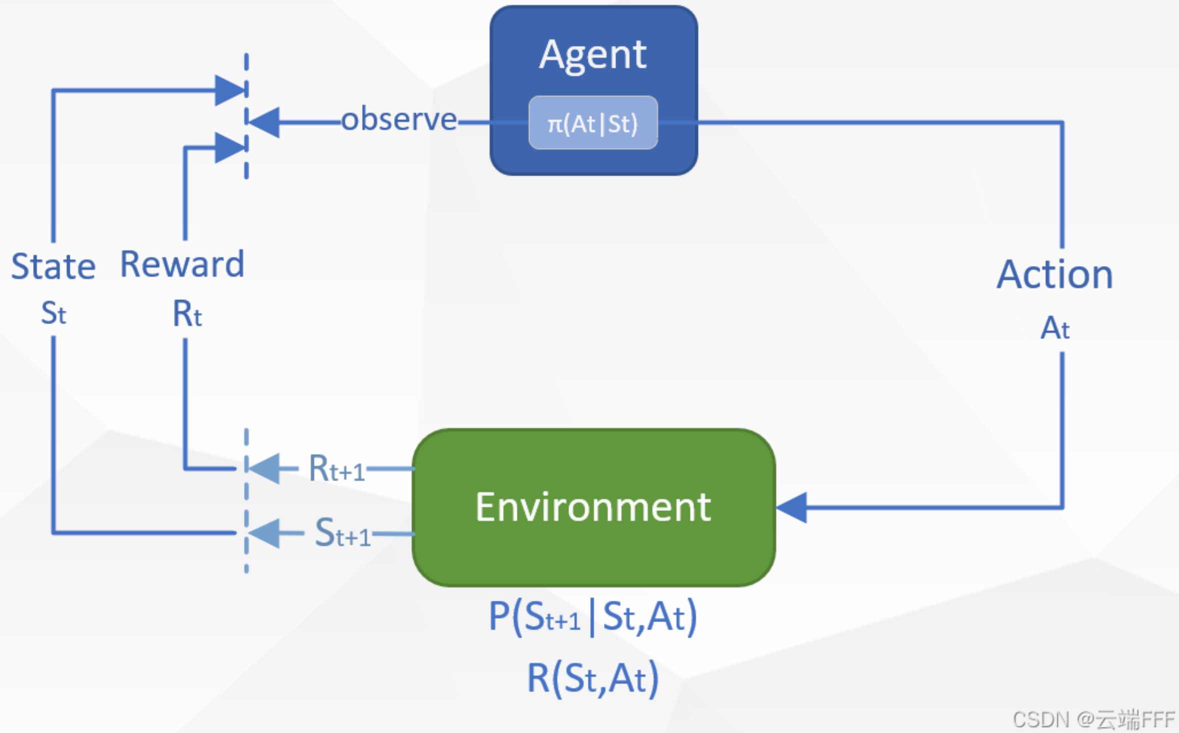
Offline RL: Benchmark and a new algorithm

seminar 2022/1/14

Online On-policy RL

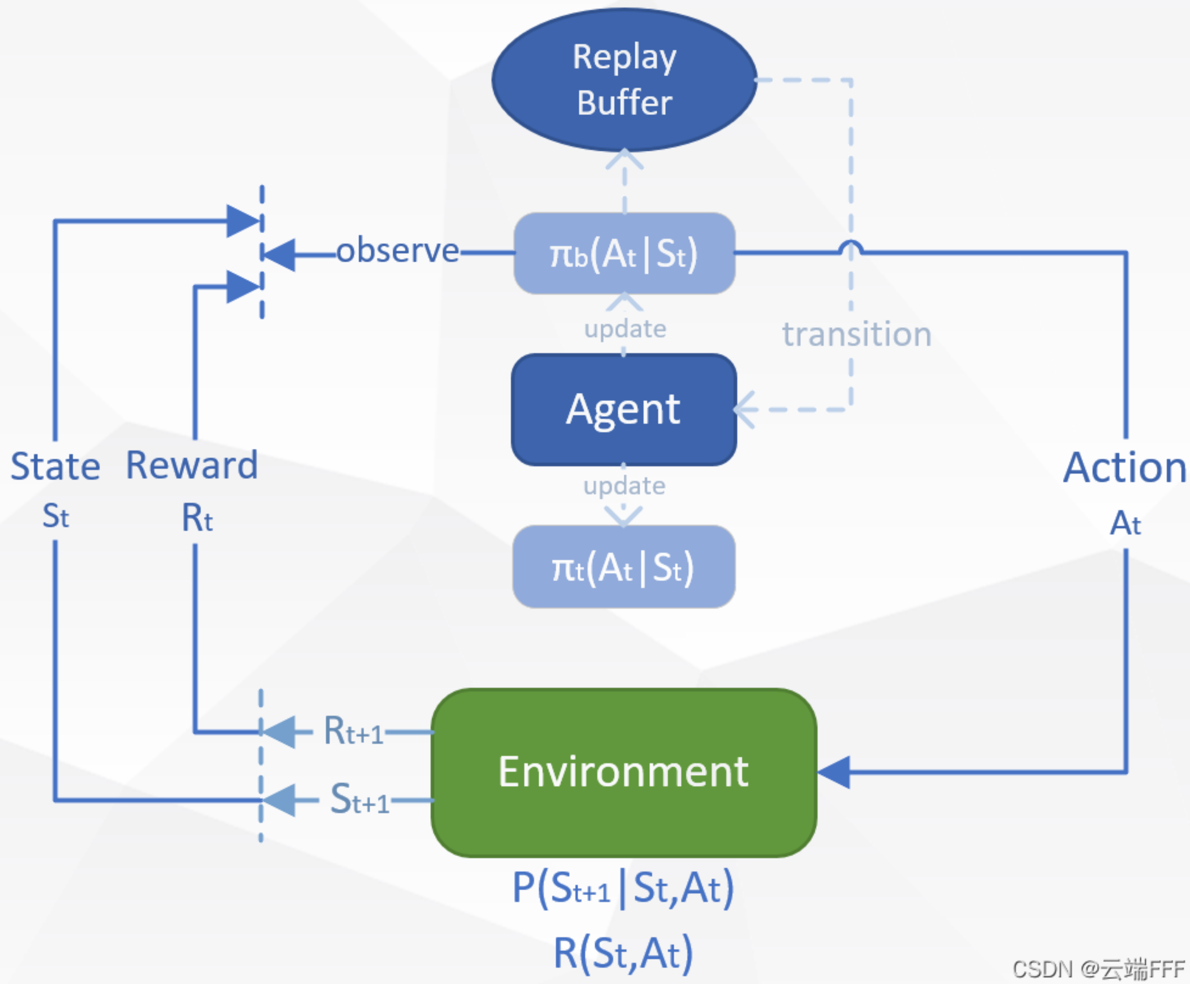
Find a optimal strategies which maximize the return



On-policy RL is usually unstable because of the exploration-exploitation dilemma, and sample-inefficient because of can't reuse past data

