Trajectory Data Suffices for Statistically Efficient Learning in Offline RL with Linear q^π– Realizability and Concentrability

Vlad Tkachuk[1], Gellért Weisz[2], Csaba Szepesvári[1,2]

1. University of Alberta, 2. Google Deepmind

Offline RL Needs Less Data if you Have Trajectories

Vlad Tkachuk[1], Gellért Weisz[2], Csaba Szepesvári[1,2]

1. University of Alberta, 2. Google Deepmind

Motivation (learning with offline data)

Safety concerns (ex: healthcare)

Motivation (learning with offline data)

Safety concerns (ex: healthcare)

ChatGPT

There is a lot of offline data available

(ex: the entire internet)

Overview

- (Setting) What is the problem?
- (Related works) What did we know?
- (Our result) What we know now!
- (Our method) How we know it...
- (Future work) What's next?

Environment: MDP

Finite-Horizon Markov Decision Process (**MDP**): $(\mathcal{S}, \mathcal{A}, P, \mathcal{R}, H, s_1)$

Environment: MDP

Finite-Horizon Markov Decision Process (**MDP**): $(\mathcal{S}, \mathcal{A}, P, \mathcal{R}, H, s_1)$

$\mathcal{S} = \bigcup_{h \in [H]} \mathcal{S}_h$ **(State space)**: where $\mathcal{S}_h$ is the set of states at stage h

$[H] = \{1, \ldots, H\}$

Environment: MDP

Finite-Horizon Markov Decision Process (**MDP**): $(\mathcal{S}, \mathcal{A}, P, \mathcal{R}, H, s_1)$

$\mathcal{S} = \bigcup_{h \in [H]} \mathcal{S}_h$ **(State space)**: where $\mathcal{S}_h$ is the set of states at stage h

$\mathcal{A}$ **(Action space)**: A finite set of actions

$[H] = \{1, \ldots, H\}$

Environment: MDP

Finite-Horizon Markov Decision Process (**MDP**): $(\mathcal{S}, \mathcal{A}, P, \mathcal{R}, H, s_1)$

$\mathcal{S} = \bigcup_{h \in [H]} \mathcal{S}_h$ **(State space)**: where $\mathcal{S}_h$ is the set of states at stage h

$\mathcal{A}$ **(Action space)**: A finite set of actions

$P : \mathcal{S}_h \times \mathcal{A} \to \mathcal{M}_1(\mathcal{S}_{h+1})$ **(Transition function)**

$[H] = \{1, \ldots, H\}$

$\mathcal{M}_1(X)$ = Set of probability distributions over the set X

Environment: MDP

Finite-Horizon Markov Decision Process (**MDP**): $(\mathcal{S}, \mathcal{A}, P, \mathcal{R}, H, s_1)$

$\mathcal{S} = \bigcup_{h \in [H]} \mathcal{S}_h$ **(State space)**: where $\mathcal{S}_h$ is the set of states at stage h

$\mathcal{A}$ **(Action space)**: A finite set of actions

$P : \mathcal{S}_h \times \mathcal{A} \to \mathcal{M}_1(\mathcal{S}_{h+1})$ **(Transition function)**

$r : \mathcal{S} \times \mathcal{A} \to [0,1]$ **(Reward function)** [Deterministic for convenience]

$[H] = \{1, \ldots, H\}$

$\mathcal{M}_1(X)$ = Set of probability distributions over the set X

Environment: MDP

Finite-Horizon Markov Decision Process (**MDP**): $(\mathcal{S}, \mathcal{A}, P, \mathcal{R}, H, s_1)$

$\mathcal{S} = \bigcup_{h \in [H]} \mathcal{S}_h$ **(State space)**: where $\mathcal{S}_h$ is the set of states at stage h

$\mathcal{A}$ **(Action space)**: A finite set of actions

$P : \mathcal{S}_h \times \mathcal{A} \to \mathcal{M}_1(\mathcal{S}_{h+1})$ **(Transition function)**

$r : \mathcal{S} \times \mathcal{A} \to [0,1]$ **(Reward function)** [Deterministic for convenience]

$H \geq 1$ **(Horizon)**

$[H] = \{1, \ldots, H\}$

$\mathcal{M}_1(X)$ = Set of probability distributions over the set X

Environment: MDP

Finite-Horizon Markov Decision Process (**MDP**): $(\mathcal{S}, \mathcal{A}, P, \mathcal{R}, H, s_1)$

$\mathcal{S} = \bigcup_{h \in [H]} \mathcal{S}_h$ **(State space)**: where $\mathcal{S}_h$ is the set of states at stage h

$\mathcal{A}$ **(Action space)**: A finite set of actions

$P : \mathcal{S}_h \times \mathcal{A} \to \mathcal{M}_1(\mathcal{S}_{h+1})$ **(Transition function)**

$r : \mathcal{S} \times \mathcal{A} \to [0,1]$ **(Reward function)** [Deterministic for convenience]

