NeurIPS 2021

Curriculum Offline Imitation Learning

Minghuan Liu[1*], Hanye Zhao[1*], Zhengyu Yang[1], Jian Shen[1],

Weinan Zhang[1], Li Zhao[2], Tie-Yan Liu[2]

*Equal contribution

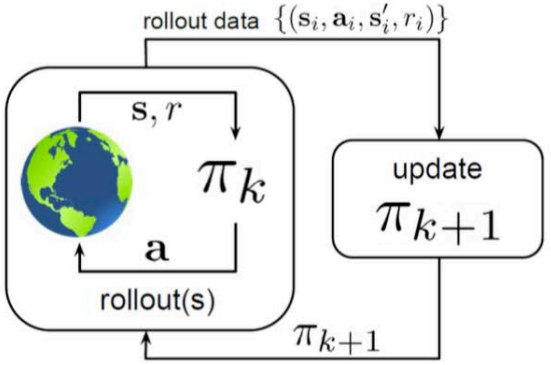
[1] Shanghai Jiao Tong University, [2] Microsoft Research Asia

minghuanliu

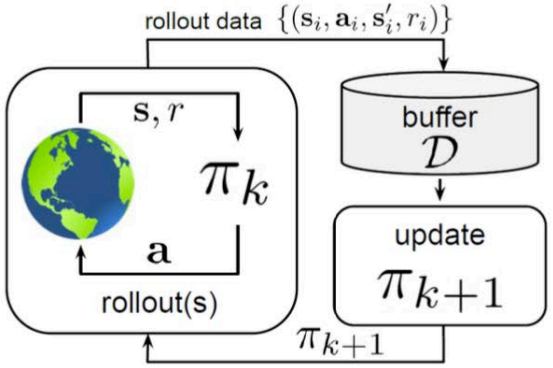
minghuanliu@sjtu.edu.cn

Online RL and Offline RL

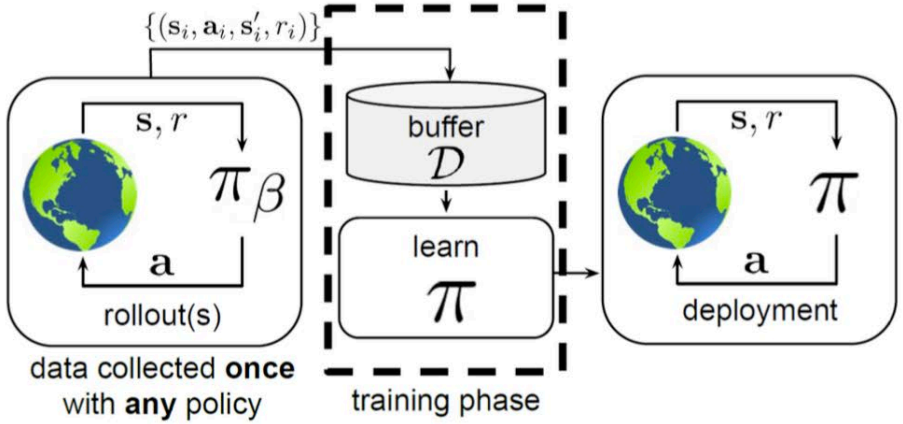
On-policy RL



Off-policy RL



Offline RL



- Offline RL focus on learning effective policies with static dataset
- Policy is trained without any deployment
- Offline RL methods are totally data-driven RL

Levine, Kumar, Tucker, Fu. **Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems.** '20

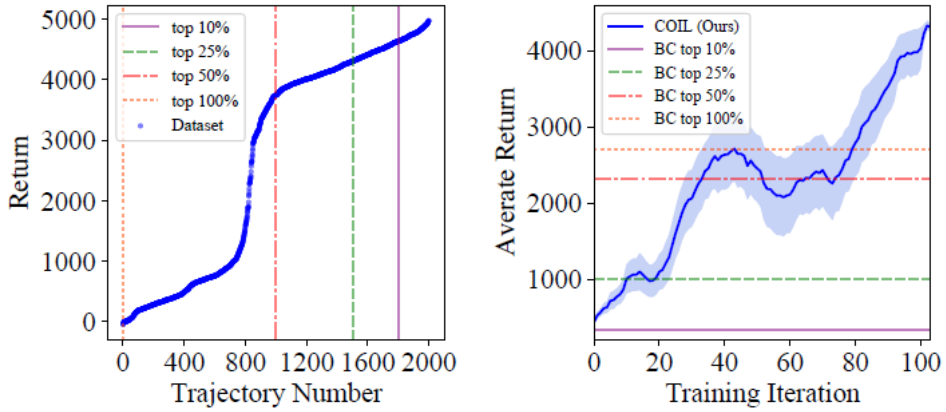
Motivation

- Challenges in offline reinforcement learning (RL)
  - Learn a value function offline cause serious extrapolation errors and **instable**
  - Directly imitate from the dataset lead to a mediocre behavior but **stable**
- What if we try stable imitation but prevent mediocre?
- What is the problem that leads to the mediocre behavior?

