# Offline Reinforcement Learning

# Deep Reinforcement Learning

Mohammad Hossein Rohban, Ph.D.

Courtesy: Some slides are adopted from CS 285 Berkeley, and CS 234 Stanford, and Pieter Abbeel's compact series on RL.


RIML Lab, CE Department, Sharif
University of Technology

Spring 2026

# The generalization gap

Tehran

Isfahan

Mnih et al. '13

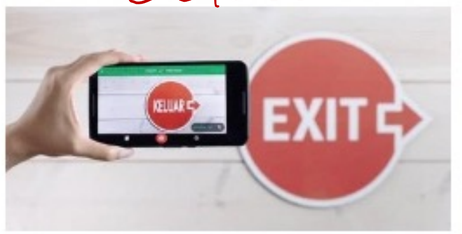
Closed World

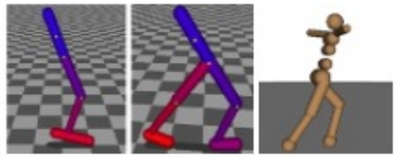
VS.

SL

Open World

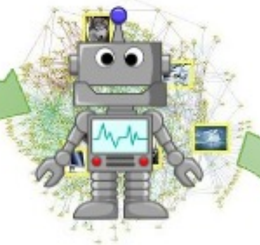
Dataset



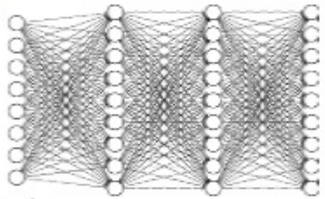



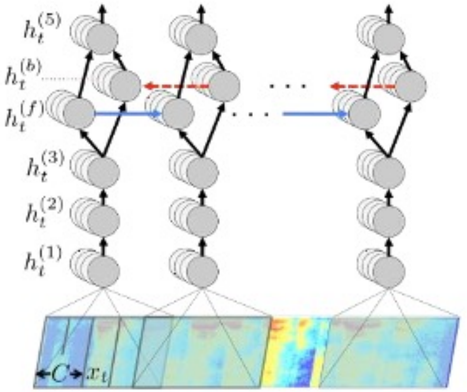
Schulman et al. '14 & '15

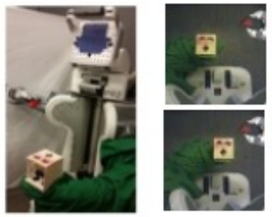
Offline RL



this is done **many** times









Levine*, Finn*, et al. '16

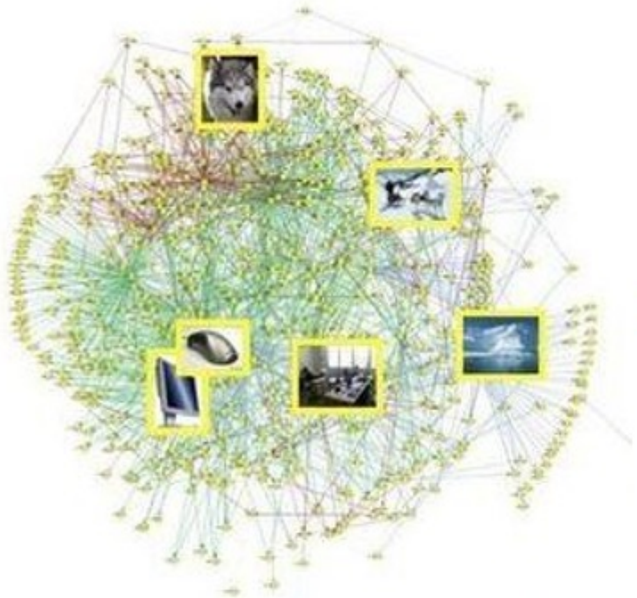
RL on Data

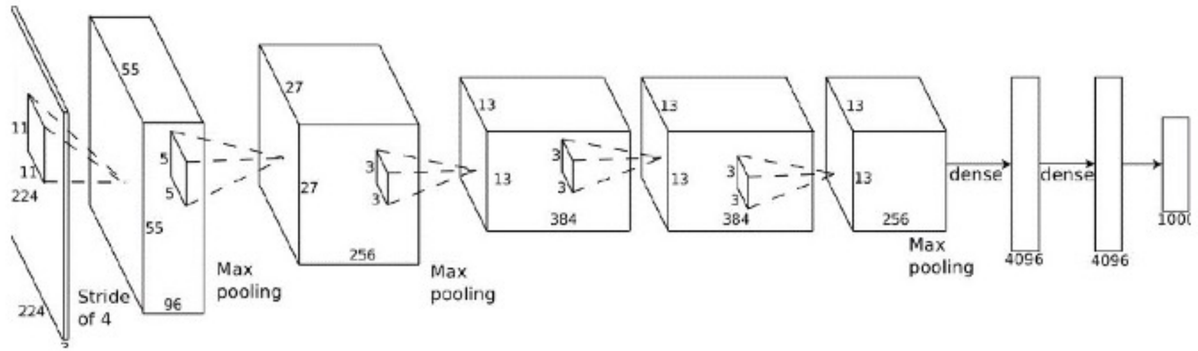
VS.