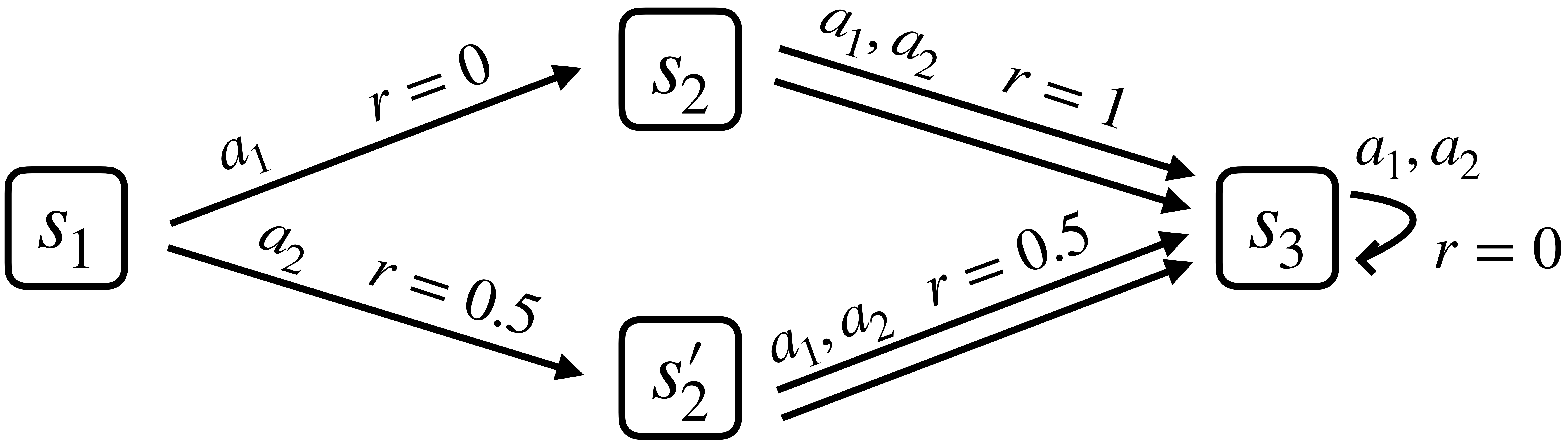
$H \geq 1$ **(Horizon)**

$s_1 \in \mathcal{S}_1$ **(Start state)**

$[H] = \{1, \ldots, H\}$

$\mathcal{M}_1(X)$ = Set of probability distributions over the set X

Example: MDP



Agent's Behaviour: Policy

$\pi : \mathcal{S} \to \mathcal{M}_1(\mathcal{A})$ **(Policy)**: A map from states to distributions over actions

$\mathcal{M}_1(X)$ = Set of probability distributions over the set X

The Problem

Definitions $\big((s_h, a_h) \in \mathcal{S}_h \times \mathcal{A}$ and $h \in [H]\big)$:

$$v^\pi(s_h) = \mathbb{E}_\pi \left[\sum_{t=h}^{H} r(S_t, A_t) \,|\, S_h = s_h \right]$$ **(State-value function)**

The Problem

Definitions $\left((s_h, a_h) \in \mathcal{S}_h \times \mathcal{A} \text{ and } h \in [H]\right)$:

$v^\pi(s_h) = \mathbb{E}_\pi\left[\sum_{t=h}^{H} r(S_t, A_t) \mid S_h = s_h\right]$ **(State-value function)**

$\pi^* = \arg\max_\pi v^\pi(s_1)$ **(Optimal policy)**

The Problem

Definitions $\left((s_h, a_h) \in \mathcal{S}_h \times \mathcal{A} \text{ and } h \in [H]\right)$:

$$v^\pi(s_h) = \mathbb{E}_\pi \left[\sum_{t=h}^{H} r(S_t, A_t) \,|\, S_h = s_h \right]$$ **(State-value function)**

$$\pi^* = \arg\max_\pi v^\pi(s_1)$$ **(Optimal policy)**

Problem: For any $\epsilon > 0$, with access to offline data of size $n = \text{poly}(1/\epsilon, H, |\mathcal{S}|, |\mathcal{A}|)$, find a policy π such that:

$$v^{\pi^*}(s_1) - v^\pi(s_1) \leq \epsilon$$

The Problem

Definitions $\left((s_h, a_h) \in \mathcal{S}_h \times \mathcal{A} \text{ and } h \in [H]\right)$:

$v^\pi(s_h) = \mathbb{E}_\pi\left[\sum_{t=h}^{H} r(S_t, A_t) \,|\, S_h = s_h\right]$ **(State-value function)**

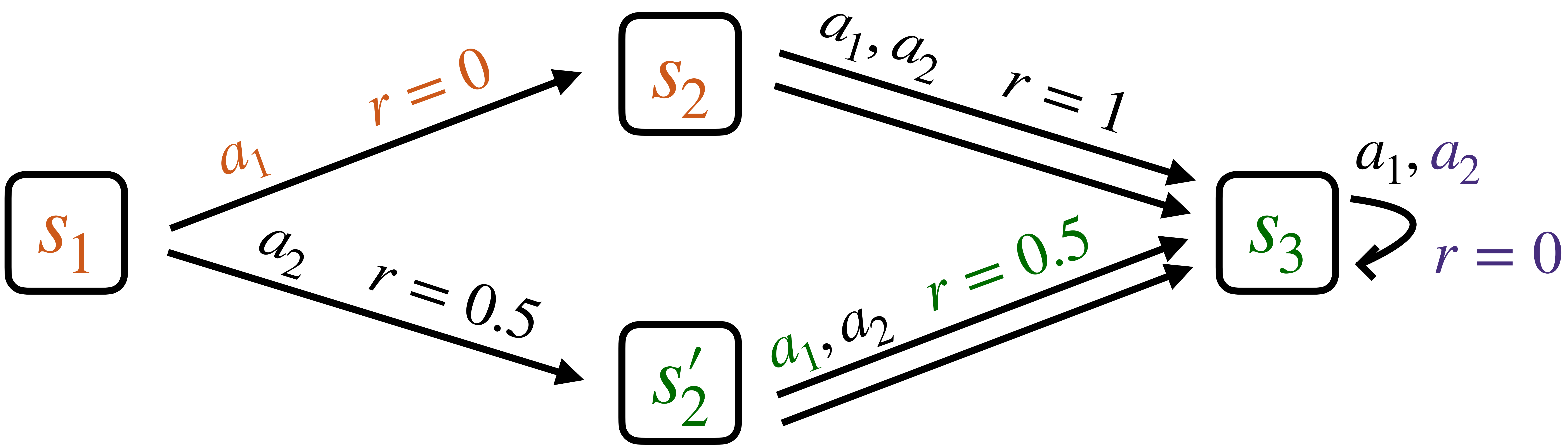
$\pi^* = \arg\max_\pi v^\pi(s_1)$ **(Optimal policy)**

