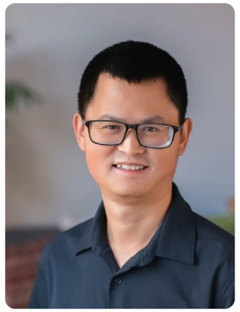
NEURAL INFORMATION PROCESSING SYSTEMS 2025

Computational Hardness of Reinforcement Learning with Partial q^{π}-realizability



Shayan Karimi Xiaoqi Tan

UNIVERSITY OF ALBERTA

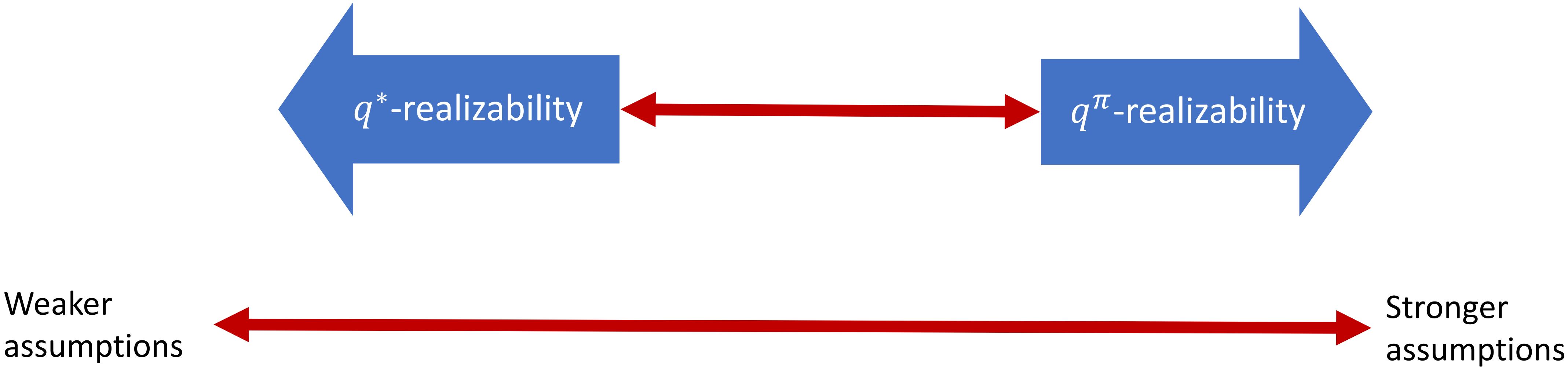
SODA LAB

amii

Overview of Research Questions and Main Results

Problems of interest: **Reinforcement learning** + **Linear function approximation**

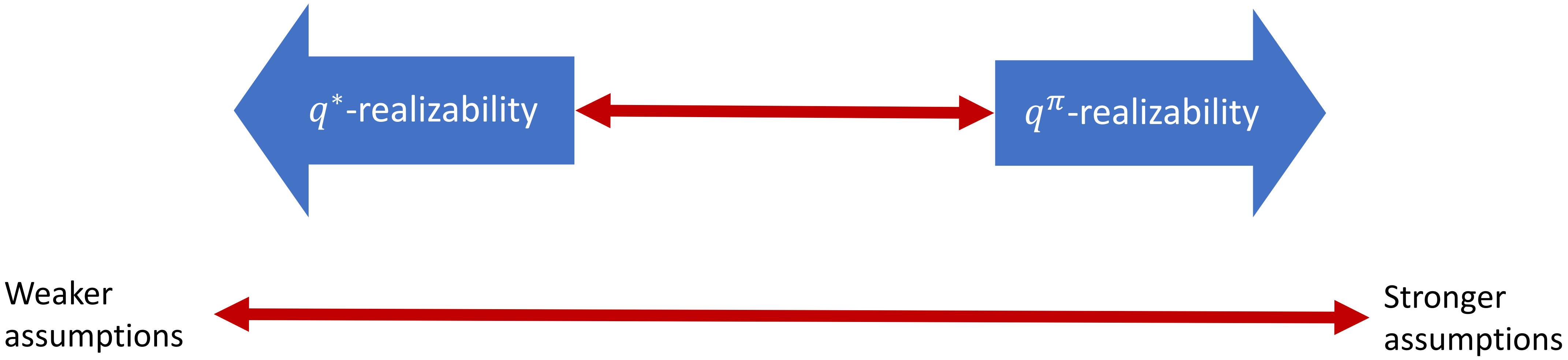
Two common problem settings: **q^*, q^π-realizability settings**