RL on Simulator

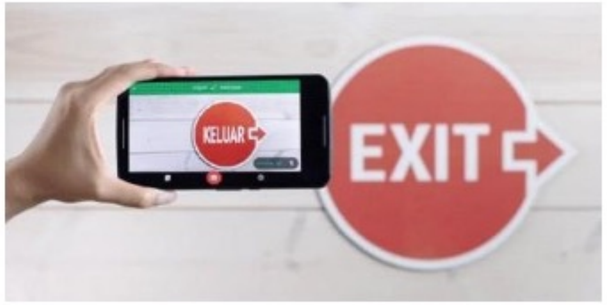
enormous gulf

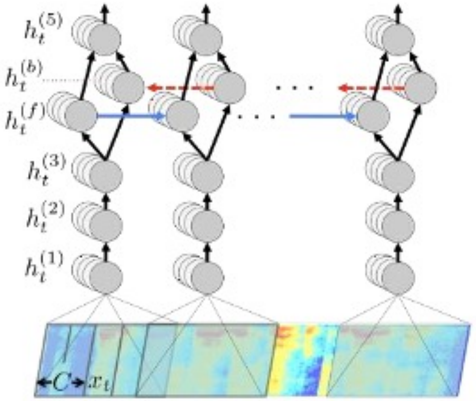

# What makes modern machine learning work?









# Can we develop data-driven RL methods?

① Scalability

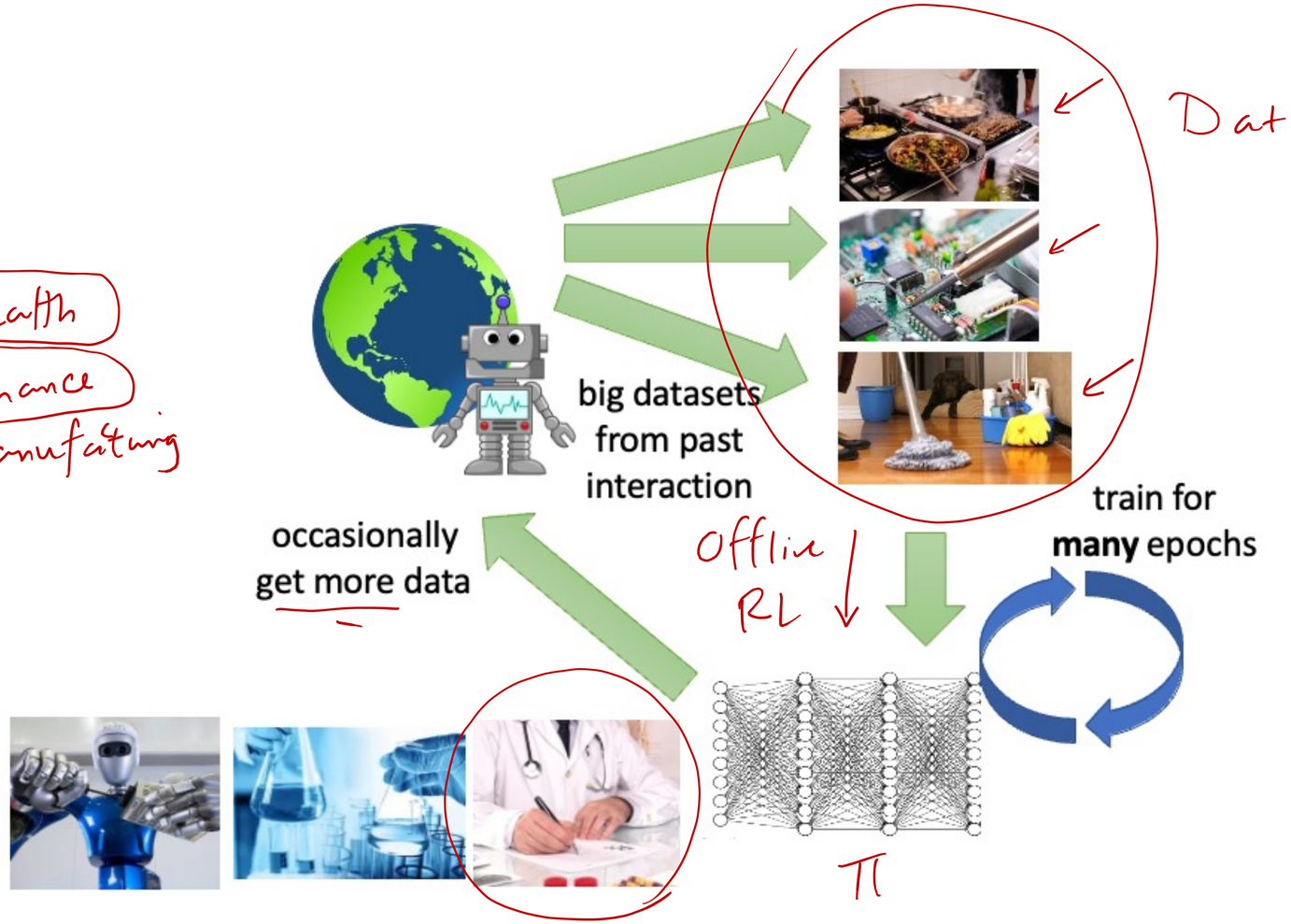
② Safety

Health

* Finance

Manufatuing

⋮




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

# What does offline RL mean?

$D = \{(s_i, a_i, s'_i, r_i)\}$

$s \sim d^{\pi_\beta}(s)$

$a \sim \pi_\beta(a|s)$ (generally not known)

$s' \sim p(s'|s, a)$

$r = r(s, a)$

$$\max_\pi \sum_{t=0}^{T} E_{s_t \sim d^\pi(s), a_t \sim \pi(a|s)}[\gamma^t r(s_t, a_t)]$$

go beyond the perf. ← achieved by $\pi_\beta$

not necessarily expert data

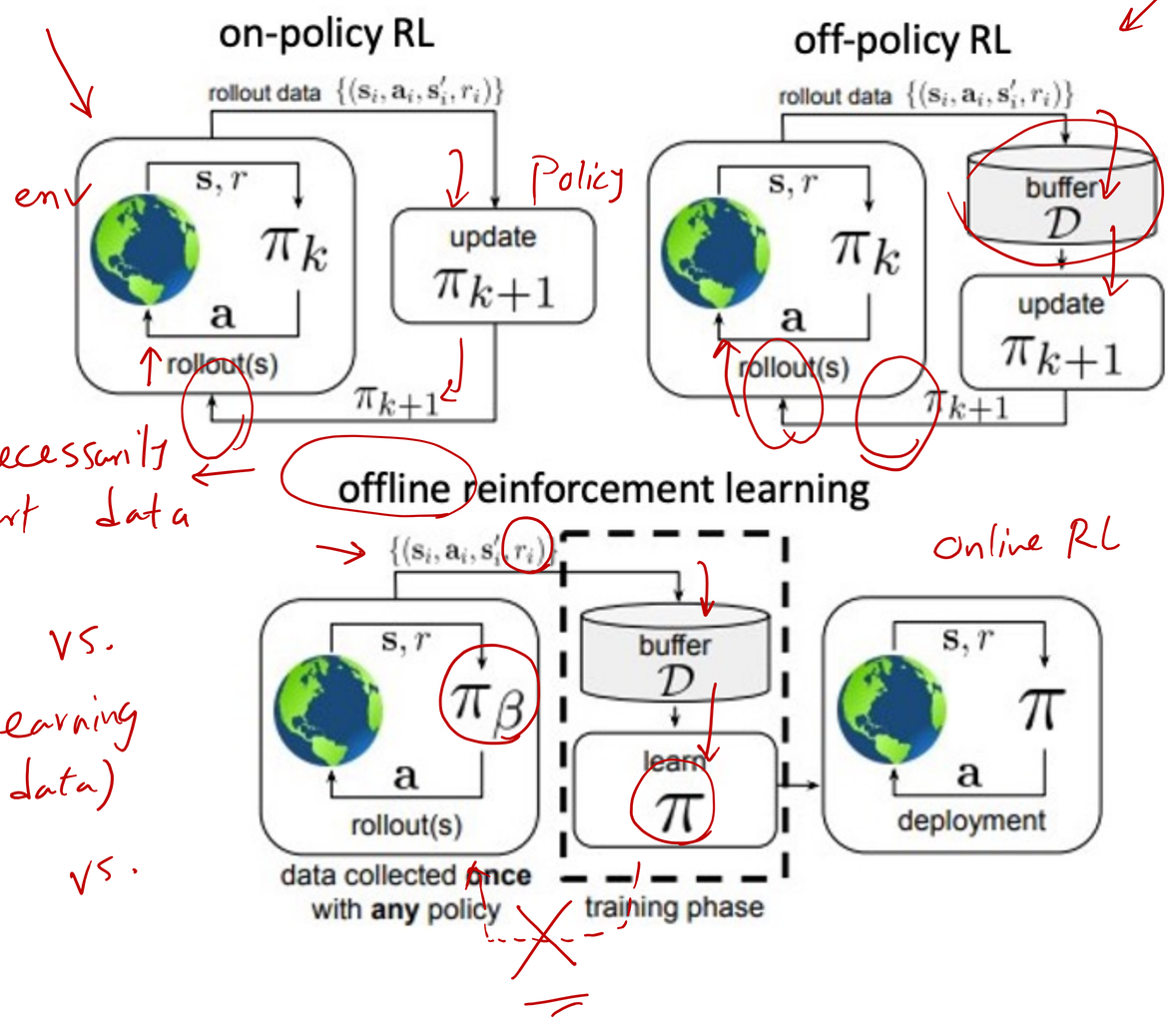
vs.

imitation learning (expert data)

vs.

SL

(e.g. policy grad)




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

# Types of offline RL problems

1. off-policy evaluation (OPE):

$$J(\pi) = E_{\pi}\left[\sum_{t=1}^{T} r(s_t, a_t)\right]$$

given $\mathcal{D}$, estimate $J(\pi)$

$$\mathcal{D} = \{(s_i, a_i, s_i', r_i)\}$$

$$s \sim d^{\pi_\beta}(s)$$

$$a \sim \pi_\beta(a|s)$$

$$s' \sim p(s'|s,a)$$

$$r \leftarrow r(s,a)$$

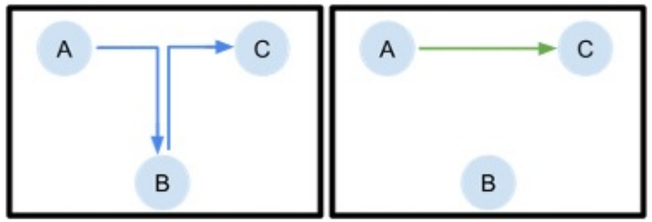
2. offline reinforcement learning: (a.k.a. batch RL, sometimes fully off-policy RL)

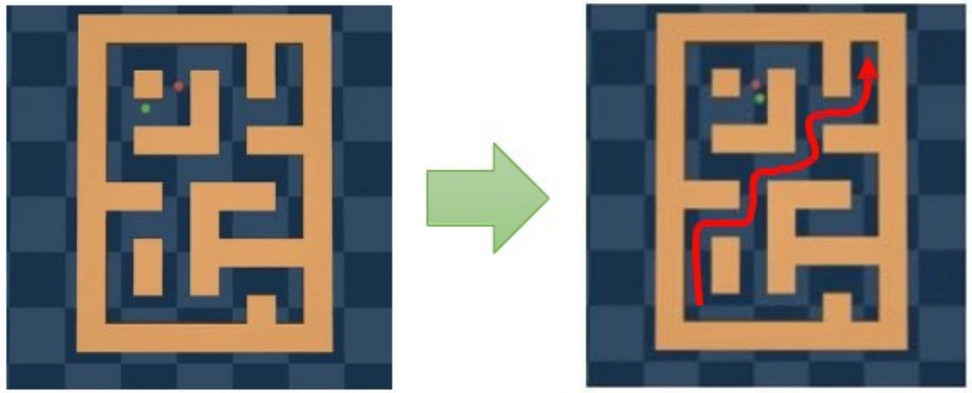
given $\mathcal{D}$, learn the best possible policy $\pi_\theta$

not necessarily obvious what this means

# How is this even possible?

1. Find the “good stuff” in a dataset full of good and bad behaviors
2. Generalization: good behavior in one place may suggest good behavior in another place
3. “Stitching”: parts of good behaviors can be recombined



# What do we expect offline RL methods to do?

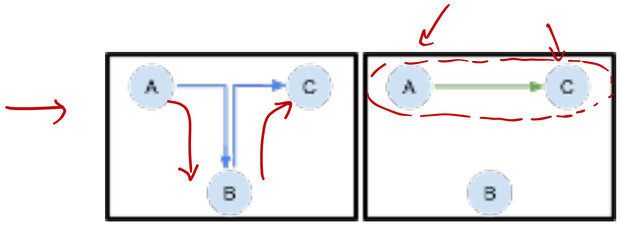
- Bad intuition: it's like imitation learning

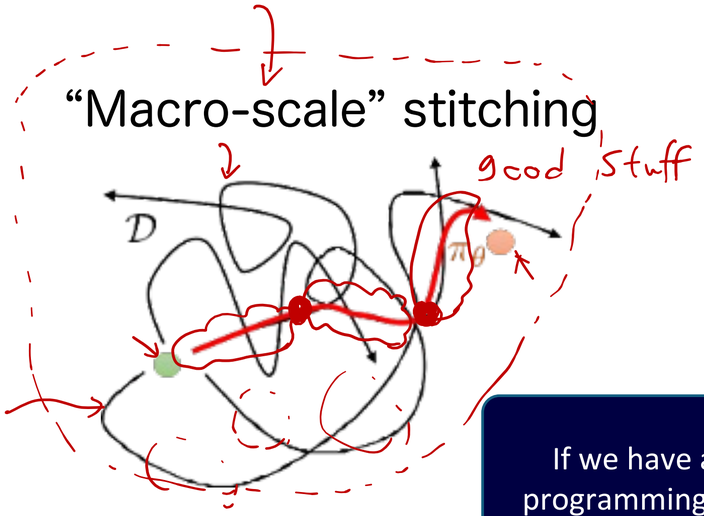
  Though it can be shown to be provably better than imitation learning even with optimal data, under some structural assumptions!