Quantity-quality dilemma on mixed dataset

- Direct imitation (behavior cloning, BC) requires both quantity and quality of the demonstration data

- An illustrative example:
  - Less data owns higher quality but less quantity, and thus cause serious compounding error problems
  - More data provides a larger quantity, yet its mean quality becomes worse
  - All lead to mediocre behaviors

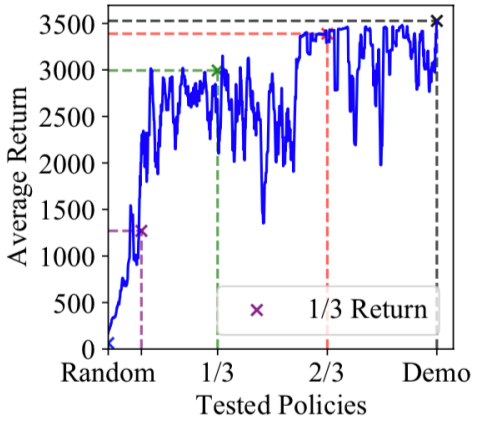


(a) Ordered trajectories. (b) Returns of BC v.s. COIL.

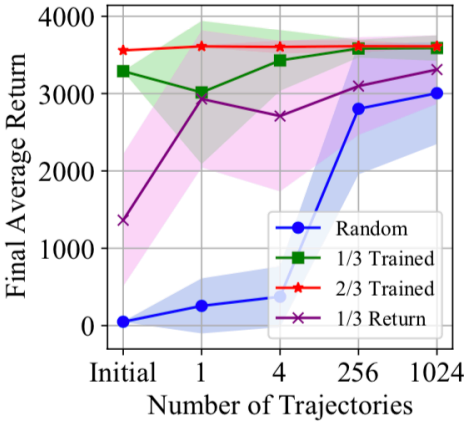
Figure 1: Examples of the quality-quantity dilemma for BC. (a) Trajectories of the Walker2d-final dataset ordered by their accumulated return. (b) Performances of behavior cloning (BC) for learning the top 10%, 25%, 50%, and 100% trajectories of the dataset.

Possible Solution: an empirical observation

- Agent can imitate a neighboring policy with much fewer samples
- BC with different initialization:
  - Online-trained policy checkpoints at different training iterations as the initiated policy to train an BC agent
  - BC from random tends to fail with a small number of high-quality demonstrations but can learn well from large-quantity and high-quality data
  - Requirement of quantity can be highly relaxed as the similarities between the demonstrated policy and the initialized imitating policy increase



(a) Online training curve.



(b) Final performance of BC.

Theoretical Explaination

- Three bound terms:
  - ***BC gap***: empirical error & the training error
  - ***data gap***: the number of samples and complexity of the state space, an intrinsic gap due to the dataset and the environment

Theorem 1 (Performance bound of BC). *Let* Π *be the set of all deterministic policy and* $|\Pi| = |A|^{|S|}$. *Assume that there does not exist a policy* $\pi \in \Pi$ *such that* $\pi(s^i) = a^i, \forall i \in \{1, \cdots, |\mathcal{D}|\}$. *Let* $\hat{\pi}_b$ *be the empirical behavior policy and the corresponding state marginal occupancy is* $\rho_{\hat{\pi}_b}$. *Suppose BC begins from initial policy* π_0, *and define* ρ_{π_0} *similarly. Then, for any* $\delta > 0$, *with probability at least* $1 - \delta$, *the following inequality holds:*

$$D_{\mathrm{TV}}(\rho_\pi(s,a) \| \rho_{\pi_b}(s,a)) \leq c(\pi_0, \pi_b, |\mathcal{D}|)$$

$$\text{where} \quad c(\pi_0, \pi_b, |\mathcal{D}|) = \frac{1}{2}\sum_{s \notin \mathcal{D}} \rho_{\pi_b}(s) + \frac{1}{2}\sum_{s \notin \mathcal{D}} |\rho_\pi(s) - \rho_{\pi_0}(s)| + \underbrace{\frac{1}{2}\sum_{s \notin \mathcal{D}} |\rho_{\pi_0}(s) - \rho_{\pi_b}(s)|}_{\textit{initialization gap}}$$

$$+ \underbrace{\frac{1}{2}\sum_{s \in \mathcal{D}} |\rho_\pi(s) - \rho_{\hat{\pi}_b}(s)| + \frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|} \mathbb{I}\left[\pi(s^i) \neq a^i\right]}_{\textit{BC gap}} + \underbrace{\left[\frac{\log|\mathcal{S}| + \log(2/\delta)}{2|\mathcal{D}|}\right]^{\frac{1}{2}} + \left[\frac{\log|\Pi| + \log(2/\delta)}{2|\mathcal{D}|}\right]^{\frac{1}{2}}}_{\textit{data gap}}$$

