# Efficient Reinforcement Learning via Representation Learning

**Wen Sun**

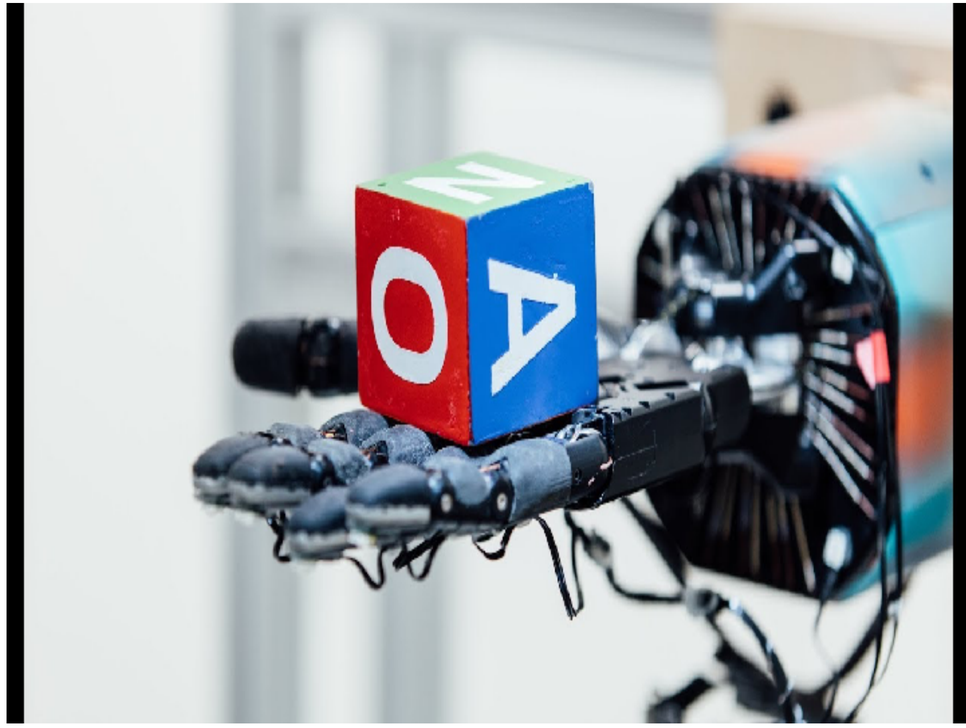
Joint work with Masatoshi Uehara (Cornell) & Xuezhou Zhang (Princeton)


Cornell University
Department of Computer Science

# Empirical RL for large-scale problems


[AlphaGo, Silver et.al, 15]


[OpenAI Five, 18]



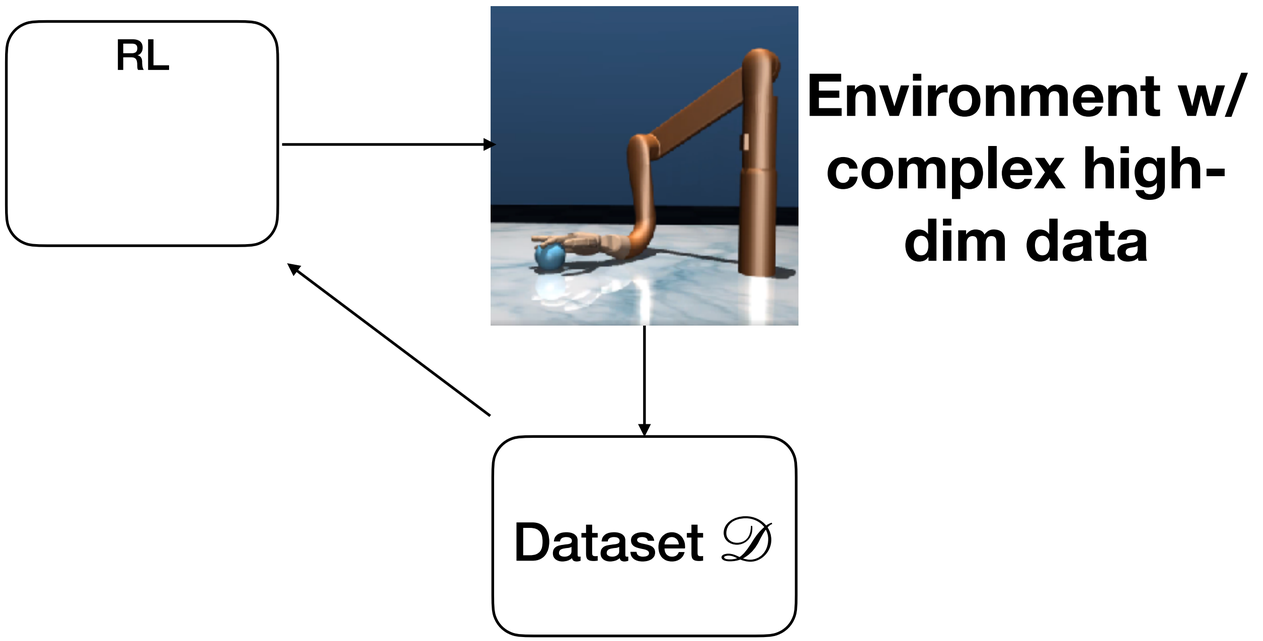
[OpenAI,19]