  See: Kumar, Hong, Singh, Levine. Should I Run Offline Reinforcement Learning or Behavioral Cloning?

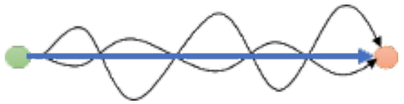

- Better intuition: get order from chaos

But this is just the clearest example!



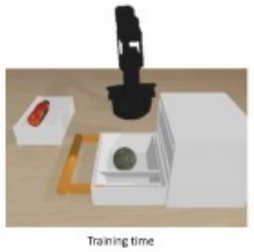

"Micro-scale" stitching:



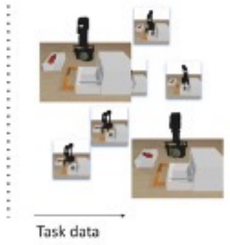
If we have algorithms that properly perform dynamic programming, we can take this idea much further and get near-optimal policies from highly suboptimal data

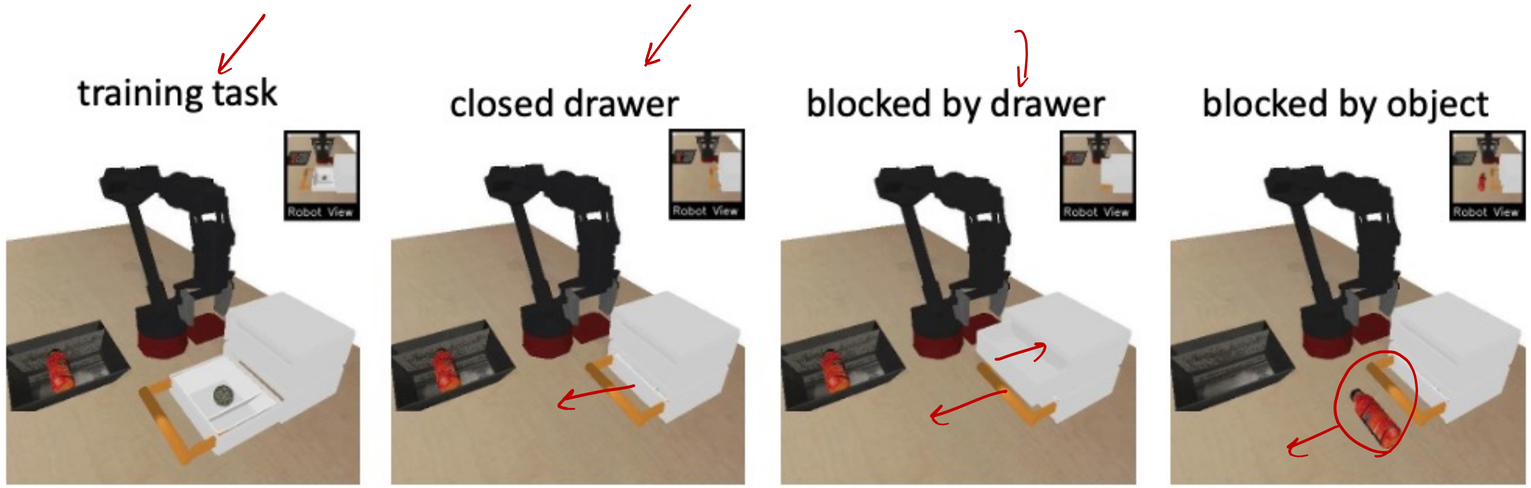
# A vivid example

RL policies typically don't generalize to initial conditions that were not seen during training



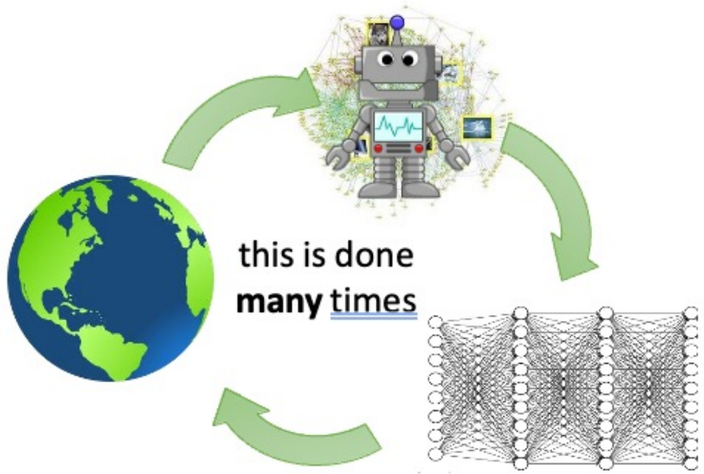

Can we use previously collected, unlabeled datasets to extend learned skills?



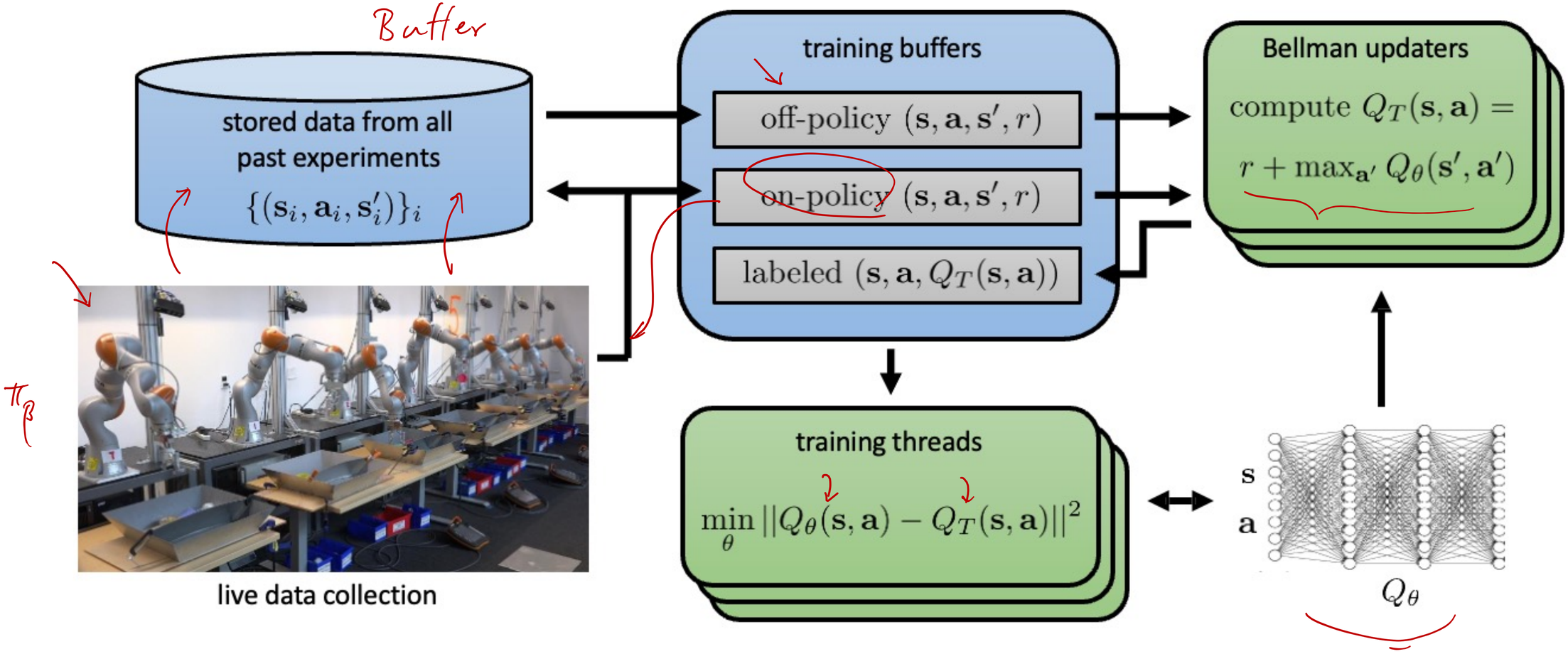

Singh, Yu, Yang, Zhang, Kumar, Levine. COG: Connecting New Skills to Past Experience with Offline Reinforcement Learning. '20