  - ***initialization gap***: distance between the state marginal distribution of the initial policy and behavior policy out of the dataset
  - ***Additional second term***: hard to analyze from theory, we estimate the empirical discrepancy outside the dataset

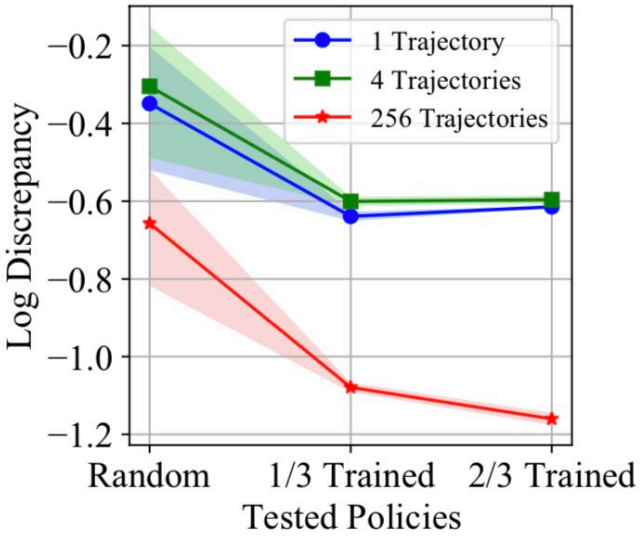
Theoretical Explaination (cont.)

- ***Additional second term***: we estimate the empirical discrepancy outside the dataset (see details in the paper)

$$\frac{1}{2}\sum_{s \neq D} |\hat{\rho}_{\pi}(s) - \hat{\rho}_{\pi_0}(s)|$$

the second term decreases as the initialized policy gets close to the demonstrated policy

Intuitively, this is because of the poor generalization on unseen states, and the error can be further reduced with a larger dataset.



(c) Empirical discrepancy.

Insight for the solution

- **Brief conclusion**:
  - The asymptotic performance of BC is highly related to the discrepancy between the initialized policy and the demonstrated policy
  - A close-to-demonstration policy can easily imitate the demonstrated policy with fewer samples
  - the distance between the initialized policy and the demonstrated policy is far, then successfully mimicking the policy will require much more samples
- **Key insight**:
  - Adaptively imitating the close policies with a small number of samples and finally terminates with the optimal behavior policy of the dataset

More formal insight from online RL

- Online RL as imitating optimal policies
  - Objective: obtaining the most accumulated rewards

$$\underset{\pi}{\text{maximize}}\, \mathbb{E}_{\tau\sim\pi}[R(\tau)]$$

  - Under the principle of maximum entropy, finding the optimal policy through RL is equivalent to imitating the optimal policy

$$P^*(\tau) \propto \exp\left(R(\tau)\right) \qquad \underset{\pi}{\text{minimize}}\, D_{KL}(P_\pi(\tau)\|P^*(\tau))$$

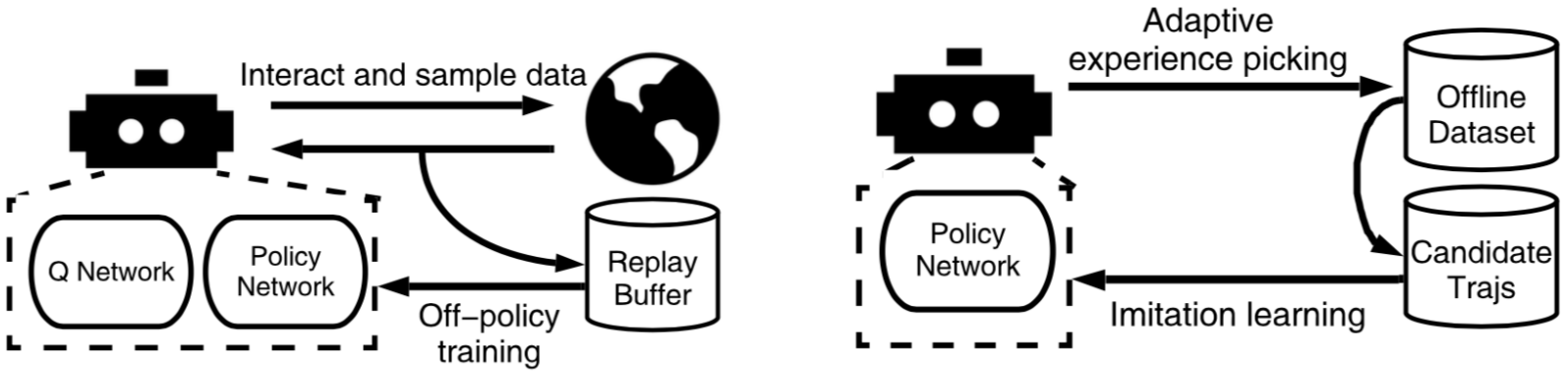
  - Thus, at each training iteration, the new policy is obtained through updating follows the direction of minimizing the KL divergence (a stage of curriculum)

