Course 3, Module 3
Control with Approximation

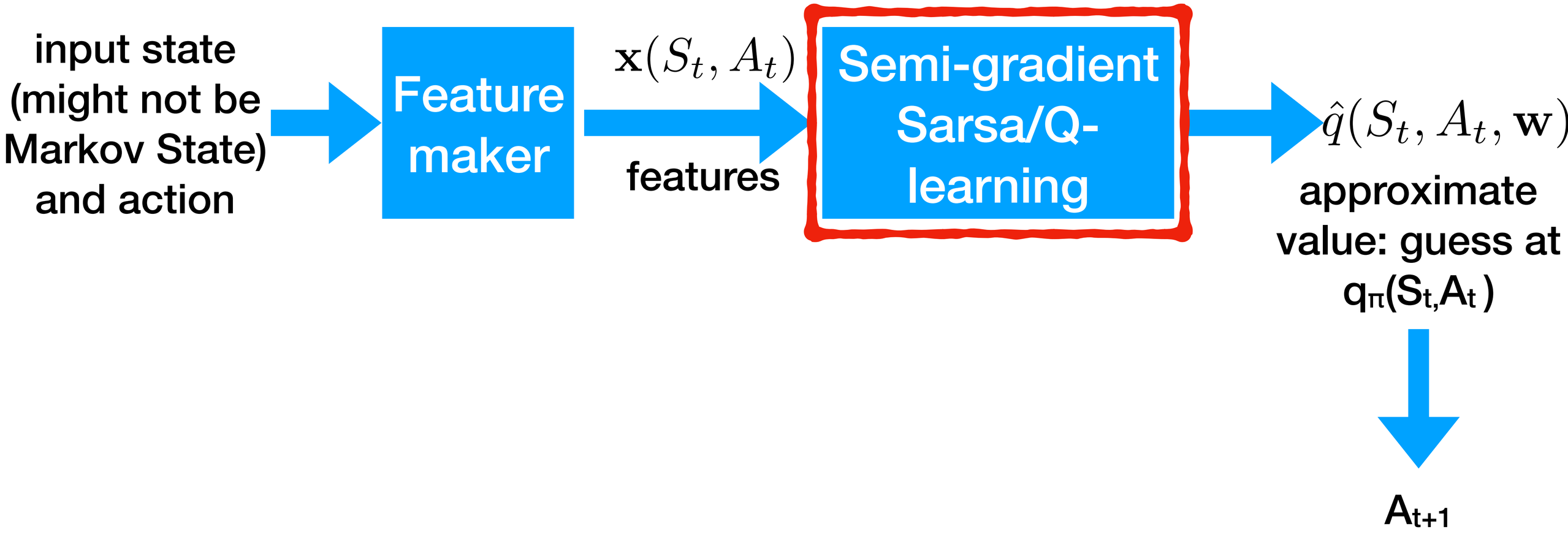
CMPUT 397
Fall 2019

Announcements

- Don't forget the course survey! Important for improving things!
- Capstone project is due this Friday at Midnight
- Either Mountain Car project I specified (in groups of 2 students)
- Or the Capstone Project on Coursera (Lunar Lander)
  - must use private session (link in eclass)
  - **must be done individually**
- Final review on Friday! Practice final released today (end of day)

- Link for questions:
  - **http://www.tricider.com/brainstorming/2q1MpvEWEG7**

Review of Course 3, Module 3
How to do control (learn a good policy) with function approximation


input state (might not be Markov State) and action
Feature maker
$\mathbf{x}(S_t, A_t)$
features
Semi-gradient Sarsa/Q-learning
$\hat{q}(S_t, A_t, \mathbf{w})$
approximate value: guess at $q_\pi(S_t, A_t)$
A_{t+1}

Video 1: Episodic Sarsa with Function Approximation

- We know how to do function approximation with TD; how about using that to learn action-values and a policy. **On-policy TD control** with approximation

- Goals:

  - Understand how to construct **action-dependent features** for approximate action-values >> stacking

