REINFORCEMENT LEARNING: WHAT, WHEN, HOW

NRC-DT Knowledge Exchange Seminar
Dec 4 2024

Abhishek Naik

Formerly:

Now:

OUTLINE

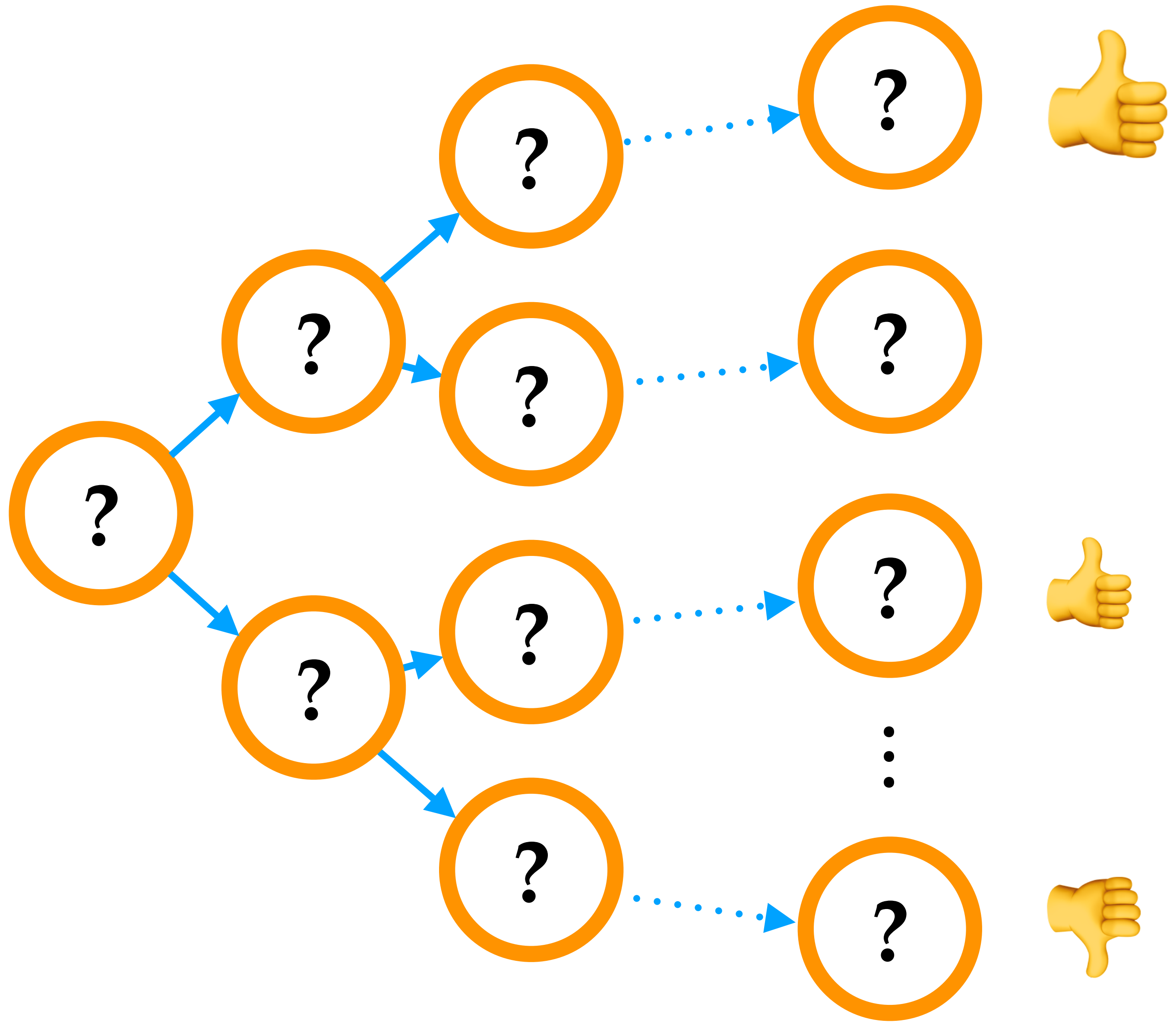
- What is reinforcement learning (RL)?
- When is it applicable?
- What is my focus?
- Should *you* be considering RL?

REINFORCEMENT LEARNING IS A PARADIGM OF LEARNING FROM INTERACTIONS

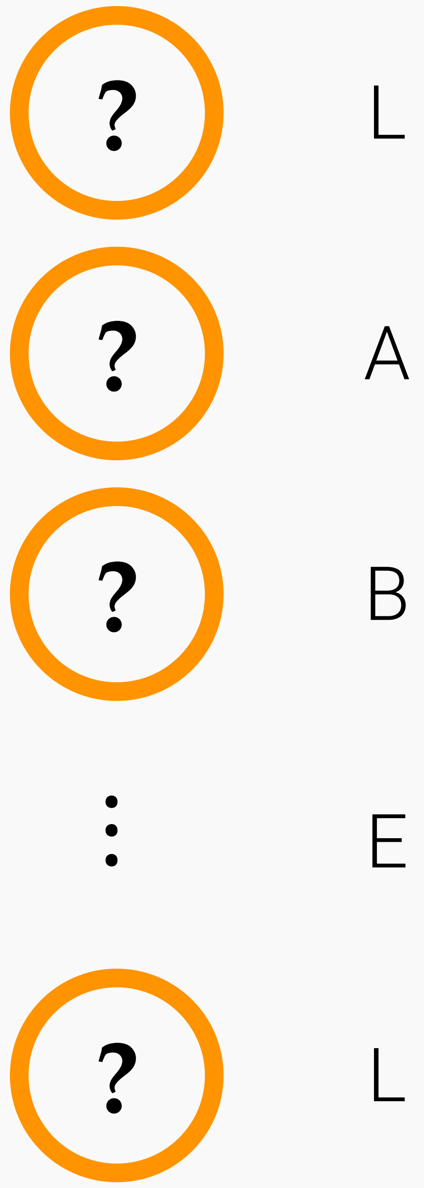
Learning from experience
by trial and error