Rich (nonlinear) function approximation + RL can work well w/ enough samples

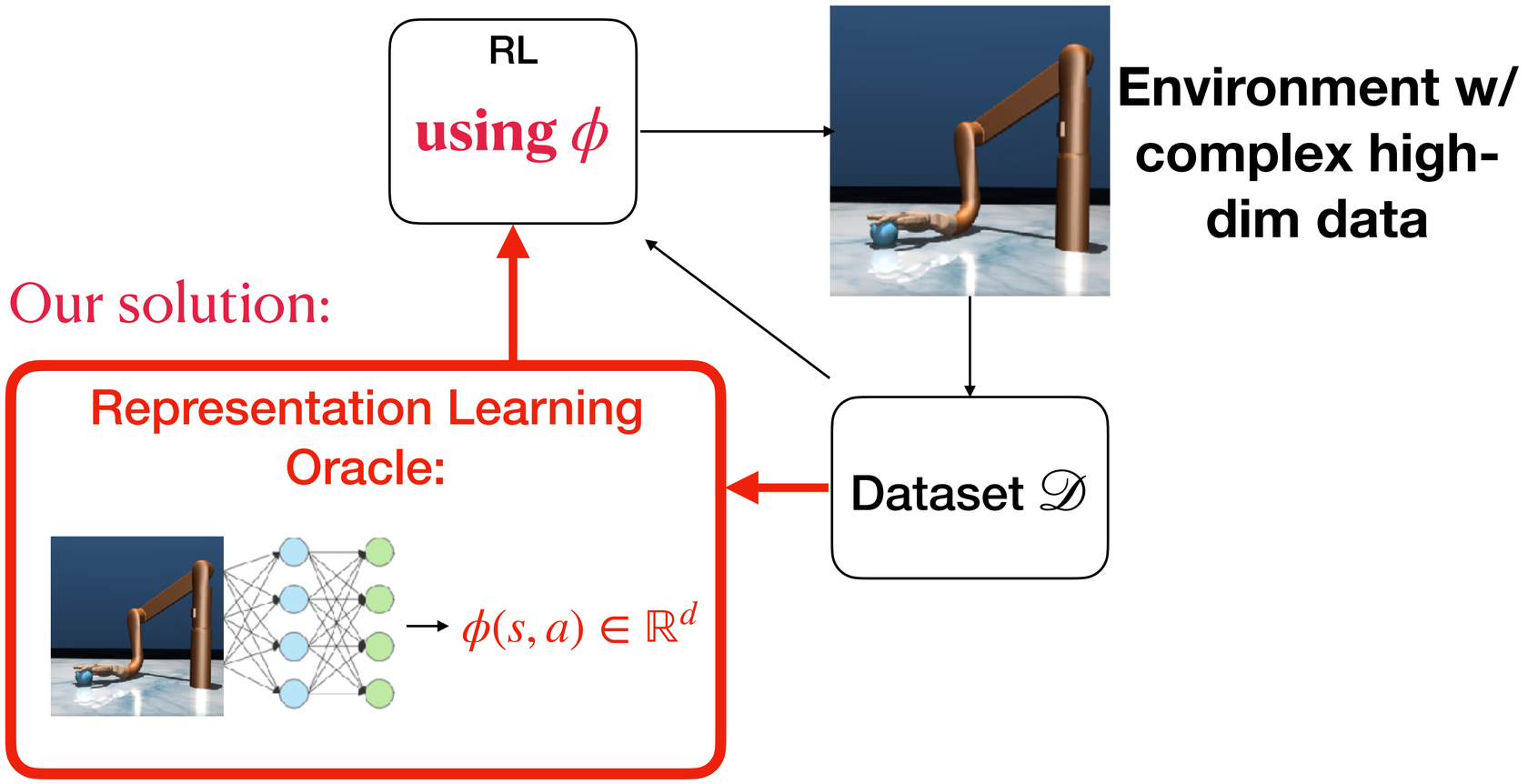
# Can we design provably efficient algorithms for

## *Rich Function Approx + RL ?*

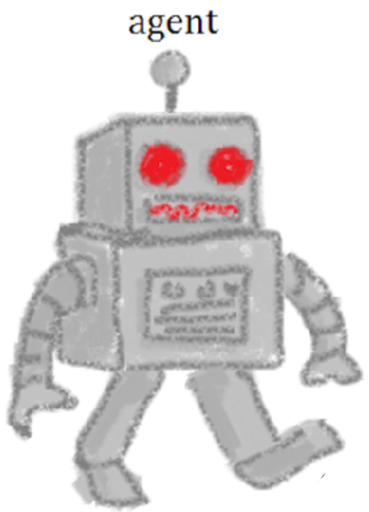
# Can we design provably efficient algorithms for *Rich Function Approx + RL ?*



# Can we design provably efficient algorithms for *Rich Function Approx + RL ?*

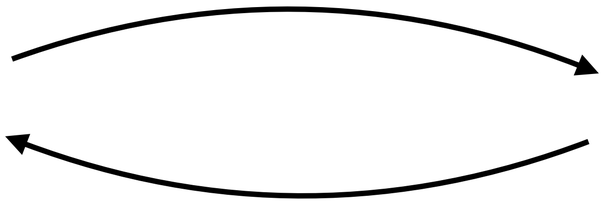


# Episodic Infinite Horizon Discounted MDPs




Policy: state to action

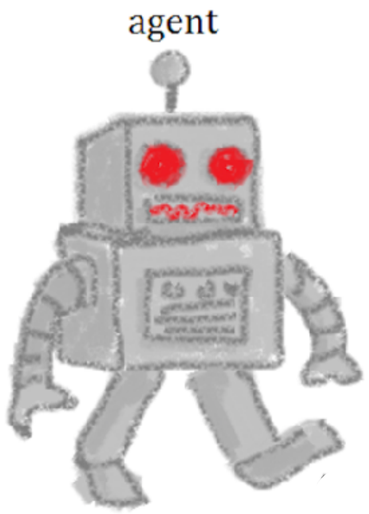
$$\pi(s) \to a$$



Reward & Next State

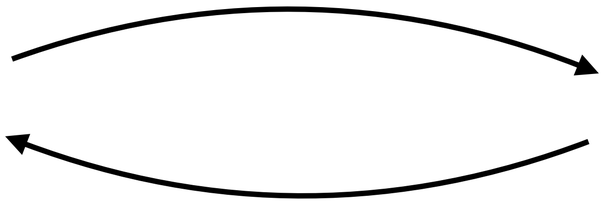
$$r(s,a), s' \sim P(\cdot \mid s,a)$$

# Episodic Infinite Horizon Discounted MDPs




Policy: state to action

$$\pi(s) \to a$$



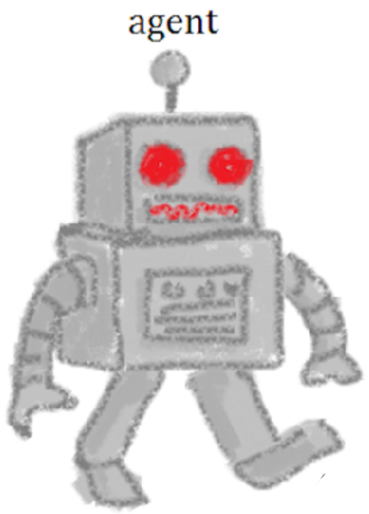
Reward & Next State

$$r(s,a), s' \sim P(\cdot \mid s,a)$$

Objective:

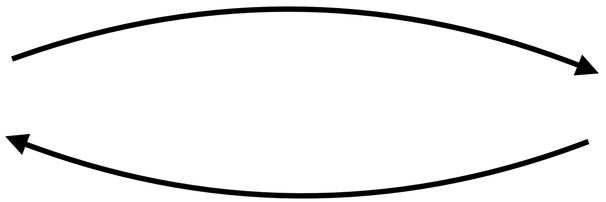
$$\max_{\pi} J(\pi; P, r), \text{ where } J(\pi; P, r) := \mathbb{E}\left[r(s_0,a_0) + \gamma r(s_1,a_1) + \gamma^2 r(s_2,a_2) + \ldots \mid a \sim \pi, P\right]$$

