Course Review

CMPUT 655
Fall 2020

Comments

- Practice midterm and practice final questions available
- Midterm will be on eClass, during class, for 1 hour 20 minutes
  - I'll show you the format with a small practice midterm before Monday
- If you want more exercise questions, see the worksheets given in CMPUT 397:
  - Schedule: https://marthawhite.github.io/rlcourse/schedule.html
  - e.g., https://marthawhite.github.io/rlcourse/docs/w-c1m3.pdf

Course Overview

Course 1 and Course 2

No
Yes
Course 1
Do we already have access to a **model**?
Yes
No
Is this a **control** problem?
Will we learn a **model**?
No
Differential Semi-Gradient SARSA
Yes
No
Yes
No
Value Iteration
Policy Iteration
Iterative Policy Evaluation
Q-Planning
Dyna-Q+
Dyna-Q
Yes
Is this a **control** problem?
Yes
No
Q-Learning
Expected SARSA
SARSA
TD

C1M1: Sequential Decision-Making

- Video 1: The K-Armed Bandit Problem
- Video 2: Estimating Action Values
  - sample-average, greedy action-selection, exploration-exploitation dilemma
- Video 3: Estimating Action Values Incrementally
- Video 4-6: The Exploration-Exploitation Trade-off and Exploration Methods
  - epsilon-greedy, optimistic initial values, upper confidence bounds

C1M2: MDPs

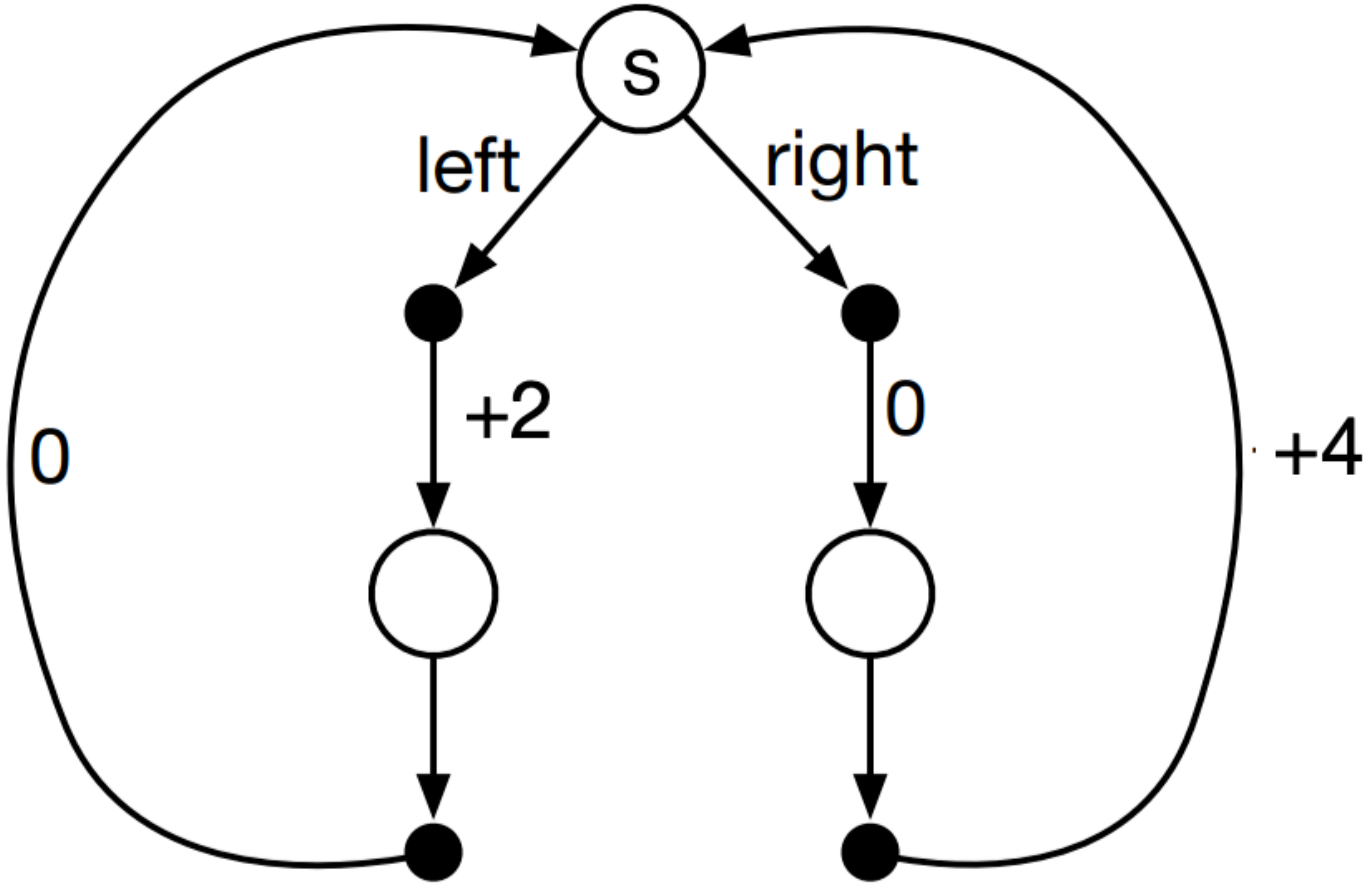
- Video 1: Markov Decision Processes
- Video 2: Examples of MDPs
- Video 3: The Goal of Reinforcement Learning (and reward hypothesis)
- Video 4: Continuing Tasks
- Video 5: Examples of Episodic Tasks and Continuing Tasks