SOME CHARACTERISTICS OF THE RL FRAMEWORK

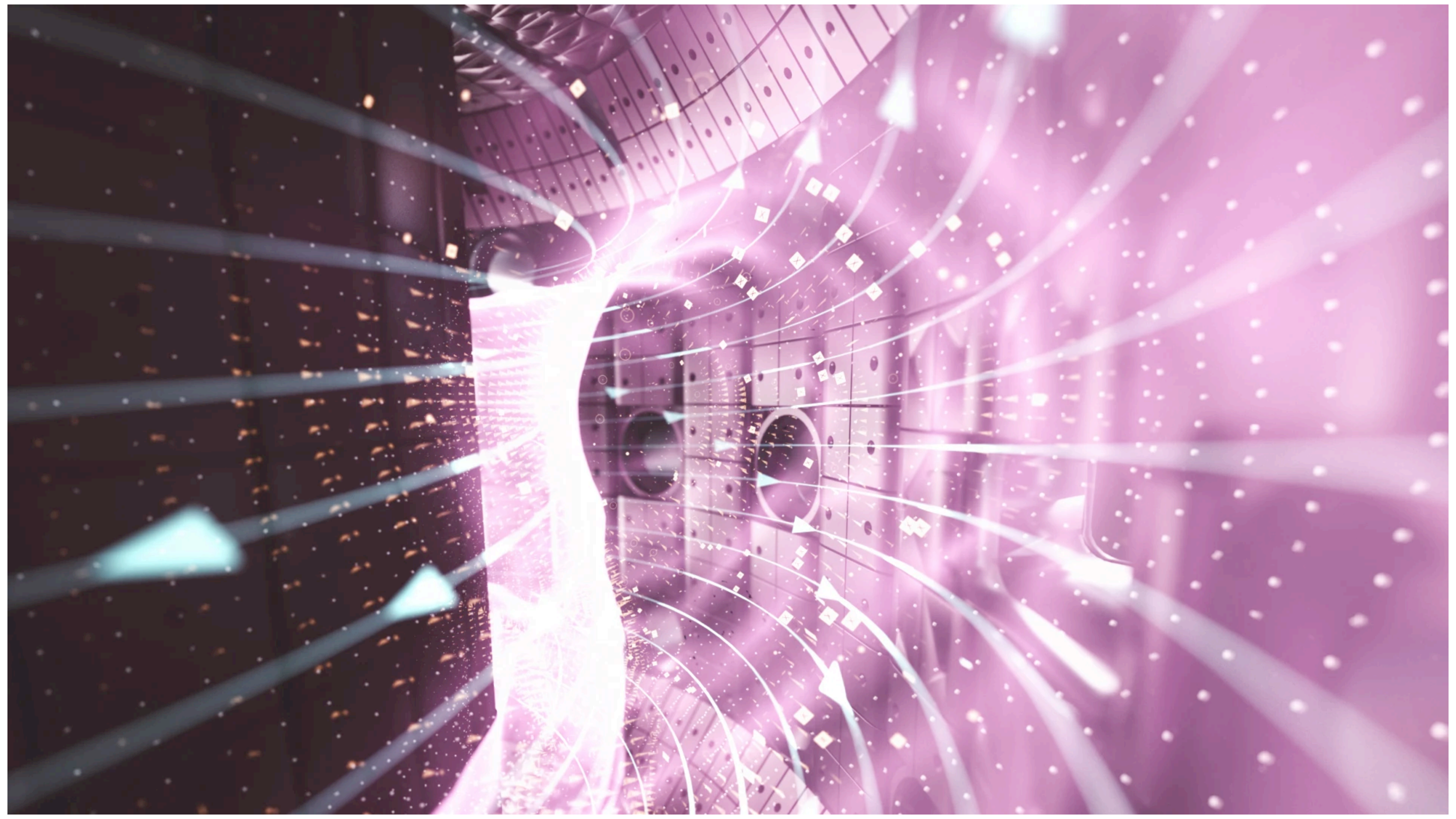
- Sequential decision-making
- Evaluative feedback
- Delayed feedback