# Episodic Infinite Horizon Discounted MDPs




Policy: state to action

$$\pi(s) \to a$$



Reward & Next State

$$r(s,a), s' \sim P(\,\cdot\,|\,s,a)$$

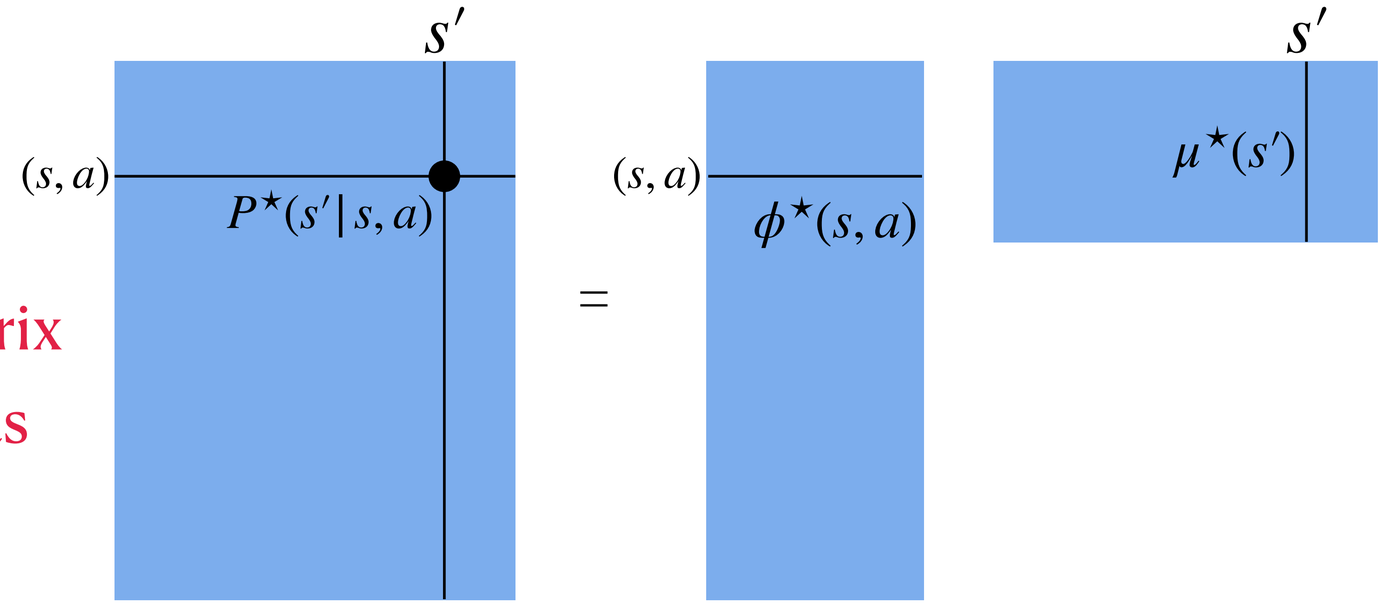
Objective:

$$\max_{\pi} J(\pi; P, r), \text{ where } J(\pi; P, r) := \mathbb{E}\left[r(s_0,a_0) + \gamma r(s_1,a_1) + \gamma^2 r(s_2,a_2) + \ldots \,|\, a \sim \pi, P\right]$$