  - and explain how to use Sarsa in episodic tasks with function approximation

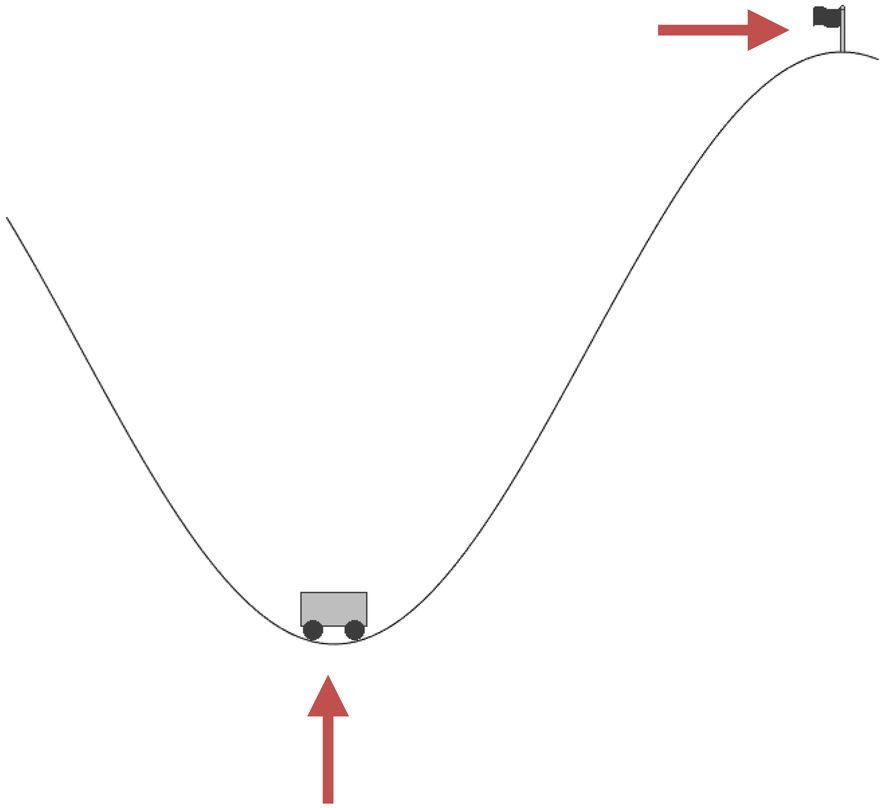
Video 2: Episodic Sarsa in Mountain Car

- **Can we do a large number of states with Semi-gradient Sarsa?** How about an infinite number of state? Yep. We do a classic control task: **Mountain Car**

- Goals:

  - gain experience analyzing the performance of an approximate TD control method

The Mountain Car environment



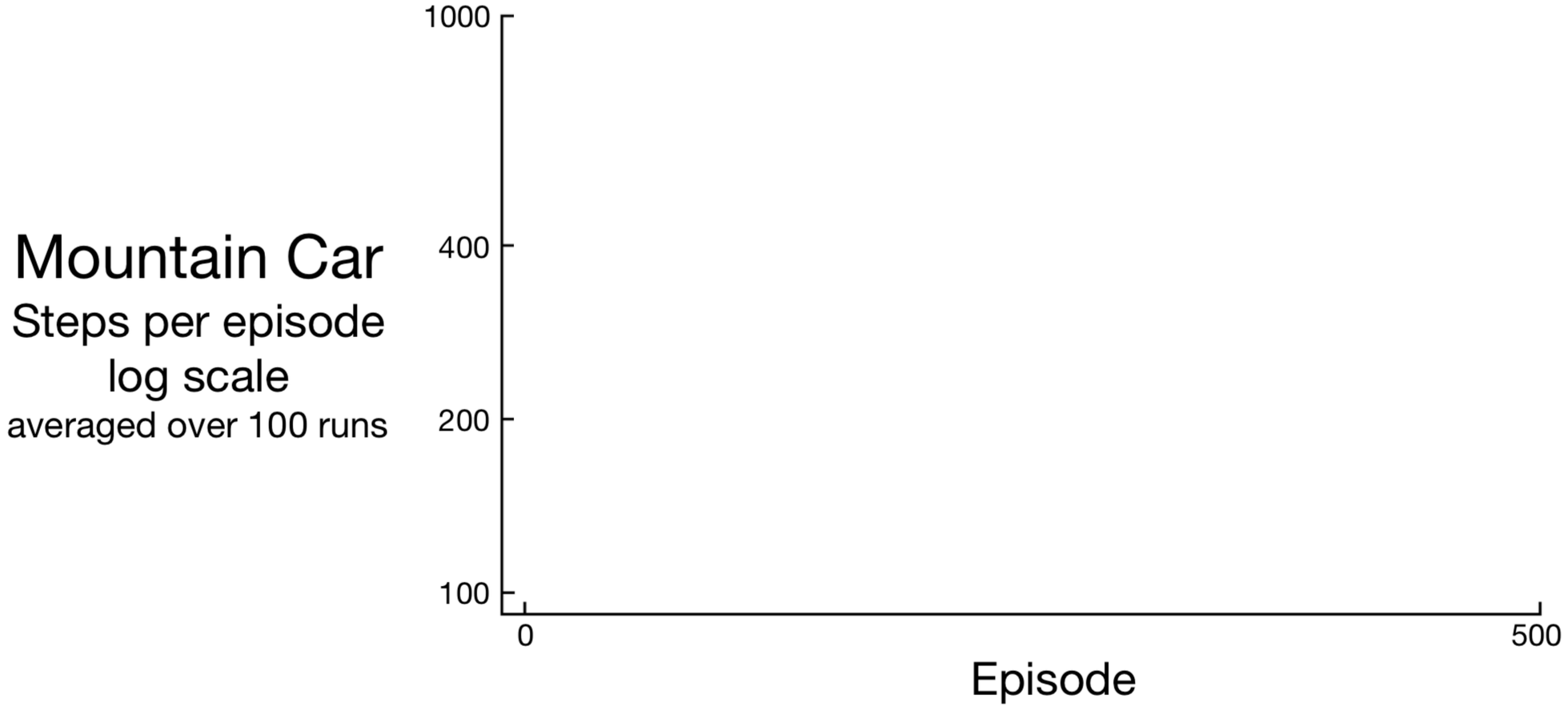
$$R_{step} = -1$$

$$\gamma = 1.0$$

State: Car position
Car velocity

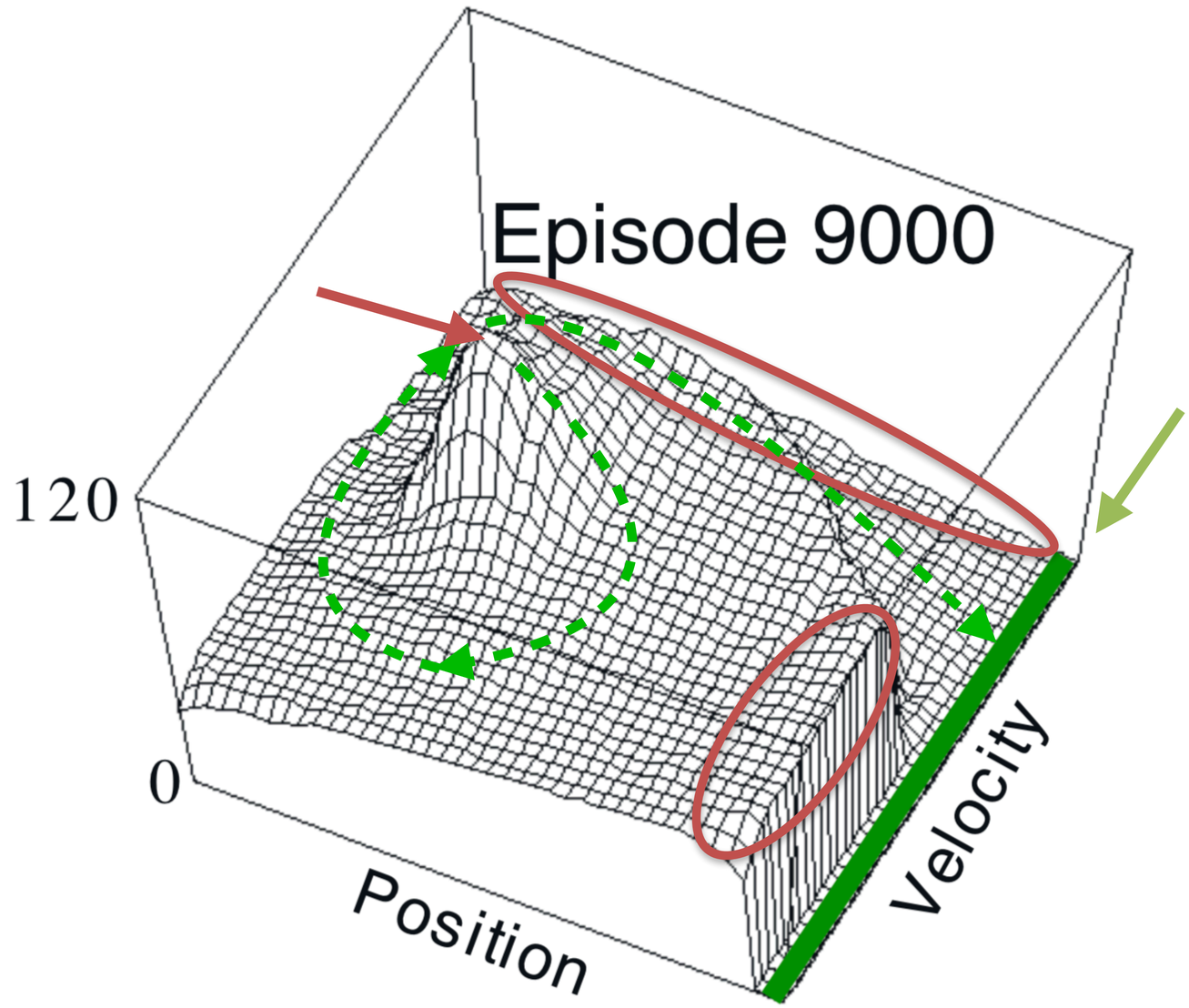
Actions: Accelerate right
Accelerate left
Coast (no acceleration)