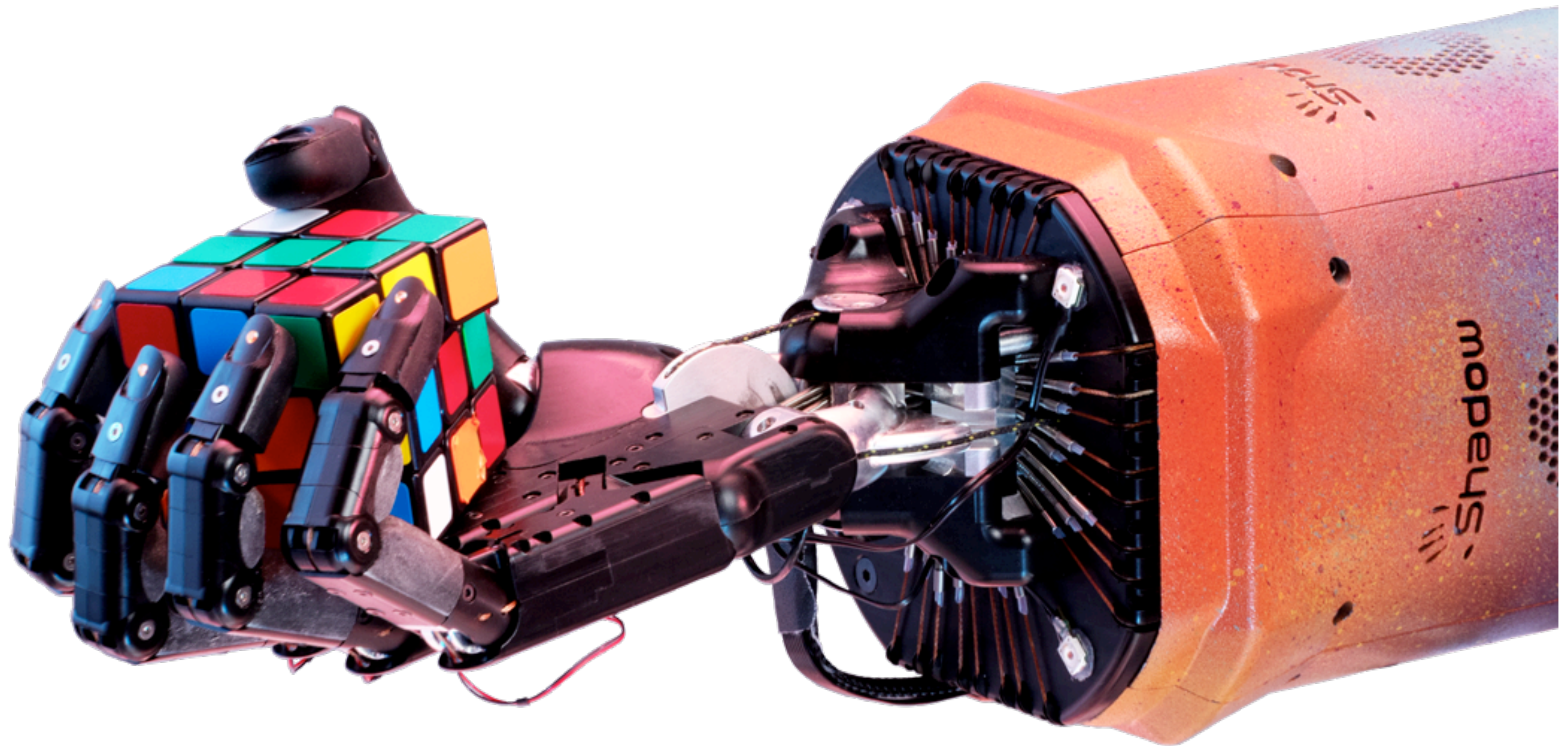


- Independent decisions
- Instructive feedback
- Immediate feedback

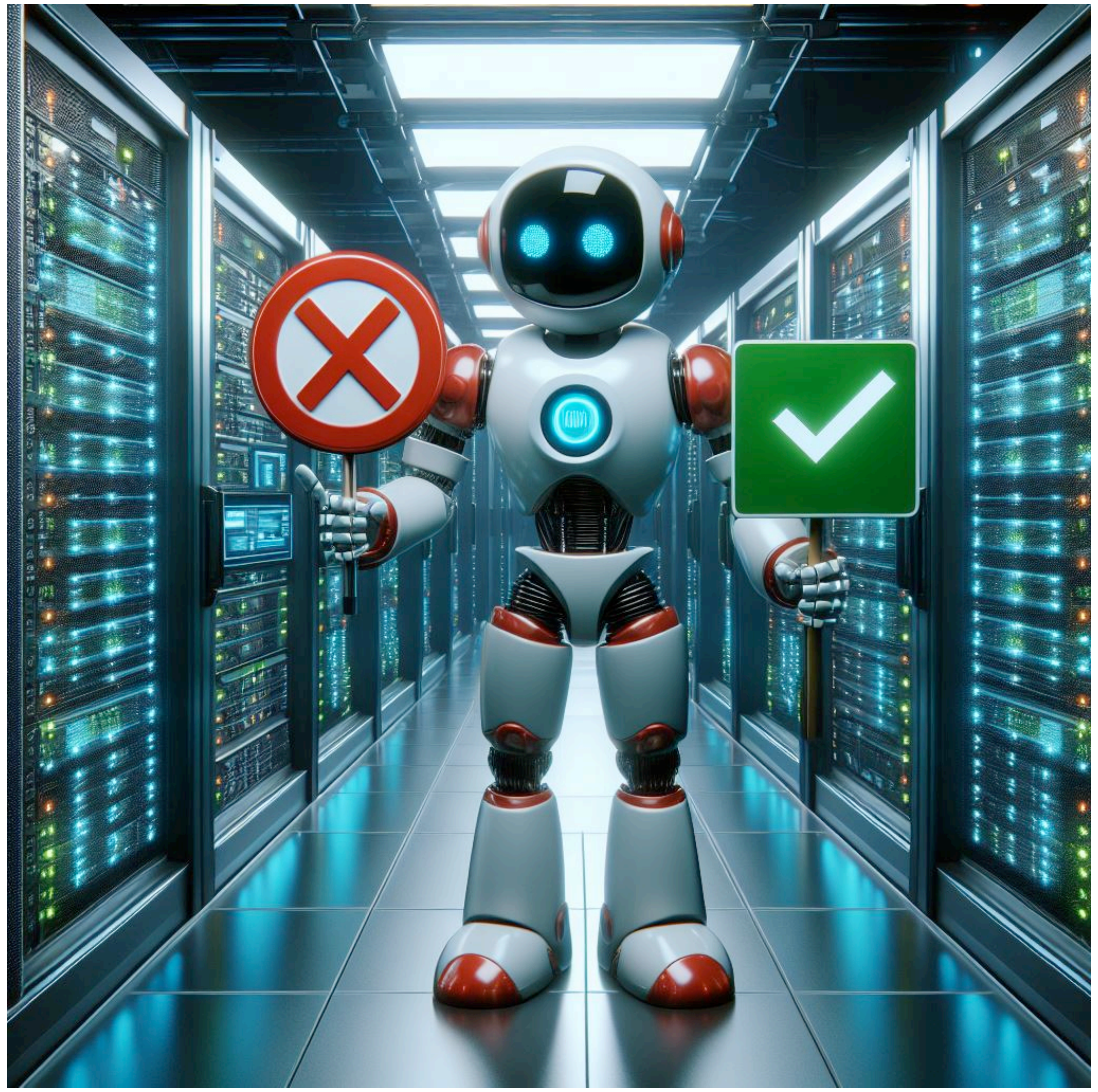


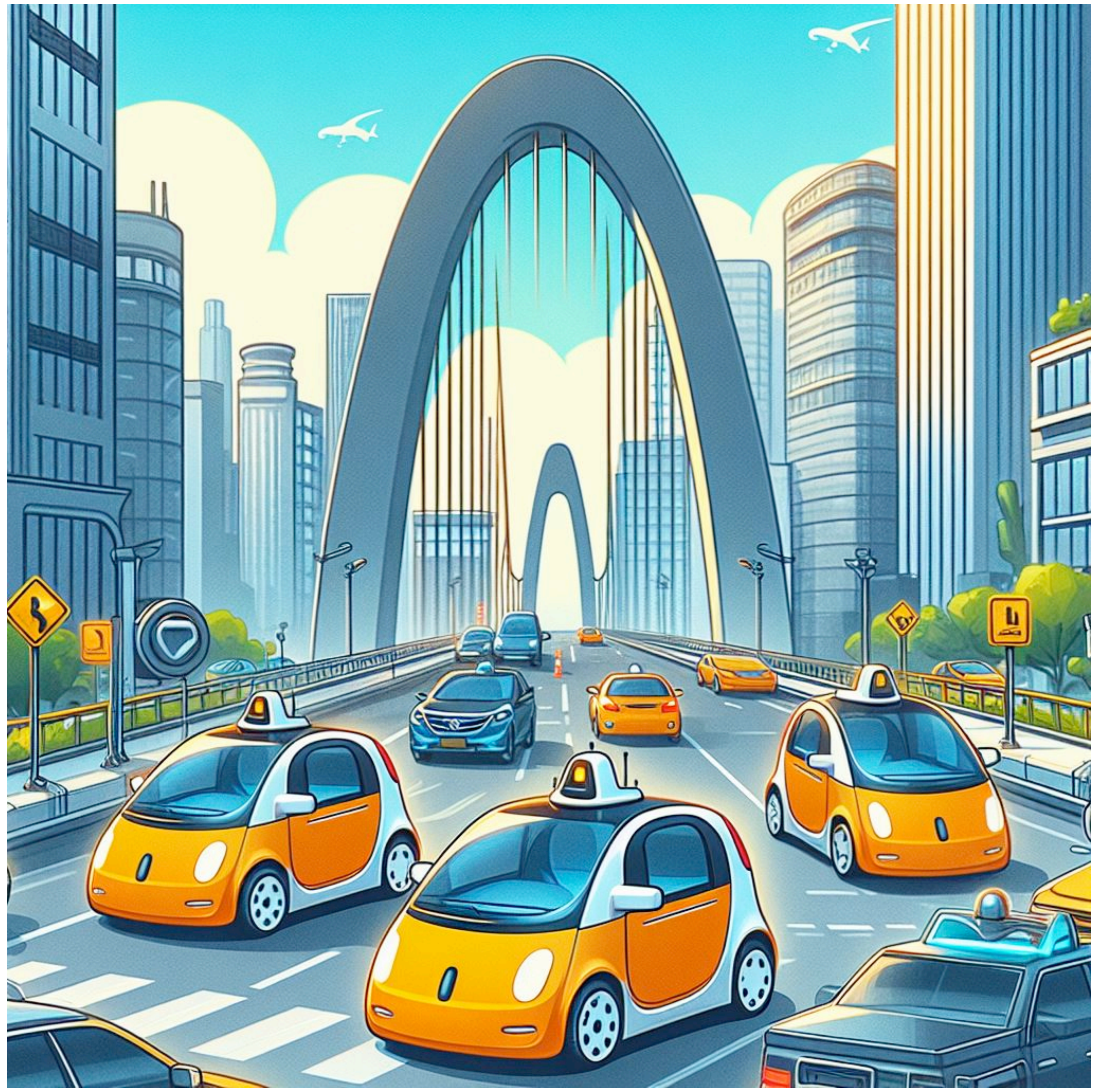
SOME IMPRESSIVE DEMONSTRATIONS OF RL

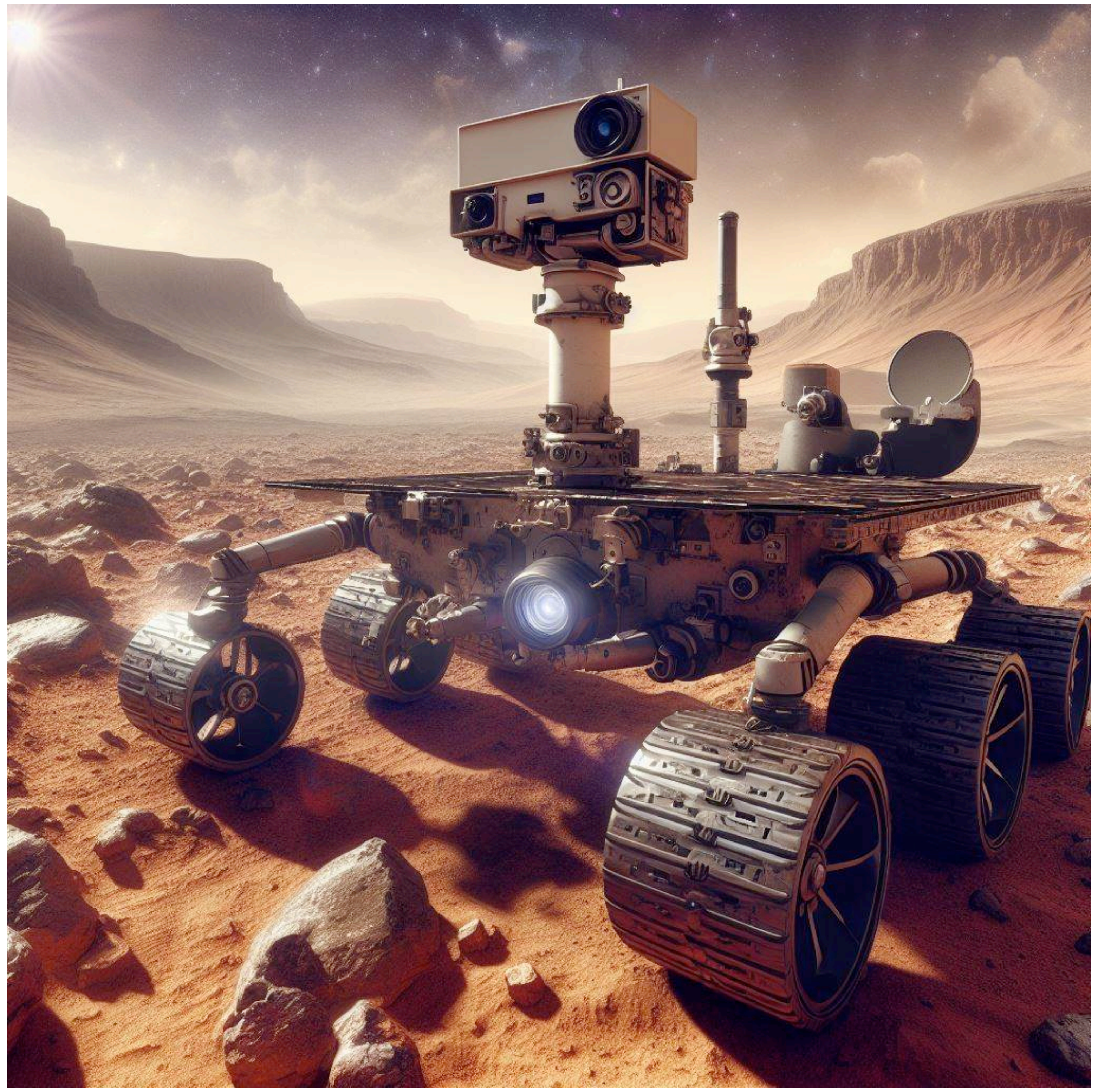




EXAMPLES OF SEQUENTIAL DECISION-MAKING PROBLEMS







Optimal allocation of solar power in a satellite, setting water-filtration-plant parameters, routing of network traffic for dynamic topologies, intelligent recommendation systems, controlling robotic limbs to perform diverse household tasks, controlling deformable mirrors for optical satellite communication, …

OUTLINE

- What is reinforcement learning (RL)?
- When is it applicable?
- What is my focus?
- Should *you* be considering RL?

