Offline Reinforcement Learning
and Offline Multi-Task RL

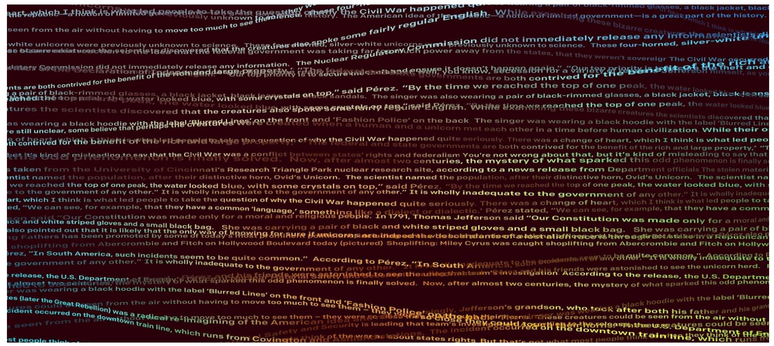
CS 330

Reminders

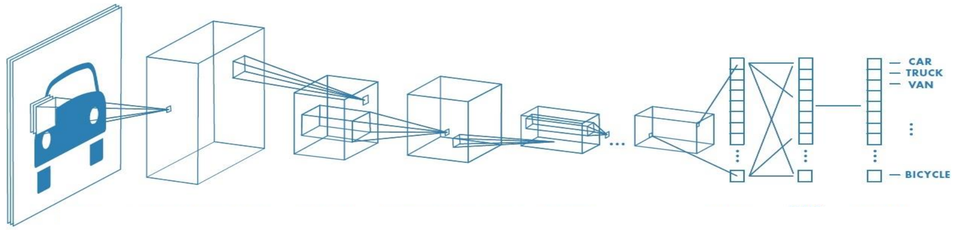
Next Monday (Nov 8th): Homework 4 (optional) is due

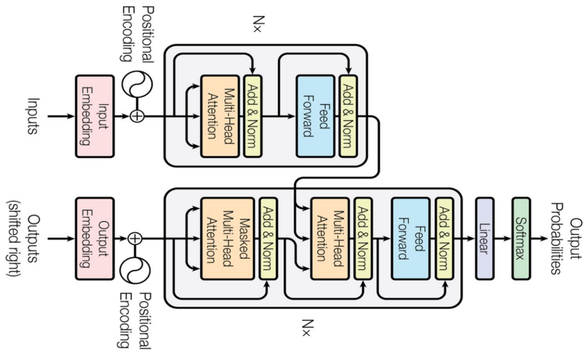
The recipe that has worked in other fields so far:

A lot of data



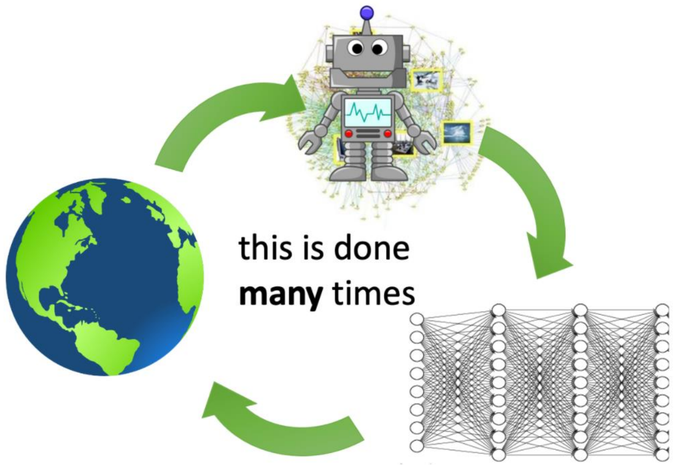
Expressive, capable models





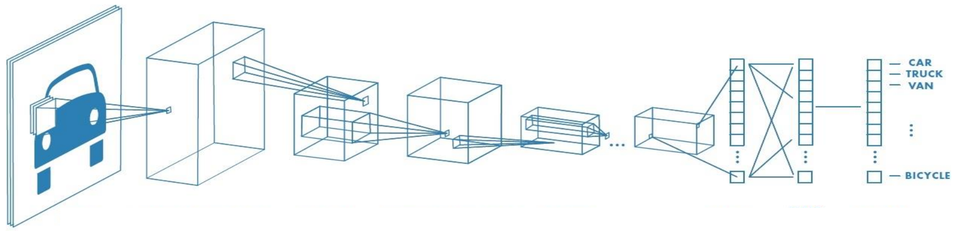
The reinforcement learning recipe so far:

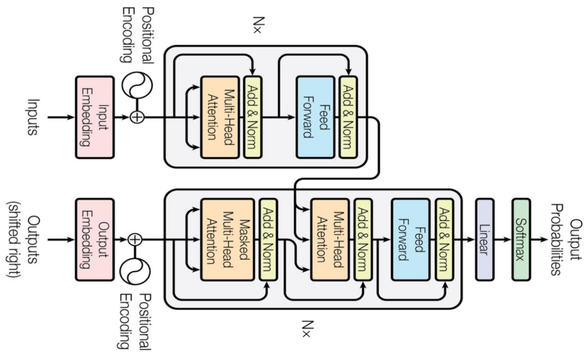
A lot of data



Hard to achieve same level of generalization with a per-experiment active learning loop!

Expressive, capable models





The Plan

Offline RL problem formulation

Offline RL solutions

Offline multi-task RL and data sharing

Offline goal-conditioned RL

The Plan

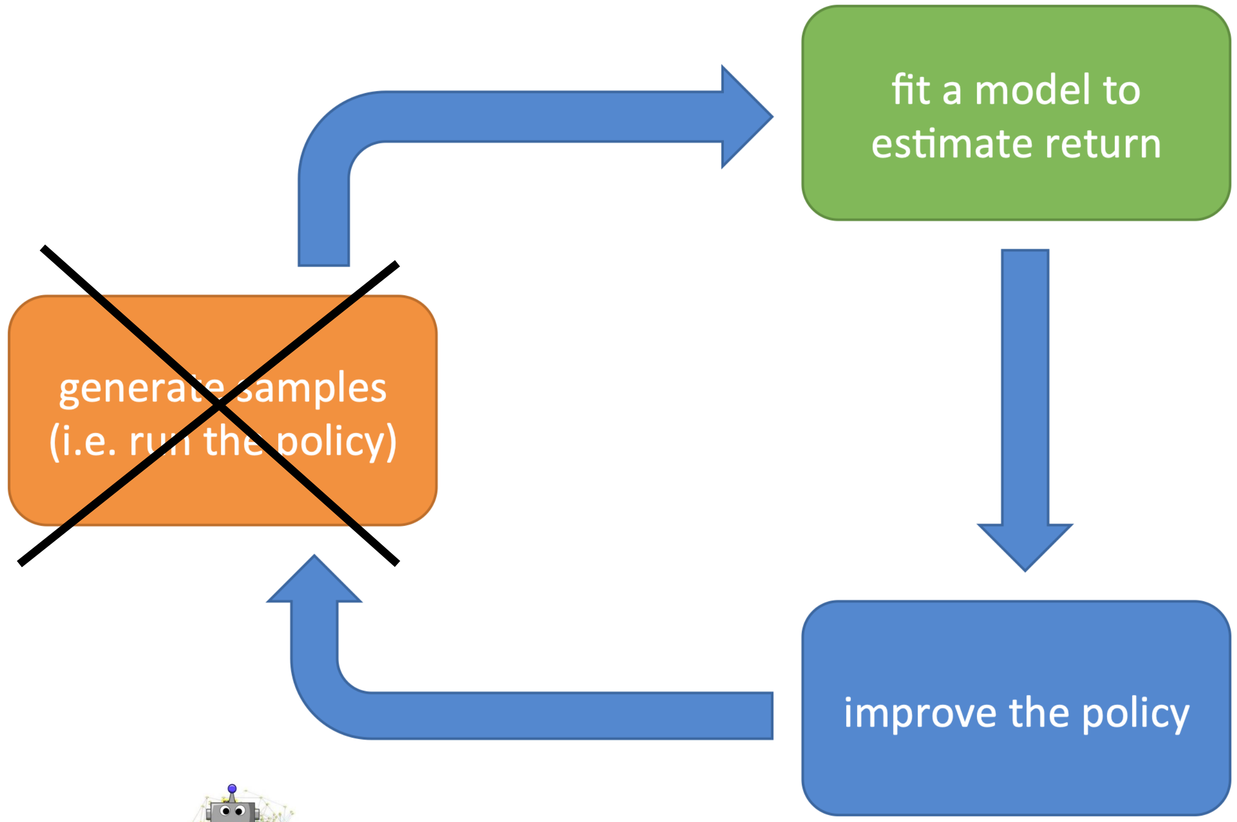
Offline RL problem formulation

Offline RL solutions

Offline multi-task RL and data sharing

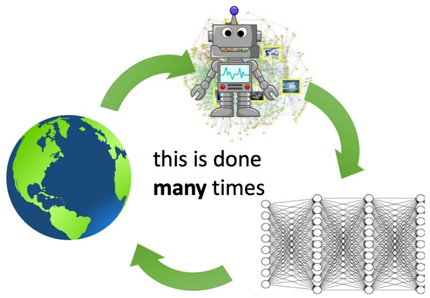
Offline goal-conditioned RL

The anatomy of a reinforcement learning algorithm



compute $\hat{Q} = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$ (MC policy gradient)
fit $Q_\phi(\mathbf{s}, \mathbf{a})$ (actor-critic, Q-learning)
estimate $p(\mathbf{s}'|\mathbf{s}, \mathbf{a})$ (model-based)

$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$ (policy gradient)
$\pi(\mathbf{s}) = \arg\max Q_\phi(\mathbf{s}, \mathbf{a})$ (Q-learning)
optimize $\pi_\theta(\mathbf{a}|\mathbf{s})$ (model-based)



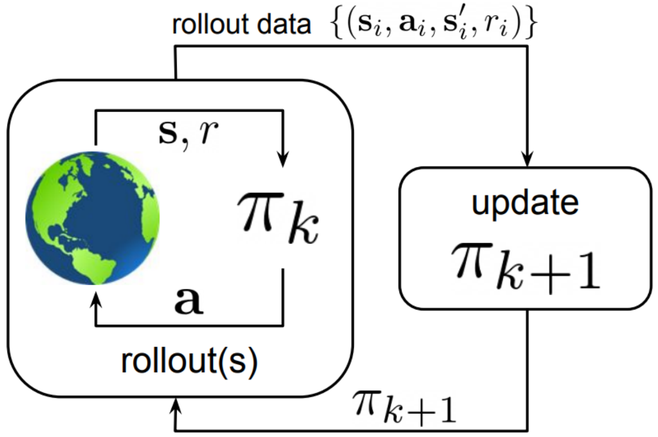
On-policy vs Off-policy

- Data comes from the current policy
- Compatible with all RL algorithms
- Can’t reuse data from previous policies

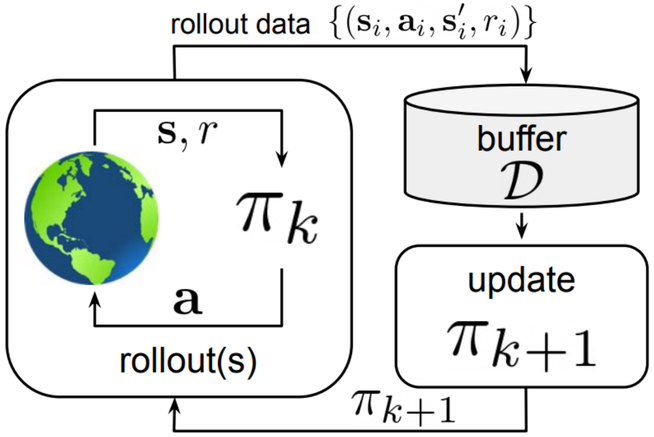
- Data comes from any policy
- Works with specific RL algorithms
- Much more sample efficient, can re-use old data