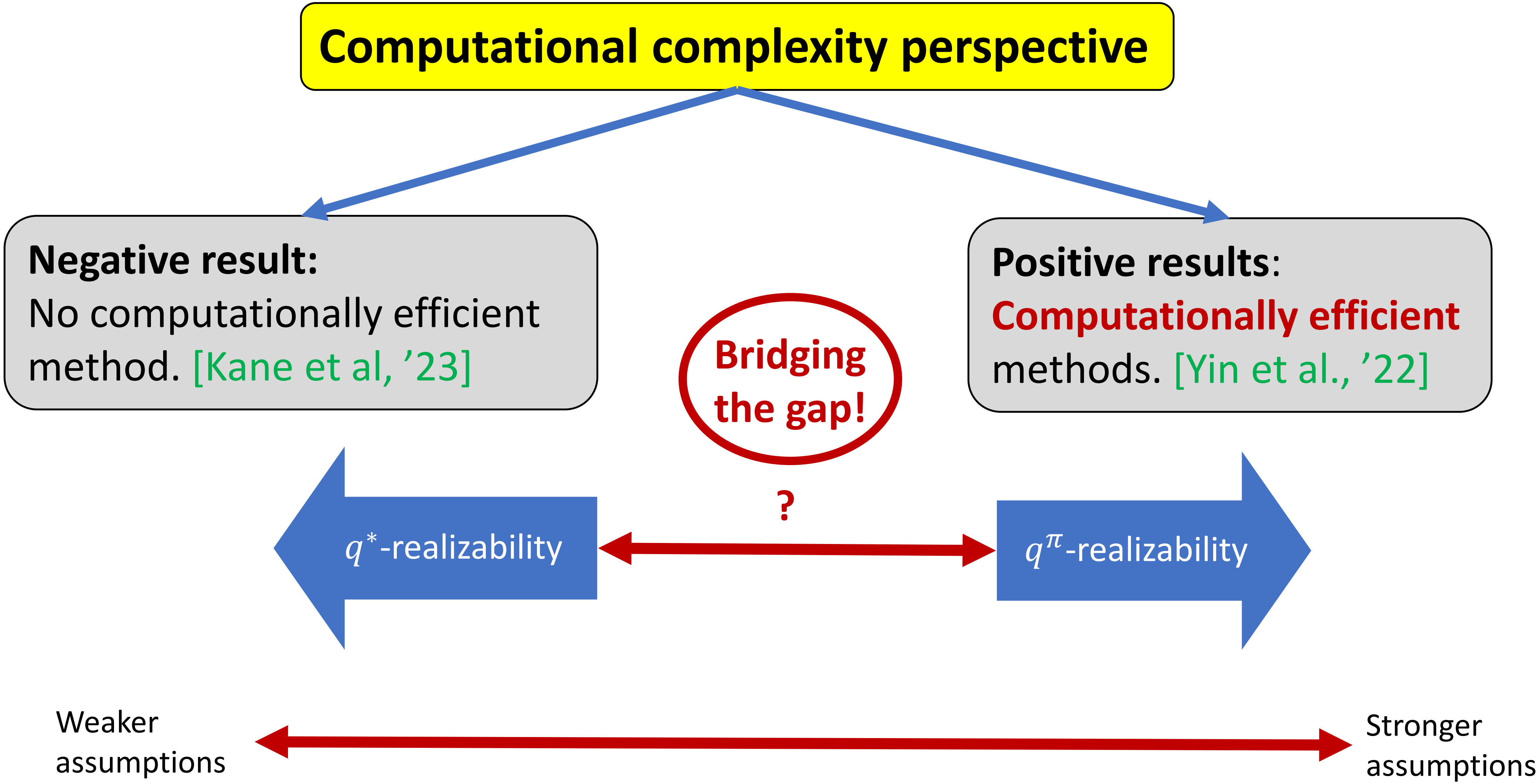


Overview of Research Questions and Main Results

Linear realizability condition holds under **optimal policy**

Linear realizability condition holds under **all policies**

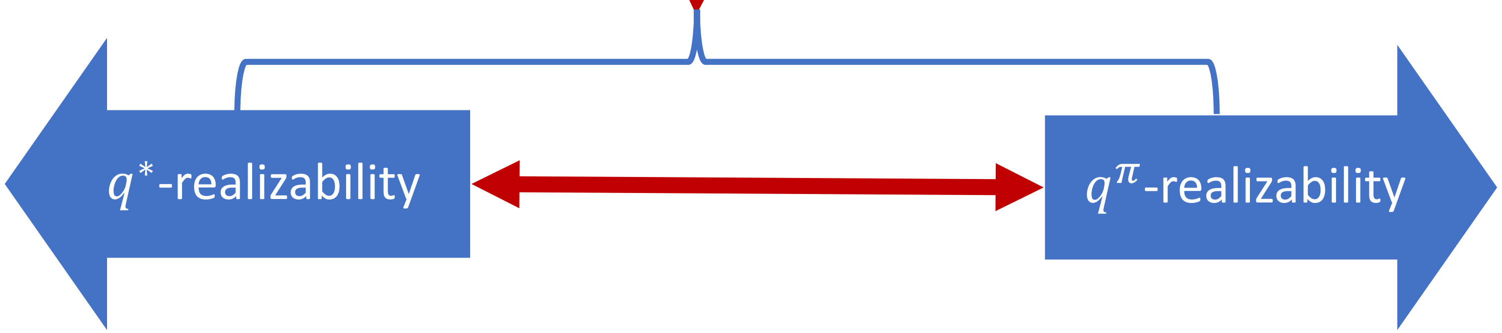



Computational complexity perspective
Negative result:
No computationally efficient method. [Kane et al, ’23]
Bridging the gap!
Positive results:
Computationally efficient methods. [Yin et al., ’22]
q^*-realizability
?
q^π-realizability
Weaker assumptions
Stronger assumptions

Question: What is the computational complexity of solving RL problems under $\boxed{\text{Partial } q^{\pi} \text{ - realizability}}$?

Negative result:
No computationally efficient method [Kane et al, ’23]

Positive results:
Computationally efficient methods. [Yin et al., ’22]



Solving this new problem is computationally hard.

Outline

- Introduction
- Problem Setting
- Main Results
- Proof Sketch
- Conclusions & Future Directions

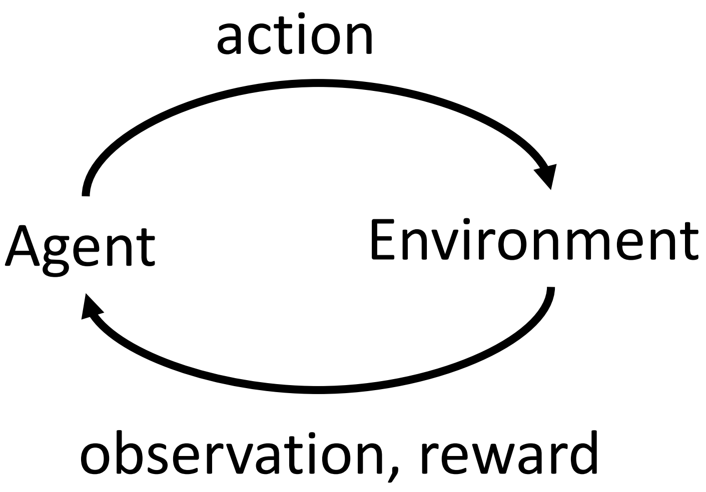
Outline

- **Introduction**
- Problem Setting
- Main Results
- Proof Sketch
- Conclusions & Future Directions

Reinforcement Learning: MDP

- In RL, agent interacts with environment in sequential manner. (here, Interactions happen in episodic way)

- Environment: A **Markov Decision Process** is defined as a tuple (S, A, P, R, H), where :