# Why should we care?



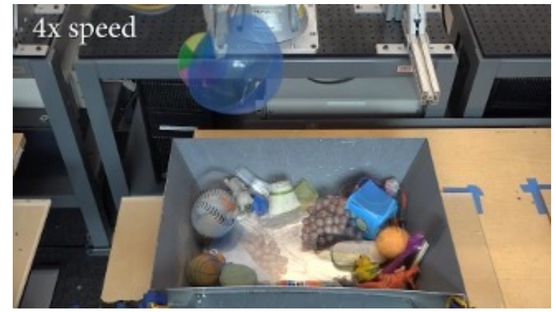


# Does it work?




Kalashnikov, Irpan, Pastor, Ibarz, Herzong, Jang, Quillen, Holly, Kalakrishnan, Vanhoucke, Levine. QT-Opt: Scalable Deep Reinforcement Learning of Vision-Based Robotic Manipulation Skills

# Does it work? (Cont.)

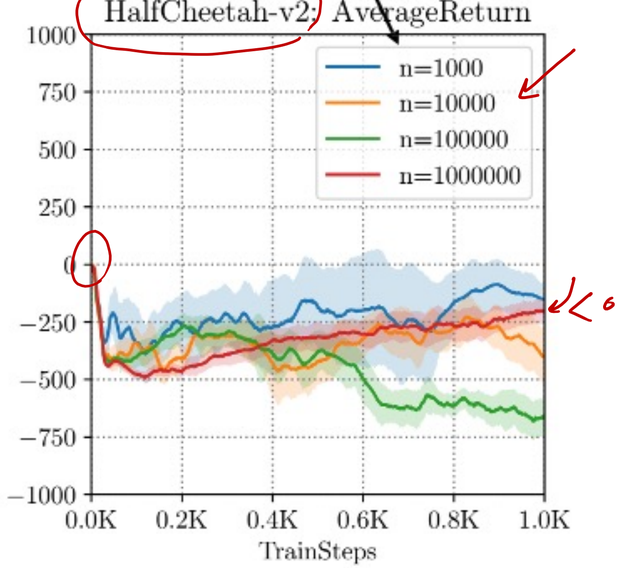

| Method | Dataset | Success | Failure |
|---|---|---|---|
| Offline QT-Opt | 580k offline | 87% | 13% |
| Finetuned QT-Opt | 580k offline + 28k online | 96% | 4% |

Kalashnikov, Irpan, Pastor, Ibarz, Herzong, Jang, Quillen, Holly, Kalakrishnan, Vanhoucke, Levine. QT-Opt: Scalable Deep Reinforcement Learning of Vision-Based Robotic Manipulation Skills

# Why is offline RL hard?

Overfitting

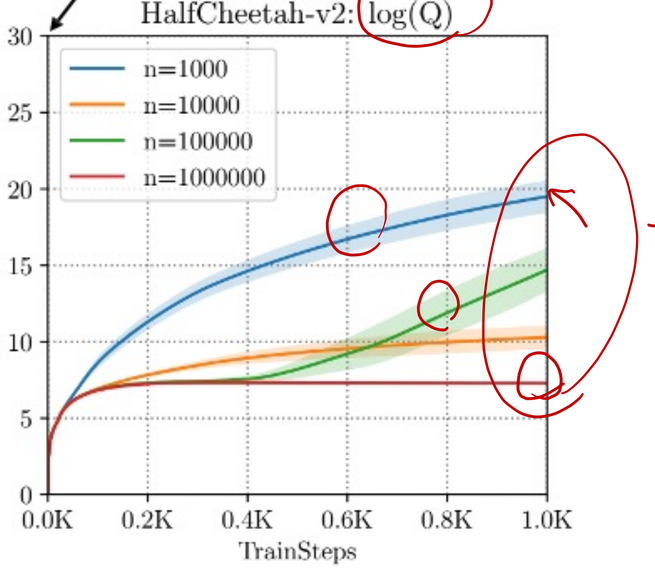
amount of data




Ret.

how well it does

log scale (massive overestimation)



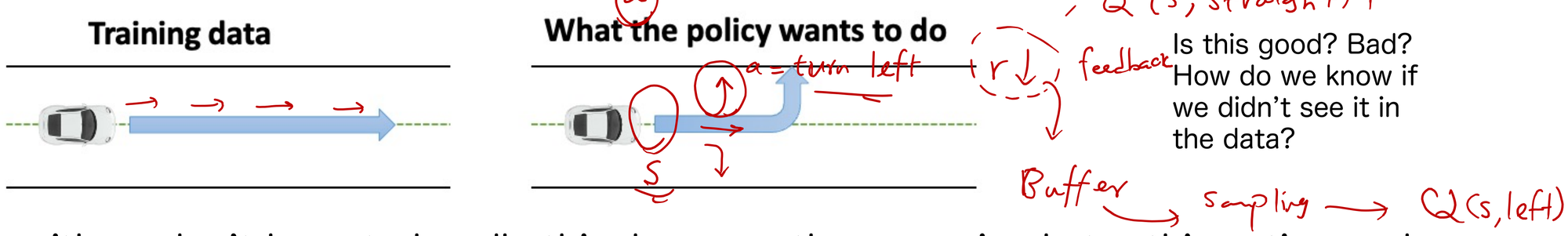

how well it *thinks* it does (Q-values)

→ They are returns that the model "thinks" it would achieve in the real world

Kumar, Fu, Tucker, Levine. Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction. NeurIPS '19

# Why is offline RL hard? (Cont.)

- Fundamental problem: counterfactual queries




- Online RL algorithms don't have to handle this, because they can simply try this action and see what happens
- Offline RL methods must somehow account for these unseen ("out-of-distribution") actions, ideally in a safe way
- …while still making use of generalization to come up with behaviors
- that are better than the best thing seen in the data!

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

# Distribution shift in a nutshell

Example empirical risk minimization (ERM) problem:

$$\theta \leftarrow \arg\min_{\theta} E_{x\sim p(x),y\sim p(y|x)}[(f_\theta(x)-y)^2]$$

given some $x^*$, is $f_\theta(x^*)$ correct?

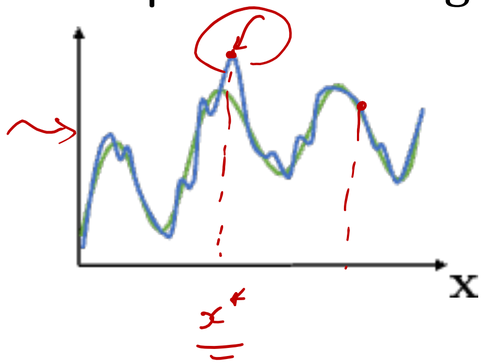
$$E_{x\sim p(x),y\sim p(y|x)}[(f_\theta(x)-y)^2] \textit{ is low}$$

$$E_{x\sim \overline{p}(x),y\sim p(y|x)}[(f_\theta(x)-y)^2] \textit{ is not, for general } \overline{p}(x) \neq p(x)$$

what if $x^* \sim p(x)$? not necessarily...

usually we are not worried – neural nets generalize well!

what if we pick $x^* \leftarrow \arg\max_x f_\theta(x)$?




Kumar, Fu, Tucker, Levine. Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction. NeurIPS '19

# Where do we suffer from distribution shift?

actual

OOD action w.r.t $\pi_\beta$

$$\cancel{Q(s,a) \leftarrow r(s,a) + \max_{a'} Q(s',a')}$$

$$Q(s,a) \leftarrow r(s,a) + \underbrace{E_{a' \sim \pi_{new}}[Q(s',a')]}_{y(s,a)}$$

under $\pi_\beta = \pi_{new}$

precise

even worse: $\pi_{new} = \text{argmax}_\pi E_{a \sim \pi(a|s)}[Q(s,a)]$

(what if we pick $x^* \leftarrow \text{argmax}_x f_\theta(x)$?

what is the objective?