Off-policy RL and offline RL

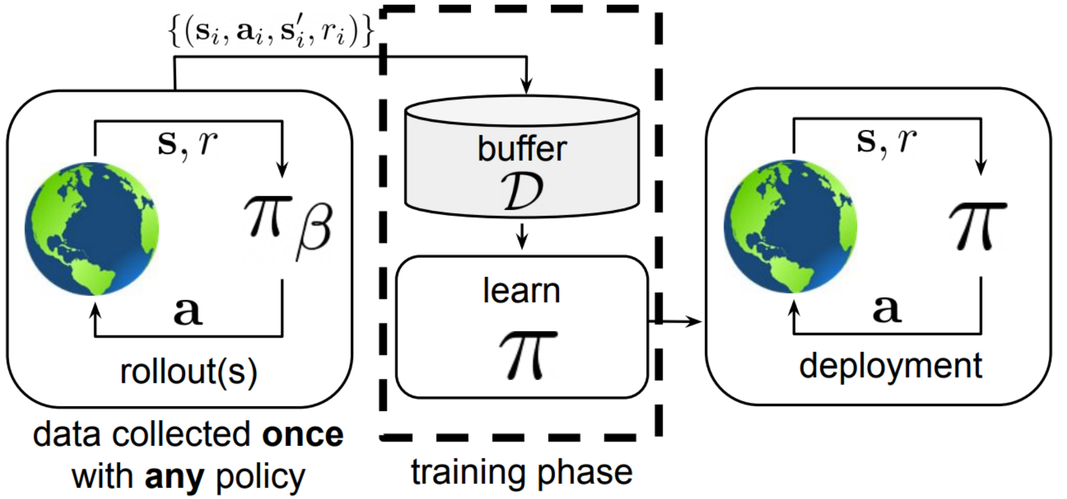
(a) online reinforcement learning



(b) off-policy reinforcement learning



(c) offline reinforcement learning



$$\mathcal{D} = \{(\mathbf{s}_i, \mathbf{a}_i, \mathbf{s'}_i, r_i)\}$$

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

$$\mathbf{a} \sim \pi_\beta(\mathbf{a}|\mathbf{s})$$ ← unknown!

$$\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{a}, \mathbf{s})$$

$$\arg\max_\theta \sum_{t=0}^{T} E_{\mathbf{s}_t \sim d^{\pi_\beta}(\mathbf{s}), \mathbf{a}_t \sim \pi_\theta(\mathbf{a}|\mathbf{s})}[\gamma^t r(\mathbf{s}_t, \mathbf{a}_t)]$$

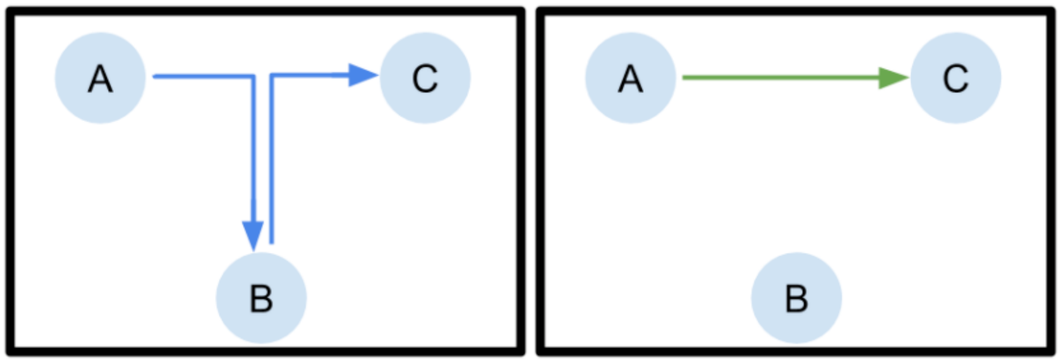
Can you think of potential applications of offline RL?

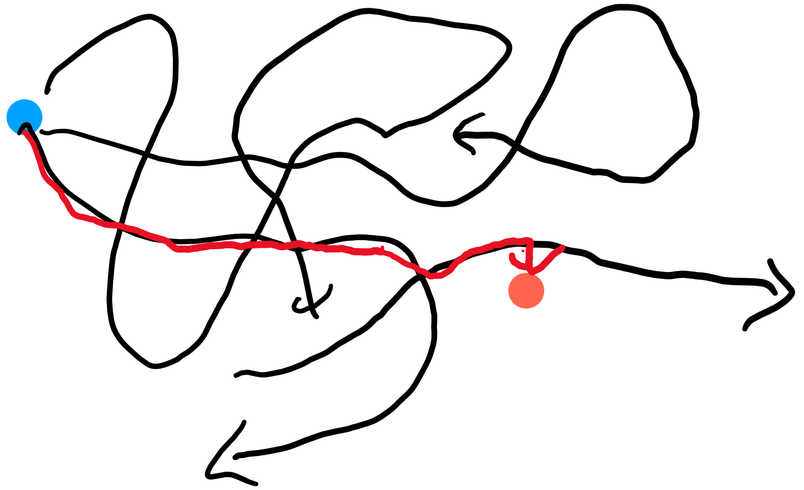
What can offline RL do?

Find best behaviors in a dataset

Generalize best behaviors to similar situations

“Stitch” together parts of good behaviors into a better behavior





Fitted Q-iteration algorithm

full fitted Q-iteration algorithm:

1. collect dataset $\{(\mathbf{s}_i, \mathbf{a}_i, \mathbf{s}'_i, r_i)\}$ using some policy
2. set $\mathbf{y}_i \leftarrow r(\mathbf{s}_i, \mathbf{a}_i) + \gamma \max_{\mathbf{a}'_i} Q_\phi(\mathbf{s}'_i, \mathbf{a}'_i)$
3. set $\phi \leftarrow \arg\min_\phi \frac{1}{2} \sum_i \left\| Q_\phi(\mathbf{s}_i, \mathbf{a}_i) - \mathbf{y}_i \right\|^2$

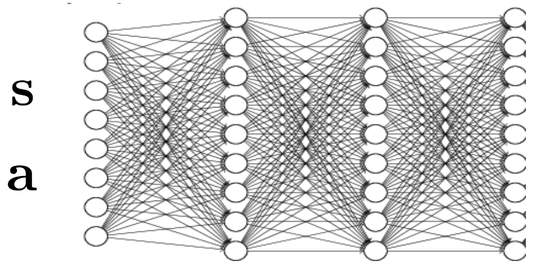
$K\times$

Algorithm hyperparameters

dataset size N, collection policy

iterations K

gradient steps S



$Q_\phi(\mathbf{s}, \mathbf{a})$
parameters ϕ

Result: get a policy $\boldsymbol{\pi(\mathbf{a}|\mathbf{s})}$ from $\boldsymbol{\arg\max_{\mathbf{a}} Q_\phi(\mathbf{s}, \mathbf{a})}$

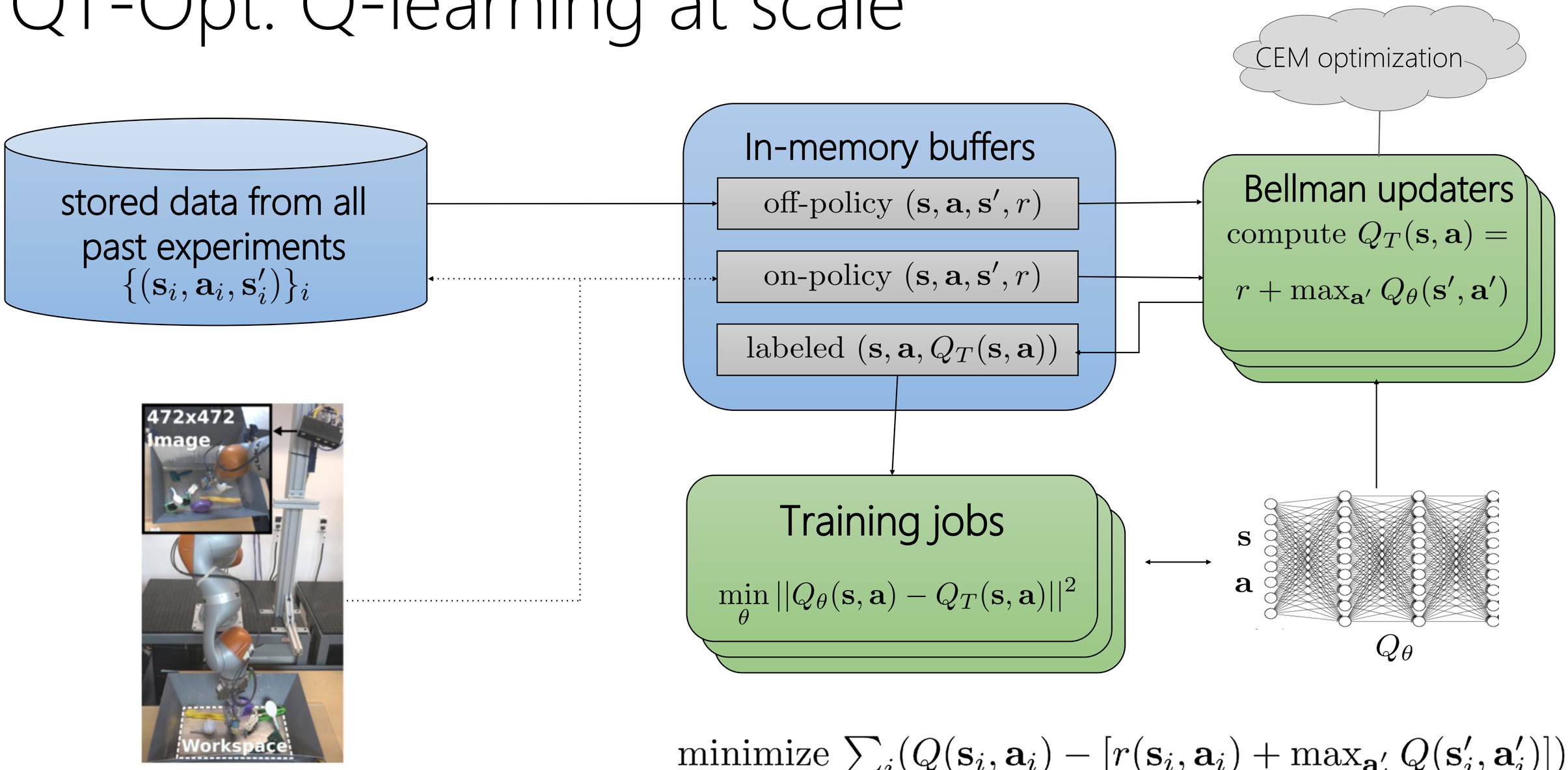
Important notes:

We can reuse data from previous policies!
an off-policy algorithm using replay buffers

This is not a gradient descent algorithm!