Learning curves



Learned values

$$-\max_{a} Q(s, a, \mathbf{w})$$



Be a good RL Scientist!

- Notice, even for this tiny problem we tried different alpha. We did **many runs**. We studied learning **speed**; **final performance**; even the value function
    - we have a good idea of how Sarsa works on this problem. It's robust and stable and pretty easy to tune it's parameters
- We want to do such careful analysis every time! Especially when comparing algorithms!
- ML and AI are growing! Lots of people want jobs
- One way to stand out, is to become a really careful empiricist! A master of **good experiments.** It's a rare skill

Video 3: Expected Sarsa with Function Approximation

- If we can do Semi-gradient Sarsa, then its just **small changes** to make Semi-gradient **Expected Sarsa** and Semi-gradient **Q-learning**!
- Goals:
  - Explain the update for Expected Sarsa with function approximation
  - And explain the update for Q-learning with function approximation

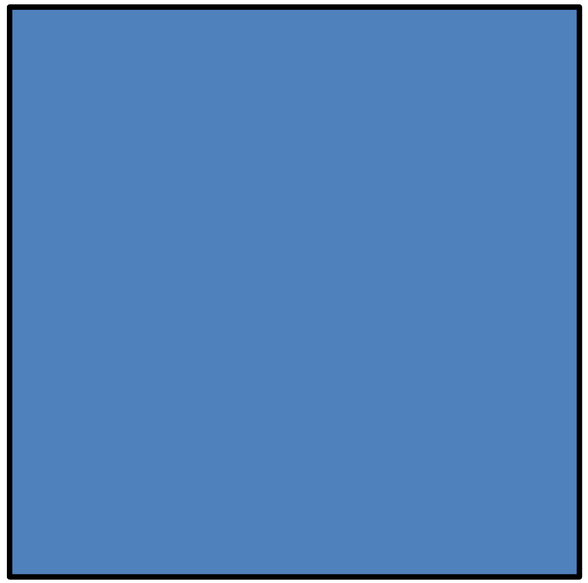
Video 4: Exploration under Function Approximation

- Balancing exploration and exploitation in RL is **hard, even in the tabular case**. Recall that some of the ideas from the Bandit problem could not be easily translated into the tabular RL problem. It is even harder in function approximation. Counting state visits? How do we do optimistic initial values with a tile coder or a NN?

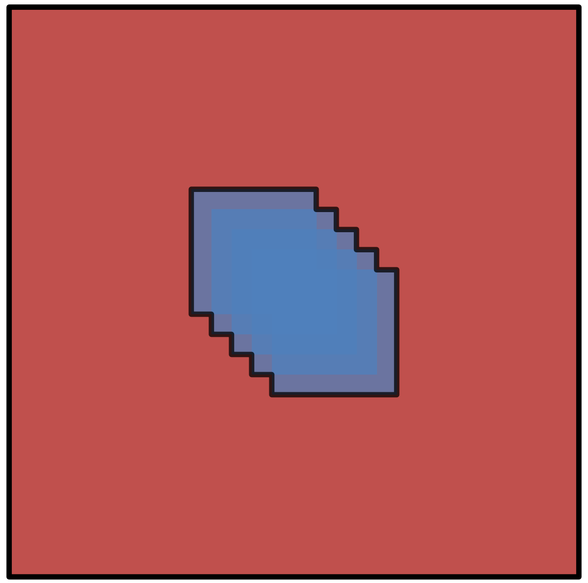
- Goals:

  - Describe how **optimistic initial values** and **epsilon-greedy** can be used with function approximation.