Problem: For any $\epsilon > 0$, with access to offline data of size $n = \text{poly}(1/\epsilon, H, |\mathcal{S}|, |\mathcal{A}|)$, find a policy π such that:

$$v^{\pi^*}(s_1) - v^\pi(s_1) \leq \epsilon$$

(i.e. Find a **good policy** with a **small amount** of **offline data**)

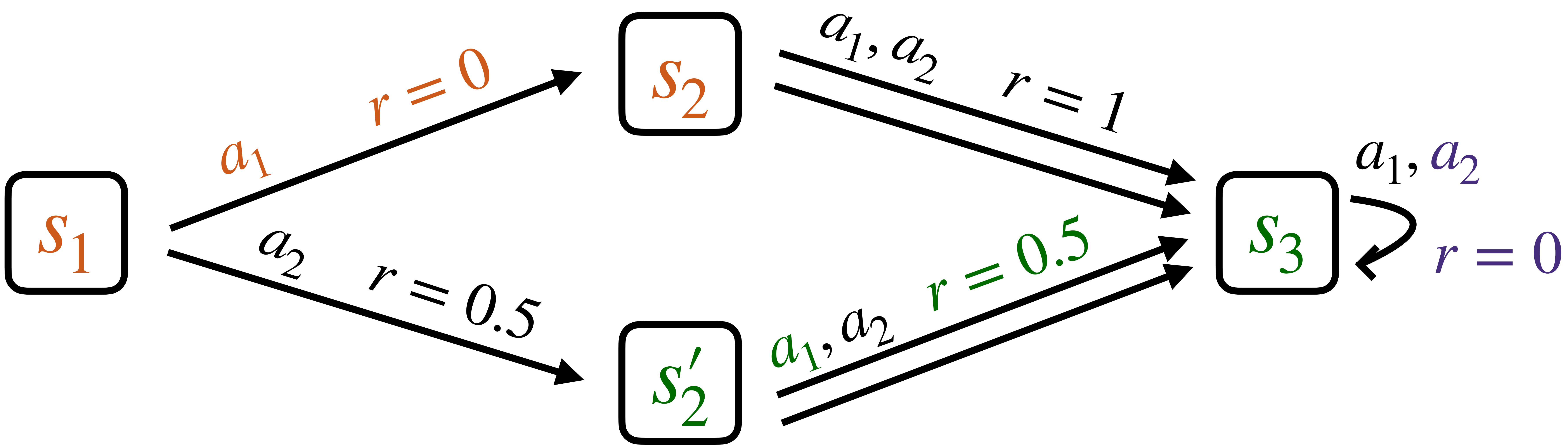
Example: Offline Data



Offline data $(n = 1)$**:** $\big((s_1, a_1, 0, s_2), (s_2', a_1, 0.5, s_3), (s_3, a_2, 0, s_3)\big)$

$h = 1$ $\quad$ $h = 2$ $\quad$ $h = 3$

Example: Offline Data



Notice $s_2 \neq s_2'$

i.e. **Not** trajectory data

Offline data $(n = 1)$: $\Big((s_1, a_1, 0, s_2), (s_2', a_1, 0.5, s_3), (s_3, a_2, 0, s_3)\Big)$

$h = 1$ $\quad h = 2$ $\quad h = 3$

The State Space is Very Very Large!

The number of states $|\mathcal{S}|$ can be very large!

Examples: **Chess, Robotics,** Go, Self-driving, etc.

The Problem

Problem: For any $\epsilon > 0$, with access to offline data of size $n = \text{poly}(1/\epsilon, H, |\mathcal{S}|, |\mathcal{A}|)$, find a policy π such that:

$$v^{\pi^*}(s_1) - v^{\pi}(s_1) \leq \epsilon$$

(i.e. Find a **good policy** with a **small amount** of **offline data**)

The Problem

Problem: For any $\epsilon > 0$, with access to offline data of size $n = \text{poly}(1/\epsilon, H, \boxed{|\mathcal{S}|,} |\mathcal{A}|)$, find a policy π such that:

$$v^{\pi^*}(s_1) - v^{\pi}(s_1) \leq \epsilon$$

(i.e. Find a **good policy** with a **small amount** of **offline data**)

The Problem

Problem: For any $\epsilon > 0$, with access to offline data of size $n = \text{poly}(1/\epsilon, H, d, |\mathcal{A}|)$, find a policy π such that:

$$v^{\pi^*}(s_1) - v^{\pi}(s_1) \leq \epsilon$$

(i.e. Find a **good policy** with a **small amount** of **offline data**)

Overview

- (Setting) What is the problem?
- (Related works) What did we know?
- (Our result) What we know now!
- (Our method) How we know it...
- (Future work) What's next?