Slide adapted from Sergey Levine

QT-Opt: Q-learning at scale



$$\text{minimize} \sum_i (Q(\mathbf{s}_i, \mathbf{a}_i) - [r(\mathbf{s}_i, \mathbf{a}_i) + \max_{\mathbf{a}'_i} Q(\mathbf{s}'_i, \mathbf{a}'_i)])^2$$

Kalashnikov, et al. QT-Opt, 2018

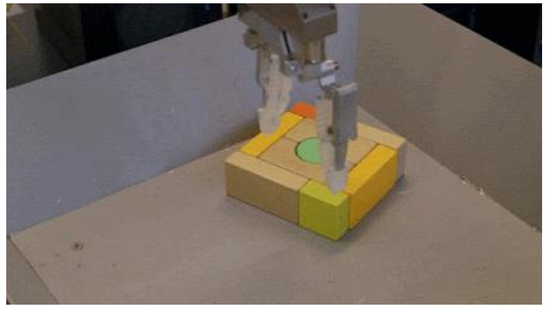
QT-Opt: setup and results

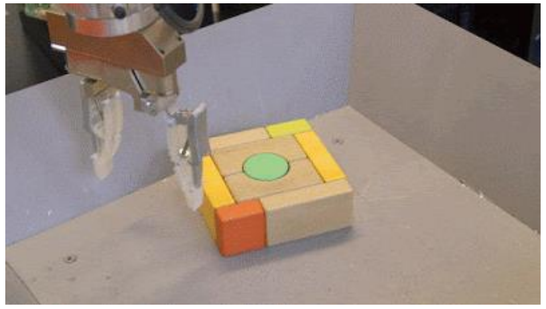
7 robots collected 580k grasps

Unseen test objects

580k offline + 28k online ⟶ 96%

580k offline ⟶ 87%





The Plan

Offline RL problem formulation

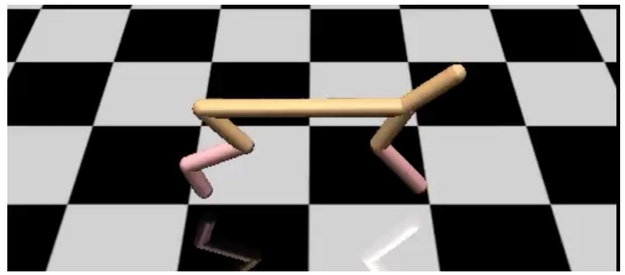
Offline RL solutions

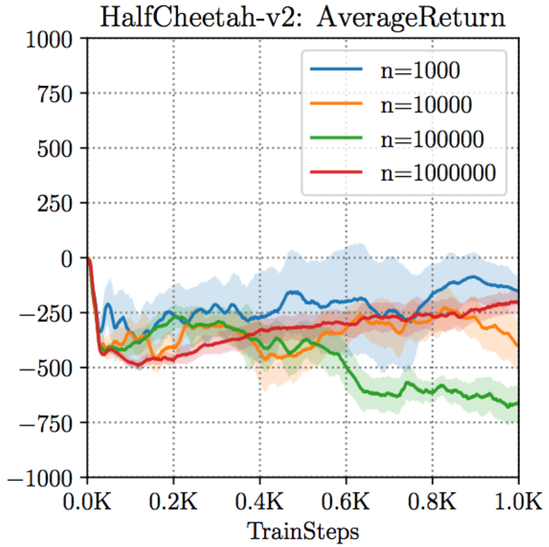
Offline multi-task RL and data sharing

Offline goal-conditioned RL

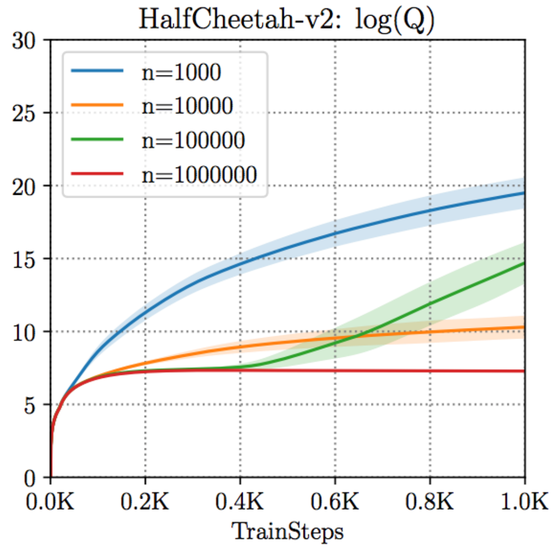
The offline RL problem

Bellman equation: $Q^\star(\mathbf{s}_t, \mathbf{a}_t) = \mathbb{E}_{\mathbf{s}' \sim p(\cdot|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a}) + \gamma \max_{\mathbf{a}'} Q^\star(\mathbf{s}', \mathbf{a}')\right]$





How well it does?



How well it thinks it does?

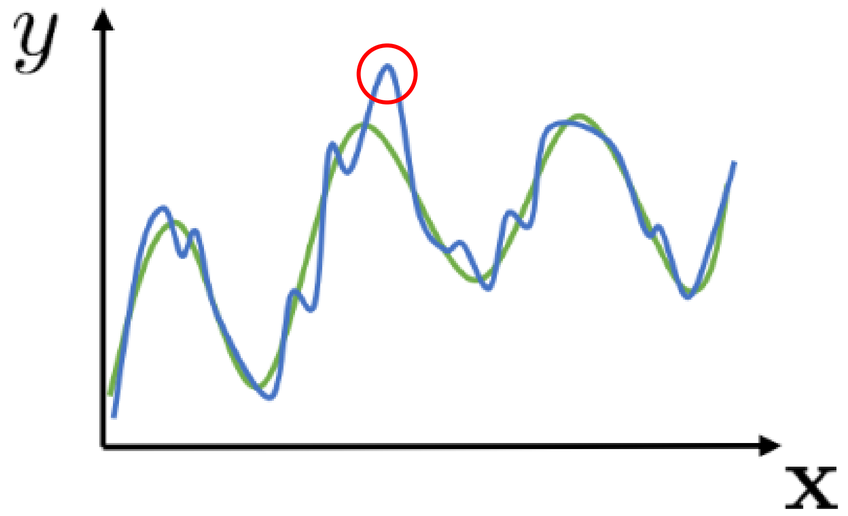
The offline RL problem

Bellman equation: $$Q^\star(\mathbf{s}_t, \mathbf{a}_t) = \mathbb{E}_{\mathbf{s}' \sim p(\cdot|\mathbf{s},\mathbf{a})}\left[r(\mathbf{s},\mathbf{a}) + \gamma \max_{\mathbf{a}'} Q^\star(\mathbf{s}', \mathbf{a}')\right]$$

We don't know the value of the actions we haven't taken (counterfactuals)

Q-learning is an adversarial algorithm!

What happens in the online case?



Solutions to the offline RL problem (explicit)

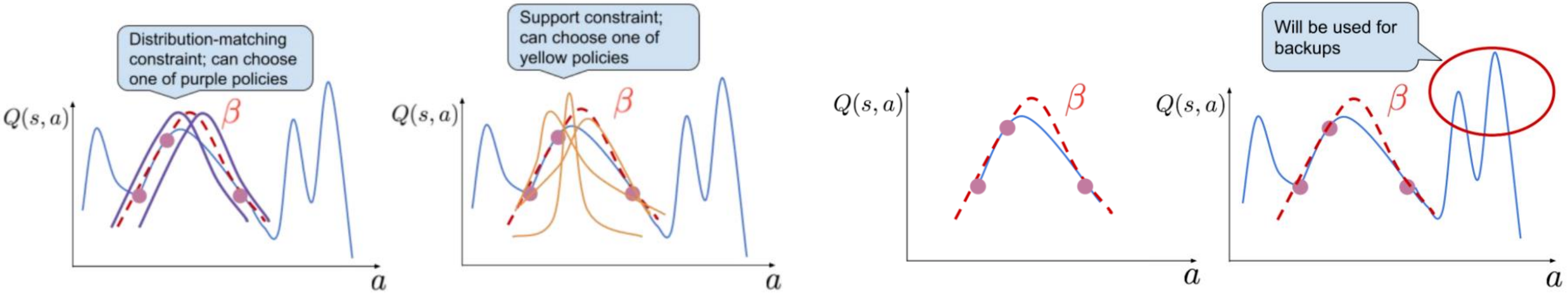
$$\pi_\phi := \arg\max_\phi E_{a\sim\pi_\phi(a|s)}[Q(s,a)] \quad \text{s.t.} \quad D(\pi_\phi(a|s), \pi_\beta(a|s)) \leq \varepsilon$$

We need a constraint

KL-divergence: $D_{KL}(\pi_\theta||\pi_\beta)$

support constraint:

$\pi_\theta(\mathbf{a}|\mathbf{s}) > 0$ only if $\pi_\beta(\mathbf{a}|\mathbf{s}) > \epsilon$



Wu et al. Behavior Regularized Offline RL, 2019

Kumar et al. Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction, 2020

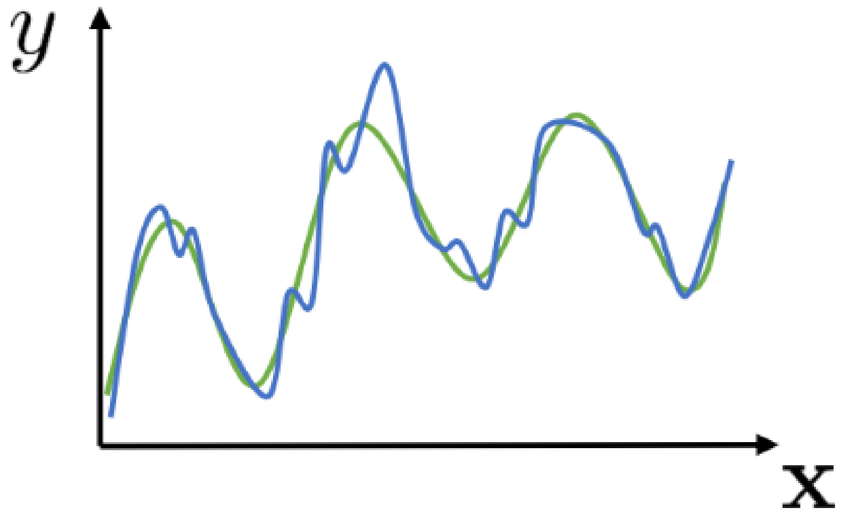
Solutions to the offline RL problem (implicit)

Solve constrained optimization via duality

$$\pi_\phi := \arg\max_\phi E_{a\sim\pi_\phi(a|s)}[Q(s,a)] \quad \text{s.t.} \quad D(\pi_\phi(a|s), \pi_\beta(a|s)) \leq \varepsilon$$

Peters et al, REPS

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



weight (s, a)

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

Peng*, Kumar* et al. Advantage-Weighted Regression, 2019

Nair et al. Accelerating Online RL with Offline Datasets, 2020

Solutions to the offline RL problem (implicit)

Conservative Q-Learning (CQL)

$$\hat{Q}^{\pi}_{\mathrm{CQL}}(s,a) \leq Q(s,a) \quad \forall s \in \mathcal{D}, a$$

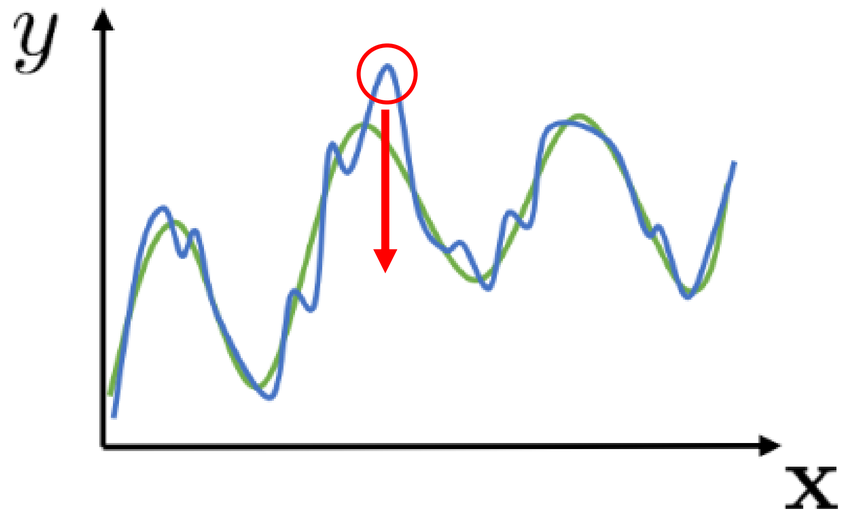
push down big Q values

$$\hat{Q}^{\pi}_{\mathrm{CQL}} := \min_{Q} \max_{\mu} \mathbb{E}_{a\sim\mu(a|s)}[Q(s,a)] + \frac{1}{2\alpha}\mathbb{E}_{s,a,s'\sim\mathcal{D}}\left[(Q(s,a) - (r(s,a) + \gamma\mathbb{E}_{a\sim\pi_\phi(a|s)}[\bar{Q}(s',a')]))^2\right]$$

push down big Q values

push up data samples

$$\hat{Q}^{\pi}_{\mathrm{CQL}} := \min_{Q} \max_{\mu} \mathbb{E}_{a\sim\mu(a|s)}[Q(s,a)] - \mathbb{E}_{a\sim\mathcal{D}(a|s)}[Q(s,a)] + \frac{1}{2\alpha}\mathbb{E}_{s,a,s'\sim\mathcal{D}}\left[(Q(s,a) - (r(s,a) + \gamma\mathbb{E}_{a\sim\pi_\phi(a|s)}[\bar{Q}(s',a')]))^2\right]$$