Assume fixed initial state $s_0$

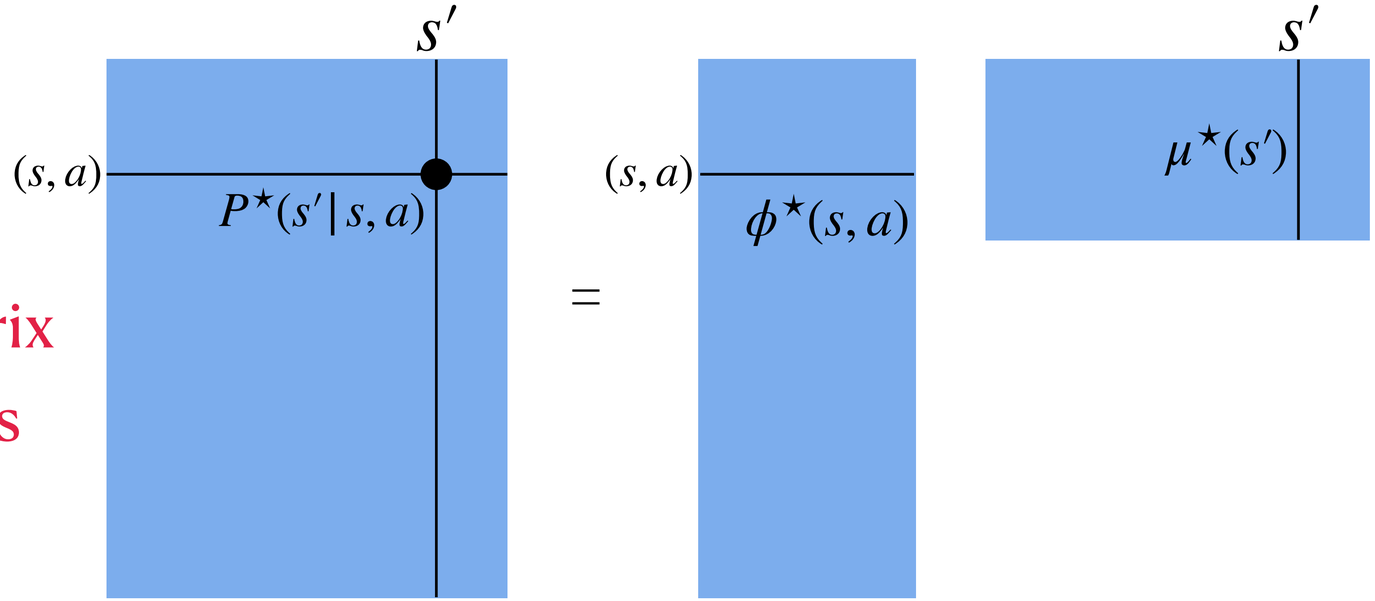
# Low-rank MDP

Transition matrix $P \in \mathbb{R}^{SA \times S}$ has rank $d$



# Low-rank MDP

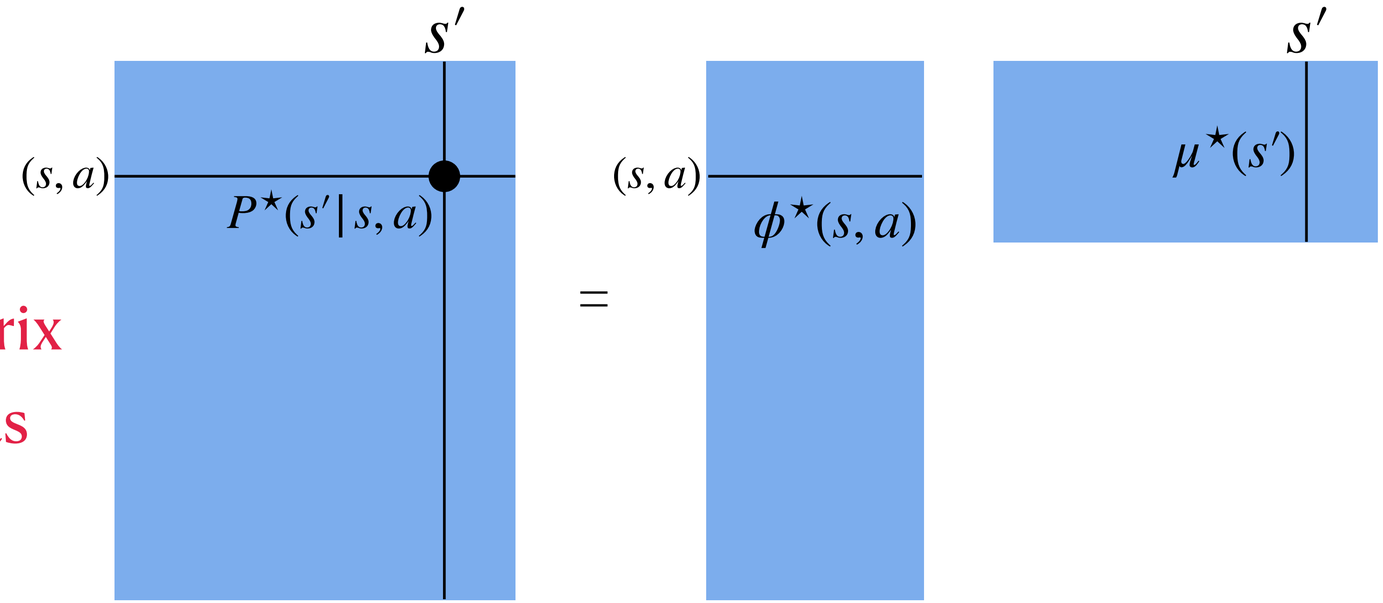
Transition matrix $P \in \mathbb{R}^{SA \times S}$ has rank $d$




$$\exists \mu^\star, \phi^\star : \quad \forall s, a, s', P^\star(s' \mid s, a) = \mu^\star(s')^\top \phi^\star(s, a)$$

# Low-rank MDP

Transition matrix $P \in \mathbb{R}^{SA \times S}$ has rank $d$




$$\exists \mu^\star, \phi^\star : \quad \forall s, a, s', P^\star(s' \mid s, a) = \mu^\star(s')^\top \phi^\star(s, a)$$

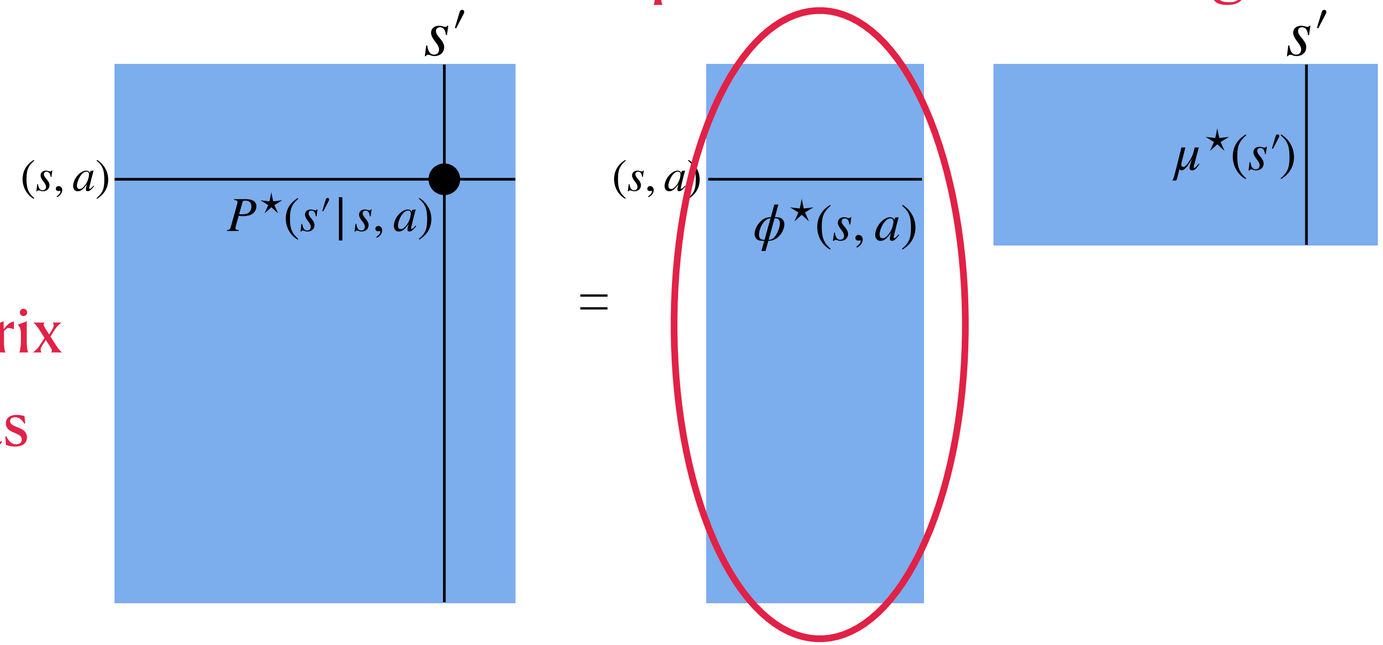
**Low-rank MDP ≠ Linear MDPs** (Jin et al, Yang & Wang)

Linear MDP = low-rank + known $\phi^\star$

# Low-rank MDP

Representation Learning

Transition matrix $P \in \mathbb{R}^{SA \times S}$ has rank $d$




$$\exists \mu^\star, \phi^\star : \quad \forall s, a, s', P^\star(s'|s,a) = \mu^\star(s')^\top \phi^\star(s,a)$$