Assumption Space

Offline data of size $n = \text{poly}(\ldots)$

All

MDP

All

Assumption Space

MDP

Data

	All
All	✗

Offline data of size $n = \text{poly}(\ldots)$

All

MDP

All

Assumption Space

MDP

Data

	All
All	✗

Offline data of size $n = \text{poly}(\ldots)$

All

Concentrability

MDP

All

Assumption Space

MDP

Data

	All
All	✗

Offline data of size $n = \text{poly}(\ldots)$

All

Concentrability

i.e. Good coverage of the state and action spaces

MDP

All

Assumption Space

	MDP: All
Data: All	✗
Data: Concentrability	✗ [1]

Offline data of size $n = \text{poly}(\ldots)$

All

Concentrability

MDP

All

1. [Foster et al., 2021]

Assumption Space

MDP

Data		All
	All	✗
	Concentrability	✗[1]

Offline data of size $n = \text{poly}(\ldots)$

All

Concentrability

MDP

All

Linearly q^π-realizable

1. [Foster et al., 2021]

Assumption Space

Data \ MDP	All
All	✗
Concentrability	✗[1]

Offline data of size $n = \text{poly}(\ldots)$

All

Concentrability

MDP

All

Linearly q^π-realizable

i.e. Value function approximation

1. [Foster et al., 2021]

Assumption Space

MDP

Data	All	Linearly q^π-realizable
All	✗	✗
Concentrability	✗[1]	✗[1]

Offline data of size $n = \text{poly}(\ldots)$

All

Concentrability

MDP

All

Linearly q^π-realizable

1. [Foster et al., 2021]

