	0.0	0.0	1.0	0.0	0.0	14.9	**47.5±9.5**
antmaze-v0 total	100.2	134.2	187.8	112.2	109.1	107.7	125.3	163.8	303.6	**378.0±49.9**
total	566.9	800.4	790.3	784.8	340.5	558.4	809.9	841.2	1002.1	**1070.4±102.0**
kitchen-v0 total	**154.5**	-	-	-	-	-	-	-	144.6	**159.8±22.6**
adroit-v0 total	104.5	-	-	-	-	-	-	-	93.6	**118.1±30.7**
total+kitchen+adroit	825.9	-	-	-	-	-	-	-	1240.3	**1348.3±155.3**
runtime	10m	10m		960m		20m	20m*	20m	80m	20m

*: Note that it is challenging to compare one-step and multi-step methods directly. Also, Brandfonbrener et al. (2021) reports results for a set of hyperparameters, such as batch and network size, that is significantly different from other methods. We report results for the original hyperparameters and runtime for a comparable set of hyperparameters.

IQL outperforms on Ant Maze Tasks

More or less Computation-efficient

- Ant Maze Task:
  - "contain very few or no near-optimal trajectories, making them very challenging for one-step methods. "

Experiments (3) IQL + online fine-tuning

Dataset	AWAC	CQL	IQL (Ours)
antmaze-umaze-v0	56.7 → 59.0	70.1 → **99.4**	**88.0** → **96.3**
antmaze-umaze-diverse-v0	49.3 → 49.0	31.1 → **99.4**	**67.0** → 49.0
antmaze-medium-play-v0	0.0 → 0.0	23.0 → 0.0	**69.0** → **89.2**
antmaze-medium-diverse-v0	0.7 → 0.3	23.0 → 32.3	**71.8** → **91.4**
antmaze-large-play-v0	0.0 → 0.0	1.0 → 0.0	**36.8** → **51.8**
antmaze-large-diverse-v0	1.0 → 0.0	1.0 → 0.0	**42.2** → **59.8**
antmaze-v0 total	107.7 → 108.3	151.5 → 231.1	**374.8** → **437.5**
pen-binary-v0	**44.6** → **70.3**	31.2 → 9.9	37.4 → 60.7
door-binary-v0	**1.3** → **30.1**	0.2 → 0.0	0.7 → **32.3**
relocate-binary-v0	**0.8** → 2.7	0.1 → 0.0	0.0 → **31.0**
hand-v0 total	**46.7** → 103.1	31.5 → 9.9	38.1 → **124.0**
total	154.4 → 211.4	182.8 → 241.0	**412.9** → **561.5**

- Offline pre-training ∈ {AWAC, CQL, IQL} → Online fine-tuning (1M steps).

Discussion

- Can we even more accelerate IQL by reducing the number of stages (*e.g.*, two-stage → one-stage), although IQL is still fast?
  - Is the time complexity of policy extraction (AWR) really faster than IQL stage?
  - Can't we use the idea of both IQL and policy extraction to devise a one-stage algorithm?
- Or any questions?