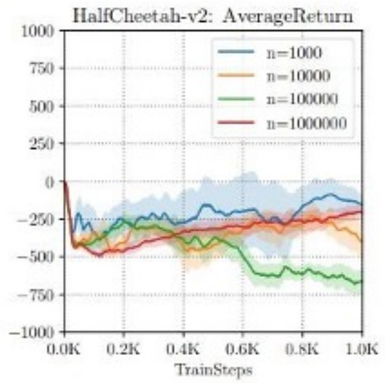
$$\min_Q E_{(s,a) \sim \pi_\beta(s,a)}[(Q(s,a) - y(s,a))^2]$$

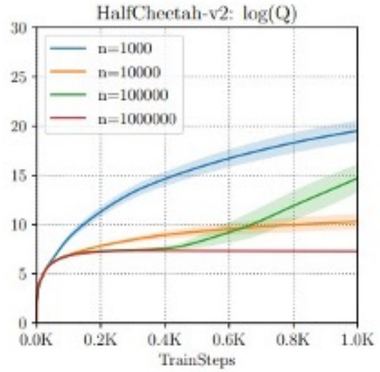
behavior policy

target value

expect good accuracy when $\pi_\beta(a|s) = \pi_{new}(a|s)$

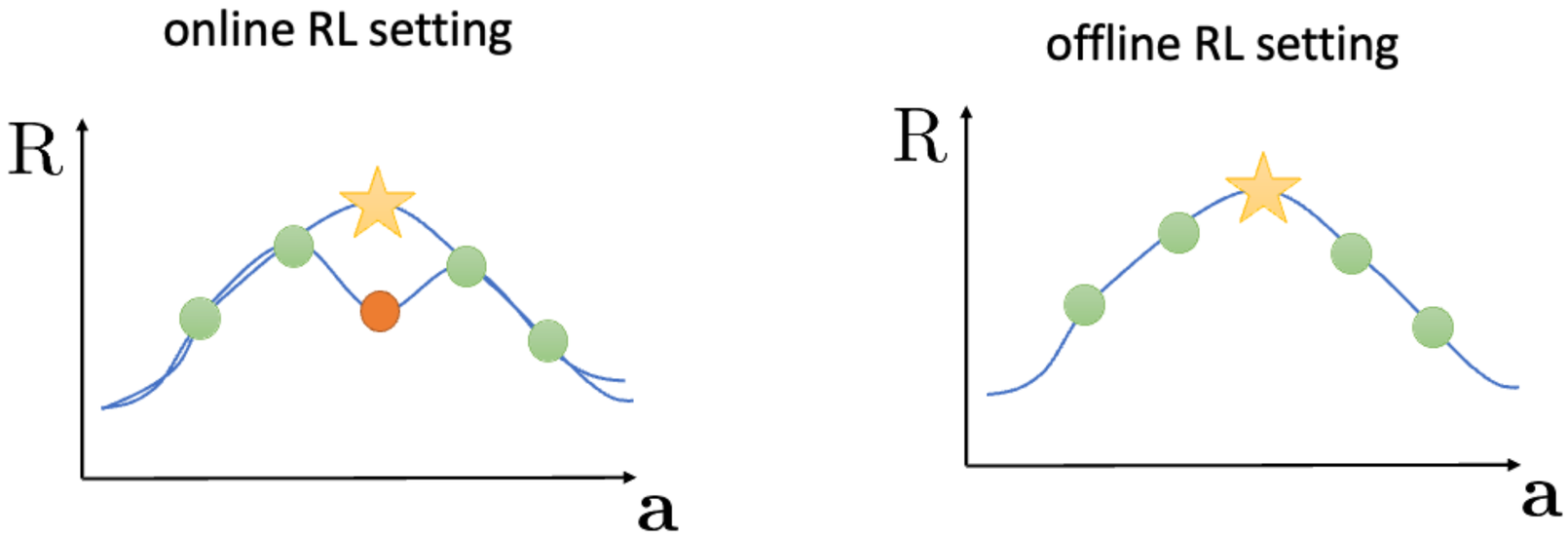
how often does that happen?




Kumar, Fu, Tucker, Levine. Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction. NeurIPS '19

# Issues with generalization are not corrected




Existing challenges with sampling error and function approximation error in standard RL become much more severe in offline RL

# Policy Constraint Methods

# How do prior methods address this?

Actor-critic

$$Q(s,a) \leftarrow r(s,a) + E_{a' \sim \pi_{new}}[Q(s',a')]$$

$$\pi_{new}(a|s) = \arg\max_{\pi} E_{a \sim \pi(a|s)}[Q(s,a)] \; s.t. \; D_{KL}(\pi \| \pi_\beta) \leq \epsilon$$

$\pi_\beta \approx \pi_{new}$

$\Rightarrow E_{a' \sim \pi_{new}}[Q(s',a')]$ precise

- This solves distribution shift, right?
- No more erroneous values?
- Issue 1: we usually don't know the behavior policy $\pi_\beta(a|s)$
  - ☑ human-provided data
  - ☑ data from hand-designed controller
  - ☑ data from many past RL runs
  - ☑ all of the above
- Issue 2: this is both too pessimistic and not pessimistic enough



$D_{KL}(\pi \| \pi_\beta)$

- "policy constraint" method
- very old idea (but it had no single name?)
- Todorov et al. [passive dynamics in linearly-solvabl
- Kappen et al. [KL-divergence control, etc.]
- trust regions, covariant policy gradients, natural po
- used in some form in recent papers:
- Fox et al. '15 ("Taming the Noise...")
- Fujimoto et al. '18 ("Off Policy...")
- Jaques et al. '19 ("Way Off Policy...")
- Kumar et al. '19 ("Stabilizing...")
- Wu et al. '19 ("Behavior Regularized...")

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

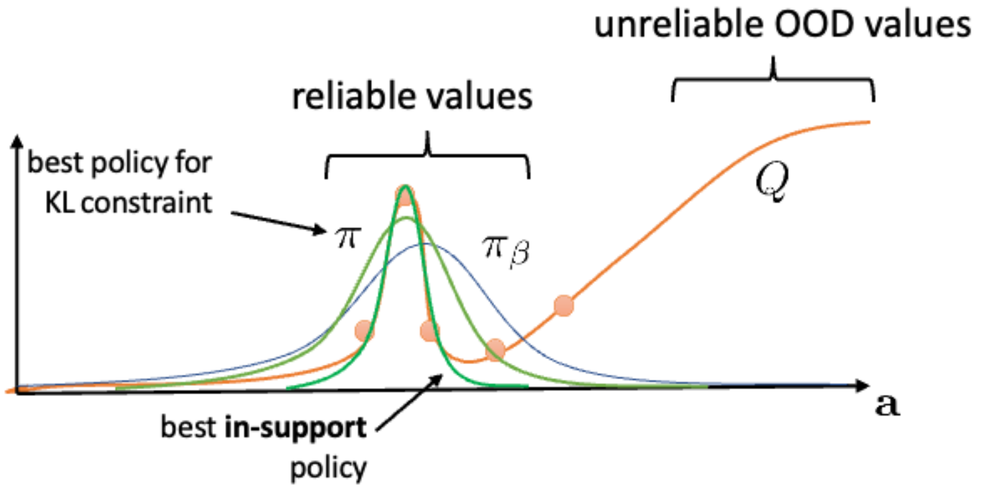
# Explicit policy constraint methods

✦ What kinds of constraints can we use?

KL-divergence: $D_{KL}(\pi \parallel \pi_\beta)$

+ easy to implement (more on this later)

- not necessarily what we want




✦ support constraint: $\pi(a|s) \geq 0 \; only \; if \; \pi_\beta(a|s) \geq \epsilon$

can approximate with MMD

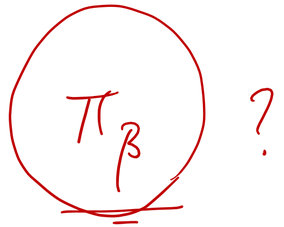
- significantly more complex to implement

+ much closer to what we really want

✦ For more information, see:

- ☑ Levine, Kumar, Tucker, Fu. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. '20
- ☑ Kumar, Fu, Tucker, Levine. Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction. '19
- ☑ Wu, Tucker, Nachum. Behavioral Regularized Offline Reinforcement Learning. '19

# Explicit policy constraint methods (Cont.)




- How do we implement constraints?
  - Modify the actor objective

Lagrange multiplier

easy to compute and differentiate for Gaussian or categorical po

~~$\theta \leftarrow \arg\max_\theta E_{s\sim D}\left[E_{a\sim\pi_\theta(a|s)}[Q(s,a)]\right]$~~

$$\theta \leftarrow \arg\max_\theta E_{s\sim D}\left[E_{a\sim\pi_\theta(a|s)}[Q(s,a) + \lambda \log \pi_\beta(a|s)]\right] + \lambda H(\pi(a|s))$$

$$D_{KL}(\pi \parallel \pi_\beta) = E_\pi[\log\pi(a|s) - \log\pi_\beta(a|s)] = -E_\pi[\log\pi_\beta(a|s)] - H(\pi)$$

  - Modify the reward function

$$\overline{r}(s,a) = r(s,a) - D(\pi, \pi_\beta)$$

simple modification to directly penalize

divergence also accounts for future divergence

See: Wu, Tucker, Nachum. Behavior Regularized Offline Reinforcement Learning. `19

generally, the best modern offline RL methods do not do either of these things

# Implicit policy constraint methods

$$\pi_{new}(\mathbf{a}|\mathbf{s}) = \arg\max_{\pi} E_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}[Q(\mathbf{s},\mathbf{a})] \; s.t. D_{KL}(\pi \| \pi_\beta) \leq \epsilon$$

$$\pi^*(\mathbf{a}|\mathbf{s}) = \frac{1}{Z(\mathbf{s})}\pi_\beta(\mathbf{a}|\mathbf{s})\exp\left(\frac{1}{\lambda}A^\pi(\mathbf{s},\mathbf{a})\right)$$

approximate via weighted max likelihood!

$$\pi_{new}(\mathbf{a}|\mathbf{s}) = \arg\max_{\pi} E_{(\mathbf{s},\mathbf{a})\sim\pi_\beta}\left[\log\pi(\mathbf{a}|\mathbf{s}) \underbrace{\frac{1}{Z(\mathbf{s})}\exp\left(\frac{1}{\lambda}A^{\pi}old(\mathbf{s},\mathbf{a})\right)}_{w(s,a)}\right]$$

samples from dataset $\mathbf{a} \sim \pi_\beta(\mathbf{a}|\mathbf{s})$

critic can be used to give us this

closed form sol.