SIMPLE AND PRACTICAL ALGORITHMS TO LEARN THROUGHOUT AN AGENT'S LIFETIME

- Find the best way to behave given constraints
- Learn continually *not learn-freeze-deploy*
- Learn online and incrementally

Use ideas developed from first principles

REWARD CENTERING

$$S_0 \; A_0 \; R_1 \; S_1 \; A_1, R_2 \ldots \; S_t \; A_t \; R_{t+1} \; S_{t+1} \; A_{t+1} \; R_{t+2} \; \ldots$$

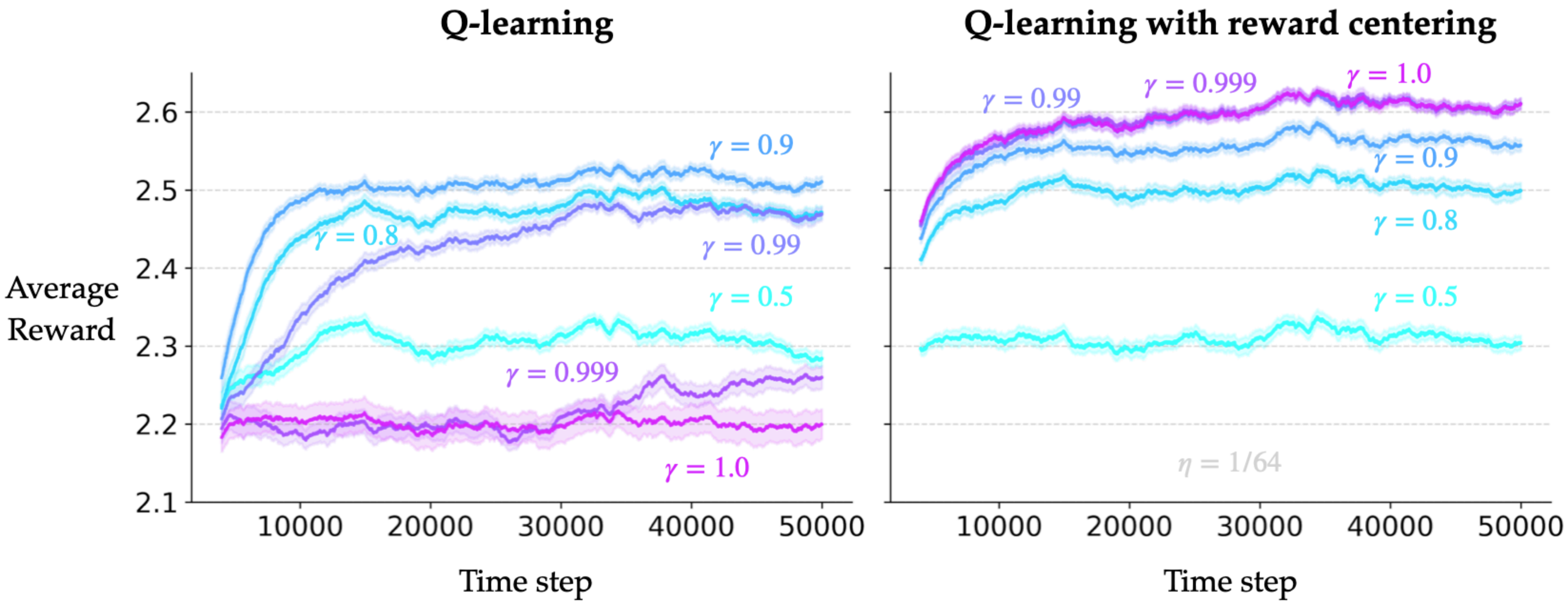
Estimate the average reward and subtract it from the observed rewards

$$Q_{t+1}(S_t, A_t) \doteq Q_t(S_t, A_t) + \alpha_t \left[R_{t+1} + \gamma \max_{a'} Q_t(S_{t+1}, a') - Q_t(S_t, A_t) \right]$$

$$\downarrow$$

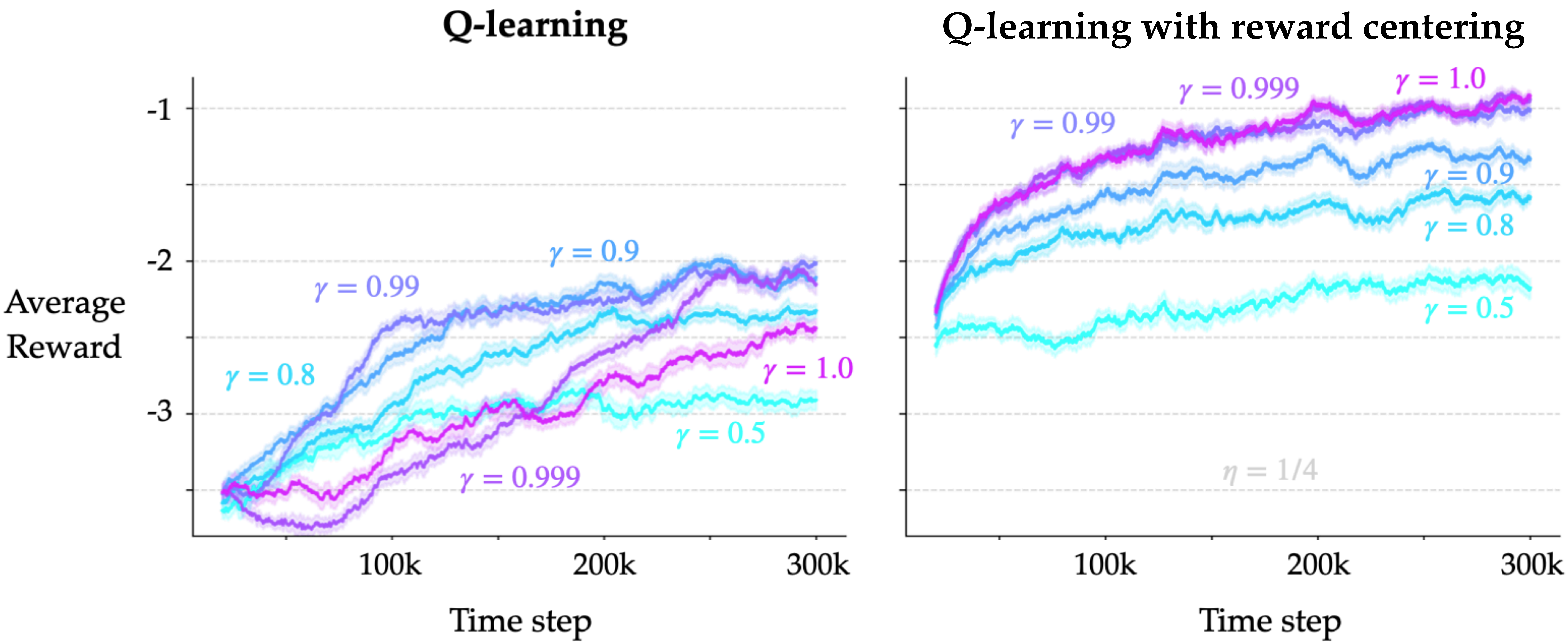
$$Q_{t+1}(S_t, A_t) \doteq Q_t(S_t, A_t) + \alpha_t \left[R_{t+1} - \bar{R}_t + \gamma \max_{a'} Q_t(S_{t+1}, a') - Q_t(S_t, A_t) \right]$$