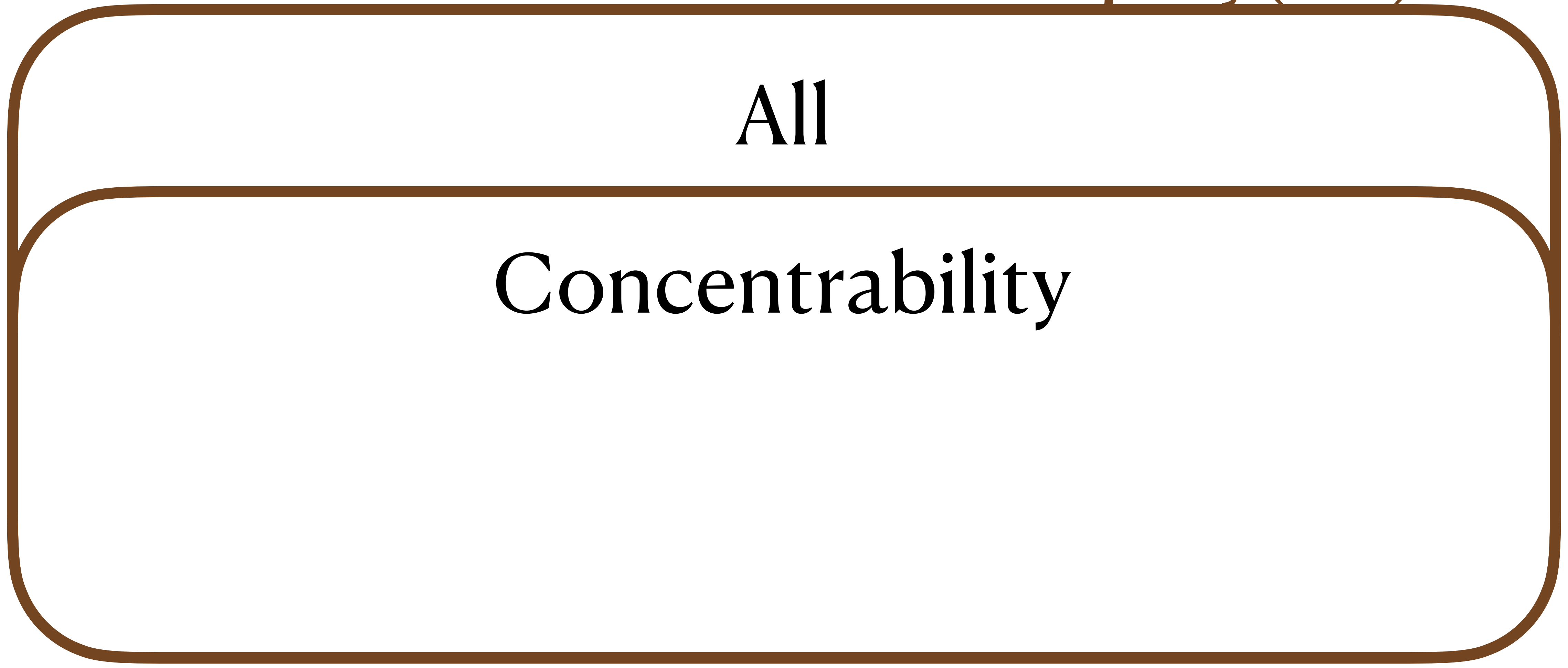
Assumption Space

MDP

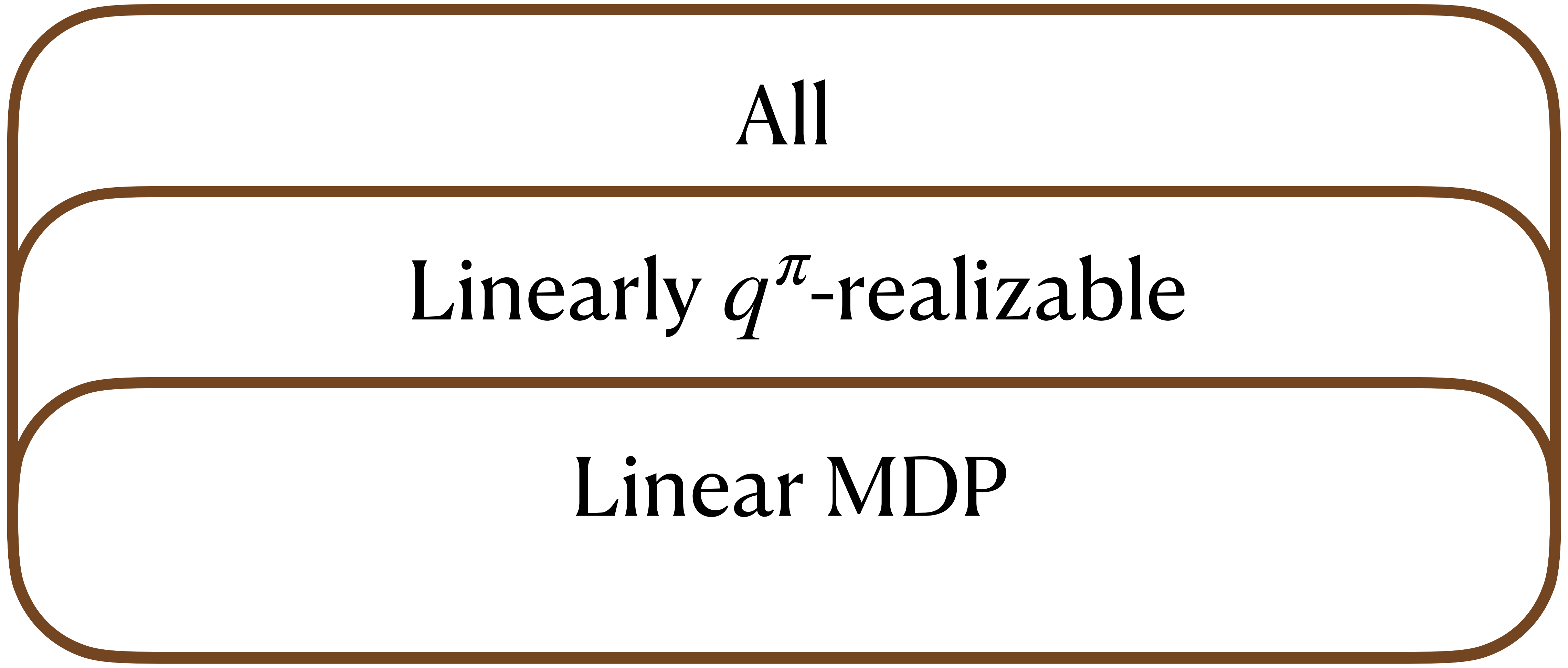
Data

	All	Linearly q^π-realizable
All	✗	✗
Concentrability	✗[1]	✗[1]

Offline data of size $n = \mathrm{poly}(\ldots)$



MDP

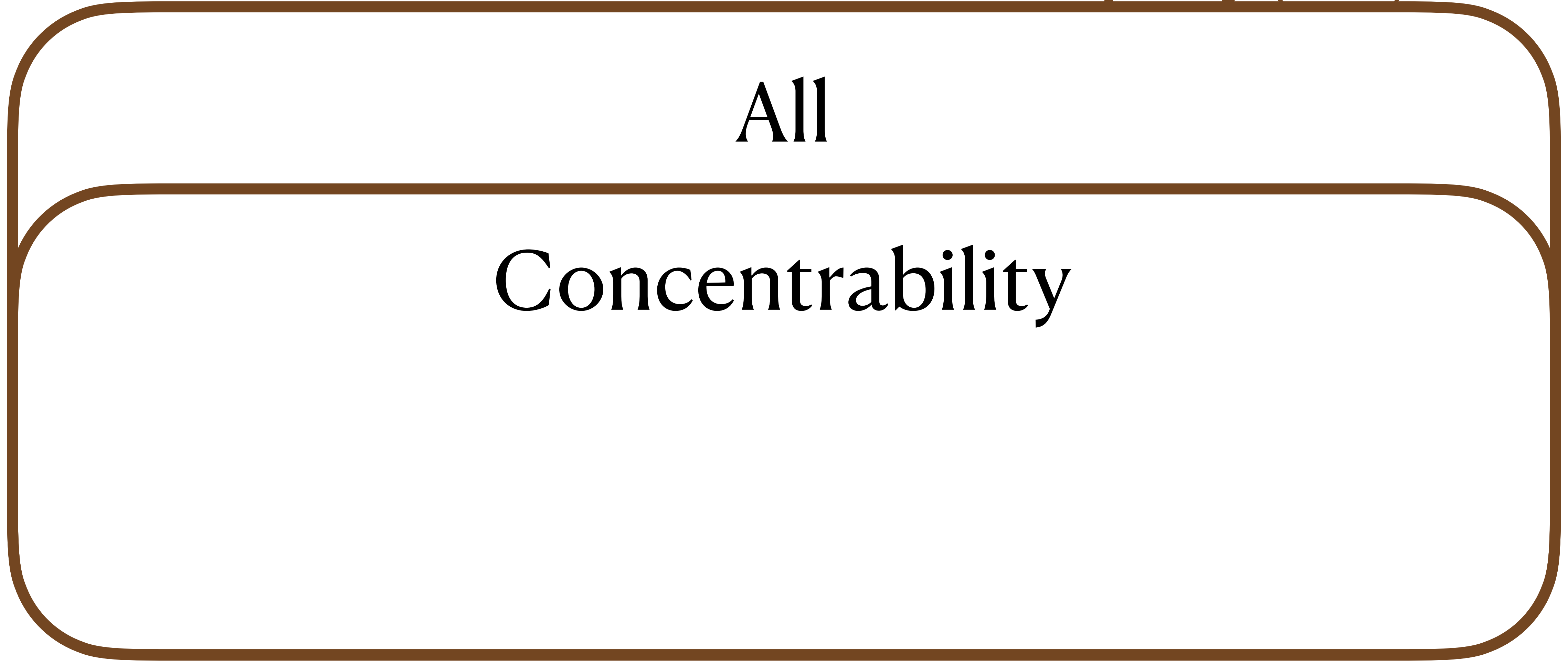


1. [Foster et al., 2021]