$\pi_\theta \leftrightarrow \pi^*$

straightforward to show via duality

See also: Peters et al. (REPS)

$\max_\theta \; \mathbb{E}_{\pi^*}[\log \pi_\theta]$

$\int \log\pi_\theta \frac{1}{Z(s)} \pi_\beta(a|s)$ (density)

$\int_a g_\theta(a) \exp\{\frac{1}{\lambda} A^{\pi_\theta}(s,a)\} da$

$= \mathbb{E}_{(s,a)\sim\pi_\beta}\{g_\theta(s,a)\}$

Peng, Kumar, Levine. Advantage-Weighted Regression. '19

# Implicit policy constraint methods (Cont.)

$$\mathcal{L}_C(\phi) = E_{(s,a,s')\sim D}\left[\left(Q_\phi(s,a) - \left(r(s,a) + \gamma E_{a'\sim\pi_\theta(a'|s')}[Q_\phi(s',a')]\right)\right)^2\right]$$

it's the same policy that maximizes the Q-func.

$$\mathcal{L}_A(\theta) = -E_{(s,a)\sim\pi_\beta}\left[\log\pi_\theta(a|s)\frac{1}{Z(s)}\exp\left(\frac{1}{\lambda}A^{\pi}old(s,a)\right)\right]$$

Offline Data

implicitly enforces $D_{KL}(\pi_\theta \| \pi_\beta) \approx 0$

1. $\phi \leftarrow \phi - \alpha\nabla_\phi\mathcal{L}_C(\phi)$
2. $\theta \leftarrow \theta - \alpha\nabla_\theta\mathcal{L}_A(\theta)$

$$Q(s,a) \leftarrow r(s,a) + E_{a'\sim\pi_{new}}[Q(s',a')]$$

$$\pi_{new}(a|s) = \arg\max_\pi E_{a\sim\pi(a|s)}[Q(s,a)] \; s.t. D_{KL}(\pi \| \pi_\beta) \leq \epsilon$$

Peng, Kumar, Levine. Advantage-Weighted Regression. '19

# Can we also avoid all OOD actions in the Q update?

$$Q(s,a) \leftarrow r(s,a) + E_{a' \sim \pi_{new}}[Q(s',a')]$$

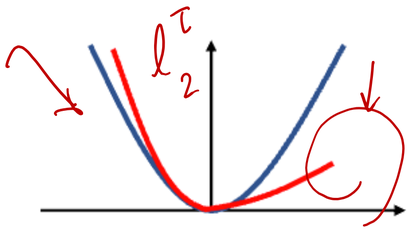
$V(s') \leftarrow$ *just another neural network*

$$V \leftarrow \text{argmin} \frac{1}{N} \sum_{i=1}^{N} \ell(V(s_i), Q(s_i,a_i))$$

this action comes from $\pi_\beta$ not from $\pi_{new}$

e.g., MSE loss $(V(s_i) - Q(s_i,a_i))^2$

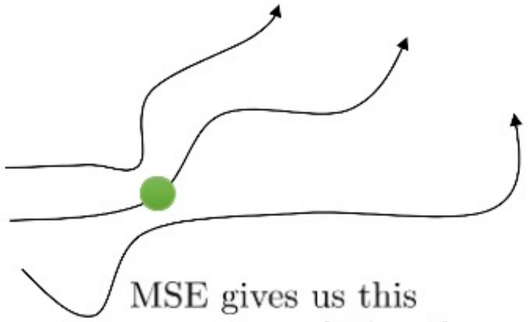
expectile: $\ell_2^\tau(x) = \begin{cases} (1-\tau)x^2 & \text{if } x > 0 \\ \tau x^2 & \text{else} \end{cases}$



$$V(s) \leftarrow \max_{a \in \Omega(s)} Q(s,a)$$

$$\Omega(s) = \{a : \pi_\beta(a|s) \geq \epsilon\}$$

if we use $\ell_2^T$ for big $\tau$




Kostrikov, Nair, Levine. Offline Reinforcement Learning with Implicit Q-Learning. '21

# Implicit Q-learning (IQL)

$$Q(s,a) \leftarrow r(s,a) + V(s')$$

$$V(s) \leftarrow \max_{a \in \Omega(s)} Q(s,a)$$

$$\Omega(s) = \{a : \pi_\beta(a|s) \geq \epsilon\}$$

if we use $\ell_2^T$ for big $\tau$

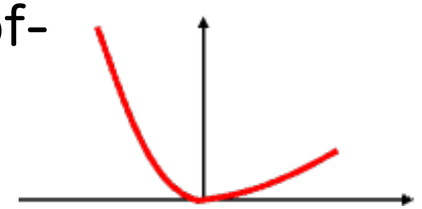
$$V \leftarrow \arg\min_V \frac{1}{N} \sum_{i=1}^{N} \ell_2^V(V(s_i), Q(s_i,a_i))$$

$$Q(s,a) \leftarrow r(s,a) + \max_{a' \in \Omega(s')} Q(s',a')$$

"implicit" policy

$$\pi_{new}(a|s) = \delta(a = \arg\max_{a \in \Omega(s)} Q(s,a))$$

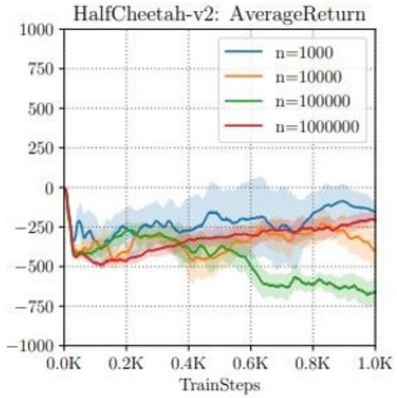
- Now we can do value function updates without ever risking out-of-distribution actions!



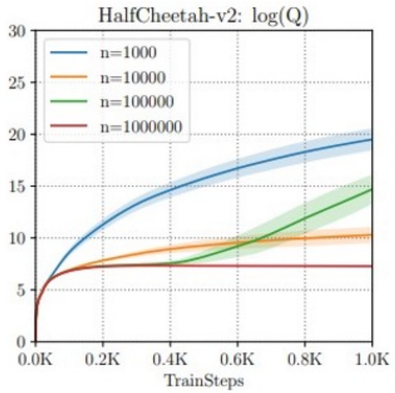
Kostrikov, Nair, Levine. Offline Reinforcement Learning with Implicit Q-Learning. '21

# Conservative Q-Learning

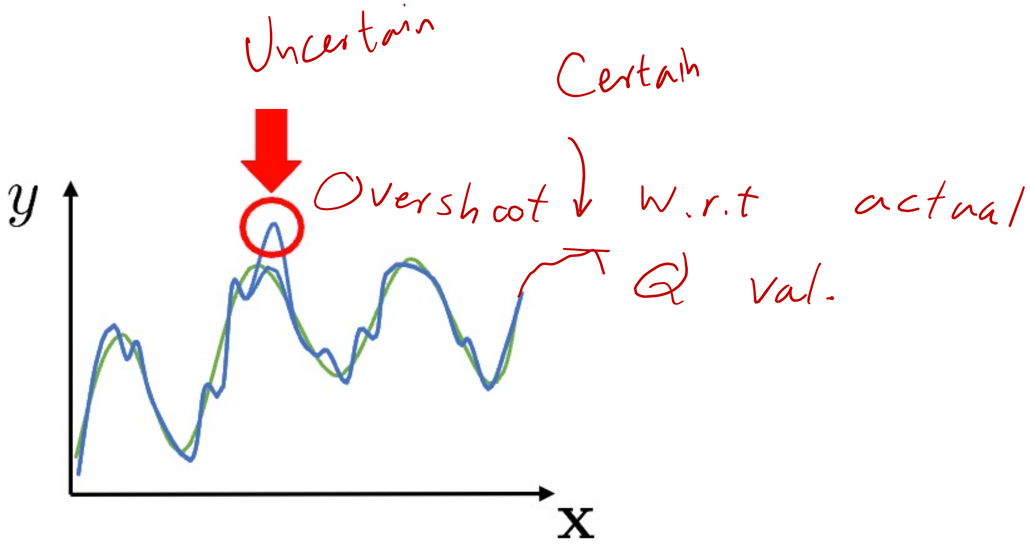
# Conservative Q-learning (CQL)




how well it does




how well it *thinks* it does (Q-values)




$$\hat{Q}^{\pi} = \arg\min_{Q} \max_{\mu} \alpha E_{s\sim D, a\sim\mu(a|s)}[Q(s,a)]$$ :term to push down big Q-values

regular objective: $+E_{(s,a,s')\sim D}[(Q(s,a) - (r(s,a) + E_{\pi}[Q(s',a')]))^2]$

can show that $\hat{Q}^{\pi} \leq Q^{\pi}$ *for large enough* $\alpha$

true Q-function

# Conservative Q-learning (CQL) - (Cont.)

A better bound:

always pushes Q-values down

push up on (s, a) samples in data

$$\hat{Q}^{\pi} = \arg\min_{Q} \max_{\mu} \alpha E_{s\sim D, a\sim\mu(a|s)}[Q(s,a)] - \alpha E_{(s,a)\sim D}[Q(s,a)] + E_{(s,a,s')\sim D}[(Q(s,a) - (r(s,a) + E_{\pi}[Q(s',a')]))^2]$$