NO INSTABILITY WITH LARGE DISCOUNT FACTORS



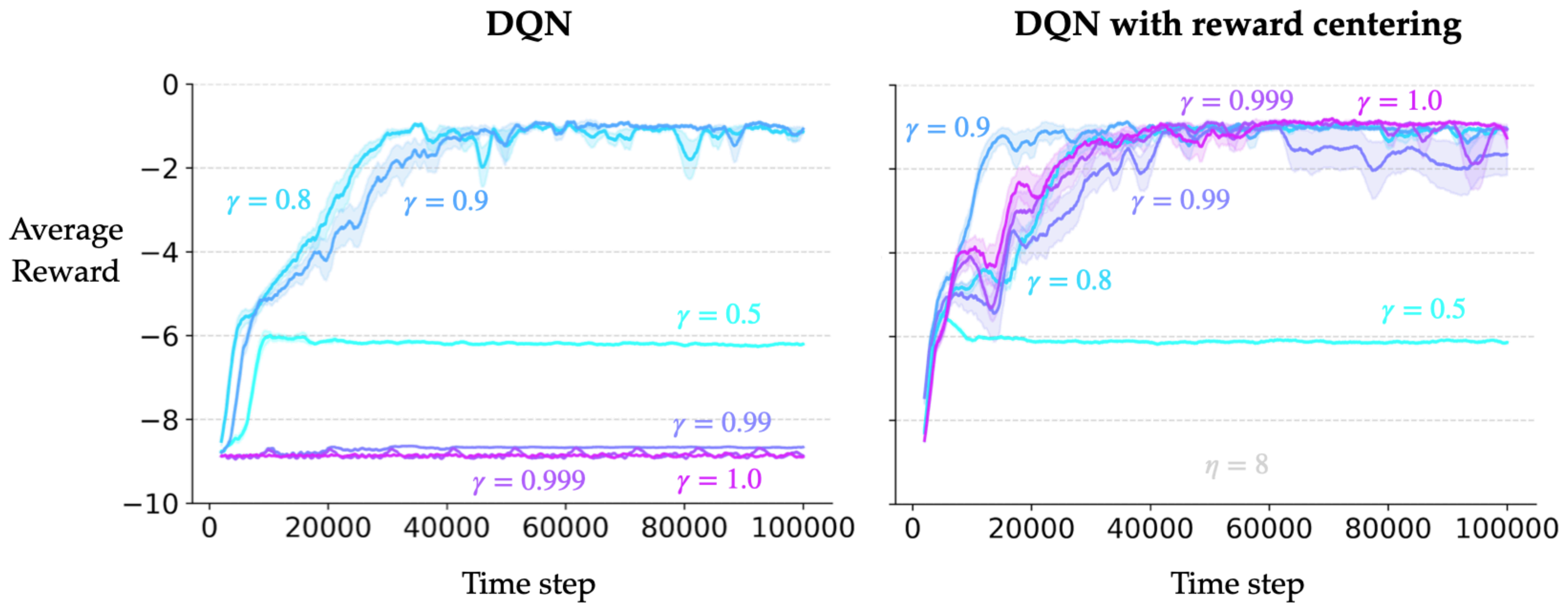
AccessControl (tabular)

NO INSTABILITY WITH LARGE DISCOUNT FACTORS



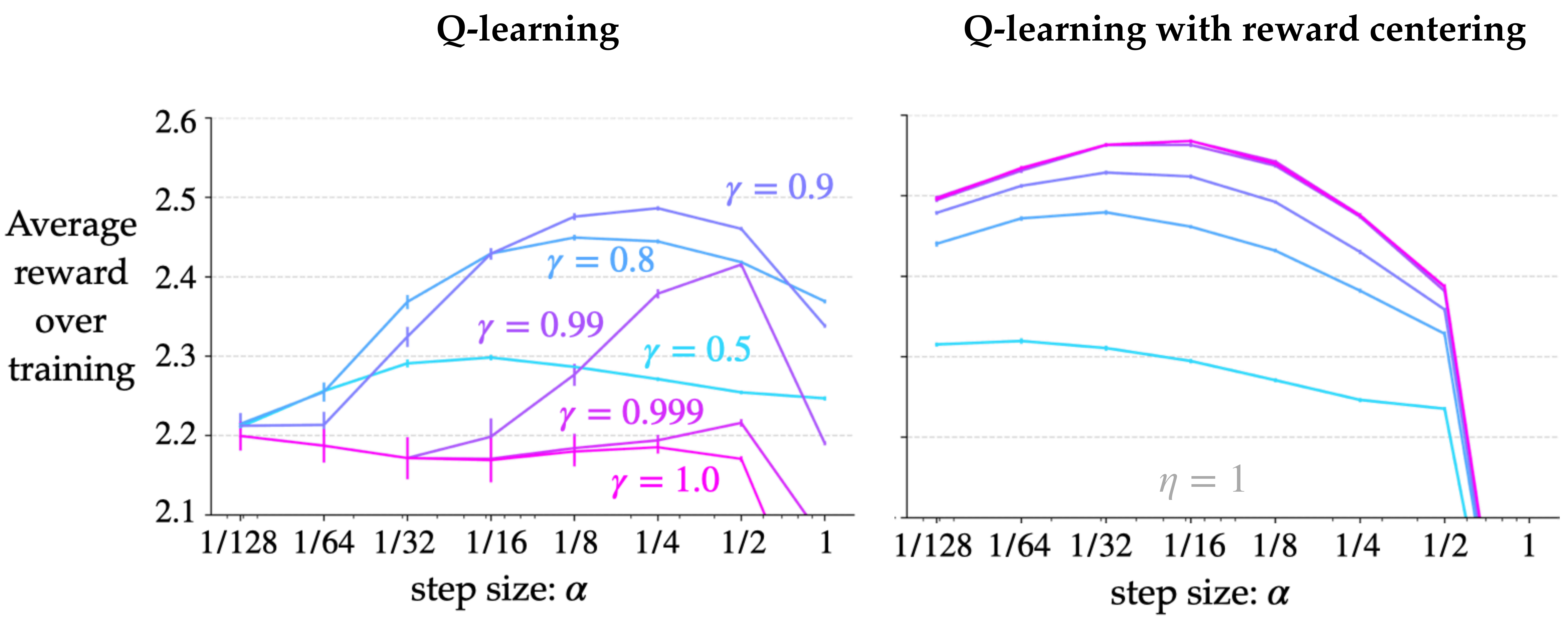
PuckWorld (linear FA)

NO INSTABILITY WITH LARGE DISCOUNT FACTORS



Pendulum (non-linear FA)

TRENDS ARE CONSISTENT ACROSS PARAMETERS



AccessControl (tabular)