  - $\boldsymbol{S}$: state space
  - $\boldsymbol{A}$: action space
  - $\boldsymbol{P}: S \times A \rightarrow \Delta(S)$ - transition dynamics
  - $\boldsymbol{R}: S \times A \rightarrow \Delta([0,1])$ - reward function
  - $\boldsymbol{H}$: horizon (length of episode)



- Agent: using policy π to interact with MDP
  - $\pi: S \rightarrow \Delta(A)$ (stochastic policy), and in case of deterministic policies $\pi: S \rightarrow \mathrm{A}$.

Reinforcement Learning: Objective

- **Value functions**: Expected sum of rewards.
  - $v^{\pi}(s) := \mathbb{E}_{\pi,s}\left[\sum_{h=1}^{H} R_t \,\middle|\, s_1 = s\right]$,
  - $q^{\pi}(s,a) := \mathbb{E}_{\pi,s,a}\left[\sum_{h=1}^{H} R_t \,\middle|\, s_1 = s, a_1 = a\right]$,
- **Objective of the learner:** finding a policy π that **maximizes** the expected cumulative reward starting from initial state s_1:

$$\pi^* = argmax_{\pi}\, v^{\pi}(s_1)$$

- How does the agent **interact** with the MDP?
  - **Generative model**: learner can query a **simulator** with any $(s,a) \in (S \times A)$ to obtain a sample (s', R), where $s' \sim P(s,a)$ and $R \sim R(s,a)$.

Linear Function Approximation in RL

- **Linear Function Approximation (LFA):** Representing the **value functions** with **linear combination of feature vector and weight vector**.