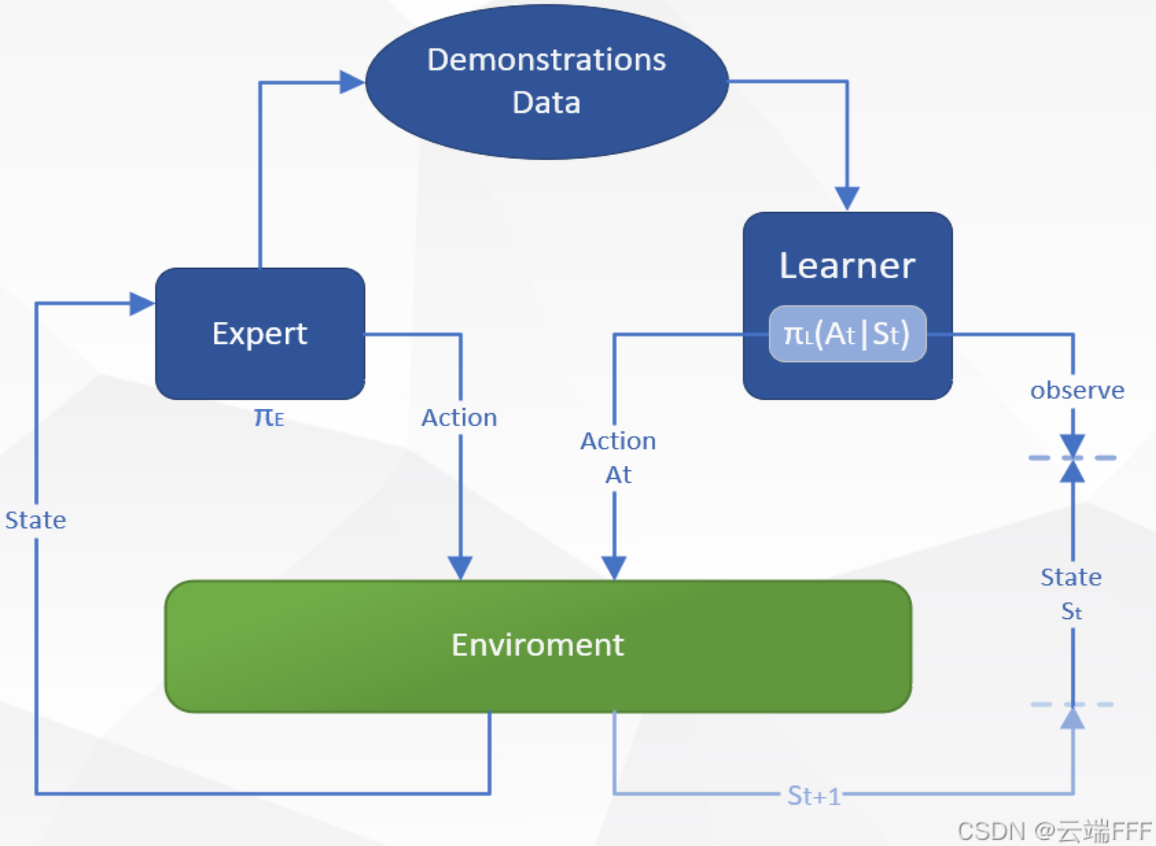
Online Off-policy RL

The policy for generating interaction samples is different from the target policy



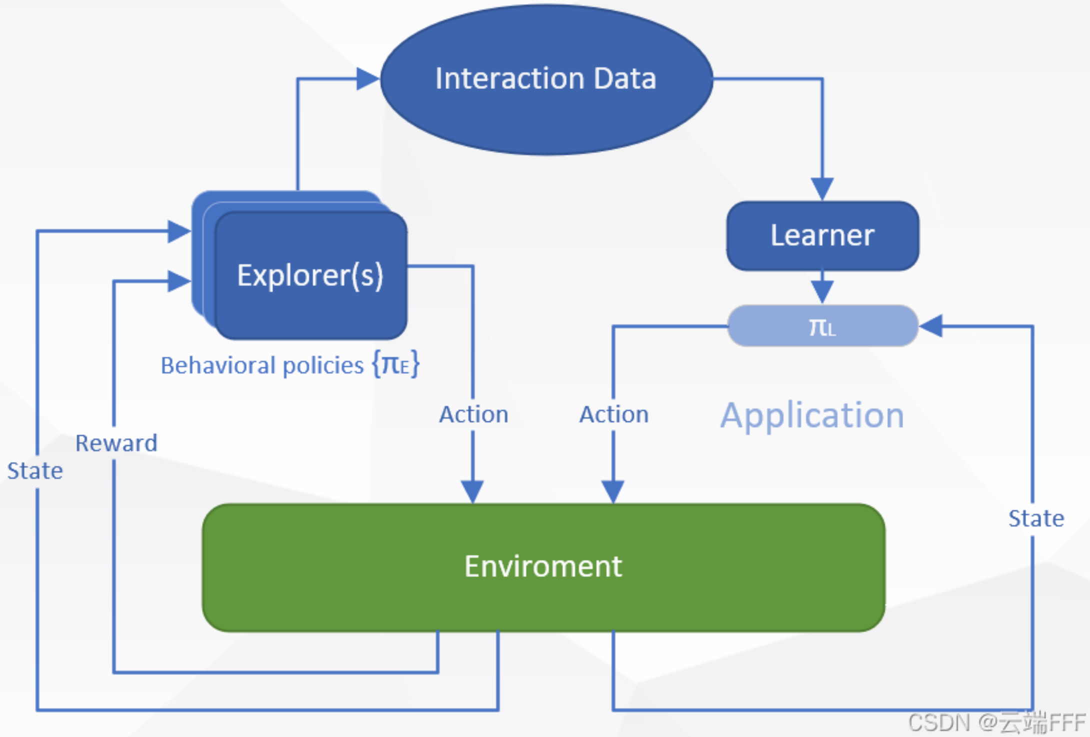
Imitation Learning

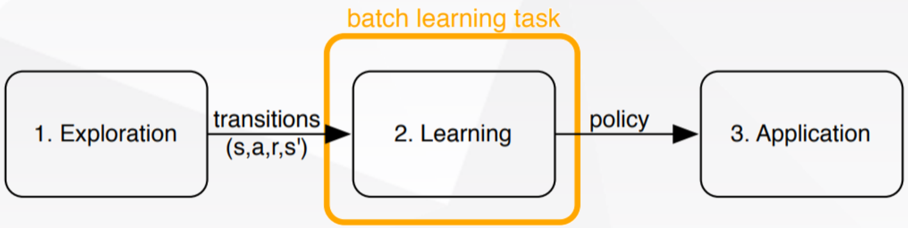
To address the challenge of reward function design, introduce expert demonstrations