$$\pi^{i+1} = \pi^i - \nabla_\pi D_{KL}(P_\pi(\tau)\|P^*(\tau))$$

Curriculum offline imitation learning

- Offline RL as adaptive imitation
  - At every training stage, the agent updates its policy by adaptively selecting trajectories from the given dataset as the imitating target (a stage of curriculum)

$$\pi^{i+1} = \pi^i - \nabla_\pi D_{KL}(P_{\tilde{\pi}^i}(\tau) \| P_\pi(\tau))$$
$$\text{s.t. } \mathbb{E}_{\tilde{\pi}} \left[D_{KL}(\tilde{\pi}^i(\cdot|s) \| \pi^i(\cdot|s)) \right] \leq \epsilon$$
$$R^i_{\tilde{\pi}} - R^i_\pi \leq \delta$$



(a) Online off-policy training. (b) Curriculum offline imitation learning.

Figure 3: Comparison between online off-policy training and curriculum offline imitation learning.

Practical solutions for the first constraint

- Adaptive experience picking by neighboring policy assessment

Observation 1. *Under the assumption that each trajectory $\tau_{\tilde{\pi}}$ in the dataset $\mathcal{D}$ is collected by an unknown deterministic behavior policy $\tilde{\pi}$ with an exploration ratio β. The requirement of the KL divergence constraint $\mathbb{E}_{\tilde{\pi}}\left[D_{KL}(\tilde{\pi}(\cdot|s)\|\pi(\cdot|s))\right] \leq \epsilon$ suffices to finding a trajectory that at least $1-\beta$ state-action pairs are sampled by the current policy π with a probability of more than ϵ_c such that $\epsilon_c \geq 1/\exp\epsilon$, i.e.:*

$$\mathbb{E}_{(s,a)\in\tau_{\tilde{\pi}}}\left[\mathbb{I}(\pi(a|s) \geq \epsilon_c)\right] \geq 1-\beta\ , \tag{8}$$

  - Therefore, to find whether a trajectory is sampled by a neighboring policy, we calculate the probability of sampling the action at each state by the current policy in the trajectory for every timestep $\{\pi(a_0|s_0),\cdots,\pi(a_h|s_h)\}$, where h is the horizon of the trajectory.
  - We set $\beta = 0.05$ and then we find N nearest policies that matches (8) instead of choosing an ϵ_c.

Practical solutions for the second constraint

- Target: refrain the performance from getting worse by imitating to a poorer target than the current level of the imitating policy
- Practical way: adopt a return filtering mechanism that filtrates the useless, poor-behaved trajectories.
  - initialize the return filter V with 0
  - update the value at each curriculum by moving average the return of the selected trajectories
  - For example, choose $\{\tau\}_n^1$ from dataset at iteration k

$$V_k = (1-\alpha)\cdot V_{k-1} + \alpha \cdot \min\{R(\tau)\}_1^n$$

Overall algorithm

Algorithm 1 Curriculum Offline Imitation Learning (COIL)

Require: Offline dataset $\mathcal{D}$, number of trajectories picked at each curriculum N, moving window of the return filter α, number of training iteration L, batch size B, number of pre-train times T, and the learning rate η.
Initialize policy π with random parameter θ.
Initialize the return filter $V = 0$.
if $\mathcal{D}$ is collected by a single policy **then**
  Do pre-training for T times using BC.
end if
while $\mathcal{D} \neq \emptyset$ **do**
  for all $\tau_i \in \mathcal{D}$ **do**
    Calculate $\tau_i(\pi) = \{\pi(a_0^i|s_0^i), \pi(a_1^i|s_1^i), \cdots, \pi(a_h^i|s_h^i)\}$.
    Sort $\tau_i(\pi)$ into $\{\pi(\bar{a}_0^i|\bar{s}_0^i), \pi(\bar{a}_1^i|\bar{s}_1^i), \cdots, \pi(\bar{a}_h^i|\bar{s}_h^i)\}$ in an ascending order, such that $\pi(\bar{a}_j^i|\bar{s}_j^i) \leq \pi(\bar{a}_{j+1}^i|\bar{s}_{j+1}^i), \quad j \in [0, h-1]$
    Choose $s(\tau_i) = \pi(\bar{a}_{\lfloor \beta h \rfloor}^i|\bar{s}_{\lfloor \beta h \rfloor}^i)$ as the criterion of τ_i.
  end for
  Select $N = \min\{N, |\mathcal{D}|\}$ trajectories $\{\bar{\tau}\}_1^N$ with the highest $s(\tau)$ as a new curriculum.
  Initialize a new replay buffer $\mathcal{B}$ with $\{\bar{\tau}\}_1^N$.
  $\mathcal{D} = \mathcal{D} \backslash \{\bar{\tau}\}_1^N$.
  for $n = 1 \rightarrow L \times N$ **do**
    Draw a random batch $\{(s, a)\}_1^B$ from $\mathcal{B}$.
    Update π_θ using behavior cloning

$$\theta \leftarrow \theta - \eta \nabla_\theta \sum_{j=1}^{B} [-\log \pi_\theta(a_j|s_j)]$$

  end for
  Update the return filter $V \leftarrow (1-\alpha)V + \alpha \cdot \min\{R(\bar{\tau})\}_1^N$.
  Filter $\mathcal{D}$ by $\mathcal{D} = \{\tau \in \mathcal{D} \mid R(\tau) \geq V\}$.
end while