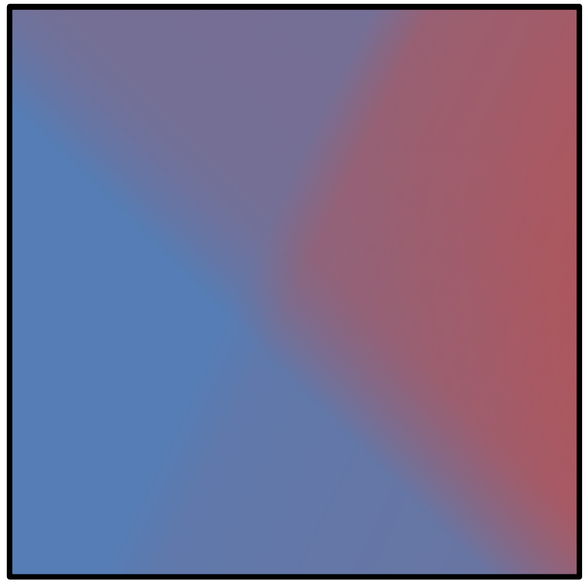
How Optimism Interacts with Generalization



Single feature



Tile coding



Neural network

Video 5: Average Reward: A New Way of Formulating Control Problems

- In some situations **discounting might not be the best choice.** For example, in **continuing tasks with function approximation.** Let's consider another way to formulate the RL task: average reward!

- Goals:

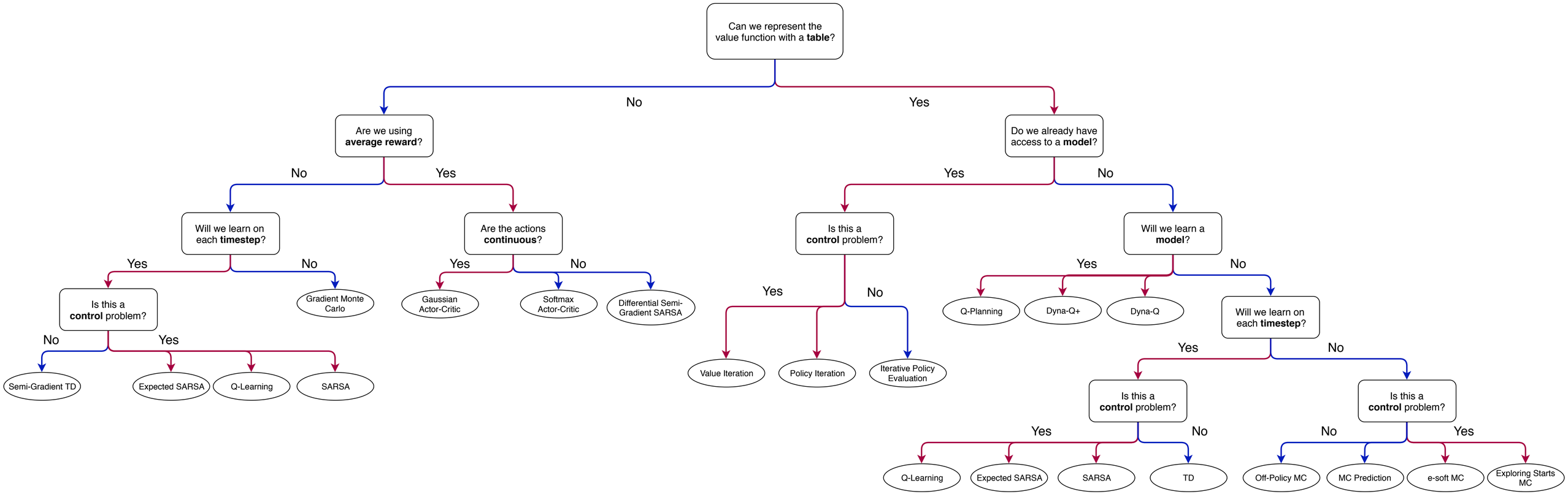
  - Describe the average reward setting

  - Explain when average reward optimal policies are different from policies obtained under discounting