Offline/Batch RL

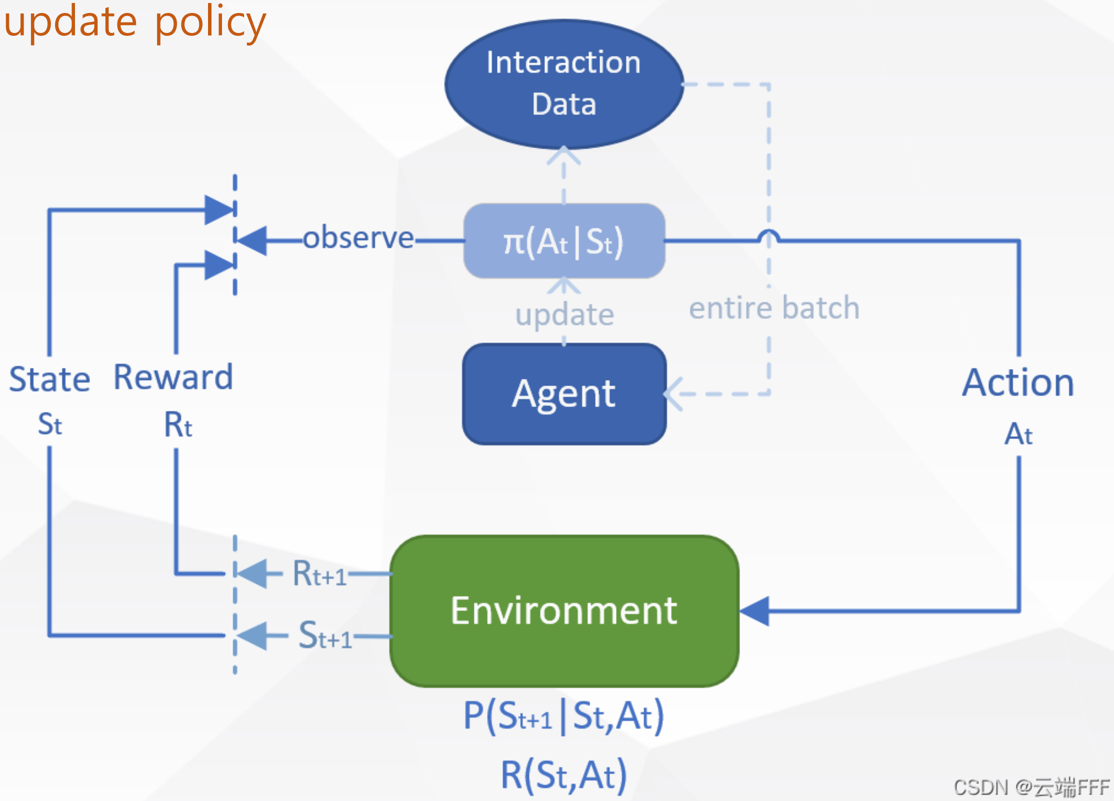
Policy is learned from a fixed dataset, enabling RL methods to take advantage of large, previously-collected datasets

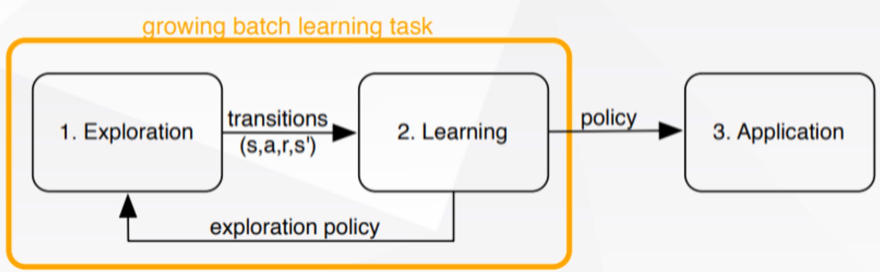




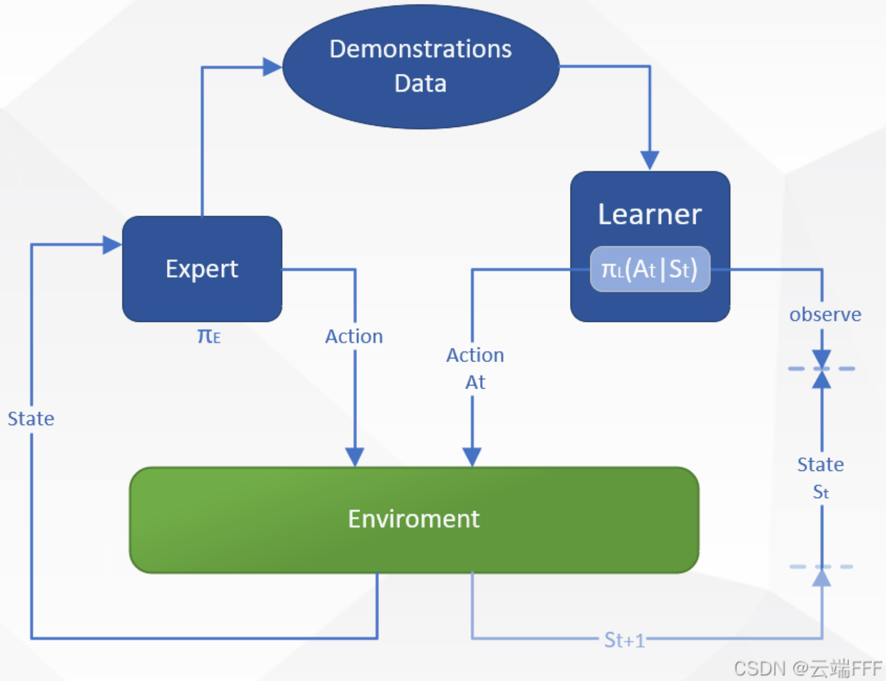
Growing Batch RL

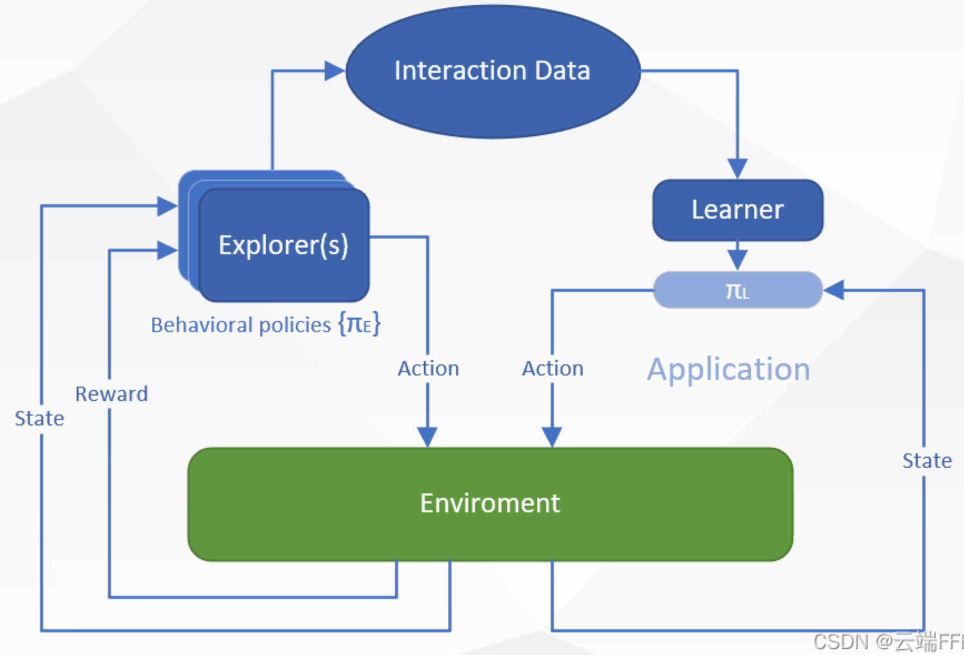
Allows online exploration to expand the Offline dataset, but still use batch RL algorithm to update policy