C1M3: Value Functions and Bellman Equations

- Video 1: Policies (stochastic and deterministic)
- Video 2: Value Functions
- Video 3: Bellman Equation Derivation
- Video 4: Why Bellman Equations?
- Video 5: Optimal Policies
- Video 6: Using Optimal Value Functions to get Optimal Policies

Selt-test: C1M3

- Is the following policy valid for this MDP (i.e. does if fit our definition of a policy): Choose left for five steps, then right for five steps, then left for five steps, and so on? Explain your answer.



C1M4: Dynamic Programming

- Video 1: Policy Evaluation vs. Control
- Video 2: Iterative Policy Evaluation (to compute a value function)
- Video 3: Policy Improvement
  - policy improvement theorem, using value functions to produce a better policy
- Video 4: Policy Iteration (to compute an optimal policy)
- Video 5: Flexibility of the Policy Iteration Framework (and GPI)
- Video 6: Efficiency of Dynamic Programming (and bootstrapping)

C1M4: Dynamic Programming

- Video 1: Policy Evaluation vs. Control
- Video 2: Iterative Policy Evaluation (to compute a value function)
- Video 3: Policy Improvement
  - policy improvement theorem, using value functions to produce a better policy
- Video 4: Policy Iteration (to compute an optimal policy)
- Video 5: Flexibility of the Policy Iteration Framework (and GPI)
- Video 6: Efficiency of Dynamic Programming (and bootstrapping)

Policy Improvement Result

$$
\begin{aligned}
v_\pi(s) &\le q_\pi(s, \pi'(s)) \\
&= \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = \pi'(s)] \\
&= \mathbb{E}_{\pi'}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s] \\
&\le \mathbb{E}_{\pi'}[R_{t+1} + \gamma q_\pi(S_{t+1}, \pi'(S_{t+1})) \mid S_t = s] \\
&= \mathbb{E}_{\pi'}[R_{t+1} + \gamma \mathbb{E}[R_{t+2} + \gamma v_\pi(S_{t+2}) | S_{t+1}, A_{t+1} = \pi'(S_{t+1})] \mid S_t = s] \\
&= \mathbb{E}_{\pi'}\big[R_{t+1} + \gamma R_{t+2} + \gamma^2 v_\pi(S_{t+2}) \bigm| S_t = s\big] \\
&\le \mathbb{E}_{\pi'}\big[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 v_\pi(S_{t+3}) \bigm| S_t = s\big] \\
&\vdots \\
&\le \mathbb{E}_{\pi'}\big[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \bigm| S_t = s\big] \\
&= v_{\pi'}(s).
\end{aligned}
$$

Difference between v and q

- “Why does the lower golf example (figure 3.3) which is supposed to be optimal have a -2 field over most of the green, where the above example with the putter has that area marked as only -1? Isn't q*() supposed to be optimal? There should be no areas where q*() has a worse result than v putt, right?”