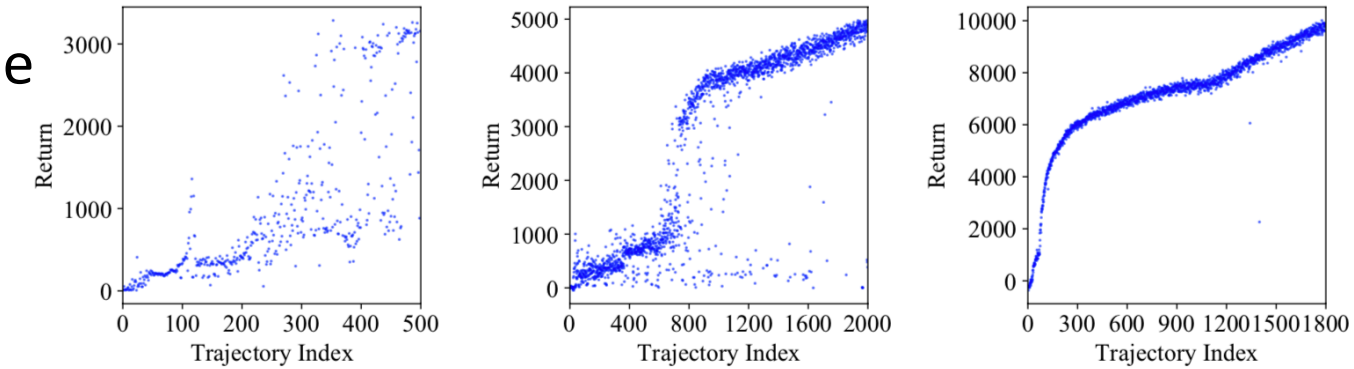
- Simple but effective curriculum offline imitation learning (COIL)
- COIL holds an experience pool that contains the candidate trajectories to be selected.
- Every training time creates a stage of the curriculum where the agent selects appropriate trajectories as the imitation target from the pool and learns them via direct BC.
- After training, the used experience will be cleaned from the pool, and the return filter also filtrates a set of trajectories.

Experimental design

- Learn from online learning experience
  - Offline dataset contains all the training experience of an SAC agent from scratch to convergence (including exploration actions)
- Benchmark scores
  - Compare with state-of-the art offline RL algorithms
- Ablation studies
  - How the important hyperparameters can be determined due to the dataset

Learn from online learning experience

- Offline dataset contains the interaction data during the whole training procedure
- Compare with imitation-based baselines:
  - BC, AWR, BAIL
- RL-based baseline
  - CQL



(a) Hopper-final. (b) Walker2d-final. (c) HalfCheetah-final.

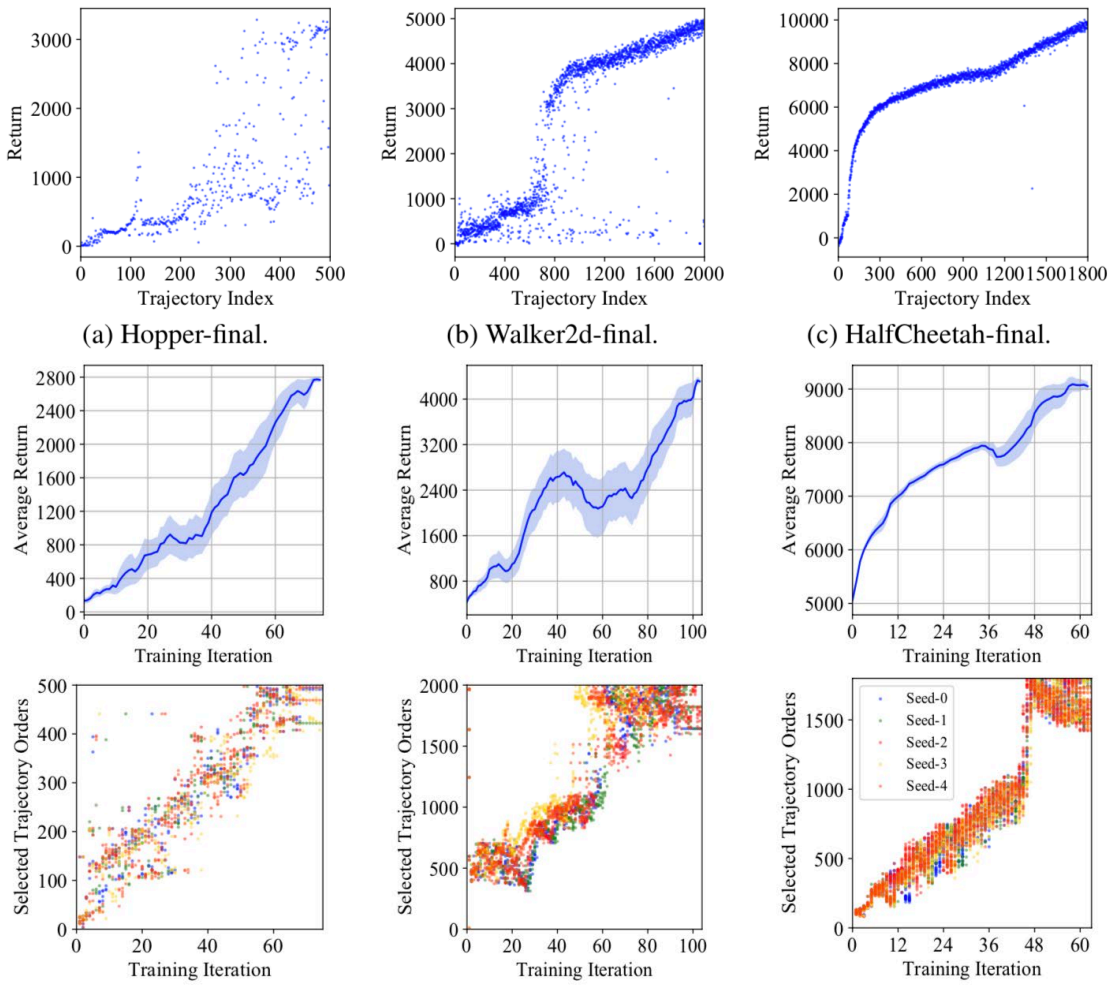
Figure 4: Trajectories in final datasets, sorted in the online training order.