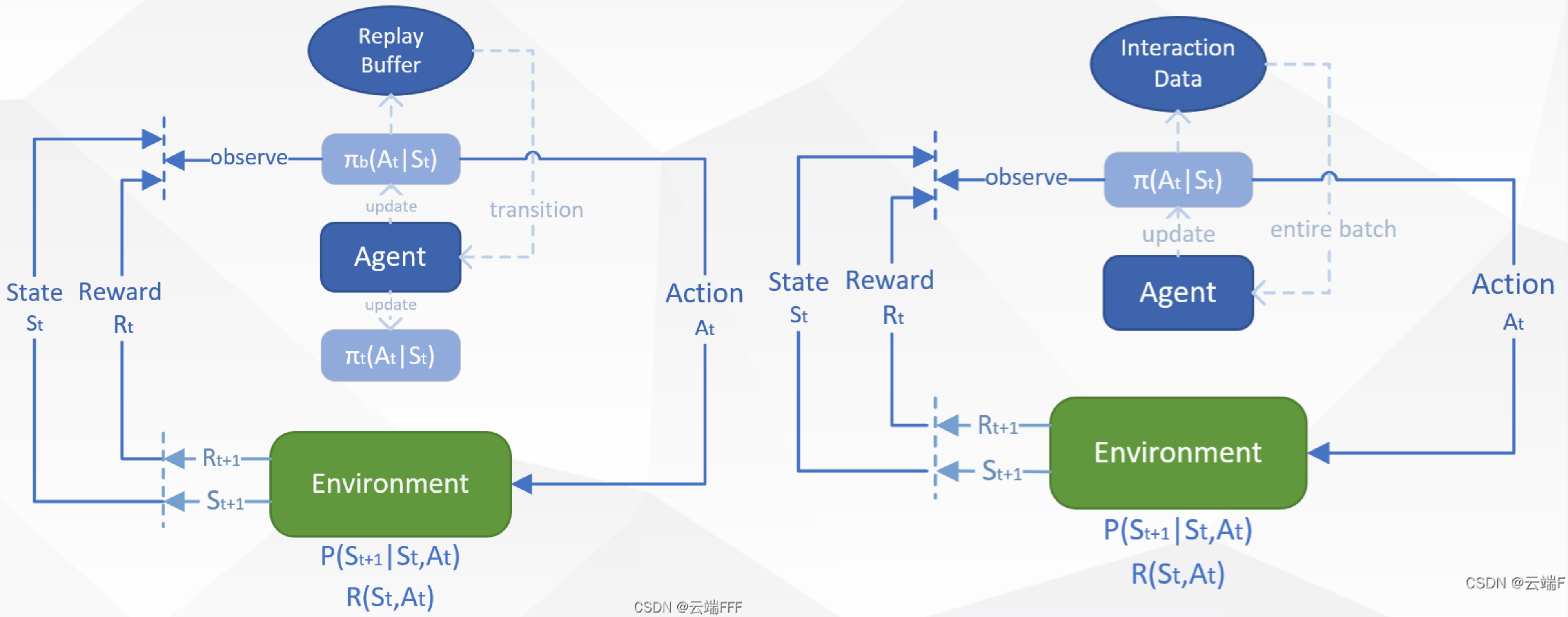
Offline RL vs. IL





- Offline RL completely forbids Learner to interact with the environment
- The policy used to generates batch dataset in Offline RL can be any policy

Growing Offline RL vs. Online RL



- The major difference is only in the method used to update the policy

All kinds of RL paradigm

分类	允许交互	区分目标策略	重用数据	更新策略使用数据量	更新策略后丢弃数据
Online (On-policy)	√	×	×	single transition	√
Online (Off-policy)	√	√	√	mini-batch	×
Semi-batch	√	×	√	mini-batch	√
Growing-batch	√	-	√	entire-batch	×
Offline/batch	×	-	√	entire-batch	×

Lange, Sascha, Thomas Gabel, and Martin Riedmiller. "Batch reinforcement learning."Reinforcement learning. Springer, Berlin, Heidelberg, 2012. 45-73.

Why Offline?

强化学习这个古老的研究领域 2016 前在国内一直比较冷的根源就是没法用。研究领域大家也都清楚强化学习算法样本利用率低，然后做出了很多改进，但是要改进到什么程度才能有用呢，其实根据我们的经验有一个标准：

零试错：一次试错不能有，上线即能发挥效果，还要明显优于基线

offline RL是个正确的方向，但是目前的主流研究也有很多明显的弯路，可能发论文与做落地本身就是不同的事，大家的关心点不可能完全一致吧

Related Work

- A naive idea is to execute online RL algorithm directly on an offline dataset, which leads to Extrapolation error[1] because of
    1. Absent Data
    2. Model Bias
    3. Training Mismatch

- Noticed that the similarity between Offline-RL, Imitation Learning and Online Off-policy RL. Two mainstream methods can be obtained from these two perspectives
    1. RL-based algorithm：BCQ、BEAR、CQL、MOPO...
    2. IL-based algorithm：MARWIL、AWR、BAIL...

1. Fujimoto, Scott, David Meger, and Doina Precup. "Off-policy deep reinforcement learning without exploration." International Conference on Machine Learning. PMLR, 2019.

D4RL: Datasets for Deep Data-Driven Reinforcement Learning

Justin Fu
UC Berkeley
justinjfu@eecs.berkeley.edu

Aviral Kumar
UC Berkeley
aviralk@berkeley.edu

Ofir Nachum
Google Brain
ofirnachum@google.com

George Tucker
Google Brain
gjt@google.com

Sergey Levine
UC Berkeley, Google Brain
svlevine@eecs.berkeley.edu

Why we need a benchmark?

One important observation we make, which was not brought to light in previous batch DRL papers, is that batches generated with different seeds but with otherwise exactly the same algorithm can give drastically different results for batch DRL[1]*.*

If different research teams use different datasets, or the implementation details of the same algorithm are different, it is not conducive to fair comparison between algorithms

A widely-accessible and reproducible benchmark can promote the research work and enhance the practicability of the algorithm applying in the real world

Chen X, Zhou Z, Wang Z, et al. BAIL: Best-action imitation learning for batch deep reinforcement learning[J]. arXiv preprint arXiv:1910.12179, 2019.

Contribution

1. **A set of baseline sequential decision tasks**, with corresponding simulation environment
2. **Open source interactive datasets of each task**. Data sources include not only Online RL Agent, but also human expert demonstration and hand-coded controller, which are closer to the data collection process in the real world
3. **Implementation of popular Offline RL algorithms** under the same specification
4. **Easy-use API** for tasks, datasets, and algorithms

Dataset properties

In order to be as close to the real-world application as possible, the Offline dataset should have the following properties

1. Narrow and biased data distributions
2. Undirected and multitask data
3. Sparse rewards
4. Suboptimal data
5. Non-representable behavior policies, non-Markovian behavior policies, and partial observability
6. Realistic domains

Environments and Tasks

-	narrow	Non-representable	Non-markovian	undirected	multitask	sparse rewards	Suboptimal	realistic	Partial observability
Maze2D			√	√	√				
AntMaze			√	√	√	√			
Gym-MuJoCo	√						√		
Adroit	√	√				√		√	
FrankaKitchen				√	√			√	
Flow		√						√	
Offline CARLA		√		√	√			√	√

Maze2D - maze2d

AntMaze - antmaze

Gym-MuJoCo - hopper/halfcheetah/walker2d

Adroit - pen/hammer/door/relocate

FrankaKitchen - kitchen

Flow - ring/merge.