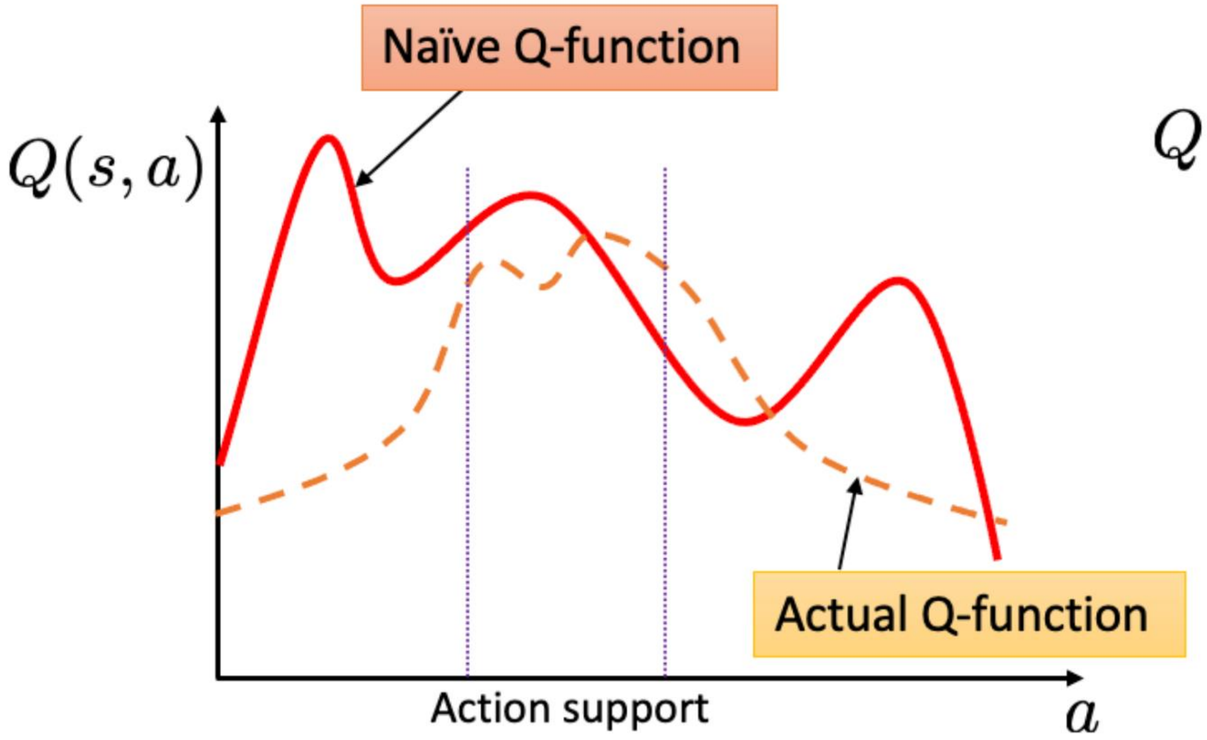
Kumar et al. Conservative Q-Learning for Offline RL, 2020

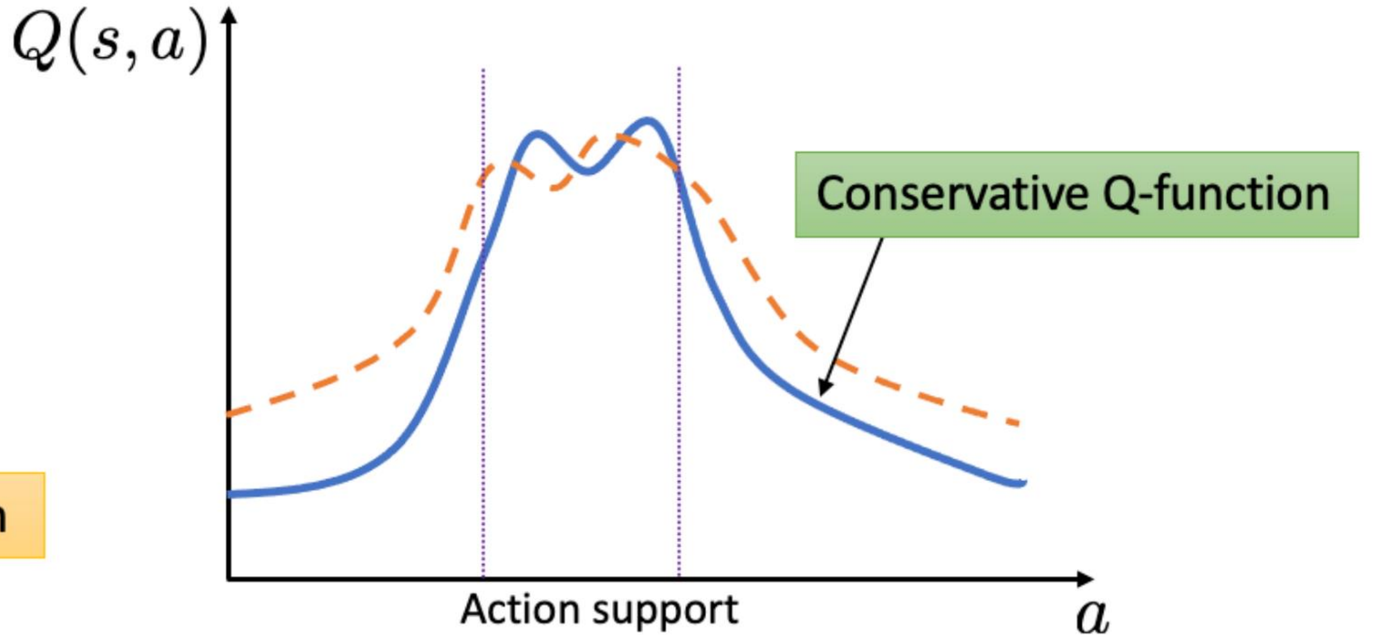
Solutions to the offline RL problem (implicit)

Conservative Q-Learning (CQL)

$$\hat{Q}^{\pi}_{\mathrm{CQL}} := \min_{Q} \max_{\mu} \mathbb{E}_{a\sim\mu(a|s)}[Q(s,a)] - \mathbb{E}_{a\sim\mathcal{D}(a|s)}[Q(s,a)]$$

$$+\frac{1}{2\alpha}\mathbb{E}_{s,a,s'\sim\mathcal{D}}\left[(Q(s,a) - (r(s,a) + \gamma\mathbb{E}_{a\sim\pi_{\phi}(a|s)}[\bar{Q}(s',a')]))^2\right]$$





Kumar et al. Conservative Q-Learning for Offline RL, 2020

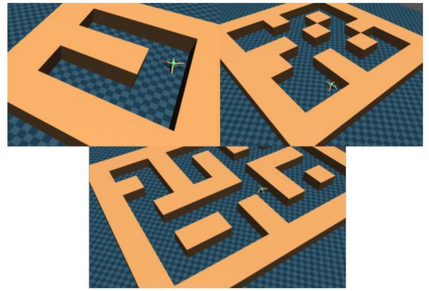
Solutions to the offline RL problem (implicit)

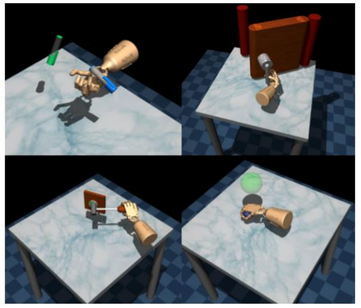
Conservative Q-Learning (CQL)

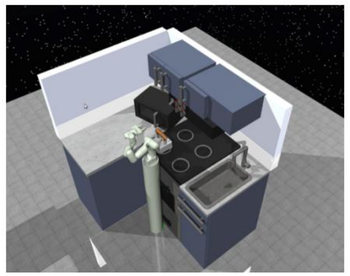
$$\hat{Q}^{\pi}_{\text{CQL}} := \min_{Q} \max_{\mu} \mathbb{E}_{a\sim\mu(a|s)}[Q(s,a)] - \mathbb{E}_{a\sim\mathcal{D}(a|s)}[Q(s,a)]$$

$$+\frac{1}{2\alpha}\mathbb{E}_{s,a,s'\sim\mathcal{D}}\left[(Q(s,a) - (r(s,a) + \gamma\mathbb{E}_{a\sim\pi_\phi(a|s)}[\bar{Q}(s',a')]))^2\right]$$

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







Domain	Task Name	BC	SAC	BEAR	BRAC-p	BRAC-v	CQL($\mathcal{H}$)	CQL(ρ)
AntMaze	antmaze-umaze	65.0	0.0	**73.0**	50.0	70.0	**74.0**	**73.5**
	antmaze-umaze-diverse	55.0	0.0	61.0	40.0	70.0	**84.0**	61.0
	antmaze-medium-play	0.0	0.0	0.0	0.0	0.0	**61.2**	4.6
	antmaze-medium-diverse	0.0	0.0	8.0	0.0	0.0	**53.7**	5.1
	antmaze-large-play	0.0	0.0	0.0	0.0	0.0	**15.8**	3.2
	antmaze-large-diverse	0.0	0.0	0.0	0.0	0.0	**14.9**	2.3
Adroit	pen-human	34.4	6.3	-1.0	8.1	0.6	37.5	**55.8**
	hammer-human	1.5	0.5	0.3	0.3	0.2	**4.4**	2.1
	door-human	0.5	3.9	-0.3	-0.3	-0.3	**9.9**	**9.1**
	relocate-human	0.0	0.0	-0.3	-0.3	-0.3	0.20	**0.35**
	pen-cloned	**56.9**	23.5	26.5	1.6	-2.5	39.2	40.3
	hammer-cloned	0.8	0.2	0.3	0.3	0.3	2.1	**5.7**
	door-cloned	-0.1	0.0	-0.1	-0.1	-0.1	0.4	**3.5**
	relocate-cloned	**-0.1**	-0.2	-0.3	-0.3	-0.3	**-0.1**	**-0.1**
Kitchen	kitchen-complete	33.8	15.0	0.0	0.0	0.0	**43.8**	31.3
	kitchen-partial	33.8	0.0	13.1	0.0	0.0	**49.8**	**50.1**
	kitchen-undirected	47.5	2.5	47.2	0.0	0.0	**51.0**	**52.4**

Kumar et al. Conservative Q-Learning for Offline RL, 2020

The Plan

Offline RL problem formulation

Offline RL solutions

Offline multi-task RL and data sharing

Offline goal-conditioned RL

Multi-task RL algorithms

Policy: $\pi_\theta(\mathbf{a}|\overline{\mathbf{s}}) \longrightarrow \pi_\theta(\mathbf{a}|\overline{\mathbf{s}}, \mathbf{z}_i)$