Definition (q^π-realizable MDP): MDP M is called q^π-realizable, if there exists $\theta_h \in \mathbb{R}^d$ for any $h \in [\mathrm{H}]$, such that $\boxed{\forall \pi}$:

$$q_h^\pi(s,a) = \langle \phi(s,a), \theta_h \rangle, \qquad \forall s \in \mathrm{S} \text{ and } \forall a \in \mathrm{A}$$

Definition (q^*-realizable MDP): MDP M is called q^*-realizable, if there exists $\theta_h \in \mathbb{R}^d$ for any $h \in [\mathrm{H}]$, such that for $\boxed{\text{optimal policy } \pi^*}$:

$$q_h^*(s,a) = \langle \phi(s,a), \theta_h \rangle, \qquad \forall s \in \mathrm{S} \text{ and } \forall a \in \mathrm{A}$$

Outline

- Introduction
- **Problem Setting**
- Main Results
- Proof Sketch
- Conclusions & Future Directions

Definition of Partial q^{π}-realizability

- **Partial q^{π}-realizability:** Given a policy set Π and a feature vector $\phi: S \times A \rightarrow \mathbb{R}^{d}$, an MDP is said to be partially q^{π}-realizable under Π if, for all $\pi \in \Pi$, there exists $\theta_h \in \mathbb{R}^{d}$ such that:

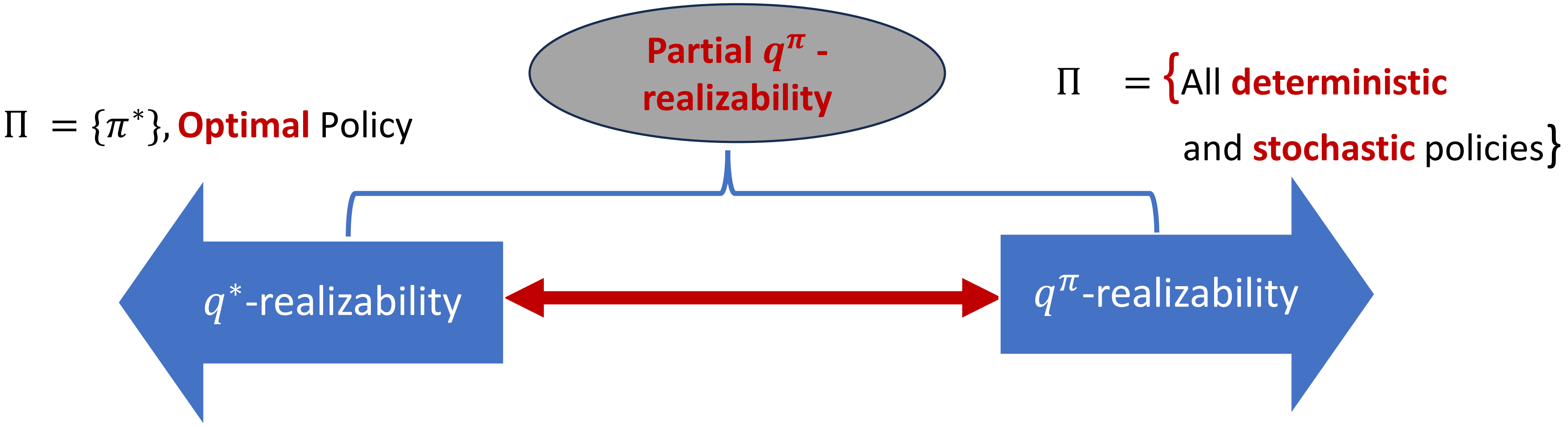
$$q_h^{\pi}(s_h, a_h) = \langle \phi(s_h, a_h), \theta_h \rangle \text{ for all } (s_h, a_h) \in S_h \times A$$

Objective under Partial q^{π}-realizability Setting

$\pi \notin \Pi$ may occur.

Objective: Given an initial state $s_1 \in S$, we want to learn a policy $\boxed{\pi}$ which is ϵ-optimal with respect to the best policy in Π with high probability:

$$V^{\pi}(s_1) \geq max_{\overline{\pi} \in \Pi} V^{\overline{\pi}}(s_1) \text{ - } \varepsilon$$



Question 1: Can we break the **hardness result** [Kane et al, '23] for the $\boldsymbol{q^*}$-**realizable** setting when considering a policy class Π with $\{\pi^*\} \subsetneq \Pi$?

Question 2: Can we still achieve **positive results** [Yin et al., '22] in the $\boldsymbol{q^{\pi}}$-**realizable** setting with a **restricted policy class $\boldsymbol{\Pi}$**?