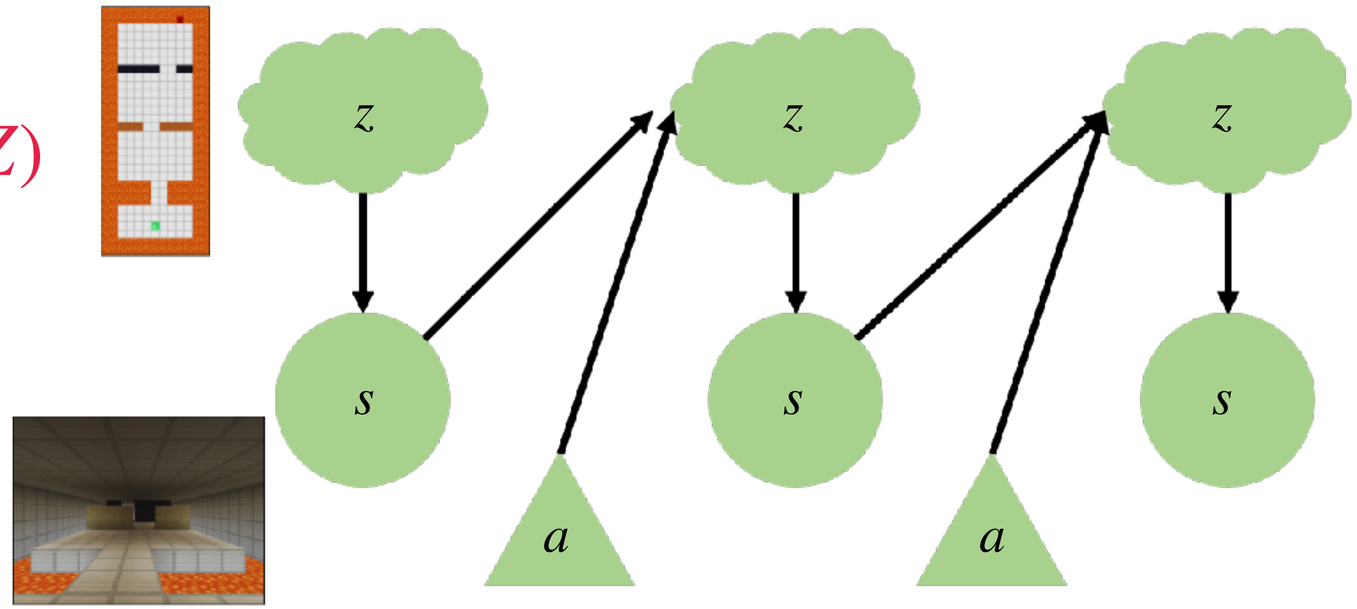
**Low-rank MDP ≠ Linear MDPs** (Jin et al, Yang & Wang)

Linear MDP = low-rank + known $\phi^\star$

# Low-rank MDP is general

e.g., Latent variable models where $Z$ is the discrete latent space

$\phi^\star(s,a) \in \Delta(Z)$

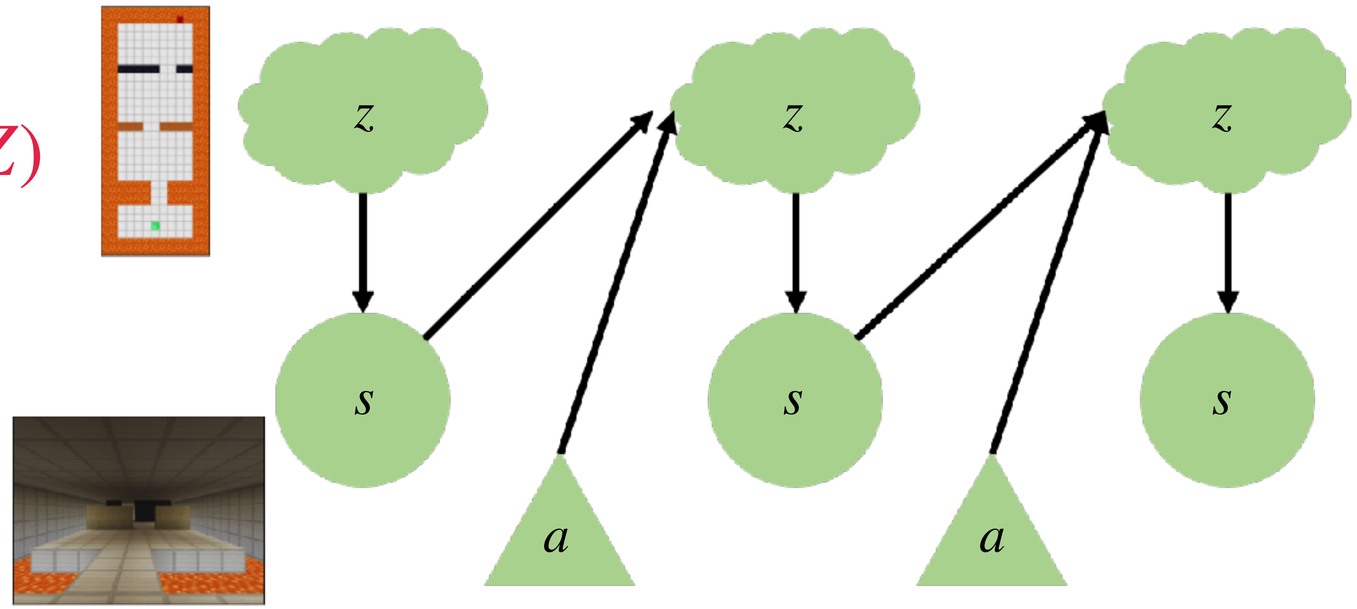



[slide credit: Akshay Krishnumurthy]

# Low-rank MDP is general

e.g., Latent variable models where $Z$ is the discrete latent space

$\phi^\star(s,a) \in \Delta(Z)$




Given $s, a$: $z \sim \phi^\star(s,a), s' \sim \nu^\star(z)$

[slide credit: Akshay Krishnumurthy]