Assumption Space

		MDP
Data	**All**	**Linearly q^{π}-realizable**
All	✗	✗
Concentrability	✗[1]	✗[1]

Offline data of size $n = \mathrm{poly}(\ldots)$



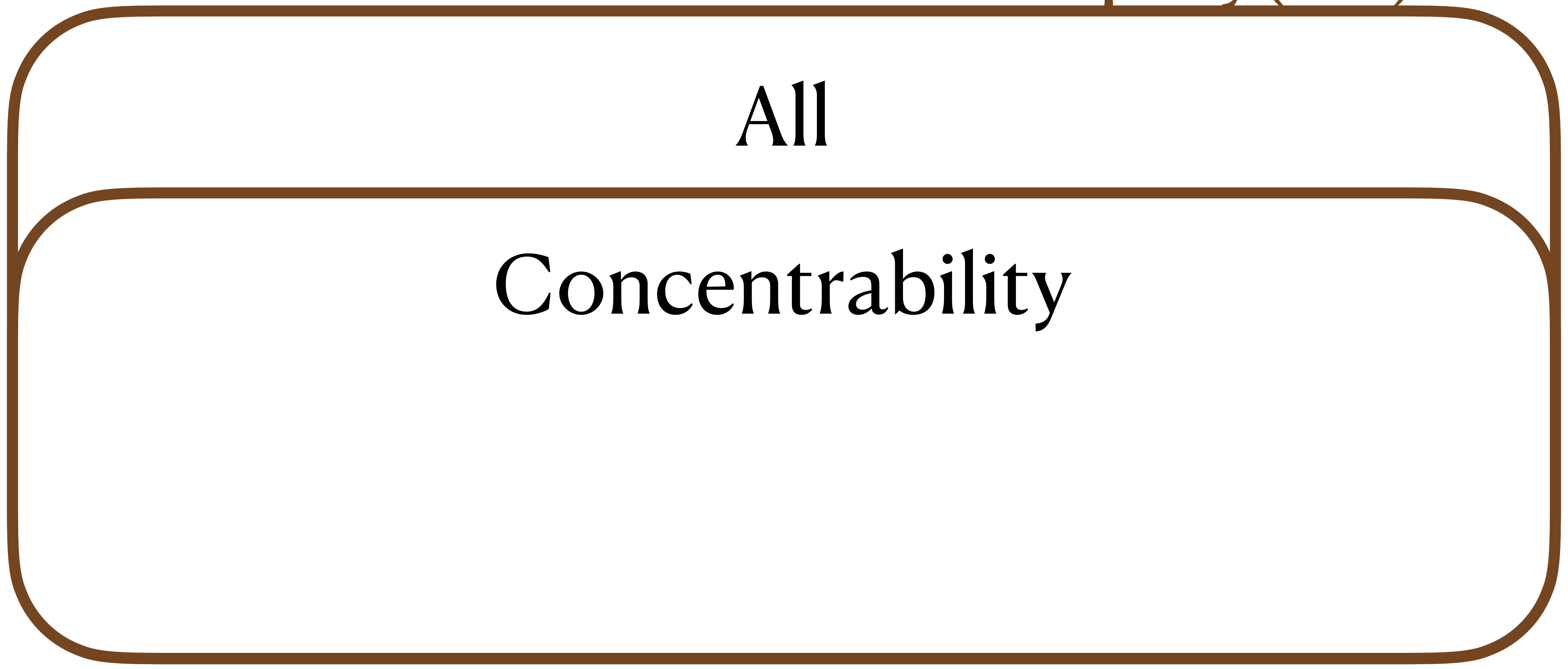
MDP

1. [Foster et al., 2021]