Table 1: Average performances on *final* datasets, the means and standard deviations are calculated over 5 random seeds. *Behavior* shows the average performance of the behavior policy that collects the data.

Dataset	Expert (SAC)	Behavior	BC	AWR	BAIL	CQL	COIL (Ours)
`hopper-final`	3163.3 (44.4)	974.5	1480.4 (800.2)	1609.7 (489.7)	2296.9 (915.9)	501.5 (227.5)	**2872.5 (133.9)**
`walker2d-final`	4866.03 (68.6)	2684.9	2099.6 (2101.3)	3213.8 (1682.9)	**4236.2 (1531.1)**	2604.3 (1937.6)	**4391.3 (697.8)**
`halfcheetah-final`	9739.1 (113.6)	7122.4	6125.6 (3910.9)	7600.9 (1153.4)	9745.0 (880.3)	**10882.0 (1042.7)**	**9328.5 (1940.6)**

Learn from online learning experience (cont.)

- Offline dataset contains the interaction data during the whole training procedure (*final* datasets)
- Learning curves shown are stable and following similar shapes as the ordered datasets.
- COIL keeps a similar training path as the online agent, thanks to the experience picking strategy and the return filter
- COIL finally terminates with a near-data-optimal policy, suggesting a nice property that the last offline model can be a great model for deployment



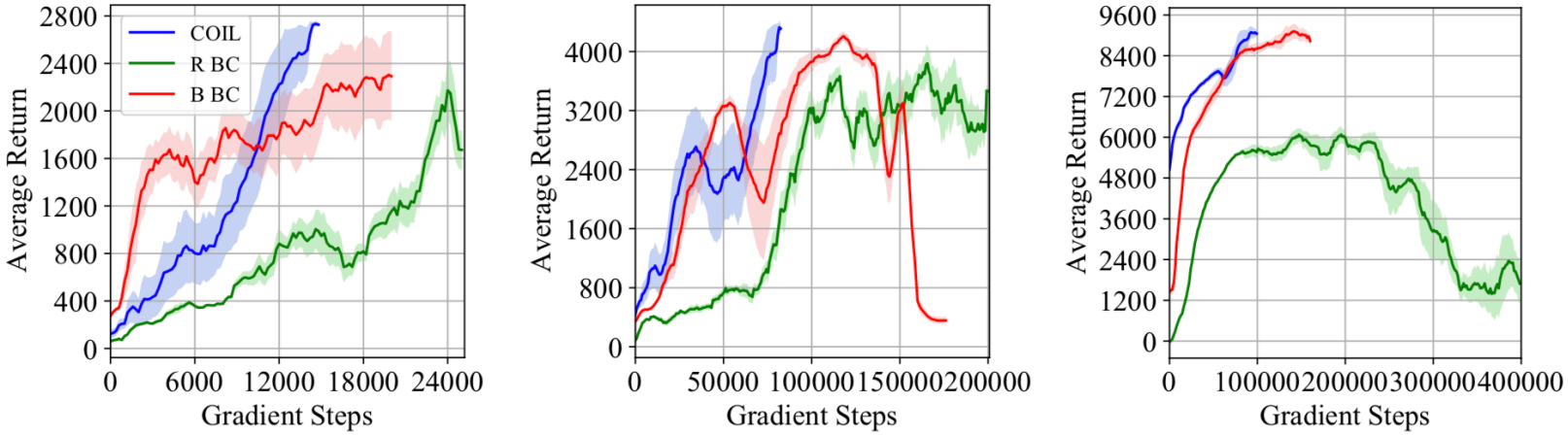
(a) Hopper-final. (b) Walker2d-final. (c) HalfCheetah-final.