# Provably efficient learning in low-rank mdp is plausible

| | Setting | Sample Complexity | Computation |
|---|---|---|---|
| **Olive [JKALS,17]** | Low Bellman rank | $\frac{d^2A}{\epsilon^2(1-\gamma)^4}$ | Inefficient |
| **Witness Rank [SJKAL,19]** | Low Witness rank | $\frac{d^2A}{\epsilon^2(1-\gamma)^4}$ | Inefficient |
| **BLin-UCB [DKLLMSW,21]** | Bilinear Class | $\frac{d^2A}{\epsilon^2(1-\gamma)^7}$ | Inefficient |
| **Moffle [MCKJA,21]** | Low-nonnegative-rank MDP | $\frac{d^6A^{13}}{\epsilon^2\eta^5(1-\gamma)^5}$ | Oracle-efficient |
| **FLAMBE [AKKS,20]** | Low-rank MDP | $\frac{d^7A^9}{\epsilon^{10}(1-\gamma)^{22}}$ | Oracle-efficient |

# Provably efficient learning in low-rank mdp is plausible

| | Setting | Sample Complexity | Computation |
|---|---|---|---|
| **Olive [JKALS,17]** | Low Bellman rank | $\frac{d^2A}{\epsilon^2(1-\gamma)^4}$ | Inefficient |
| **Witness Rank [SJKAL,19]** | Low Witness rank | $\frac{d^2A}{\epsilon^2(1-\gamma)^4}$ | Inefficient |
| **BLin-UCB [DKLLMSW,21]** | Bilinear Class | $\frac{d^2A}{\epsilon^2(1-\gamma)^7}$ | Inefficient |
| **Moffle [MCKJA,21]** | Low-nonnegative-rank MDP | $\frac{d^6A^{13}}{\epsilon^2\eta^5(1-\gamma)^5}$ | Oracle-efficient |
| **FLAMBE [AKKS,20]** | Low-rank MDP | $\frac{d^7A^9}{\epsilon^{10}(1-\gamma)^{22}}$ | Oracle-efficient |

FLAMBE is oracle-efficient and was state-of-art on low-rank MDP

# Our learning setting

1. Realizable hypothesis classes $\Gamma$, $\Phi$

$$\mu^\star \in \Gamma, \phi^\star \in \Phi$$

# Our learning setting

## 1. Realizable hypothesis classes $\Gamma$, $\Phi$

$$\mu^\star \in \Gamma, \phi^\star \in \Phi$$

## 2. Computation oracle:

Maximum Likelihood Estimation (MLE):

$$(\hat{\mu}, \hat{\phi}) := \arg\max_{\mu,\phi} \sum_{i=1}^{n} \ln\left(\mu(s_i')\phi(s_i, a_i)\right)$$

# Our learning setting

## 1. Realizable hypothesis classes $\Gamma$, $\Phi$

$$\mu^\star \in \Gamma, \phi^\star \in \Phi$$

## 2. Computation oracle:

Maximum Likelihood Estimation (MLE):

$$(\hat{\mu}, \hat{\phi}) := \arg\max_{\mu,\phi} \sum_{i=1}^{n} \ln\left(\mu(s_i')\phi(s_i, a_i)\right)$$

## 3. Learning Goal:

Finding near-optimal policy w/ (tight) $\text{poly}(A, d, 1/(1-\gamma), \ln(|\Phi||\Gamma|))$

# Our algorithm: Rep-UCB

## (UCB-driven Representation Learning for online RL)

At iteration n:

# Our algorithm: Rep-UCB

## (UCB-driven Representation Learning for online RL)

At iteration n:

Data generation from $\pi^n$:

$$s \sim d^{\pi^n}, a \sim \text{Uniform}(\mathcal{A}), s' \sim P^\star(\,.\,|s,a)$$
$$a' \sim \text{Uniform}(\mathcal{A}), s'' \sim P^\star(\,.\,|s',a')$$

# Our algorithm: Rep-UCB

## (UCB-driven Representation Learning for online RL)

At iteration n:

Data generation from $\pi^n$:

$$s \sim d^{\pi^n}, a \sim \text{Uniform}(\mathscr{A}), s' \sim P^\star(.|s,a)$$
$$a' \sim \text{Uniform}(\mathscr{A}), s'' \sim P^\star(.|s',a')$$

Data Aggregation:

$$\mathscr{D}_n = \mathscr{D}_{n-1} + \{s, a, s'\}$$
$$\mathscr{D}'_n = \mathscr{D}'_{n-1} + \{s', a', s''\}$$

# Our algorithm: Rep-UCB

## (UCB-driven Representation Learning for online RL)

At iteration n:

Data generation from $\pi^n$:

$$s \sim d^{\pi^n}, a \sim \text{Uniform}(\mathscr{A}), s' \sim P^\star(.\,|s,a)$$
$$a' \sim \text{Uniform}(\mathscr{A}), s'' \sim P^\star(.\,|s',a')$$

Data Aggregation:

$$\mathscr{D}_n = \mathscr{D}_{n-1} + \{s, a, s'\}$$
$$\mathscr{D}'_n = \mathscr{D}'_{n-1} + \{s', a', s''\}$$

Representation / model Learning (MLE)

$$\hat{P} := (\hat{\mu}, \hat{\phi}) = \arg\max_{\mu,\phi} \mathbb{E}_{\mathscr{D}_n + \mathscr{D}'_n} \ln(\mu(s')^\top \phi(s,a))$$

# Our algorithm: Rep-UCB

## (UCB-driven Representation Learning for online RL)

At iteration n:

Data generation from $\pi^n$:

$$s \sim d^{\pi^n}, a \sim \text{Uniform}(\mathscr{A}), s' \sim P^\star(.\,|s,a)$$
$$a' \sim \text{Uniform}(\mathscr{A}), s'' \sim P^\star(.\,|s',a')$$

Data Aggregation:

$$\mathscr{D}_n = \mathscr{D}_{n-1} + \{s, a, s'\}$$
$$\mathscr{D}'_n = \mathscr{D}'_{n-1} + \{s', a', s''\}$$

Representation / model Learning (MLE)

$$\hat{P} := (\hat{\mu}, \hat{\phi}) = \arg\max_{\mu,\phi} \mathbb{E}_{\mathscr{D}_n + \mathscr{D}'_n} \ln(\mu(s')^\top \phi(s,a))$$

(Linear bandit style) bonus under $\hat{\phi}$:

$$b(s,a) = c\sqrt{\hat{\phi}(s,a)\Sigma^{-1}\hat{\phi}(s,a)}$$

$$\Sigma = \sum_{s,a \in \mathscr{D}_n} \hat{\phi}(s,a)\hat{\phi}(s,a)^\top$$

# Our algorithm: Rep-UCB

## (UCB-driven Representation Learning for online RL)

At iteration n:

Data generation from $\pi^n$:

$$s \sim d^{\pi^n}, a \sim \text{Uniform}(\mathscr{A}), s' \sim P^\star(.\,|s,a)$$
$$a' \sim \text{Uniform}(\mathscr{A}), s'' \sim P^\star(.\,|s',a')$$