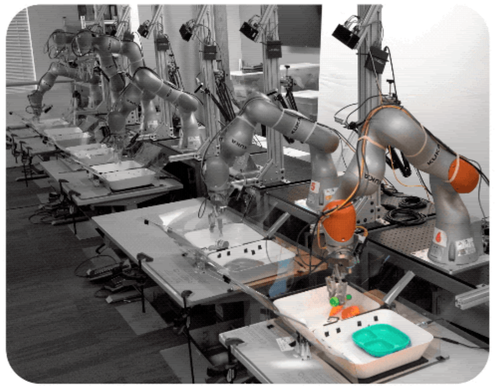
Q-function: $Q_\phi(\overline{\mathbf{s}}, \mathbf{a}) \longrightarrow Q_\phi(\overline{\mathbf{s}}, \mathbf{a}, \mathbf{z}_i)$

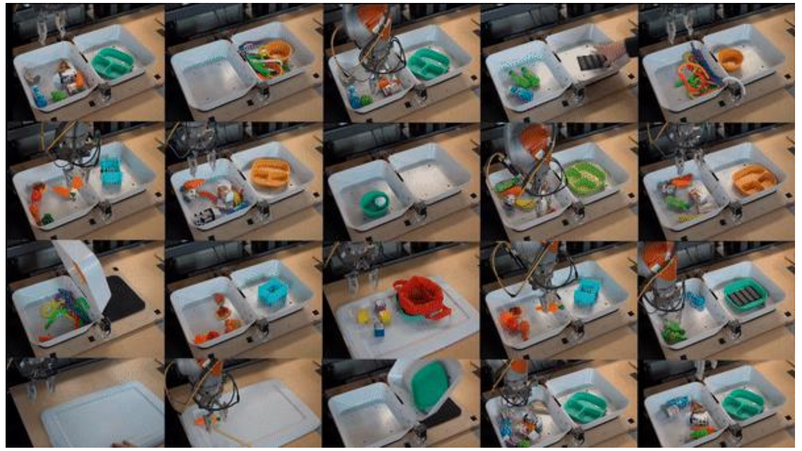
What is different about reinforcement learning?

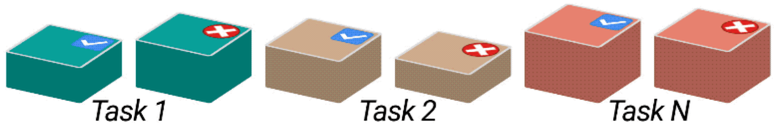
The data distribution is
controlled by the agent!

Should we share data in addition to sharing weights?

Example of multi-task Q-learning applied to robotics: MT-Opt



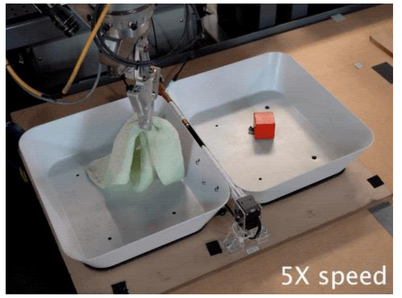




80% avg improvement over baselines across all the ablation tasks (4x improvement over single-task)

~4x avg improvement for tasks with little data

Fine-tunes to a new task (to 92% success) in 1 day



Kalashnikov et al. MT-Opt. CoRL '21

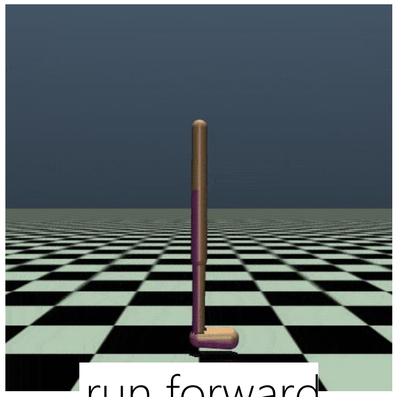
Can we share data across distinct tasks?

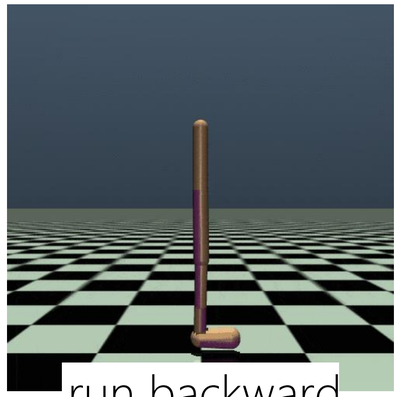
$\mathcal{D} = \{(\mathbf{s}_i, \mathbf{a}_i, \mathbf{s'}_i, r_i)\}$

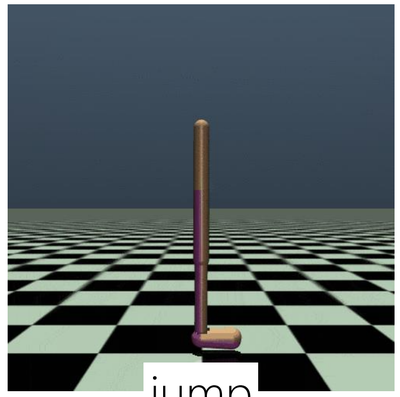
$\mathbf{s} \sim d^{\pi_\beta}(\mathbf{s})$

$\mathbf{a} \sim \pi_\beta(\mathbf{a}|\mathbf{s})$ ← unknown!

$\mathbf{s}' \sim p(\mathbf{s}'|\mathbf{a}, \mathbf{s})$


run forward


run backward


jump

Dataset types / Tasks	Dataset Size	Avg Return		$D_{\mathrm{KL}}(\pi, \pi_\beta)$	
		No Sharing	**Sharing All**	**No Sharing**	**Sharing All**
medium-replay / `run forward`	109900	**998.9**	966.2	**3.70**	10.39
medium-replay / `run backward`	109980	**1298.6**	1147.5	**4.55**	12.70
medium-replay / `jump`	109511	**1603.1**	1224.7	**3.57**	15.89
average task performance	N/A	**1300.2**	1112.8	**3.94**	12.99
medium / `run forward`	27646	297.4	**848.7**	**6.53**	11.78
medium / `run backward`	31298	207.5	**600.4**	**4.44**	10.13
medium / `jump`	100000	351.1	**776.1**	**5.57**	21.27
average task performance	N/A	285.3	**747.7**	**5.51**	14.39
medium-replay / `run forward`	109900	590.1	**701.4**	**1.49**	7.76
medium / `run backward`	31298	614.7	**756.7**	**1.91**	12.2
expert / `jump`	5000	**1575.2**	885.1	**3.12**	27.5
average task performance	N/A	**926.6**	781	**2.17**	15.82

- Sharing data generally helps
- It can **hurt performance** in some cases
- Can we characterize **why** it hurts performance?

Sharing data exacerbates distribution shift

Yu*, Kumar*, Chebotar, Hausman, Levine, Finn. *Multi-Task Offline Reinforcement Learning with Conservative Data Sharing*, 2021

Sharing data while reducing distributional shift - Conservative Data Sharing

We assume that relabeling data $\mathcal{D}_j$ from task j to task i generates a dataset $\mathcal{D}_{j \to i}$, which is then additionally used to train on task i. Thus, the effective dataset for task i after relabeling is given by: $\mathcal{D}_i^{\text{eff}} := \mathcal{D}_i \cup (\cup_{j \neq i} \mathcal{D}_{j \to i})$.

That means that we can control the dataset/behavior policy itself!

Yu*, Kumar*, Chebotar, Hausman, Levine, Finn. *Multi-Task Offline Reinforcement Learning with Conservative Data Sharing*, 2021

Can we automatically identify how to share data?

Standard offline RL:

$$\pi^*(\mathbf{a}|\mathbf{s}) = \arg\max_{\pi} J_{\mathcal{D}}(\pi) - \alpha\mathcal{D}(\pi, \pi_\beta)$$

Maximize reward

Regularize towards the data (behavior policy π_β)

Can we optimize the data distribution?

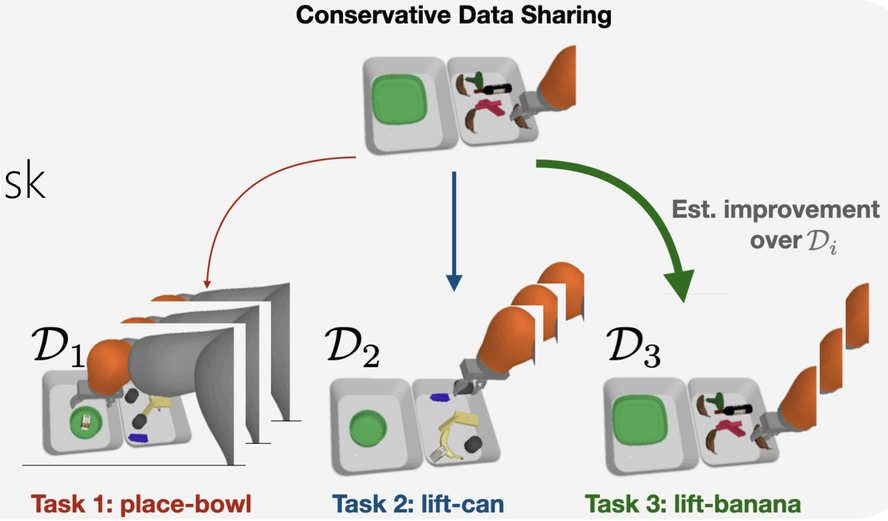
$$\pi^*(\cdot|\cdot, i) := \arg\max_{\pi} \max_{\pi_\beta^{\text{eff}} \in \Pi_{\text{relabel}}} \left[J_{\mathcal{D}_\text{i}^{\text{eff}}}(\pi) - \alpha\mathcal{D}(\pi, \pi_\beta^{\text{eff}}; i) \right]$$

Optimize for the effective behavior policy to **maximize reward** and **minimizes distribution shift**

Conservative data sharing (CDS)

Share data $(\mathbf{s}, \mathbf{a})$ when conservative Q-value will increase for that task

$$\hat{Q}^\pi(\mathbf{s}, \mathbf{a}, i) - E_{\mathbf{s}', \mathbf{a}' \sim \mathcal{D}_\text{i}} \left[\hat{Q}^\pi(\mathbf{s}', \mathbf{a}', i) \right] \geq 0$$



Does CDS prevent excessive distributional shift?