$\mathcal{L}_{CQL}\hat{Q}^{\pi}$

no longer guaranteed that $\hat{Q}^{\pi}(s,a) \leq Q^{\pi}(s,a) \, for \, all (s,a)$

but guaranteed that $E_{\pi(a|s)}\left[\hat{Q}^{\pi}(s,a)\right] \leq E_{\pi(a|s)}[Q^{\pi}(s,a)] \, for \, all s \in D$

Kumar, Zhou, Tucker, Levine. Conservative Q-Learning for Offline Reinforcement Learning. '20

# Conservative Q-learning (CQL) - (Cont.)

1. Update $\hat{Q}^{\pi}$ $w.r.t. \mathcal{L}_{CQL}(\hat{Q}^{\pi})$ $using \mathcal{D}$
2. Update policy $\pi$

$\hat{Q} \leq Q$

if actions are discrete:

$$\pi(\mathbf{a}\,|\,\mathbf{s}) = \begin{cases} 1 & \text{if } \mathbf{a} = \arg\max_{\mathbf{a}} \hat{Q}(\mathbf{s}, \mathbf{a}) \\ 0 & \text{otherwise} \end{cases}$$

if actions are continuous:

$$\theta \leftarrow \theta + \alpha \nabla_{\theta} \sum_{i} E_{\mathbf{a} \sim \pi_{\theta}(\mathbf{a}|\mathbf{s}_i)}[\hat{Q}(\mathbf{s}_i, \mathbf{a})]$$

Kumar, Zhou, Tucker, Levine. Conservative Q-Learning for Offline Reinforcement Learning. '20

# Conservative Q-learning (CQL) - (Cont.)

bi-level

$$\hat{Q}^{\pi} = \arg\min_{Q} \max_{\mu} \alpha E_{s\sim D, a\sim\mu(a|s)}[Q(s,a)] - \alpha E_{(s,a)\sim D}[Q(s,a)] + \mathcal{R}(\mu)$$

$$+ E_{(s,a,s')\sim D}[(Q(s,a) - (r(s,a) + E_{\pi}[Q(s',a')]))^2]$$

$\mathcal{L}_{CQL}\hat{Q}^{\pi}$

$D_{KL}(\mu \| \rho)$

$\mathcal{H}(\mu)$

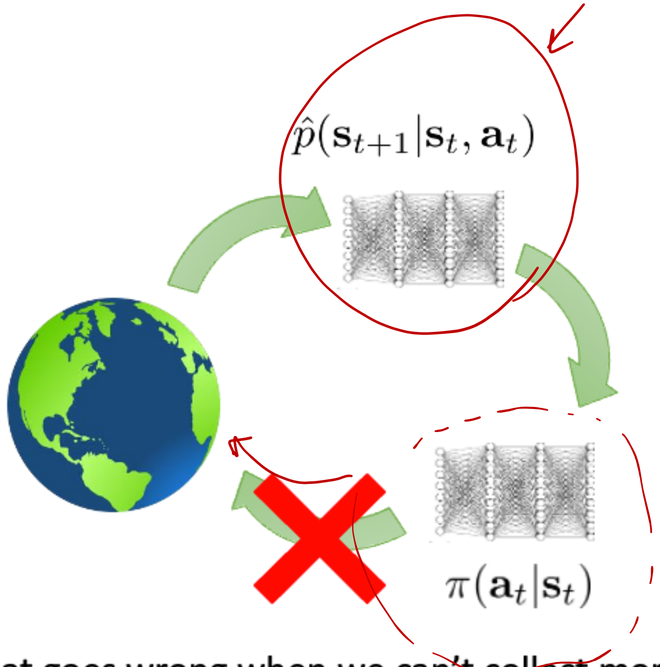
$\mu(a|s) \stackrel{?}{=} \frac{e^{\beta Q(s,a)}}{\sum_{a'} e^{\beta Q(s,a')}}$

common choice: $\mathcal{R} = E_{s\sim D}[\mathcal{H}(\mu(\cdot|s))]$ maximum entropy regularization

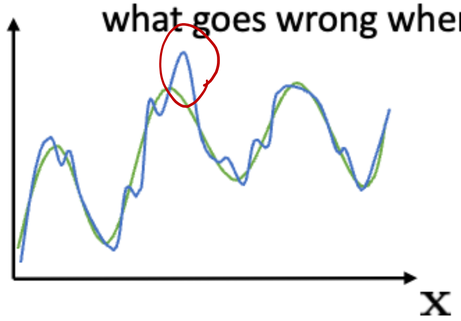
Kumar, Zhou, Tucker, Levine. Conservative Q-Learning for Offline Reinforcement Learning. '20

# Model-Based Offline RL

# How does model-based RL work?

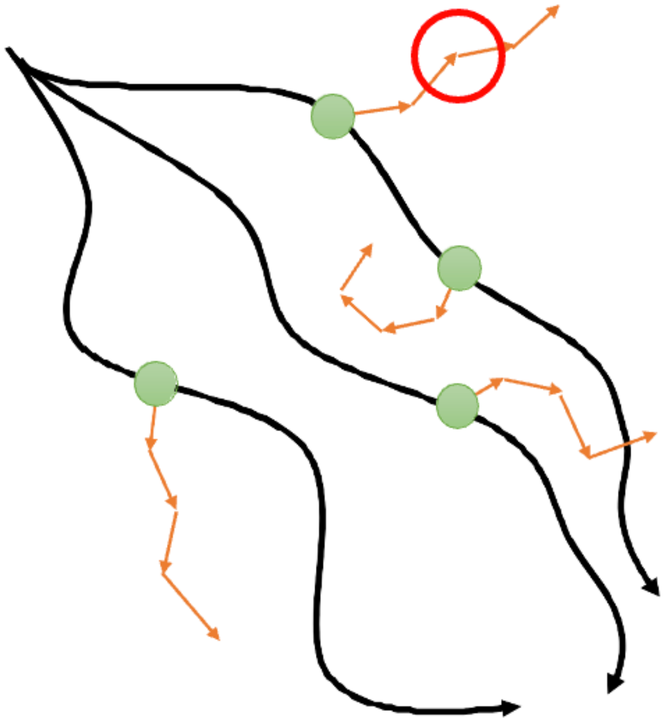



what goes wrong when we can't collect more data?




...so the model's predictions are invalid

these states are OOD



the model answers "what if" questions

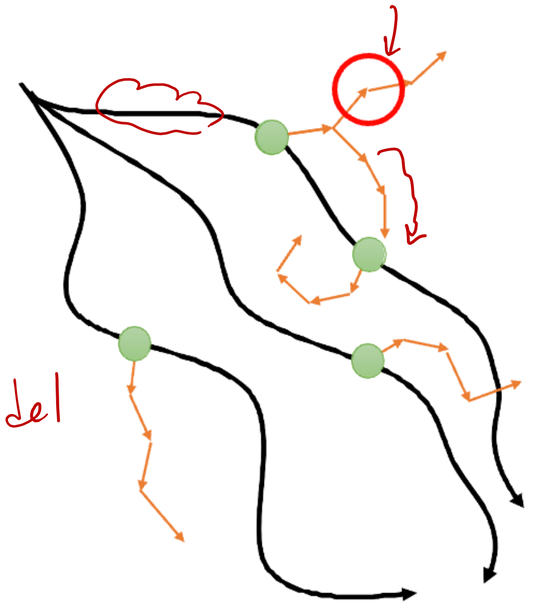
# Model-Based Offline RL

solution: “punish” the policy for exploiting

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

uncertainty penalty

Ensemble of World Model



…and then use any existing model-based RL algorithm

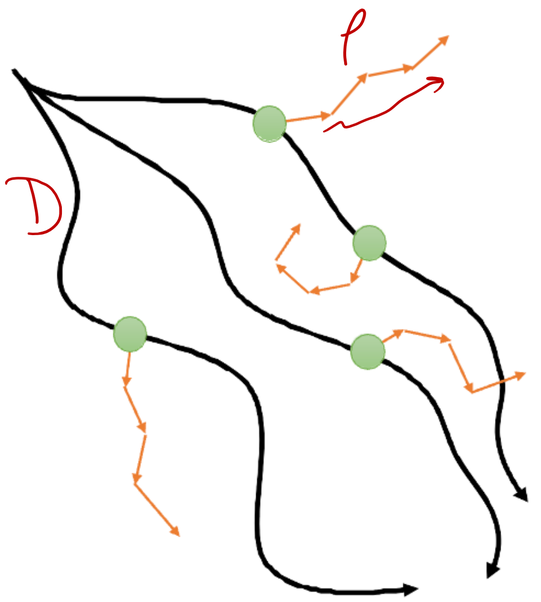
Yu, Thomas, Yu, Ermon, Zou, Levine, Finn, Ma. MOPO: Model-Based Offline Policy Optimization. ‘20

See also: Kidambi et al., MOReL: Model-Based Offline Reinforcement Learning. ‘20 (concurrent)

# Conservative Model-Based RL

Basic idea: just like CQL minimizes Q-value of policy actions, we can minimize Q-value of model state-action tuples

$\rho = \hat{M} \cdot \pi$




state-action tuples from the model

$$Q^{k+1} \leftarrow \arg\min_{Q} \beta\left(\mathbb{E}_{s,a\sim\rho(s,a)}[Q(s,a)] - \mathbb{E}_{s,a\sim D}[Q(s,a)]\right)$$

$$+\frac{1}{2}\mathbb{E}_{s,a,s'\sim d_f}\left[\left(Q(s,a) - B^{\pi}\hat{Q}^{k}(s,a)\right)^2\right].$$

Intuition: if the model produces something that looks clearly different from real data, it's easy for the Q-function to make it look bad

Yu, Kumar, Rafailov, Rajeswaran, Levine, Finn. COMBO: Conservative Offline Model-Based Policy Optimization. 2021.