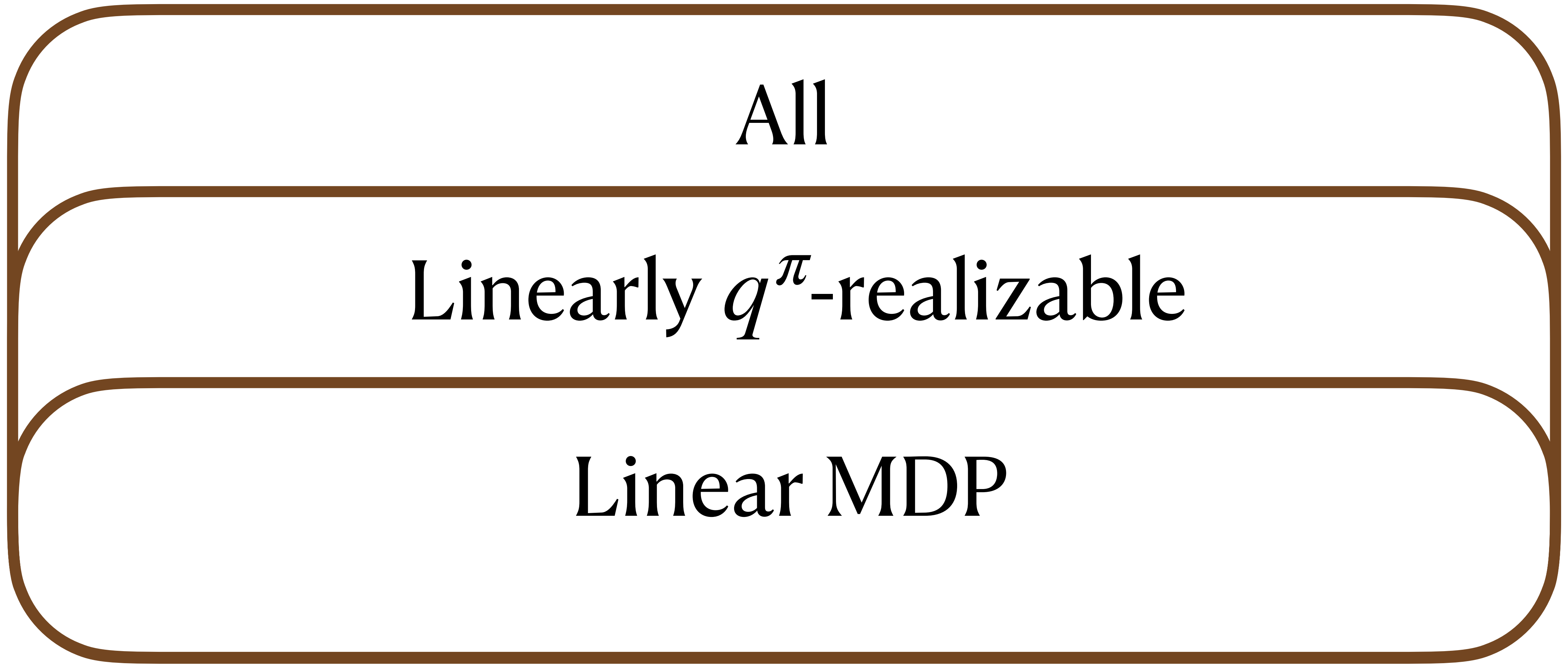
Assumption Space

Data \ MDP	All	Linearly q^{π}-realizable	Linear MDP
All	✗	✗	✗
Concentrability	✗[1]	✗[1]	✓[2]

Offline data of size $n = \text{poly}(\ldots)$



MDP



1. [Foster et al., 2021], 2.[Chen and Jiang, 2019]

Assumption Space

MDP

Data

	All	Linearly q^{π}-realizable	Linear MDP
All	✗	✗	✗
Concentrability	✗[1]	✗[1]	✓[2]

Offline data of size $n = \text{poly}(\ldots)$

All

Concentrability

Something stronger than concentrability

MDP

All

Linearly q^{π}-realizable

Linear MDP

1. [Foster et al., 2021], 2.[Chen and Jiang, 2019]

Assumption Space

MDP

Data	All	Linearly q^π-realizable	Linear MDP
All	✗	✗	✗
Concentrability	✗[1]	✗[1]	✓[2]

Offline data of size $n = \text{poly}(\ldots)$

All

Concentrability

Concentrability & Trajectory data

MDP

All

Linearly q^π-realizable

Linear MDP

1. [Foster et al., 2021], 2.[Chen and Jiang, 2019]

Assumption Space

	MDP		
Data	**All**	**Linearly q^π-realizable**	**Linear MDP**
All	✗	✗	✗
Concentrability	✗[1]	✗[1]	✓[2]
Concentrability & Trajectory data	✗	?	✓[2]

Offline data of size $n = \text{poly}(\ldots)$

All

Concentrability

Concentrability & Trajectory data

MDP

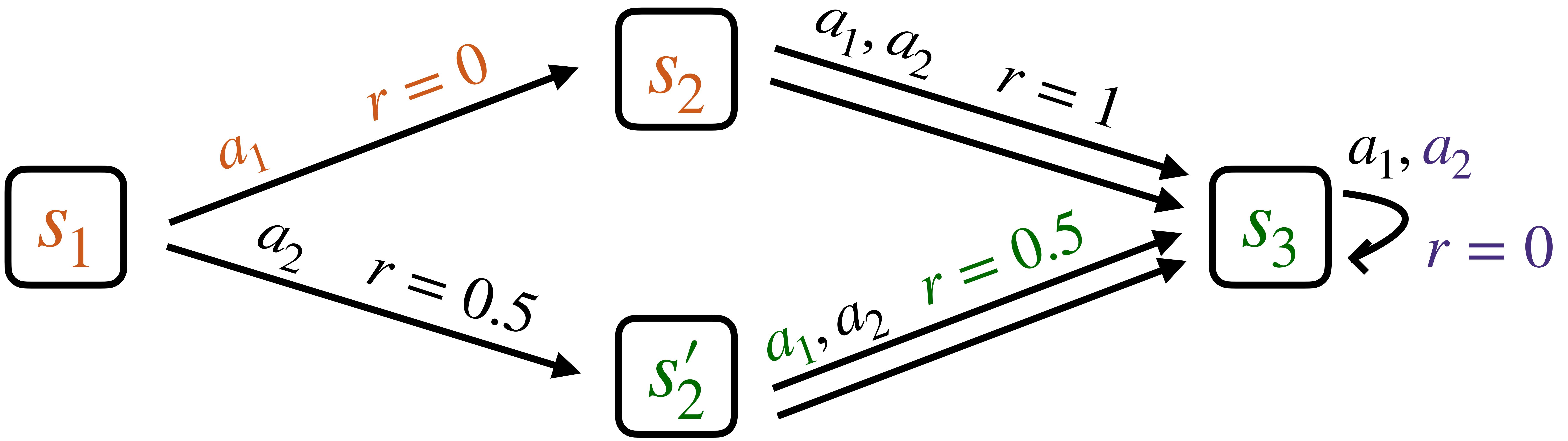
All

Linearly q^π-realizable

Linear MDP

1. [Foster et al., 2021], 2.[Chen and Jiang, 2019]