Need for Well-defined Policy Sets

To have a well-defined RL problem under **partial q^{π}-realizability**, we need to specify the policy set Π.

- **Partial q^{π}-realizability:** Given a policy set $\boxed{\Pi}$ and a feature vector $\phi : S \times A \rightarrow \mathbb{R}^{d}$, an MDP is said to be partially q^{π}-realizable under Π if, for all $\pi \in \Pi$, there exists $\theta_h \in \mathbb{R}^{d}$ such that:

$$q_h^{\pi}(s_h, a_h) = \langle \phi(s_h, a_h), \theta_h \rangle \text{ for all } (s_h, a_h) \in S_h \times A$$

Our Focus: Two Policy Sets

Greedy policy set (Π^{g}) and **Softmax policy set (Π^{sm})**

- **Partial q^π-realizability:** Given a policy set $\boxed{\Pi}$ and a feature vector $\phi : S \times A \to \mathbb{R}^d$, an MDP is said to be partially q^π-realizable under Π if, for all $\pi \in \Pi$, there exists $\theta_h \in \mathbb{R}^d$ such that:

$$q_h^\pi(s_h, a_h) = \langle \phi(s_h, a_h), \theta_h \rangle \text{ for all } (s_h, a_h) \in S_h \times A$$

Greedy Policy Set

Greedy Policy: Let $\phi' : S \times A \to \mathbb{R}^{d'}$ be a feature vector with dimension $d' \in \mathbb{N}$. For any $h \in [H]$ and $\theta' \in \mathbb{R}^{d'}$, let $\pi_{\theta'} : S_h \to A$ be defined as follows:

$$\pi_{\theta'}(s_h) := argmax_{a \in A} \langle \phi'(s_h, a), \theta' \rangle \qquad \text{for all} \quad s_h \in S_h$$

Greedy Policy Set:

$$\Pi^g := \{\pi_{\theta'} | \theta' \in \mathbb{R}^{d'}\}$$

Remark: $\phi' \in \mathbb{R}^{d'}$ and $\phi \in \mathbb{R}^{d}$, where $\phi \neq \phi'$.

Softmax Policy Set

Softmax Policy: Let $\phi' : S \times A \rightarrow \mathbb{R}^{d'}$ be a feature vector with dimension d' $\in \mathbb{N}$. For any $h \in [H]$ and $\theta' \in \mathbb{R}^{d'}$, let $\pi_{\theta'}: S_h \rightarrow \Delta(A)$ be defined as follows:

$$\pi_{\theta'}(a|s_h) = \frac{e^{\phi'(s_h,a)^T \theta'}}{\sum_{i=1}^{\kappa} e^{\phi'(s_h,a_i)^T \theta'}} \qquad \forall s_h \in S_h$$

Softmax Policy Set: $\Pi^{sm} \coloneqq \{\pi_{\theta'} | \theta' \in \mathbb{R}^{d'}\}$

Outline

- Introduction
- Problem Setting
- **Main Results**
- Proof Sketch
- Conclusions & Future Directions

Theorem 1 (NP-hardness under greedy policy set; informal):

Consider a partially q^{π}-realizable instance with $\pi \in \Pi^{g}$. Then, for some specific constant ϵ (with ϵ sufficiently small), no polynomial-time algorithm can compute an ϵ-optimal policy, unless P=NP.

Theorem 2 (Hardness under softmax policy set; informal):

Consider a partially q^{π}-realizable instance with $\pi \in \Pi^{sm}$. Then, for some specific constant ϵ (with ϵ sufficiently small), no randomized subexponential time algorithm with low error probability can compute an ϵ-optimal policy, under rETH.

Question 1: Can we break the hardness result [Kane et al, '23] for the q^{*}-realizable setting when considering a policy class Π with $\{\pi^{*}\} \subsetneq \Pi$?

Question 2: Can we still achieve positive results [Yin et al., '22] (in terms of computational efficiency) in the q^{π}-realizable setting with a restricted policy class Π?

Outline

- Introduction
- Problem Setting
- Main Results
- **Proof Sketch**
- Conclusions & Future Directions