(a) Hopper-final. (b) Walker2d-final. (c) HalfCheetah-final.

Figure 5: Training curves and orders of selected trajectories.

Compared with naïve strategies

- *Return-ordered BC (RBC)*: picks N trajectories with the lowest returns for each curriculum to perform behavioral cloning, and then removes them from the dataset.
- *Buffer-shrinking BC (BBC)*: begin its training with the entire dataset in the buffer; after a fixed number of gradient steps, it shrinks the buffer by discarding p% of trajectories with the lowest returns



(a) Hopper-final. (b) Walker2d-final. (c) HalfCheetah-final.

Figure 8: Comparison of training curves between COIL and Return-ordered BC (*R BC*) and Buffer-shrinking BC (*B BC*) on *final* datasets with the same batch size. Different strategies terminate with different gradient steps.

D4RL benchmarks

- BC is able to approach or outperform the performance of the behavior policy on the datasets generated from a single policy
- COIL achieves the performance of the optimal behavior policy on most datasets, and doing so will allow COIL to beat or compete with the state-of-the-art results

Table 2: Average performance on D4RL datasets. Results in gray columns is our implementation that are tested among 5 random seeds. The other results are based on numbers reported in D4RL among three random seeds without standard deviations. *Best 1%* shows the average return of the top 1% best trajectories, representing the performance of the optimal behavior policy; *Behavior* shows the average performance of the dataset.

Dataset	Expert (D4RL)	Behavior	Best 1%	BC (D4RL)	BC (Ours)	COIL (Ours)	BAIL	MOPO	SoTA (D4RL)
hopper-random	3234.3	295.1	340.4	299.4	330.1 (3.5)	378.5 (15.2)	318.0 (5.1)	**432.6**	376.3
hopper-medium	3234.3	1018.1	3076.4	923.5	1690.1 (852.0)	**3012.0 (332.2)**	1571.5 (900.7)	862.1	2557.3
hopper-medium-replay	3234.3	466.9	1224.8	364.4	853.6 (397.5)	1333.7 (271.1)	808.7 (192.5)	**3009.6**	1227.3
hopper-medium-expert	3234.3	1846.8	3735.7	**3621.2**	**3527.4 (504.1)**	**3615.5 (168.9)**	2435.9 (1265.2)	1682.0	3588.5
walker2d-random	4592.3	1.1	25.0	73.0	171.0 (59.3)	320.5 (70.7)	130.8 (87.2)	**597.1**	336.3
walker2d-medium	4592.3	496.4	3616.8	304.8	1521.9 (1381.3)	2184.5 (1279.2)	1242.4 (1545.7)	643.0	**3725.8**
walker2d-medium-replay	4592.3	356.6	1593.7	518.6	715.0 (406.5)	1439.9 (347.0)	532.9 (359.0)	**1961.1**	1227.3
walker2d-medium-expert	4592.3	1059.7	5133.4	297.0	**3488.6 (1815.1)**	**4012.3 (1463.0)**	**3633.9 (1839.7)**	2526.0	**5097.3**
halfcheetah-random	12135.0	-302.6	-85.4	-17.9	-124.3 (60.6)	-0.3 (0.7)	-96.4 (49.7)	3957.2	**4114.8**
halfcheetah-medium	12135.0	3944.9	4327.7	4196.4	3276.4 (1500.7)	4319.6 (243.7)	4277.6 (564.9)	4987.5	**5473.8**
halfcheetah-medium-replay	12135.0	2298.2	4828.4	4492.1	4035.7 (365.4)	4812.0 (148.7)	3854.8 (966.3)	**6700.6**	5640.6
halfcheetah-medium-expert	12135.0	8054.4	12765.4	4169.4	633.2 (2152.9)	**10535.6 (3334.9)**	**9470.3 (4178.9)**	7184.7	7750.8

Ablation studies

- Critical hyperparameter: α
  - α: moving average rate of the return filter
  - Determines the rate of filtering out the bad trajectories as the agent imitates a better policy where small α corresponds to a high filtering rate
  - Highly influenced by the changes in returns of the trajectories in the dataset
  - Two datasets, showing that
    - a small α should be assigned for a dataset where the return of trajectories changes rapidly
    - A large α should be assigned to a dataset where the returns are almost the same