Data Aggregation:

$$\mathscr{D}_n = \mathscr{D}_{n-1} + \{s, a, s'\}$$
$$\mathscr{D}'_n = \mathscr{D}'_{n-1} + \{s', a', s''\}$$

Representation / model Learning (MLE)

$$\hat{P} := (\hat{\mu}, \hat{\phi}) = \arg\max_{\mu,\phi} \mathbb{E}_{\mathscr{D}_n + \mathscr{D}'_n} \ln(\mu(s')^\top \phi(s,a))$$

(Linear bandit style) bonus under $\hat{\phi}$:

$$b(s,a) = c\sqrt{\hat{\phi}(s,a)\Sigma^{-1}\hat{\phi}(s,a)}$$

$$\Sigma = \sum_{s,a \in \mathscr{D}_n} \hat{\phi}(s,a)\hat{\phi}(s,a)^\top$$

Plan w/ reward + bonus

$$\pi^{n+1} = \max_\pi J(\pi; \hat{P}, r + b)$$

# PAC-Bound of Rep-UCB in low-rank MDP

Assume trajectory-reward is normalized in [0,1]. W/ high probability, it finds an $\epsilon$ near optimal policy, with # of samples:

$$\widetilde{O}\left(\frac{d^4A^2}{\epsilon^2(1-\gamma)^5}\cdot\ln\left(|\Gamma||\Phi|\right)\right)$$

# PAC-Bound of Rep-UCB in low-rank MDP

Assume trajectory-reward is normalized in [0,1]. W/ high probability, it finds an $\epsilon$ near optimal policy, with # of samples:

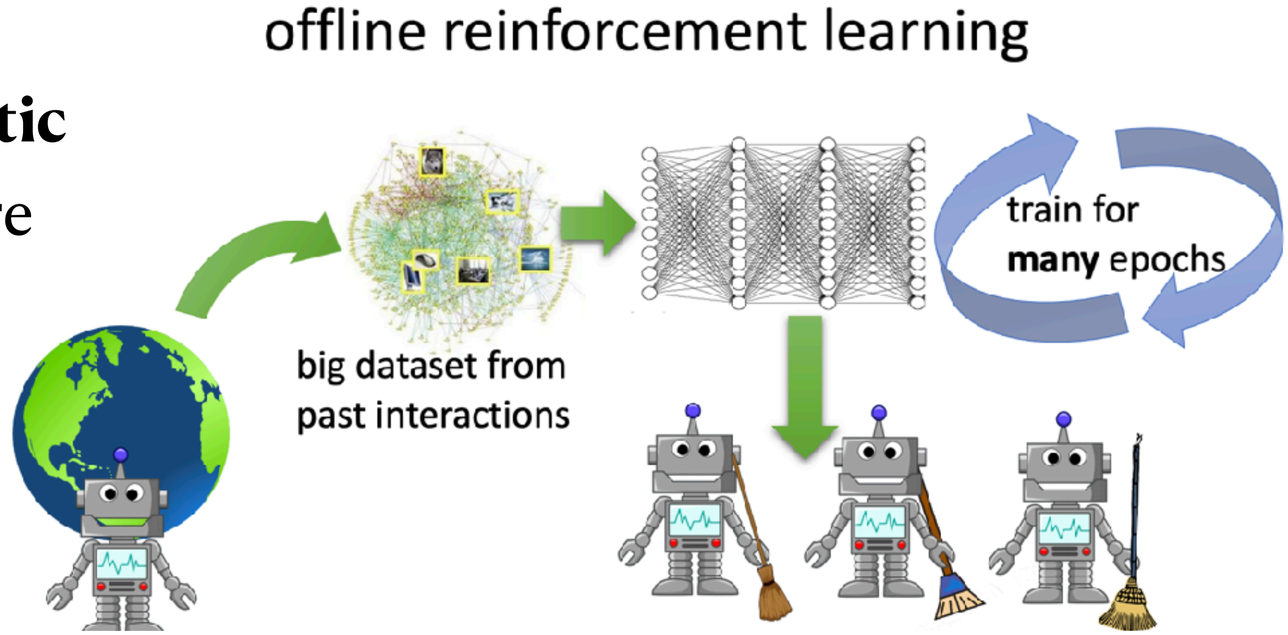
$$\widetilde{O}\left(\frac{d^4A^2}{\epsilon^2(1-\gamma)^5}\cdot\ln\left(|\Gamma||\Phi|\right)\right)$$

For reference, prior SOTA
FLAMBE has the following bound:

$$\widetilde{O}\left(\frac{d^7A^9}{\epsilon^{10}(1-\gamma)^{22}}\cdot\ln\left(|\Gamma||\Phi|\right)\right)$$

# Applying our new techniques to Offline RL

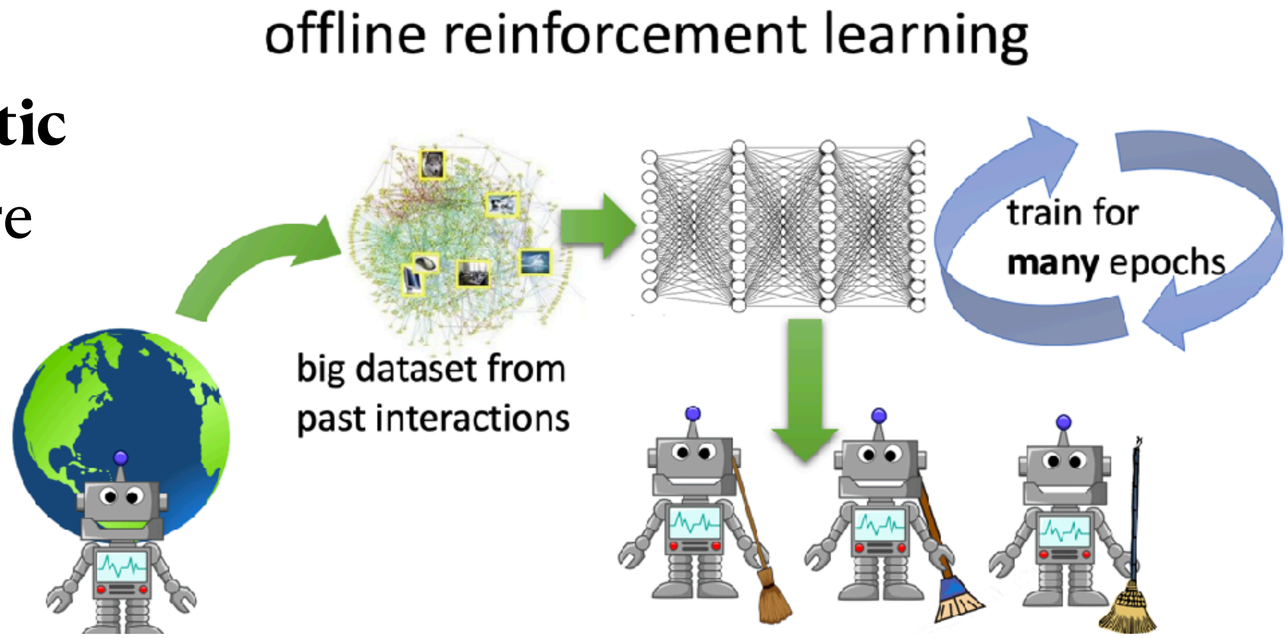
Offline RL: we only have a **static dataset** $\mathcal{D} = \{s, a, s'\}$, where $(s, a) \sim \pi_b, s' \sim P^\star(\,.\,|\,s, a)$