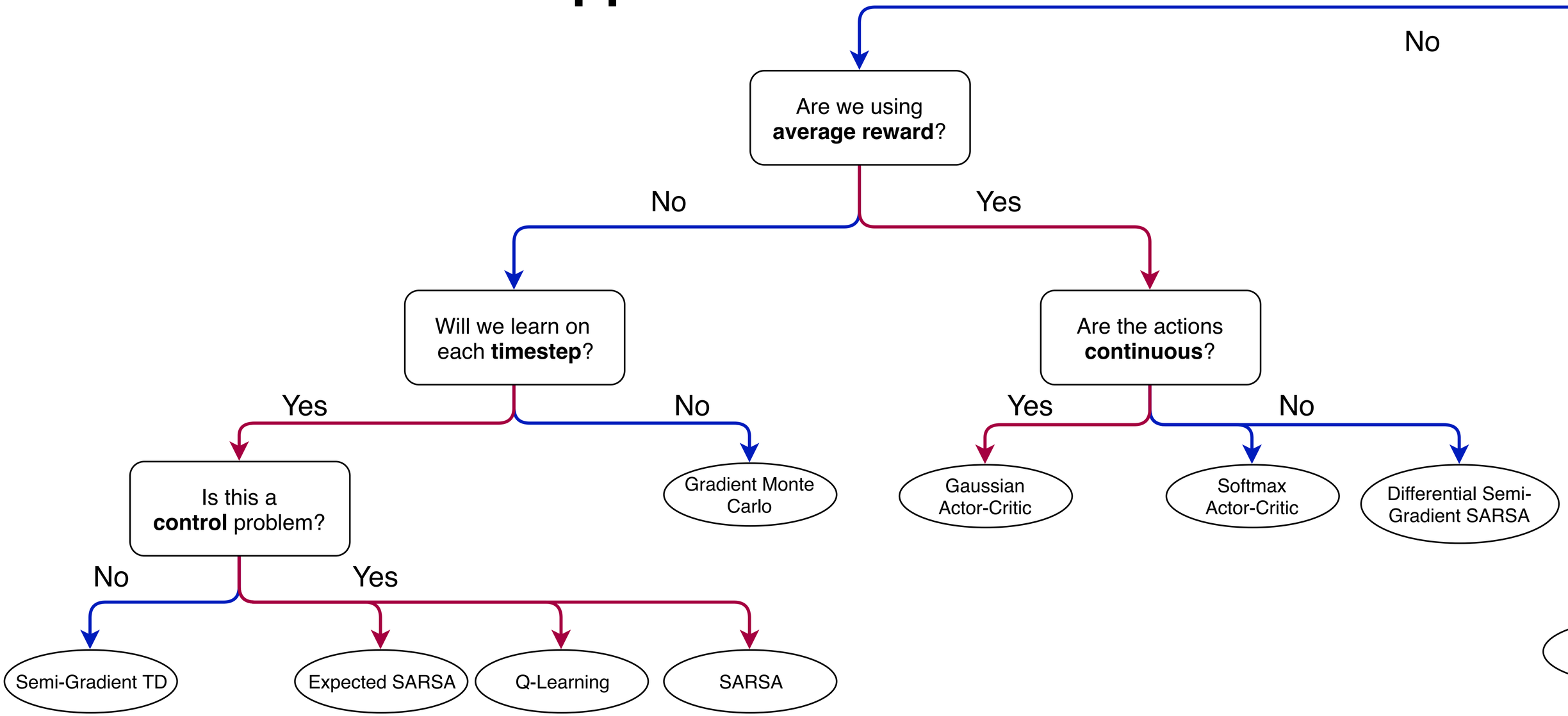
  - And understand differential value functions.

Where are we?