Environment	Dataset types / Tasks	$D_{\mathrm{KL}}(\pi, \pi_\beta)$ No Sharing	Sharing All	CDS (ours)
walker2d	medium-replay / run forward	**1.49**	7.76	**1.49**
	medium / run backward	**1.91**	12.2	6.09
	expert / jump	3.12	27.5	**2.91**

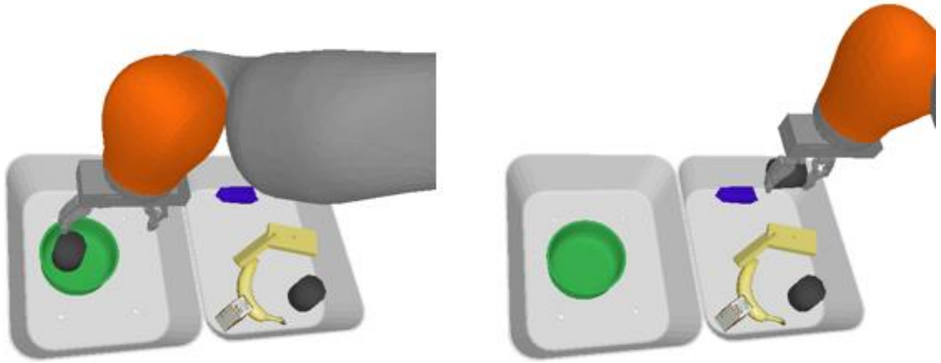
CDS reduces the KL divergence between the **data distribution** and the **learned policy**

This translates to improved performance

Environment	Tasks / Dataset type	CDS (ours)	Sharing All	No Sharing
walker2d	run forward / medium-replay	**1100.7**	701.4	590.1
	run backward / medium	638.4	**756.7**	614.7
	jump / expert	1538.4	885.1	1575.2
	average	**1092.5**	781	926.6

Yu*, Kumar*, Chebotar, Hausman, Levine, Finn. *Multi-Task Offline Reinforcement Learning with Conservative Data Sharing*, 2021

Experiments on vision-based robotic manipulation



simulated object manipulation tasks

Comparisons:

HIPI: GCRL sharing strategy based on highest return
Skill: *domain-specific* approach based on human intuition

Task Name	CDS (ours)	HIPI [15]	Skill [32]	Sharing All	No Sharing
`lift-banana`	**54.0%**	39.7%	33.6%	45.6%	12.6%
`lift-bottle`	**76.3%**	58.5%	53.3%	42.8%	44.5%
`lift-sausage`	**75.9%**	65.6%	62.5%	73.8%	55.2%
`lift-milk`	**82.7%**	75.3%	62.8%	68.9%	58.9%
`lift-food`	**70.3%**	64.6%	23.1%	64.9%	29.5%
`lift-can`	**76.1%**	70.8%	37.6%	49.4%	41.8%
`lift-carrot`	**80.4%**	70.1%	69.4%	72.2%	63.1%
`place-bowl`	84.4%	72.0%	**84.5%**	64.7%	77.0%
`place-plate`	**86.9%**	82.2%	79.5%	75.1%	82.2%
`place-divider-plate`	**89.4%**	72.6%	81.0%	79.3%	84.7%
average	**77.6%**	67.2%	58.7%	63.7%	55.0%

Yu*, Kumar*, Chebotar, Hausman, Levine, Finn. *Multi-Task Offline Reinforcement Learning with Conservative Data Sharing*, 2021

The Plan

Offline RL problem formulation

Offline RL solutions

Offline multi-task RL and data sharing

Offline goal-conditioned RL

Goal-conditioned RL with hindsight relabeling

1. Collect data $\mathcal{D}_k = \{(\mathbf{s}_{1:T}, \mathbf{a}_{1:T}, \mathbf{s}_g, r_{1:T})\}$ using some policy
2. Store data in replay buffer $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_k$
3. Perform hindsight relabeling:
   a. Relabel experience in $\mathcal{D}_k$ using last state as goal: $\mathcal{D}'_k = \{(\mathbf{s}_{1:T}, \mathbf{a}_{1:T}, \mathbf{s}_T, r'_{1:T}\}$ where $r'_t = -d(\mathbf{s}_t, \mathbf{s}_T)$
   b. Store relabeled data in replay buffer $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}'_k$
4. Update policy using replay buffer $\mathcal{D}$

$k++$

g strategies?

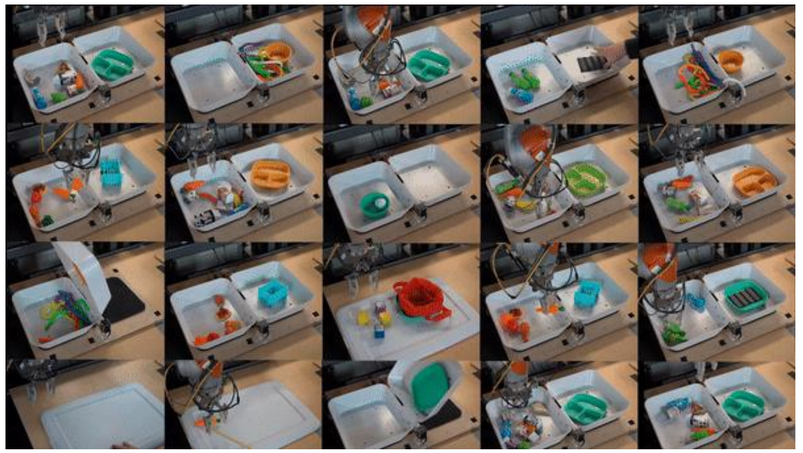
e trajectory

What if we do it fully offline?

Kaelbling. Learning to Achieve Goals. IJCAI '93
Andrychowicz et al. Hindsight Experience Replay. NeurIPS '17

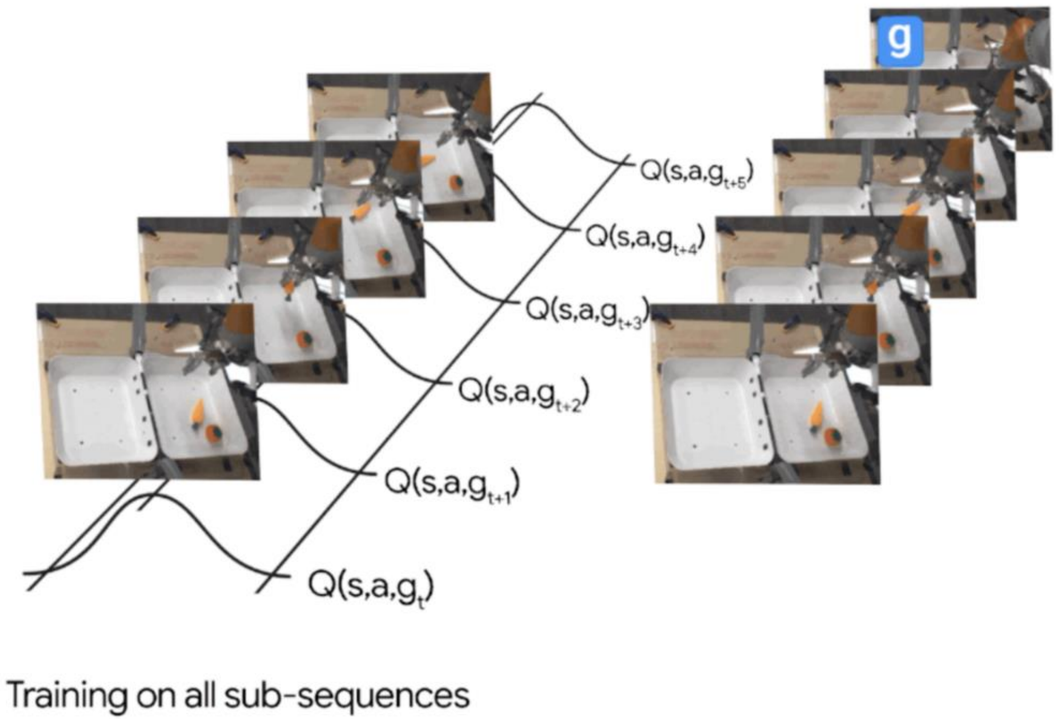
Actionable Models: Moving Beyond Tasks

- At scale, task definitions become a bottleneck
- **Goal** state is a task!
  Rewards through **hindsight relabeling**
- **Conservative Q-learning** to create artificial negative examples



Actionable Models, Chebotar, Hausman, Lu, Xiao, Kalashnikov, Varley, Irpan, Eysenbach, Julian, Finn, Levine, ICML 2021

Actionable Models: Moving Beyond Tasks



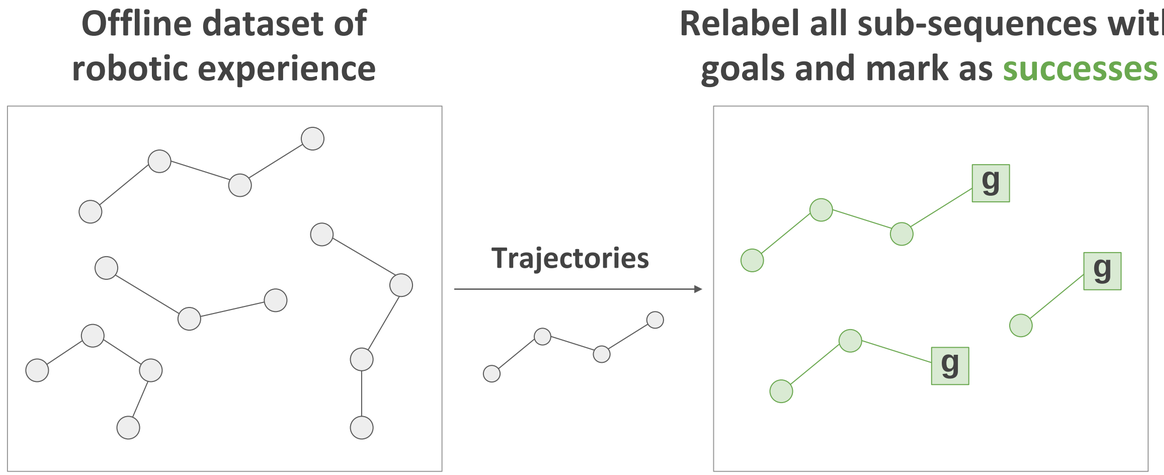
Training on all sub-sequences

- At scale, task definitions become a bottleneck
- **Goal** state is a task!
  Rewards through **hindsight relabeling**
- **Conservative Q-learning** to create artificial negative examples

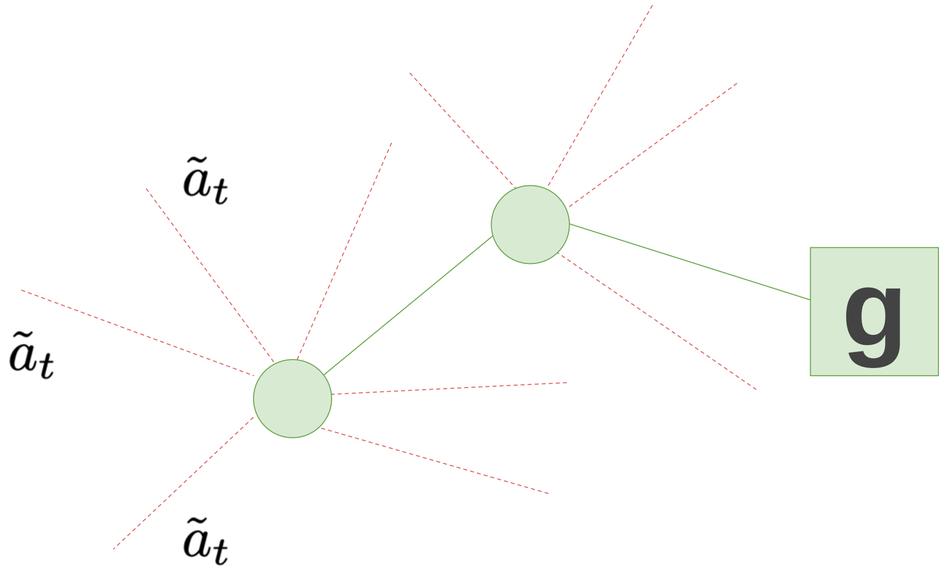
- **Functional understanding** of the world:
  a world model that also provides
  an **actionable policy**
- **Unsupervised** objective for Robotics?
  - Zero-shot visual tasks
  - Downstream fine-tuning

Actionable Models, Chebotar, Hausman, Lu, Xiao, Kalashnikov, Varley, Irpan, Eysenbach, Julian, Finn, Levine, ICML 2021