[Image from BAIR blog post: https://bair.berkeley.edu/blog/2020/12/07/offline/]

# Applying our new techniques to Offline RL

Offline RL: we only have a **static dataset** $\mathcal{D} = \{s, a, s'\}$, where $(s, a) \sim \pi_b, s' \sim P^\star(\,.\,|\,s, a)$




Goal: learn to find some high quality policy solely from $\mathcal{D}$

[Image from BAIR blog post: https://bair.berkeley.edu/blog/2020/12/07/offline/]

# Coverage condition of the offline data

A comparator policy $\pi$ is covered by offline data if the relative condition number is bounded:

$$C_{\pi^*} := \max_x \frac{x^\top \left(\mathbb{E}_{s,a\sim d^\pi} \phi^\star(s,a)\phi^\star(s,a)^\top\right) x}{x^\top \left(\mathbb{E}_{s,a\sim d^{\pi_b}} \phi^\star(s,a)\phi^\star(s,a)^\top\right) x} < \infty$$

Note coverage is wrt true representation only!

# Coverage condition of the offline data

A comparator policy $\pi$ is covered by offline data if the relative condition number is bounded:

$$C_{\pi^*} := \max_x \frac{x^\top \left(\mathbb{E}_{s,a\sim d^\pi} \phi^\star(s,a)\phi^\star(s,a)^\top\right) x}{x^\top \left(\mathbb{E}_{s,a\sim d^{\pi_b}} \phi^\star(s,a)\phi^\star(s,a)^\top\right) x} < \infty$$

Note coverage is wrt true representation only!

Goal is to learn **robustly**, i.e., as long as there is a high quality policy that is covered by $d^{\pi_b}$, we want to compete against it!

# The Rep-LCB Algorithm

## (Low confidence bound driven offline RL)

1. Representation / model Learning (MLE) under $\mathscr{D}$

$$\hat{P} := (\hat{\mu}, \hat{\phi}) = \arg\max_{\mu,\phi} \mathbb{E}_{\mathscr{D}} \ln(\mu(s')^\top \phi(s,a))$$

# The Rep-LCB Algorithm

## (Low confidence bound driven offline RL)

2. Penalty w/ $\hat{\phi}$

$$b(s,a) = c\sqrt{\hat{\phi}(s,a)\Sigma^{-1}\hat{\phi}(s,a)}$$

$$\Sigma = \sum_{s,a \in \mathscr{D}_n} \hat{\phi}(s,a)\hat{\phi}(s,a)^\top$$

1. Representation / model Learning (MLE) under $\mathscr{D}$

$$\hat{P} := (\hat{\mu}, \hat{\phi}) = \arg\max_{\mu,\phi} \mathbb{E}_{\mathscr{D}} \ln(\mu(s')^\top \phi(s,a))$$

# The Rep-LCB Algorithm

## (Low confidence bound driven offline RL)

1. Representation / model Learning (MLE) under $\mathscr{D}$

$$\hat{P} := (\hat{\mu}, \hat{\phi}) = \arg\max_{\mu,\phi} \mathbb{E}_{\mathscr{D}} \ln(\mu(s')^\top \phi(s,a))$$

2. Penalty w/ $\hat{\phi}$

$$b(s,a) = c\sqrt{\hat{\phi}(s,a)\Sigma^{-1}\hat{\phi}(s,a)}$$

$$\Sigma = \sum_{s,a \in \mathscr{D}_n} \hat{\phi}(s,a)\hat{\phi}(s,a)^\top$$

3. Conservative Plan w/ reward + penalty

$$\hat{\pi} = \max_{\pi} J(\pi; \hat{P}, r - b)$$

# The guarantee of Rep-LCB

Assume the behavior policy $\pi_b(a \mid s) \geq w, \forall s$; W/ high probability, for **ALL comparator policy** $\pi^*$ (include **history-dependent ones**):

$$J(\pi^*; r) - J(\hat{\pi}; r) \leq \widetilde{O}\left(\frac{d^2}{(1-\gamma)^{1.5}}\sqrt{\frac{wC_{\pi^*}}{n}} \cdot \ln(|\Phi||\Gamma|)\right)$$

# The guarantee of Rep-LCB

Assume the behavior policy $\pi_b(a \mid s) \geq w, \forall s$; W/ high probability, for **ALL comparator policy** $\pi^*$ (include **history-dependent ones**):

$$J(\pi^*; r) - J(\hat{\pi}; r) \leq \widetilde{O}\left( \frac{d^2}{(1-\gamma)^{1.5}} \sqrt{\frac{wC_{\pi^*}}{n} \cdot \ln(|\Phi||\Gamma|)} \right)$$

# The guarantee of Rep-LCB

Assume the behavior policy $\pi_b(a \mid s) \geq w, \forall s$; W/ high probability, for **ALL comparator policy** $\pi^*$ (include **history-dependent ones**):

$$J(\pi^*; r) - J(\hat{\pi}; r) \leq \widetilde{O}\left( \frac{d^2}{(1-\gamma)^{1.5}} \sqrt{\frac{wC_{\pi^*}}{n}} \cdot \ln(|\Phi||\Gamma|) \right)$$

(prior work CPPO [Uehara & Sun, 21] can achieve similar guarantee, but is a version-space alg)

# Summary

**1. Improved online Representation Learning algorithm for low-rank MDP:**

Oracle-efficient + tight sample complexity

**2. New offline RL algorithm for low-rank MDP:**

Partial coverage + Oracle-efficient

Rep-UCB / LCB: https://arxiv.org/pdf/2110.04652.pdf