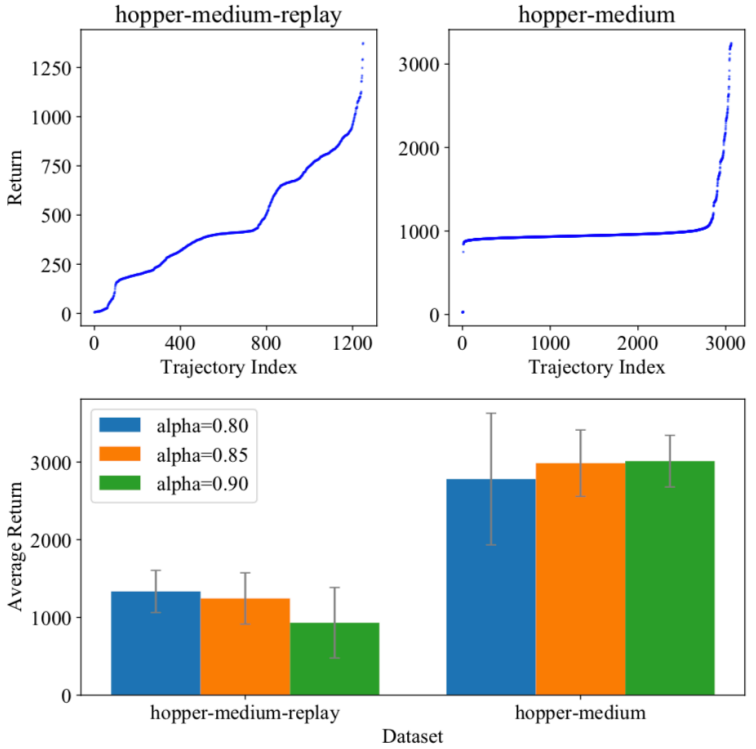
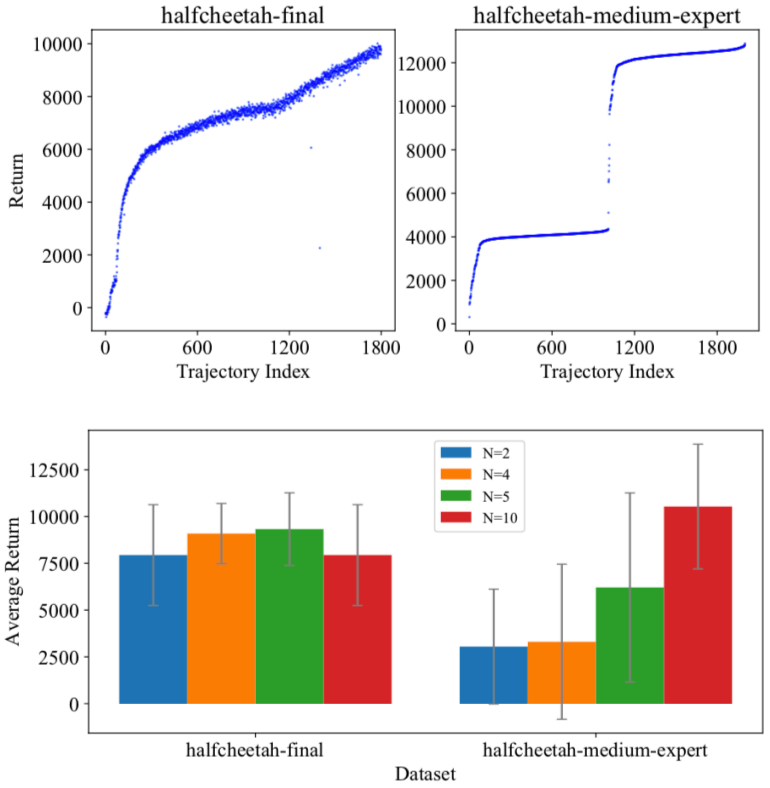


Figure 6: Returns of trajectories in `hopper-medium-replay` and `hopper-medium`, and final performances of COIL with different α.

Ablation studies (cont.)

- Critical hyperparameter: N
  - N: the number of chosen trajectory
  - Influenced by the discrepancy between the behavior policies in two consecutive curriculums
  - Two datasets, showing that
    - A small N should be chosen for a dataset where the trajectories in the dataset are dense such that the distance between the behavior policies are close
    - A large N should be chosen to a dataset where the trajectories are sparse arranged



Conclusion

- The experimental and a theoretical analysis for the quantity-quality dilemma of behavior cloning (BC) motivates us to propose the curriculum offline imitation learning (COIL) for offline RL
- COIL takes advantage of imitation learning by improving the current policy with adaptive neighboring policies
- COIL has several good properties and can compete against state-of-the-art offline RL algorithms
- **COIL is simple, effective, stable, and probably practical** for offline RL!

Thanks for listening

NeurIPS 2021

Curriculum Offline Imitation Learning

Minghuan Liu[1], Hanye Zhao[1], Zhengyu Yang[1], Jian Shen[1],
Weinan Zhang[1], Li Zhao[2], Tie-Yan Liu[2]

[1] Shanghai Jiao Tong University, [2] Microsoft Research Asia

minghuanliu

minghuanliu@sjtu.edu.cn