Offline CARLA - lane

Curriculum Offline Imitating Learning

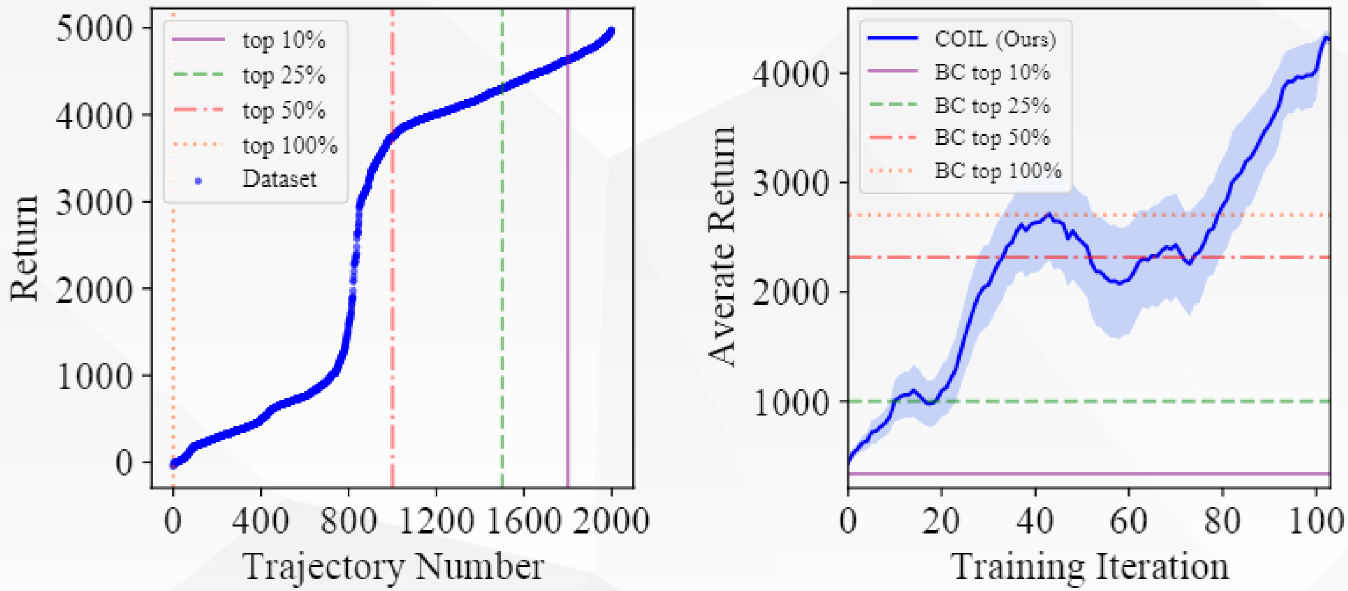
Minghuan Liu[1*] **Hanye Zhao**[1*] **Zhengyu Yang**[1] **Jian Shen**[1]
Weinan Zhang[1†] **Li Zhao**[2] **Tie-Yan Liu**[2]
[1] Shanghai Jiao Tong University, [2] Microsoft Research
{minghuanliu, fineartz, zyyang, rockyshen, wnzhang}@sjtu.edu.cn,
{lizo,tyliu}@microsoft.com

Quantity-quality dilemma

- Offline imitation learning can always stably learn to perform as the behavioral policy, which may be helpful under single-behavior datasets. However, BC may fail in learning a good behavior under a diverse dataset containing a mixture of policies (both goods and bads)

- **Quantity-quality dilemma：** On mix dataset,
  1. Top data owns higher quality but less quantity, and thus cause serious compounding error problems
  2. More data provides a larger quantity, yet its mean quality becomes worse.

Verification Experiment



(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.

Key Idea

- Important observation：Under RL scenarios, the agent can imitate a neighboring policy with much fewer samples (by BC).
- Extend to a solution for mixed datasets：The agent can adaptively imitate the better neighboring policies step by step and finally reach the optimal behavior policy of the dataset

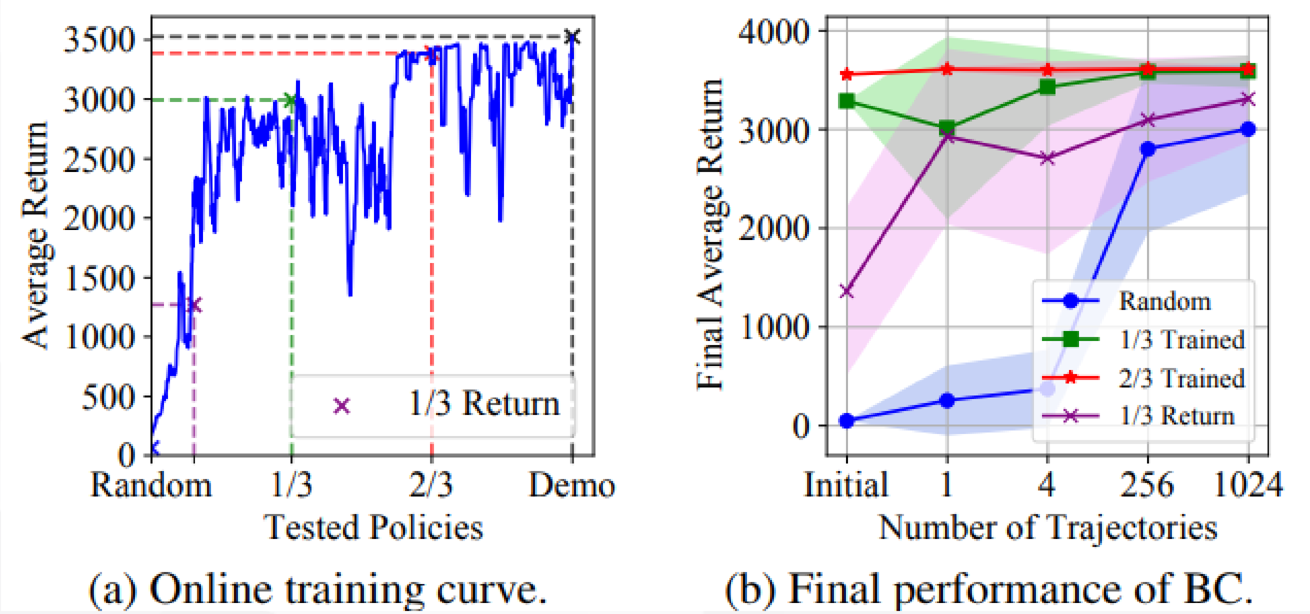


Figure 2: (a) Online training curves of an SAC agent trained on the Hopper environment, where the crosses and dashed lines indicate the stage of selected policies. (b) Final performances achieved by imitating the demo policy using BC, initialized with different stages of policies. The curves depict the fact that close-to-demonstration policy can easily imitate the demonstrated policy with fewer samples.