Actionable Models: Hindsight Relabeling

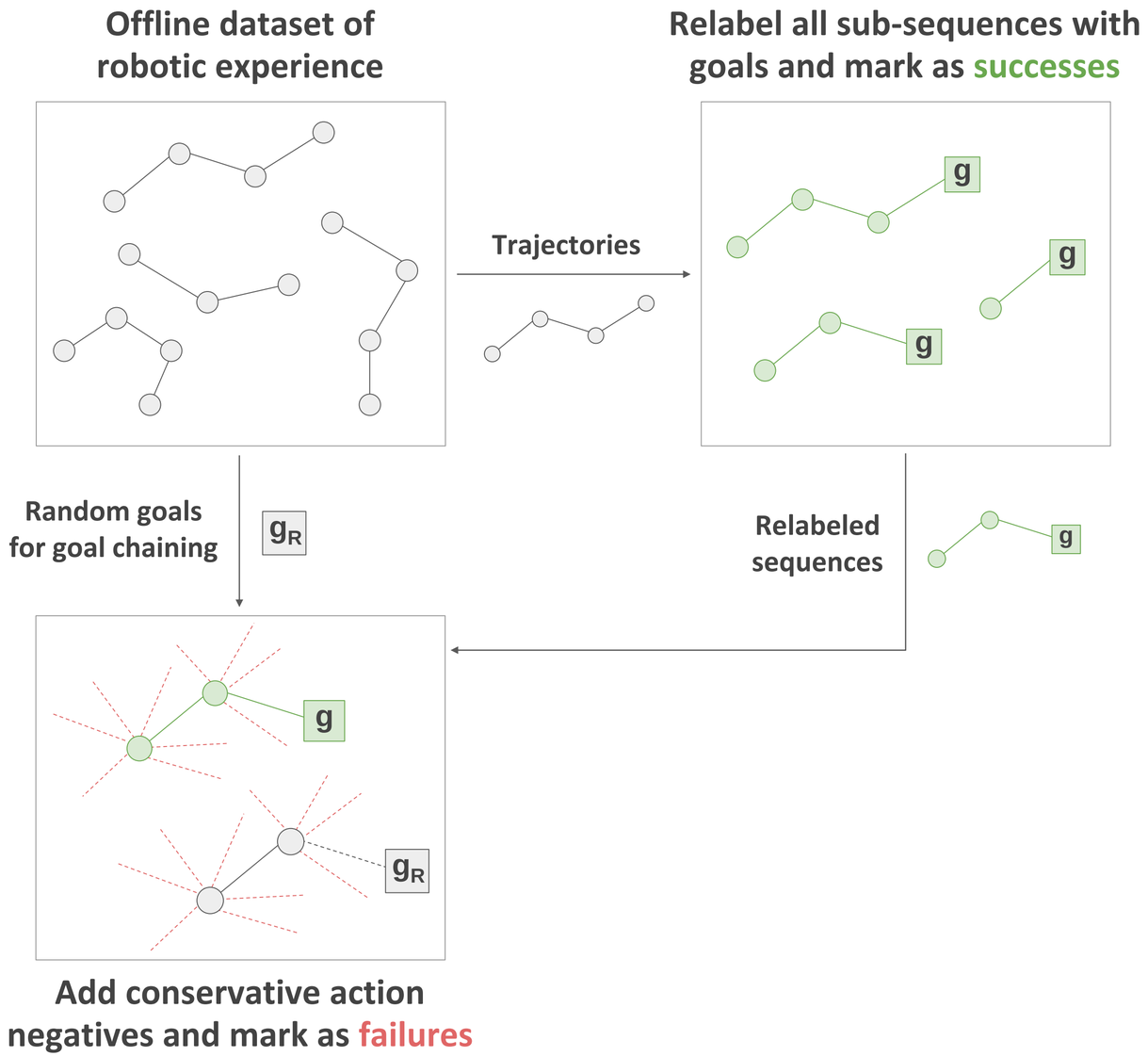


Actionable Models: Artificial Negatives

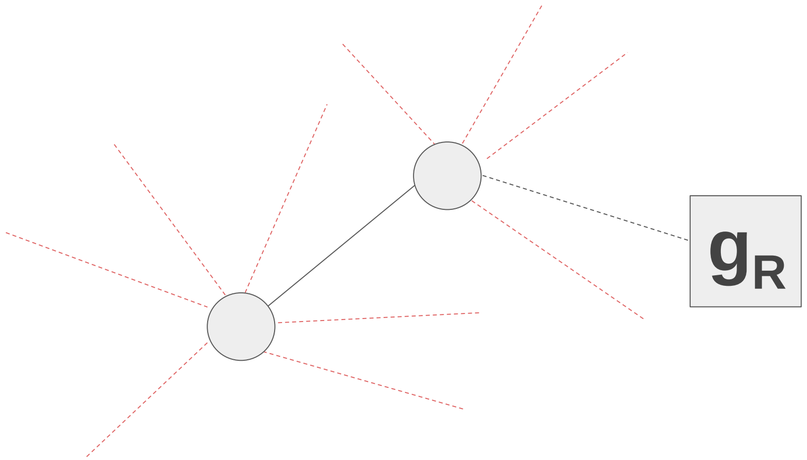


- **Offline hindsight relabeling:**
  only positive examples → need negatives
- **Conservative strategy**:
  minimize Q-values of unseen actions
- Sample **contrastive** artificial negative actions: $\tilde{a}_t \sim \exp(Q^\pi(s_t, \tilde{a}_t, g))$

Actionable Models: Goal Chaining

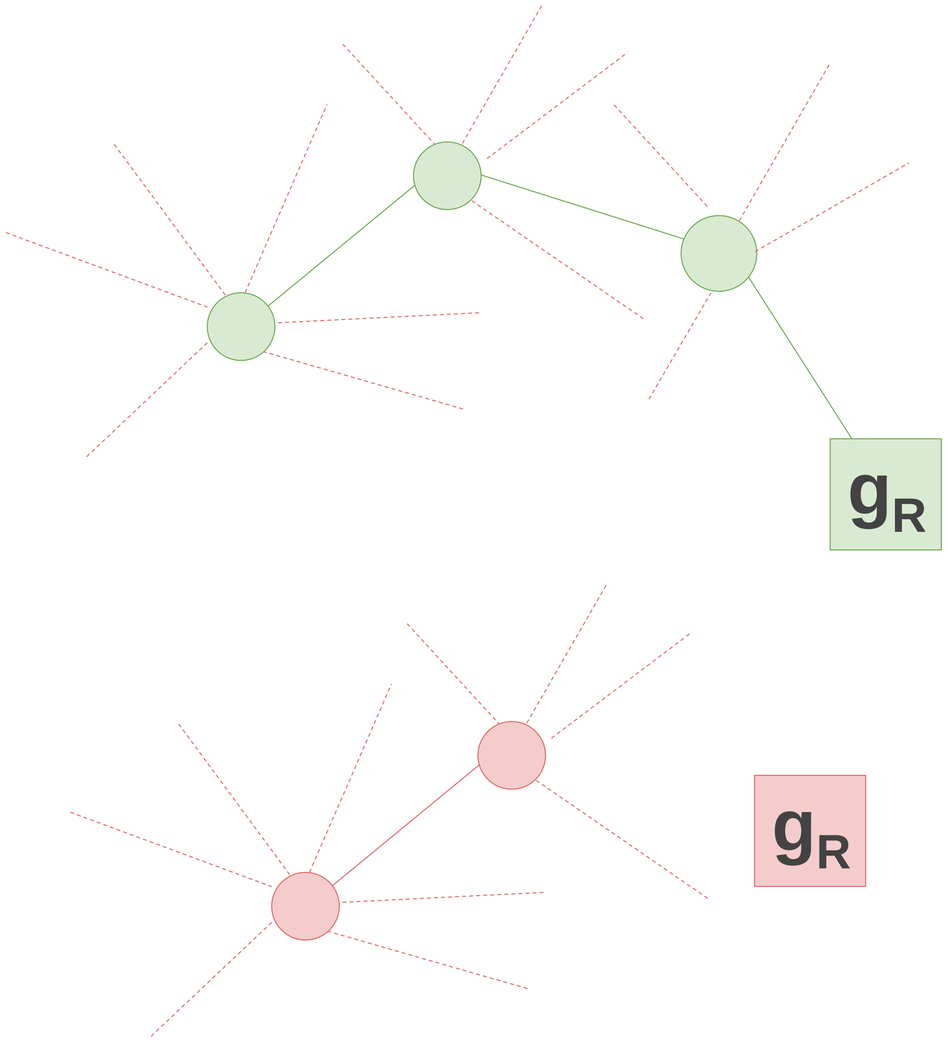


Actionable Models: Goal Chaining



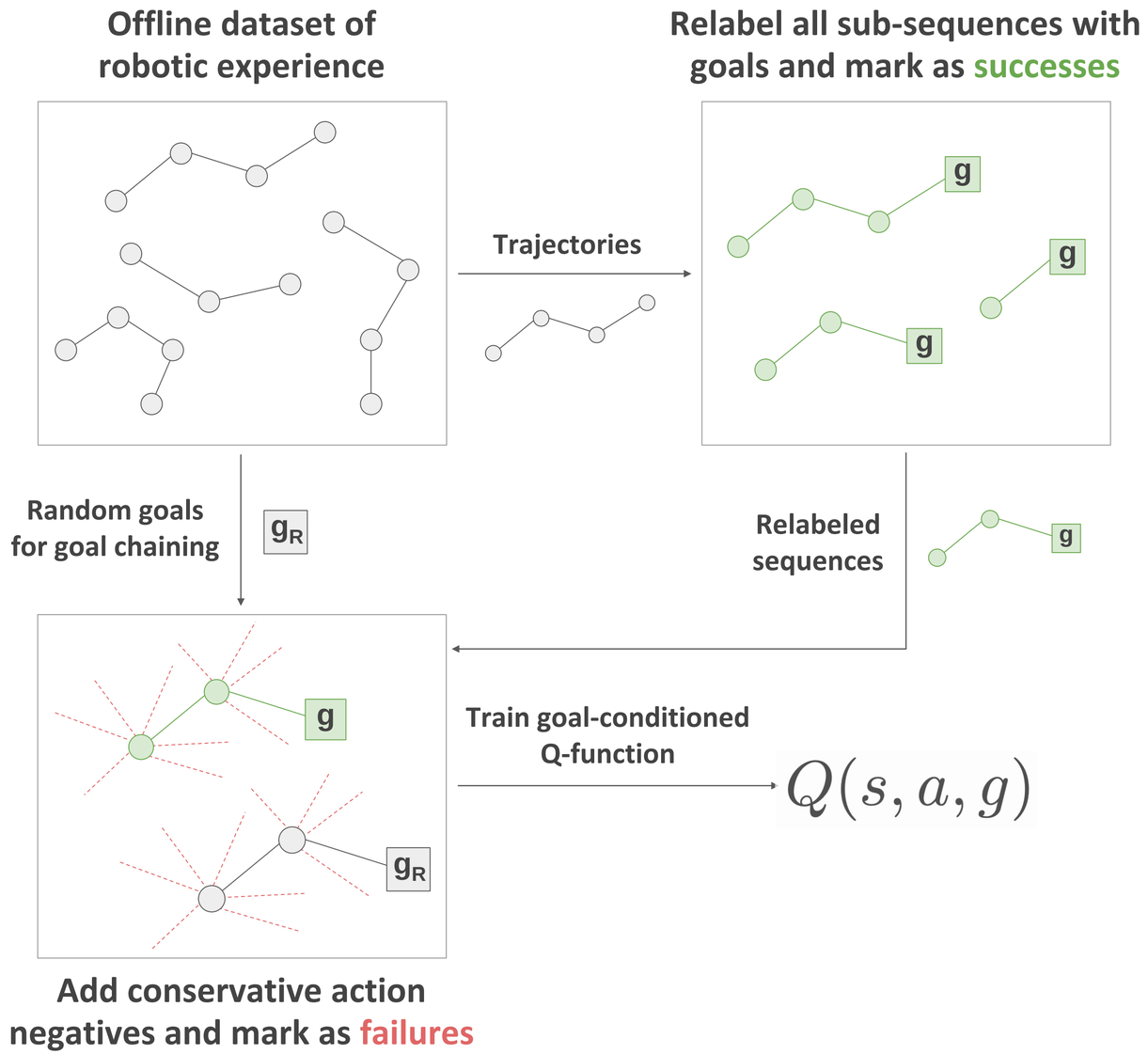
- Recondition on random goals to enable **chaining** goals **across episodes**