Overview of the Proof

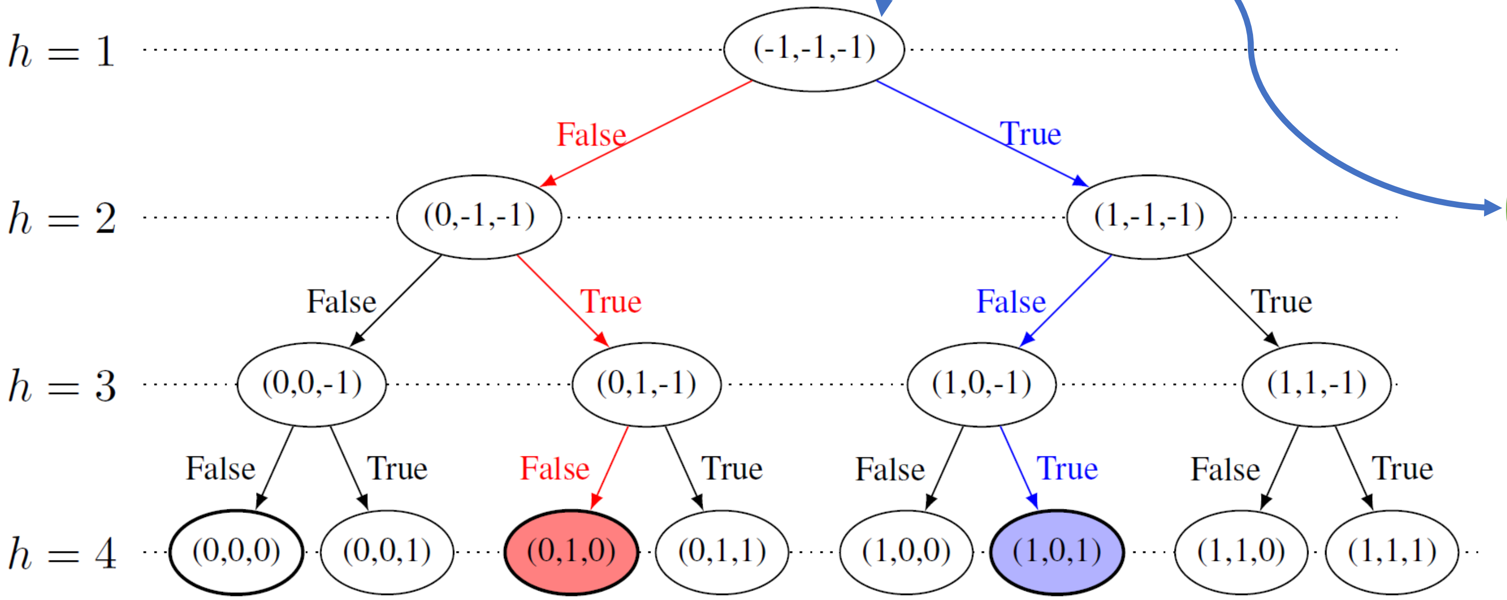
- **Deterministic reduction**: δ-MAX-3SAT $\leq_p$ Greedy-Linear-2-RL .
- Two main steps for proving hardness result for our complexity problems:

Step 1 (Polynomial Transformation):
Design an MDP instance under partial q^π-realizability (**Greedy-Linear-2-RL**) from a given complexity problem (**δ-MAX-3SAT**) in polynomial time.

Step 2 (Algorithmic connection):
Show that the method for solving our RL problem (**Greedy-Linear-2-RL**) can be used for solving the NP-hard problem instance (**δ-MAX-3SAT**).

Example of MDP Instance Design

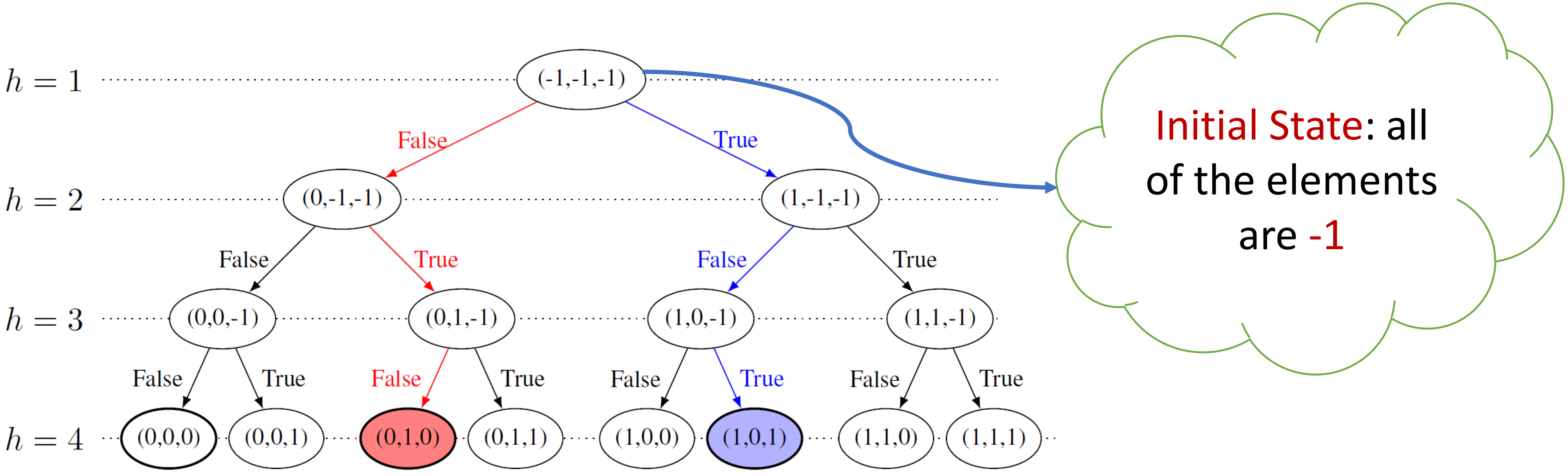
- Given δ-MAX-3SAT instance $\boldsymbol{\varphi}$: $\boxed{(\boldsymbol{x_1} \vee \neg \boldsymbol{x_2} \vee \boldsymbol{x_3})} \wedge (\neg \boldsymbol{x_1} \vee \boldsymbol{x_2} \vee \neg \boldsymbol{x_3})$, our MDP is as follows:



We have 3 variables, and state is a tuple with three elements

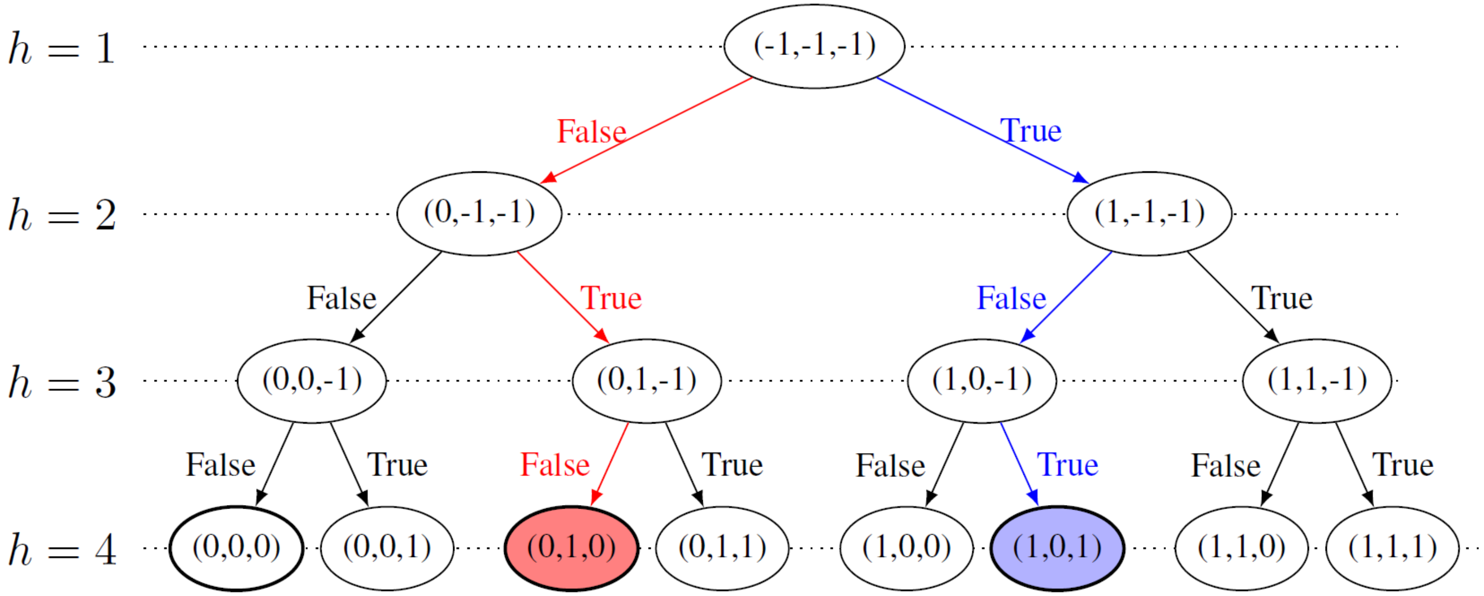
Example of MDP Instance Design

- Given δ-MAX-3SAT instance $\boldsymbol{\varphi}$: $(\boldsymbol{x_1} \vee \neg \boldsymbol{x_2} \vee \boldsymbol{x_3}) \wedge (\neg \boldsymbol{x_1} \vee \boldsymbol{x_2} \vee \neg \boldsymbol{x_3})$, our MDP is as follows:



Example of MDP Instance Design

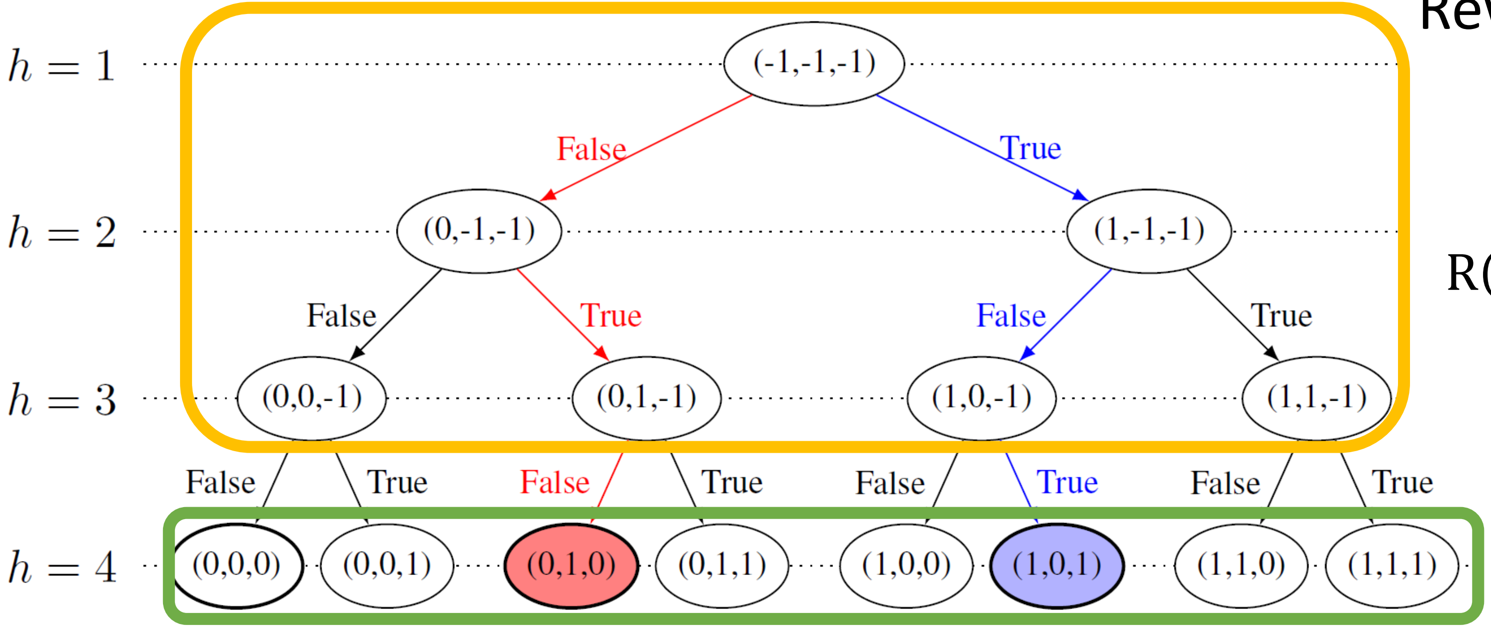
- Given δ-MAX-3SAT instance $\boldsymbol{\varphi}$: $(\boldsymbol{x_1} \lor \neg\boldsymbol{x_2} \lor \boldsymbol{x_3}) \land (\neg\boldsymbol{x_1} \lor \boldsymbol{x_2} \lor \neg\boldsymbol{x_3})$, our MDP is as follows:



Two actions:
True or **False**

Example of MDP Instance Design

- Given δ-MAX-3SAT instance $\boldsymbol{\varphi}$: $(\boldsymbol{x_1} \vee \neg\boldsymbol{x_2} \vee \boldsymbol{x_3}) \wedge (\neg\boldsymbol{x_1} \vee \boldsymbol{x_2} \vee \neg\boldsymbol{x_3})$, our MDP is as follows:



Reward is **Zero**

$$R(s_h) = \begin{cases} \frac{|C_{true}(s_h)|}{|C|} & \text{if } h = H, \\ 0 & \text{otherwise.} \end{cases}$$

Terminal states:
we get our rewards

Outline

- Introduction
- Problem Setting
- Main Results
- Proof Sketch
- **Conclusions & Future Directions**

Conclusions

- Introducing **partial q^{π}-realizability** setting **which bridges the gap** between **q^{π}-realizability and q^{*}-realizability assumptions**.

- Obtaining **hardness result** for this new problem setting under different **policy sets (greedy policy set Π^{g} and softmax policy set Π^{sm})**.

- Our results show that enlarging the policy class **beyond the optimal policy π^{*}** **does not eliminate the fundamental computational challenges.**

Thanks so much for your attention!