Example: Offline Data



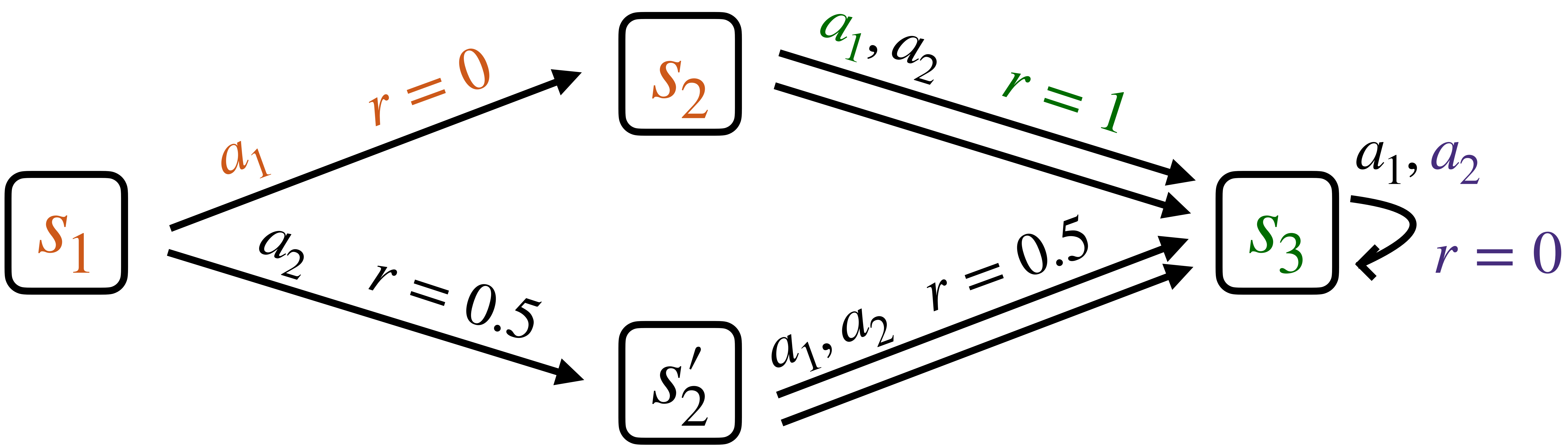
Notice $s_2 \neq s_2'$

i.e. **Not** trajectory data

Offline data ($n = 1$): $\Big((s_1, a_1, 0, s_2), (s_2', a_1, 0.5, s_3), (s_3, a_2, 0, s_3)\Big)$

$h = 1$ $\quad$ $h = 2$ $\quad$ $h = 3$

Example: Offline Trajectory Data



Notice $\boxed{s_2 = s_2}$

i.e. **trajectory data**

$$\textbf{Offline data } (n=1)\textbf{:}\ \Big(\underbrace{(s_1, a_1, 0, \boxed{s_2})}_{h=1}^{a_1 \sim \pi_g(s_1)}, \underbrace{(\boxed{s_2}, a_1, 1, s_3)}_{h=2}^{a_1 \sim \pi_g(s_2)}, \underbrace{(s_3, a_2, 0, s_3)}_{h=3}^{a_2 \sim \pi_g(s_3)} \Big)$$

Overview

- (Setting) What is the problem?
- (Related works) What did we know?
- (Our result) What we know now!
- (Our method) How we know it...
- (Future work) What's next?

(Our result) What we know now!

Data \ MDP	All	Linearly q^{π}-realizable	Linear MDP
All	✗	✗	✗
Concentrability	✗[1]	✗[1]	✓[2]
Concentrability & Trajectory data	✗	✓	✓[2]

Offline data of size $n = \text{poly}(\ldots)$

All

Concentrability

Concentrability & Trajectory data

MDP

All

Linearly q^{π}-realizable

Linear MDP

1. [Foster et al., 2021], 2.[Chen and Jiang, 2019]

(Our result) What we know now!

Theorem

Theorem [This work]: For any $\epsilon > 0$, with linear q^π-realizability and access to offline trajectory data (satisfying *concentrability*) of size $n = \text{poly}(1/\epsilon, H, d, C)$, our algorithm outputs a policy π such that:

$$v^{\pi^*}(s_1) - v^{\pi}(s_1) \leq \epsilon$$

Overview

• (Setting)	What is the problem?
• (Related works)	What did we know?
• (Our result)	What we know now!
• (Our method)	How we know it…
• (Future work)	What's next?

(Our method) How we know it...

Our Algorithm (roughly):

Modify the linearly q^{π}-realizable MDP to be a linear MDP

Run an algorithm that works in linear MDPs

Overview

- (Setting) What is the problem?
- (Related works) What did we know?
- (Our result) What we know now!
- (Our method) How we know it…
- (Future work) What's next?

(Future work) What's next?

Our algorithm isn't computationally efficient (not $\mathrm{poly}(1/\epsilon, H, d, C)$)

Open problem: Can the problem be solved computationally efficiently?

(Future work) What's next?

Our algorithm isn't computationally efficient (not poly$(1/\epsilon, H, d, C)$)

Open problem: Can the problem be solved computationally efficiently?

We require $n = \tilde{\Omega}\left(C^4 H^7 d^4 / \epsilon^2\right)$

Open problem: What is the best possible n?

Thank You :)

References

J. Chen and N. Jiang. Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning*, pages 1042–1051. PMLR, 2019.

D. J. Foster, A. Krishnamurthy, D. Simchi-Levi, and Y. Xu. Offline reinforcement learning: Fundamental barriers for value function approximation. *arXiv preprint arXiv*:2111.10919, 2021.

G. Weisz, A. György, and C. Szepesvári. Online rl in linearly qpi-realizable mdps is as easy as in 368 linear mdps if you learn what to ignore. *arXiv preprint arXiv*:2310.07811, 2023.