# Summary, Applications, Open Questions

# Which offline RL algorithm do I use?

If you want to only train offline…

Conservative Q-learning + just one hyperparameter + well understood and widely tested Implicit Q-learning + more flexible (offline + online) - more hyperparameters

If you want to only train offline and finetune online

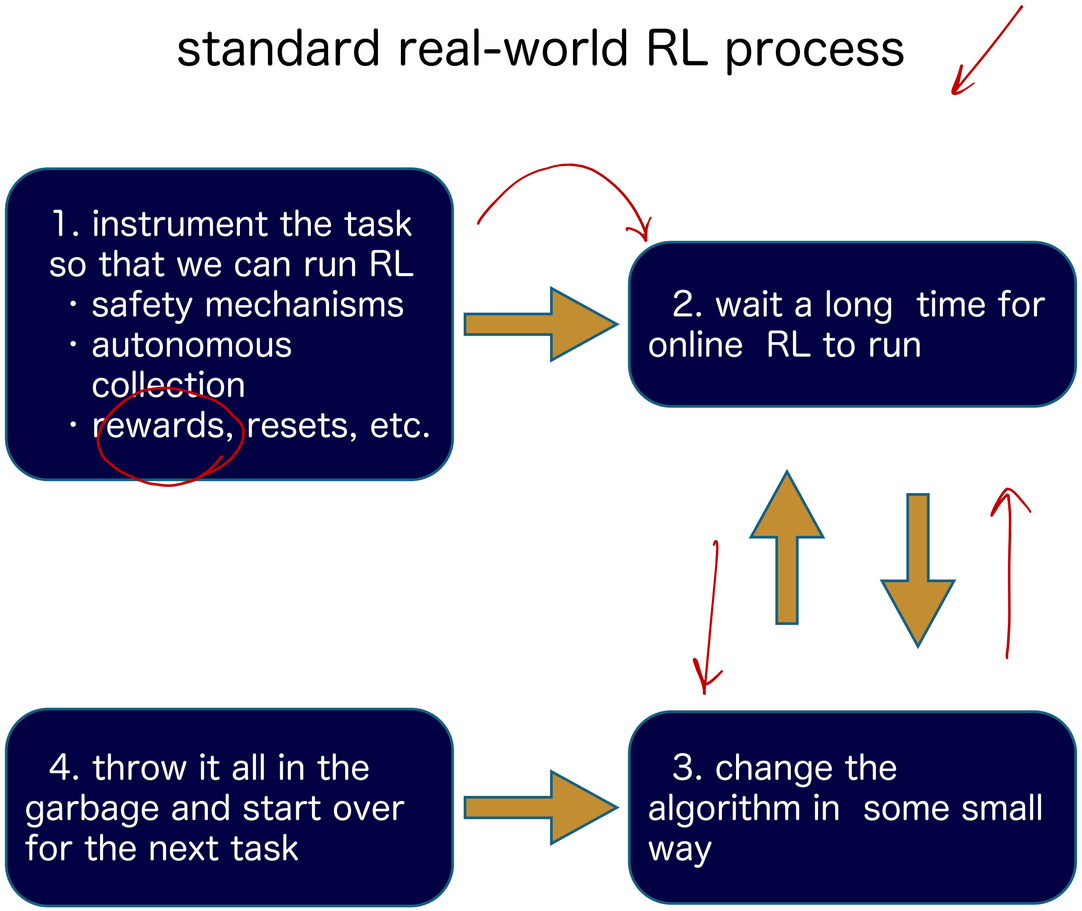
Advantage-weighted actor-critic (AWAC) + widely used and well tested Implicit Q-learning + seems to perform much better!

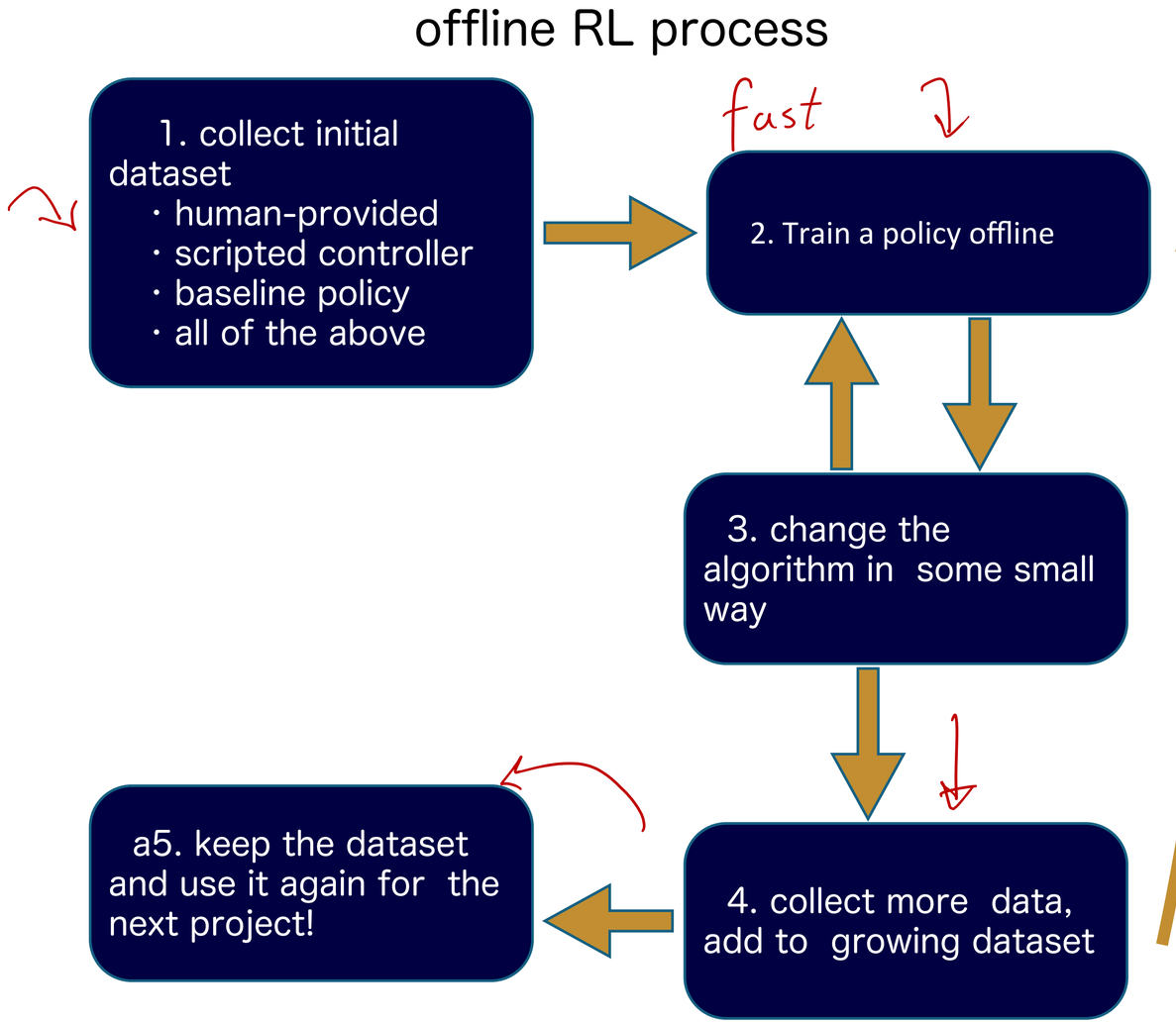
If you have a good way to train models in your domain

COMBO + similar properties as CQL, but benefits from models

- not always easy to train a good model in your domain!

# The power of offline RL



# Takeaways, conclusions, future directions

current offline RL algorithms

"The Gap"

"the dream"

1. Collect a dataset using any policy or mixture of policies
2. Run offline RL on this dataset to learn a policy
3. Deploy the policy in the real world

- An offline RL workflow
  - Supervised learning workflow: train/test split
  - Offline RL workflow: ???
- Statistical guarantees
  - Biggest challenge: distributional shift/counterfactuals
  - Can we make any guarantees?
- Scalable methods, large-scale applications
  - Dialogue systems
  - Data-driven navigation and driving