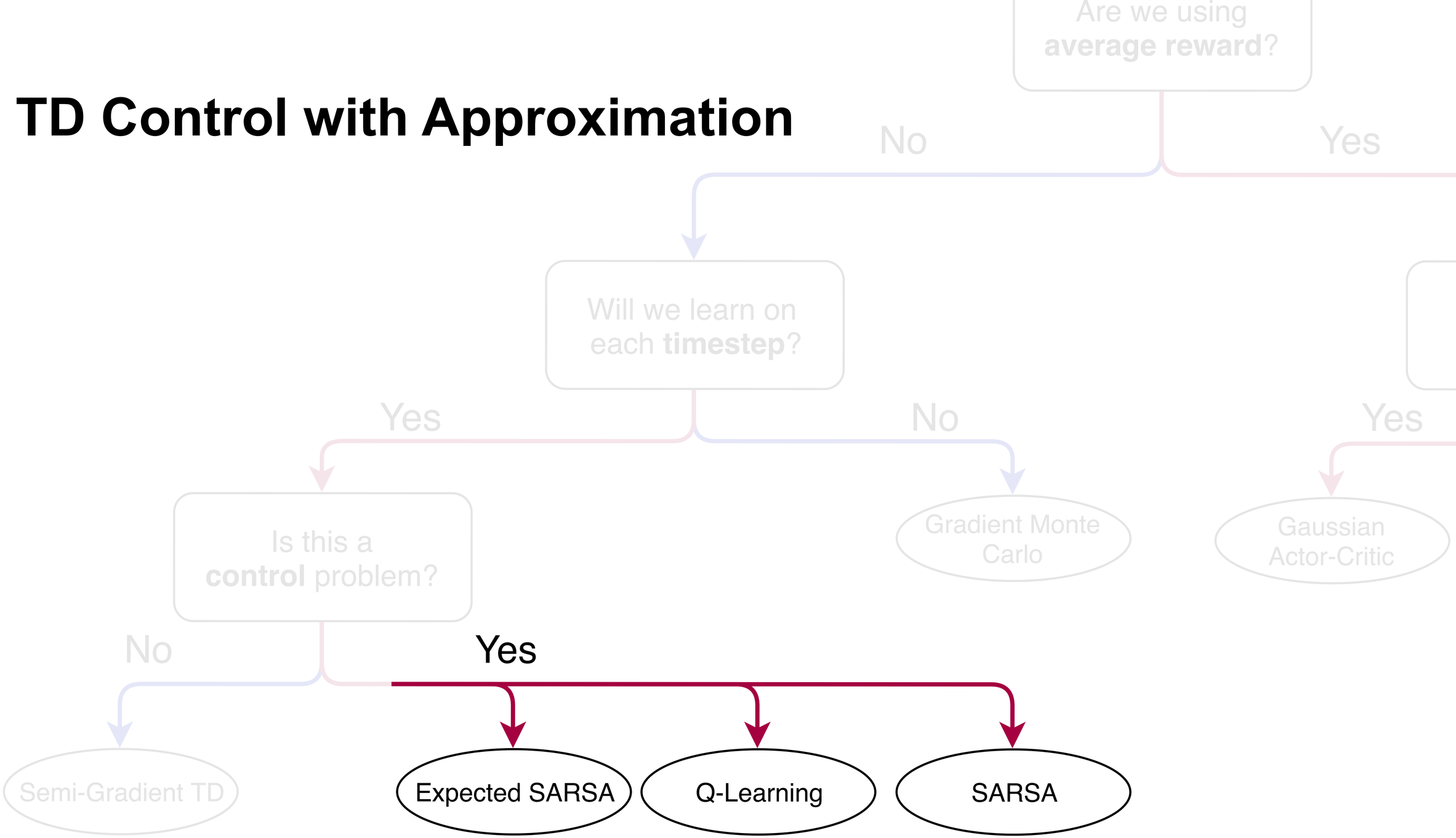
TD Control with Approximation



TD Control with Approximation



TD Control with Approximation



Question 6. [15 MARKS]

We discussed two strategies for exploration: ϵ-greedy and optimistic initial values. Assume that the agent is in a tabular setting.

Part (a) [5 MARKS]

Does ϵ-greedy exploration ensure every state will be visited at least once, in the limit (i.e., after many many steps of interaction)?

Question 6. [15 MARKS]

We discussed two strategies for exploration: ϵ-greedy and optimistic initial values. Assume that the agent is in a tabular setting.

Part (b) [5 MARKS]

Do optimistic initial values ensure every state will be visited at least once, in the limit (i.e., after many many steps of interaction)?

Part (c) [5 MARKS]

Now instead consider the function approximation setting. Does your answer to (a) or (b) changes? Explain why or why not.