Key Idea

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 as well as 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:*

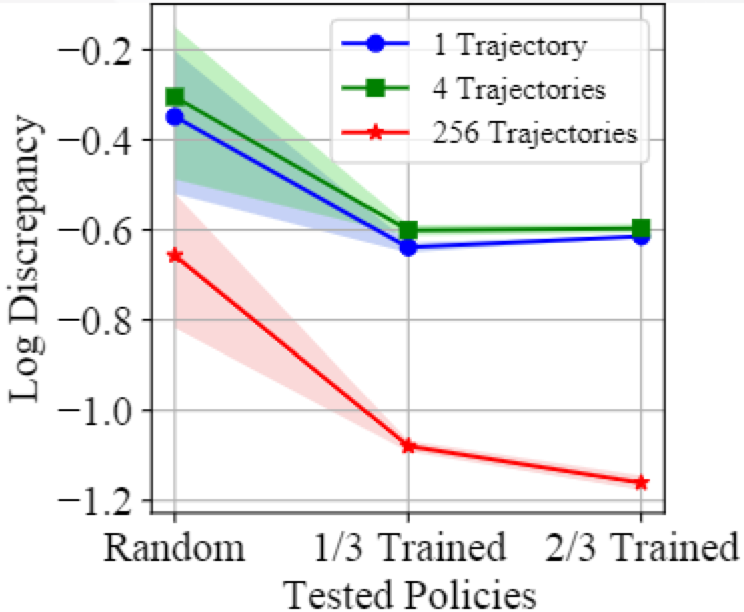
$$D_{\mathrm{TV}}(\rho_\pi(s,a) \| \rho_{\pi_b}(s,a)) \leq c(\pi_0, \pi_b, |\mathcal{D}|)$$

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

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

Analysis of the second item

$$\frac{1}{2}\sum_{s\notin\mathcal{D}}|\rho_\pi(s)-\rho_{\pi_0}(s)|$$

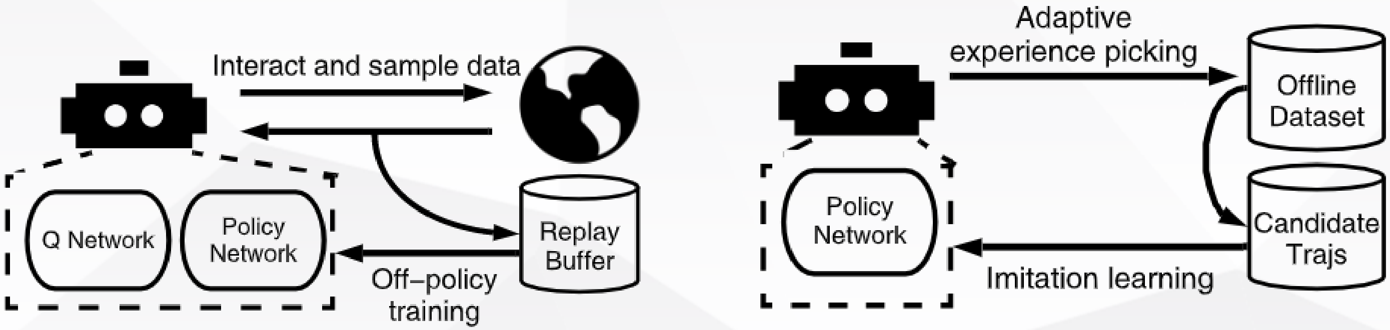


(c) Empirical discrepancy.

(c) Empirical estimation on the discrepancy between the initialized policy and the trained policy outside the support of the demonstrations. Initialized with a closer-to-demo policy always enjoys more minor discrepancy.

Key Idea

- Generally, given the same discrepancy c(π0, πb, |D|) = C, if the initialized policy narrows down the *initialization gap* as is close to the demonstrated policy, then the requirement for more samples to minimize the *data gap* can be relaxed.



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

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

Online RL as Imitating Optimal Policies

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

principle of maximum entropy

$$P^*(\tau) \propto \exp\left(R(\tau)\right)$$

$$\underset{\pi}{\text{minimize}}\, D_{KL}(P_\pi(\tau)\|P^*(\tau))$$

$$P_\pi(\tau) = \rho(s_0)\sum_{t=0}^{T} \mathcal{T}(s_{t+1}|s_t, a_t)\pi(a_t|s_t)$$

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

Offline RL as Adaptive Imitation

- Construct a finite policy sequence $\pi^0, \tilde{\pi}^1, \pi^1, \tilde{\pi}^2, \pi^2, \cdots, \tilde{\pi}^N, \pi^N$ such that $\pi^i \preceq \pi^{i+1}$, where $\tilde{\pi}^i$ is characterized by its trajectory, which is picked from database based on the current policy π^{i-1}, π^i, is the imitation result taken $\tilde{\pi}^i$ as the target policy.
- Formally, with dataset $\{\mathcal{D}\}_1^N$, at every training stage i, the agent updates its policy π^i by adaptively selecting $\tau \sim \tilde{\pi}^{i+1}$ from $\mathcal{D}$ as the imitating target such that

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

Similar to Online RL

$$s.t.\ \mathbb{E}_{\tilde{\pi}}\left[D_{KL}(\tilde{\pi}^{i+1}(\cdot|s)||\pi^i(\cdot|s))\right] \le \epsilon$$

must be neighboring policy

$$R_{\tilde{\pi}}^i - R_{\pi}^i \ge \delta$$

must be better policy

Debug :)

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

晓晨你好，

谢谢你的指出，我们仔细查看了论文，发现这里确实是写反了。之前没有reviewer提出，我们自己在检查的时候也没有发现，非常抱歉带来理解上的困扰。

我们已经在arxiv上上传了修正后的版本，过几天应该可以更新。

再次抱歉，非常感谢指出这个问题。

刘明桓

上海交通大学

Select episodes

Select trajectories which are sampled by a neighboring policy

$$\mathbb{E}_{\tilde{\pi}}\left[D_{KL}\left(\tilde{\pi}(\cdot|s)||\pi(\cdot|s)\right)\right] \leq \epsilon$$

Importance sampling ratio $\approx$ 1

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.:*

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

Set β = 0.05 as an intuitive ratio of exploration. As for ϵ_c , we let the agent choose the value through finding N appropriate trajectories

Return Filter

- Refrain the performance from getting worse by imitating to a poorer target than the current level of the imitating policy

$$R_{\tilde{\pi}} - R_{\pi} \geq \delta$$

- Evaluate current policy's performance based on the learned curriculum

$$V_k = (1-\alpha) \cdot V_{k-1} + \alpha \cdot \min\{R(\tau)\}_1^n$$
$$\mathcal{D} = \{\tau \in \mathcal{D} \mid R(\tau) \geq V\}$$

Pseudo code

Algorithm 1 Curriculum Offline Imitation Learning (COIL)

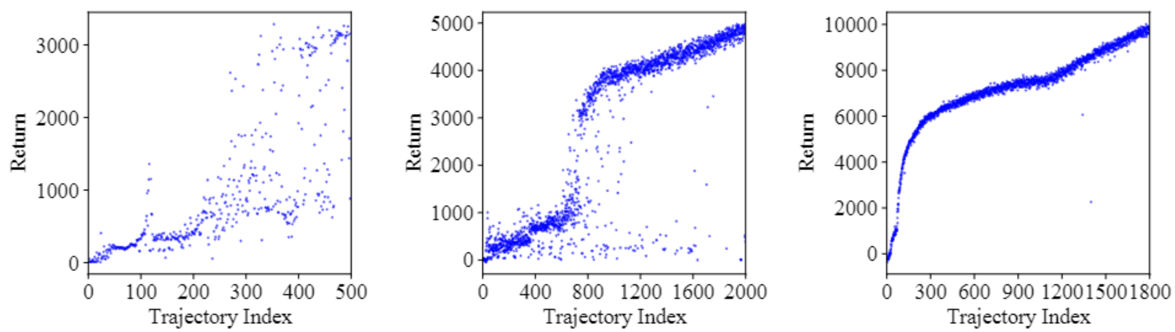
Require: Offline dataset $\mathcal{D}$, number of trajectories picked at each curriculum N, moving window of the return filter α, number of training iteration L, batch size B, number of pre-train times T, and the learning rate η.