Actionable Models: Goal Chaining

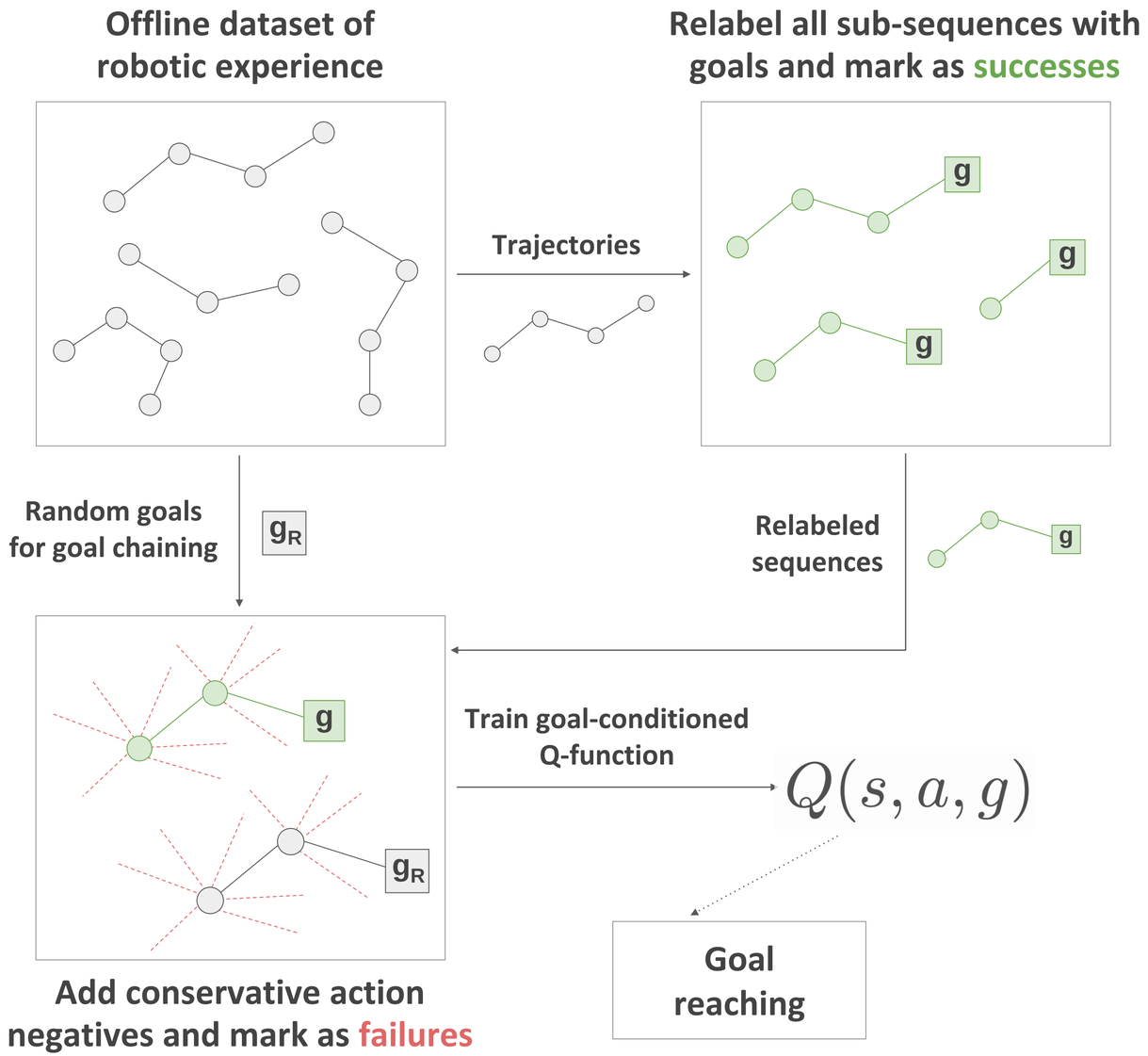


- Recondition on random goals to enable **chaining** goals **across episodes**
- If **pathway to a goal** exists:
  dynamic programming will propagate reward
- **No pathway to the goal**:
  conservative strategy will minimize Q-values

Actionable Models



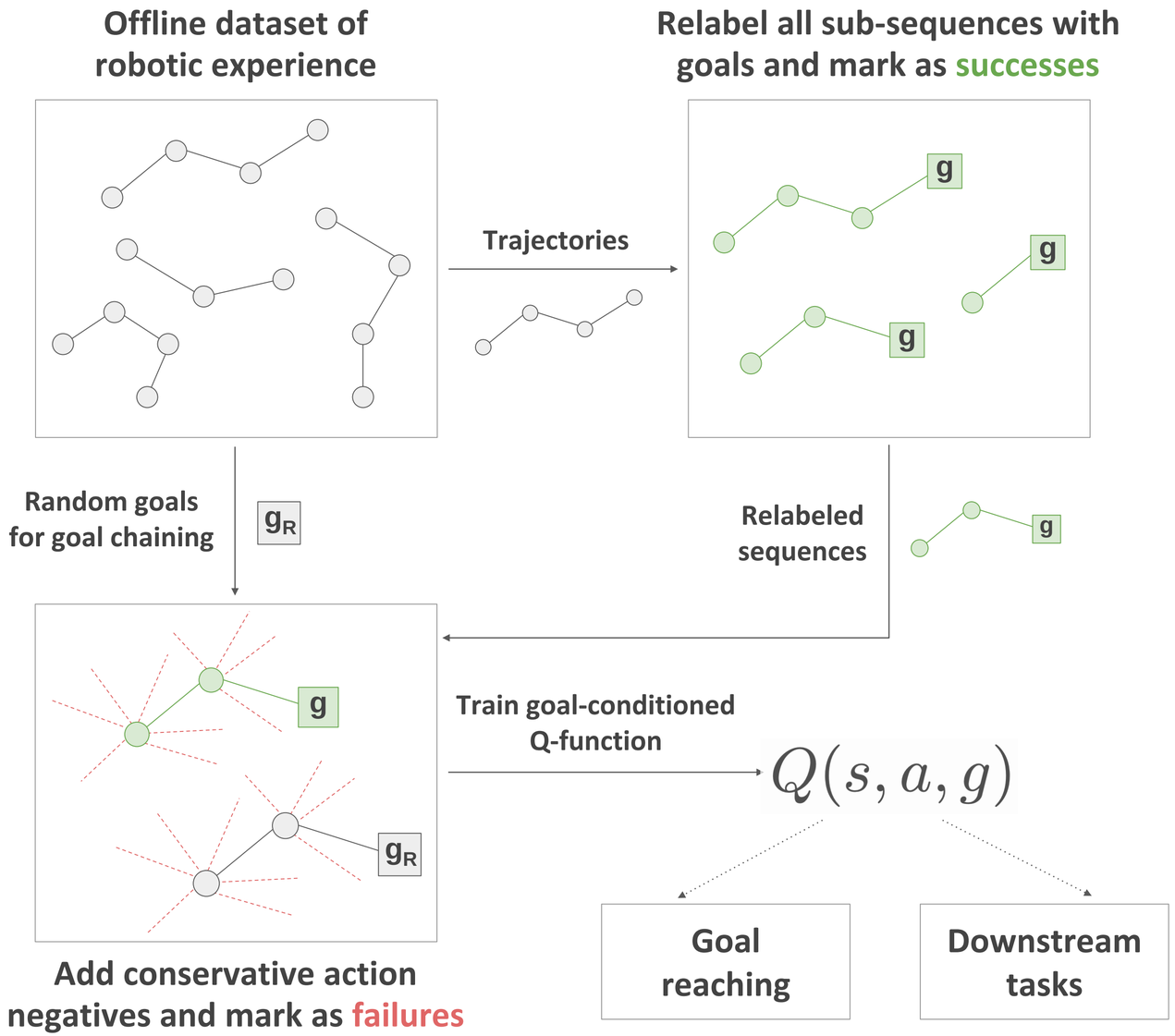
Actionable Models



Actionable Models: Real world goal reaching

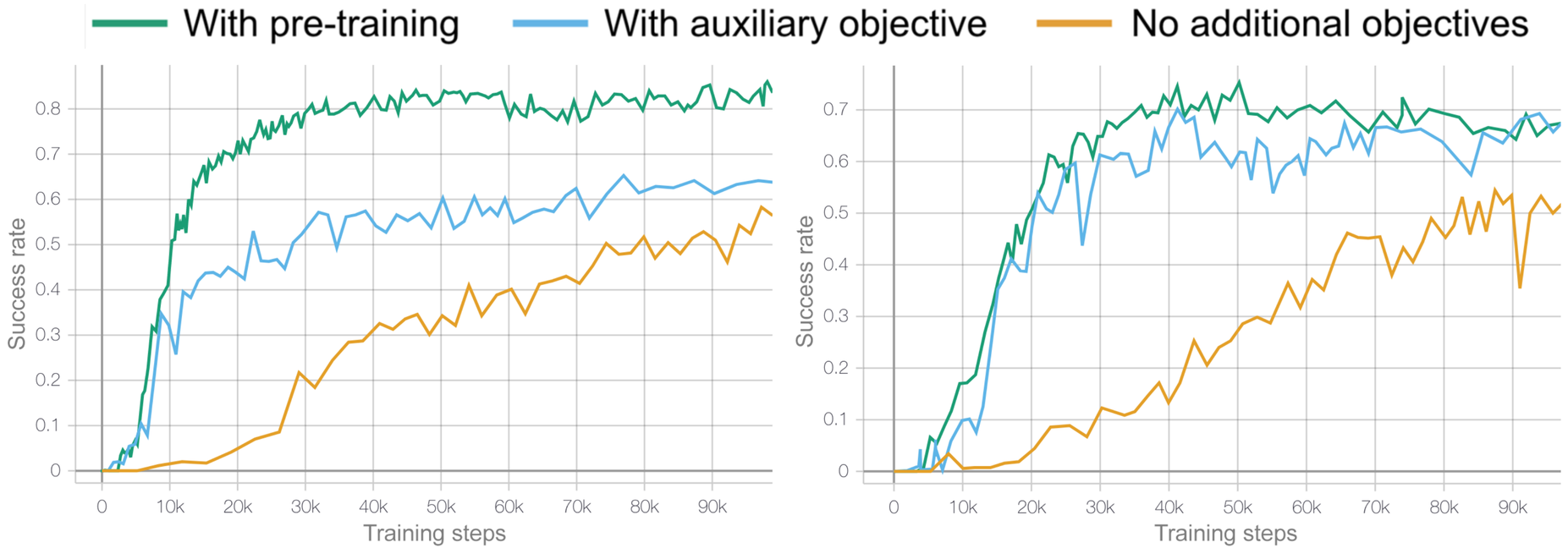
Task	Success rate
Instance grasping	92%
Rearrangement	74%
Container placing	66%

Actionable Models



Actionable Models: Downstream tasks

Simulated ablations



Real-world fine-tuning with a small amount of data

Task	No pre-training	With pre-training
Grasp box	0%	27%
Grasp banana	4%	20%
Grasp milk	1%	20%

The Plan

Offline RL problem formulation

Offline RL solutions

Offline multi-task RL and data sharing

Offline goal-conditioned RL

Next time

What about long-horizon tasks?

Hierarchical RL

Skill discovery

Reminder

Homework 4 (optional) is due next Monday!