THE SPECIAL CASE OF $\gamma = 1$

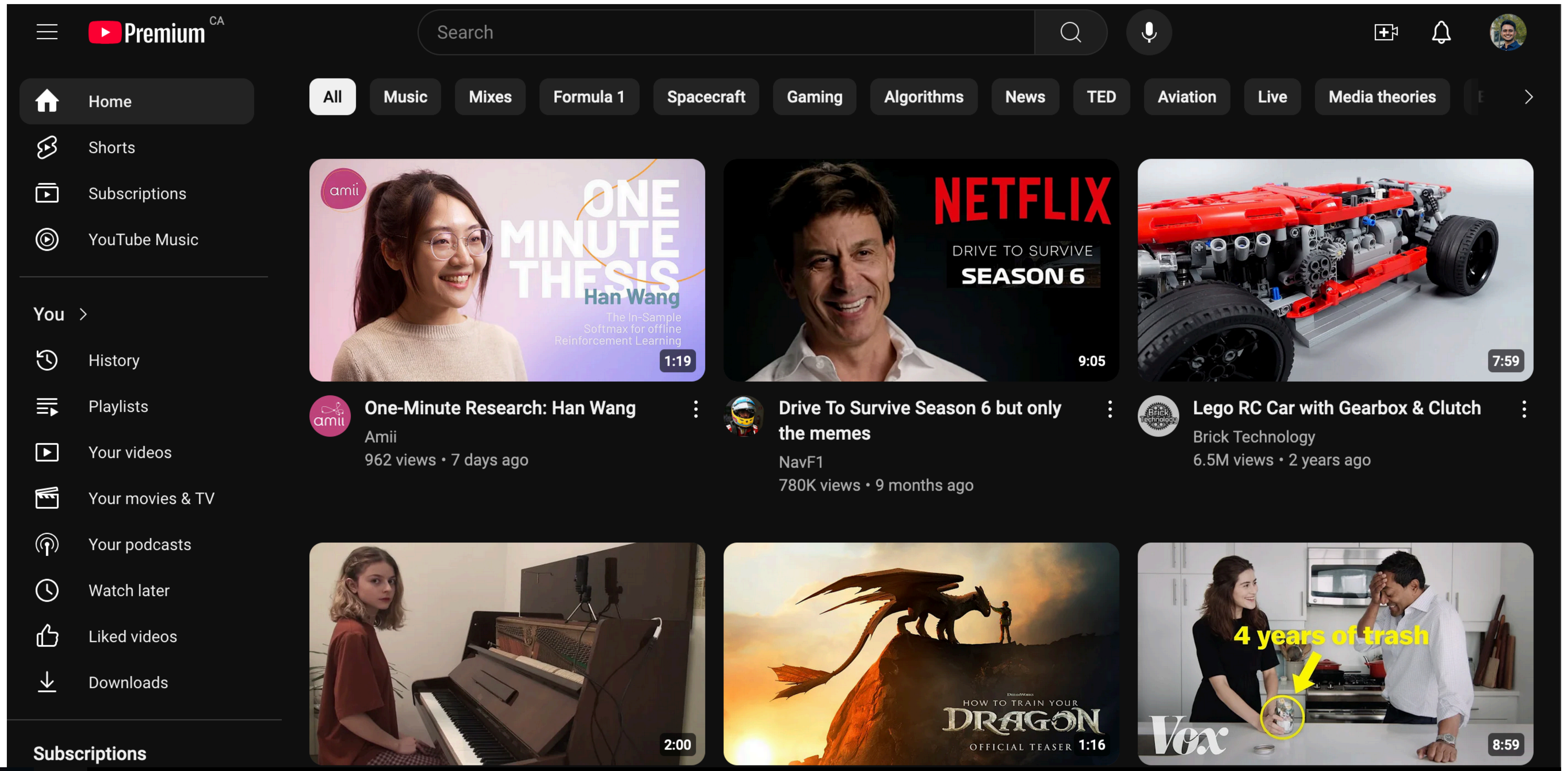
$$S_0\ A_0\ R_1\ S_1\ A_1, R_2 \ldots\ S_t\ A_t\ R_{t+1}\ S_{t+1}\ A_{t+1}\ R_{t+2}\ \ldots$$