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 $s(\tau_i) = \pi(\bar{a}^i_{\lfloor \beta h \rfloor}|\bar{s}^i_{\lfloor \beta h \rfloor})$ 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

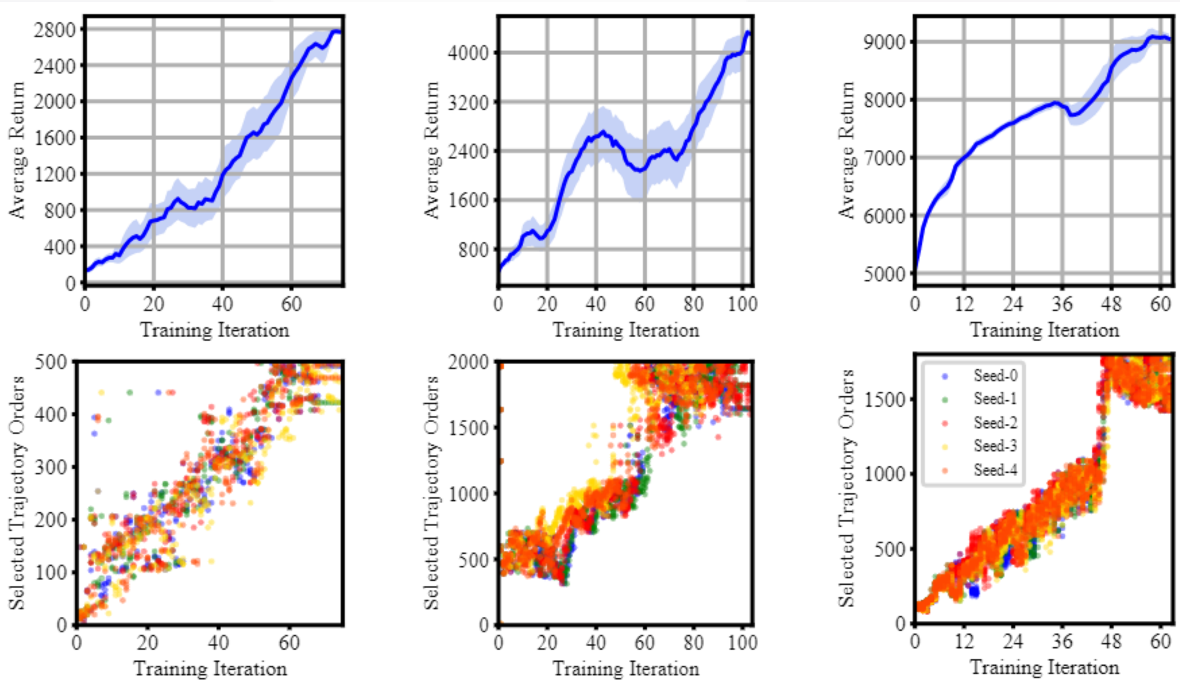
$$\theta \leftarrow \theta - \eta \nabla_\theta \sum_{j=1}^{B} [-\log \pi_\theta(a_j|s_j)]$$

  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

Experiments



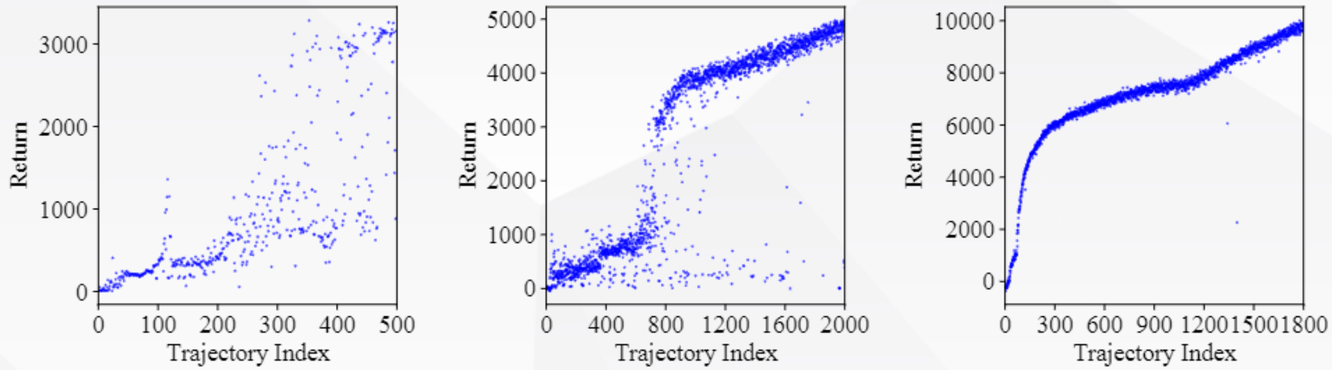
(a) Hopper-final. (b) Walker2d-final. (c) HalfCheetah-final.



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

- 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

Experiments



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

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)**

- COIL substantially outperforms the other baselines for the final buffer dataset. COIL reaches the performance close to the optimal policy.
- BAIL and AWR can not always find the optimal behavior due to the difficulty of its hyperparameters tuning and value regression.
- BC that learns a mediocre policy

Experiments on D4RL

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

- BC is able to approach or outperform the performance of the behavior policy on the datasets generated from a single policy (random/medium), but still remains a gap between the optimal behavior policy (Best 1%)
- 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
- For model-based algorithm like MOPO, it behaves well on the medium-replay datasets due to the sufficient data to learn a good environment model; but it can hardly outperform SoTA model-free results on other datasets