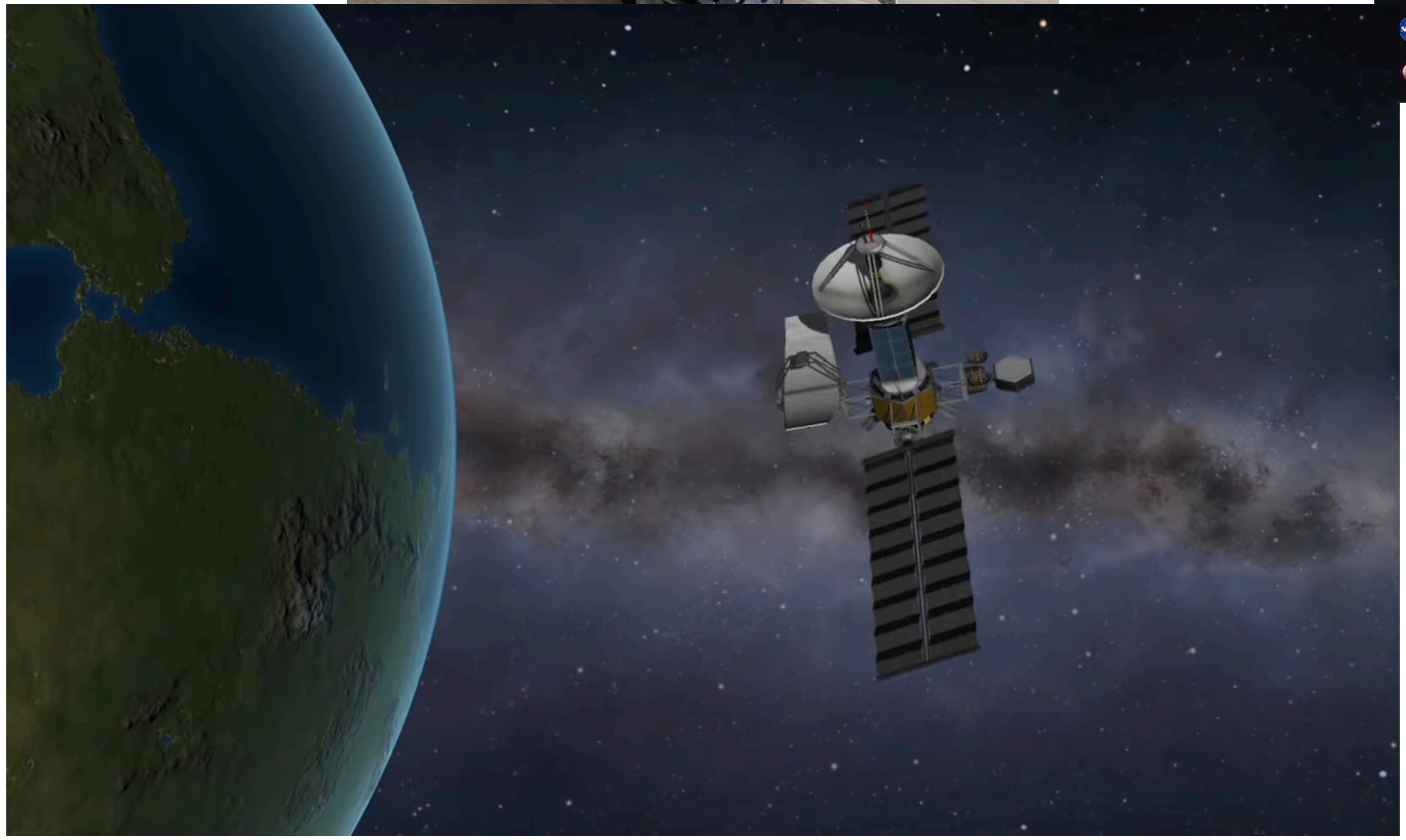
$$\max_{\pi}\ r(\pi)$$

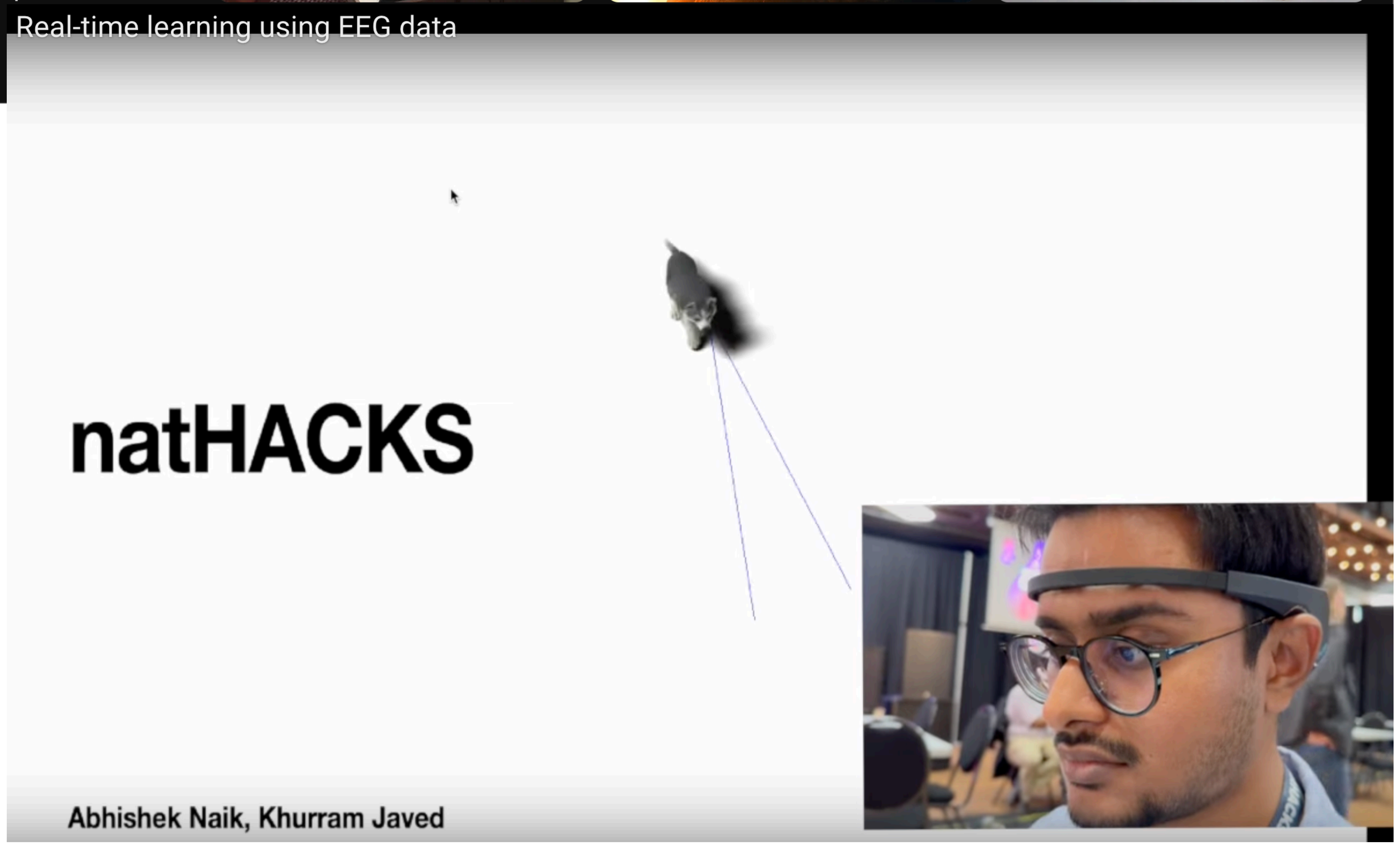
$$r(\pi) \doteq \lim_{n\to\infty} \frac{1}{n} \mathbb{E}_{\pi}\Big[\sum_{t=1}^{n} R_t\Big]$$

- Fundamental one-step average-reward algorithms
  - learning and planning
  - on- and off-policy
  - prediction and control
- More efficient multi-step versions using traces
- All the extensions to the options framework

SOME APPLICATION-ORIENTED PROJECTS







SHOULD YOU BE CONSIDERING RL?

- RL is a framework for sequential decision-making problems
  - Actions can have long-term consequences
  - Feedback is evaluative in nature
  - The agent generates its own data
- RL algorithms enable *learning* the best way to behave, via trial and error

THANK YOU

Questions?

STRETCH SLIDES

TEMPORAL-DIFFERENCE LEARNING: AN ALGORITHM TO MAXIMIZE LONG-TERM REWARD

$$P_{new} = (1 - \alpha)P_{old} + \alpha(P_{correct})$$

$$P_{new} = (1 - \alpha)P_{old} + \alpha(P_{better})$$

$$= P_{old} + \alpha(P_{better} - P_{old})$$

$$V_{new}(s) = V_{old}(s) + \alpha(\underbrace{R + V_{old}(s') - V_{old}(s)}_{\text{TD error}})$$

inspired from psychology and constrained by computation

TD LEARNING BEST FITS VARIOUS PSYCH/NEURO DATA

- explains blocking and higher-order conditioning
- predicted the reversal of blocking — later confirmed by Kehoe et al. (1987)
- experimental support for the reward-prediction-error hypothesis: Schultz et al. (1997)
- causal support using optogenetics: Steinberg et al. (2013)