Ablation Study

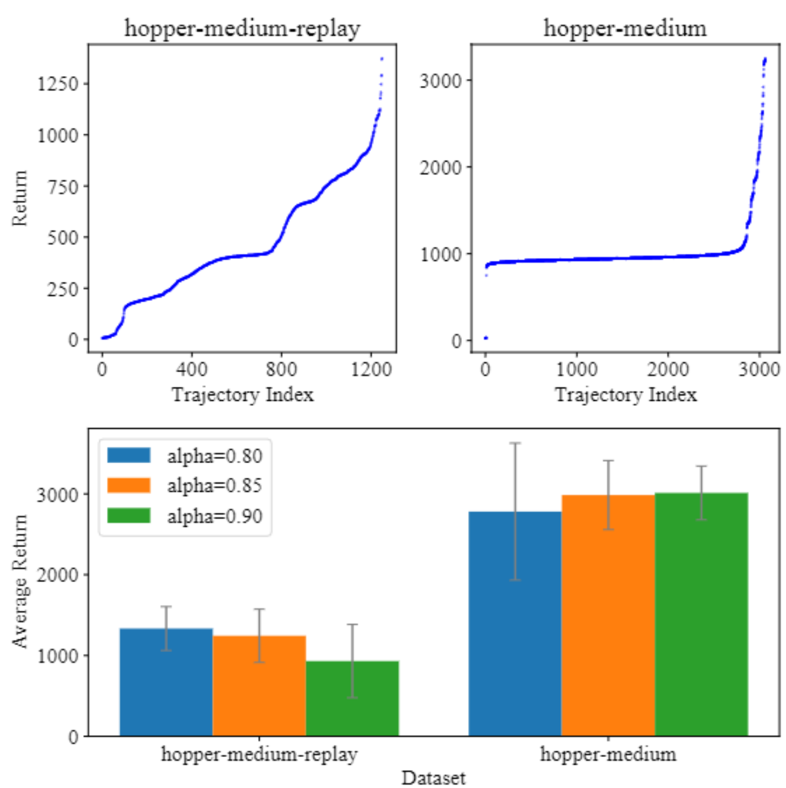
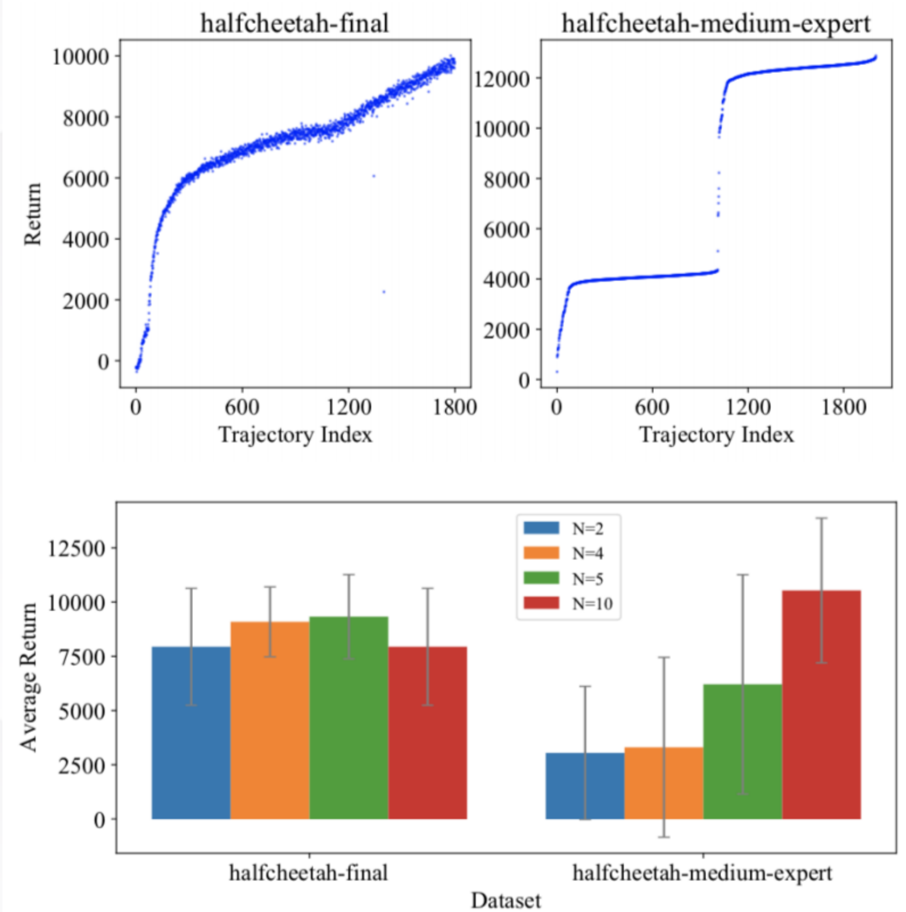


Figure 6: Returns of trajectories in `hopper-medium-replay` and `hopper-medium`, and final performances of COIL with different α.



Compared with Naive 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

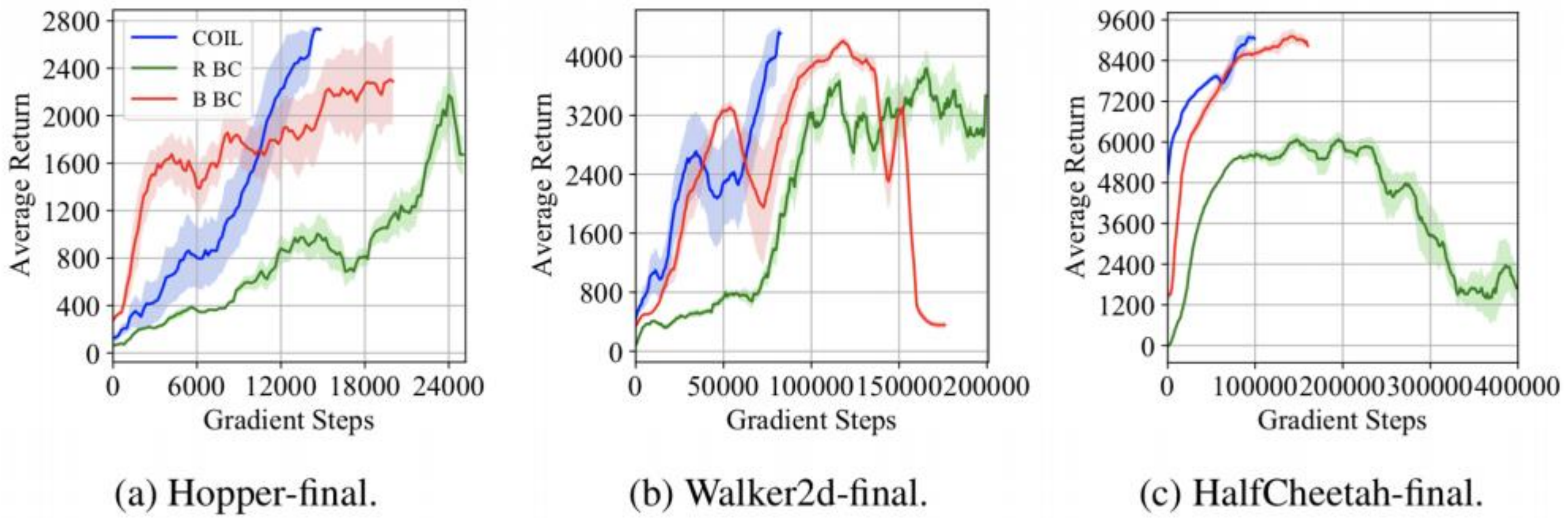


Figure 8: Comparison of training curves between COIL and Return-ordered BC (*R BC*) and Buffer-shrinking BC (*B BC*) on *final* datasets with the same batch size. Different strategies terminate with different gradient steps.

Conclusion

- Analyze the quantity-quality dilemma of behavior cloning (BC) from both an experimental and a theoretical point of view, and propose COIL
- Experiments show good properties of COIL with competitive evaluation results against SOTA offline RL algorithm
- COIL can stops automatically with a good policy, which make it easier to be applied in real-world applications without online evaluation to find the stop point as the previous algorithms do
- The best performence of COIL is limited into the performance of the dataset

The End

thanks