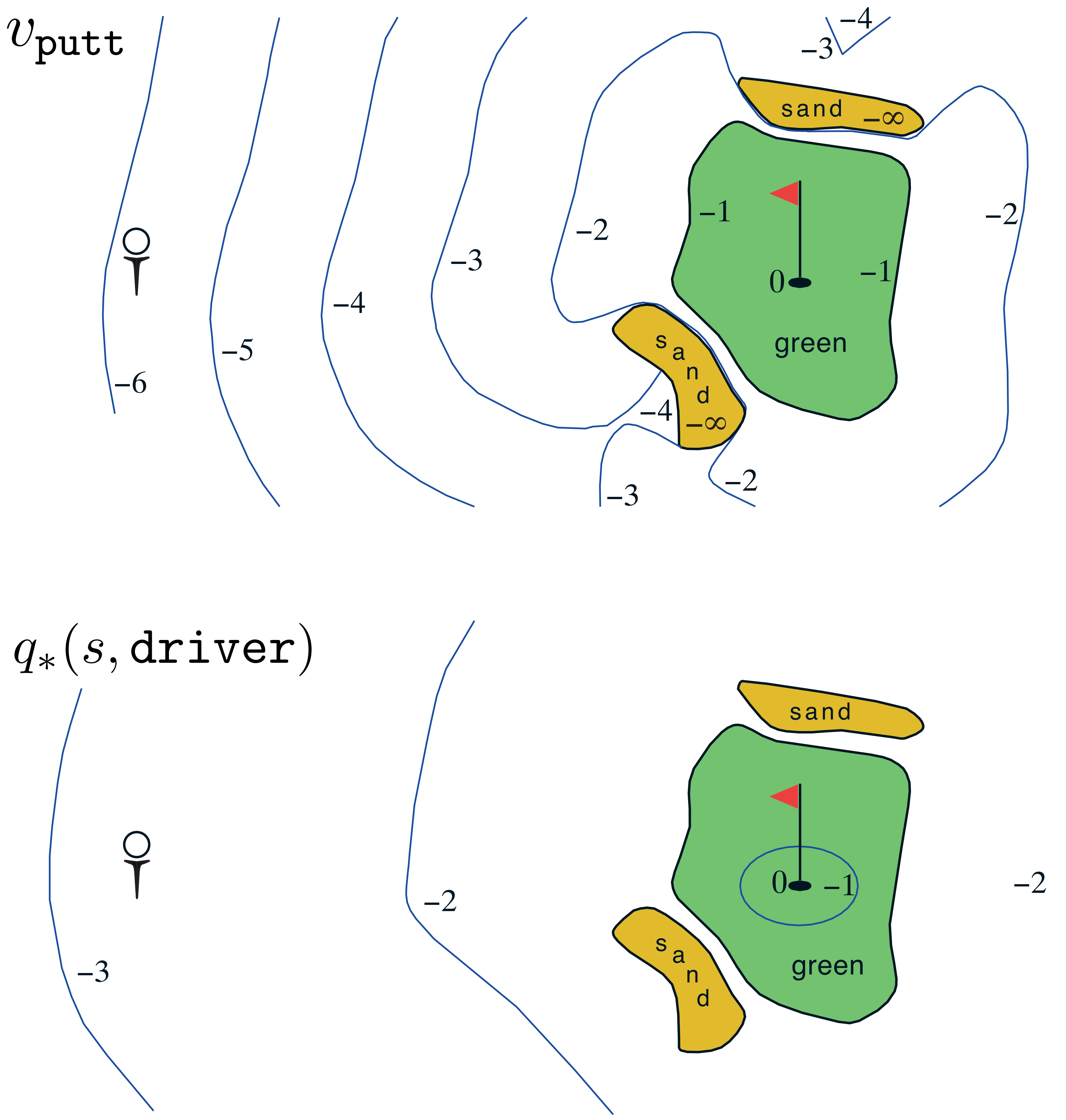
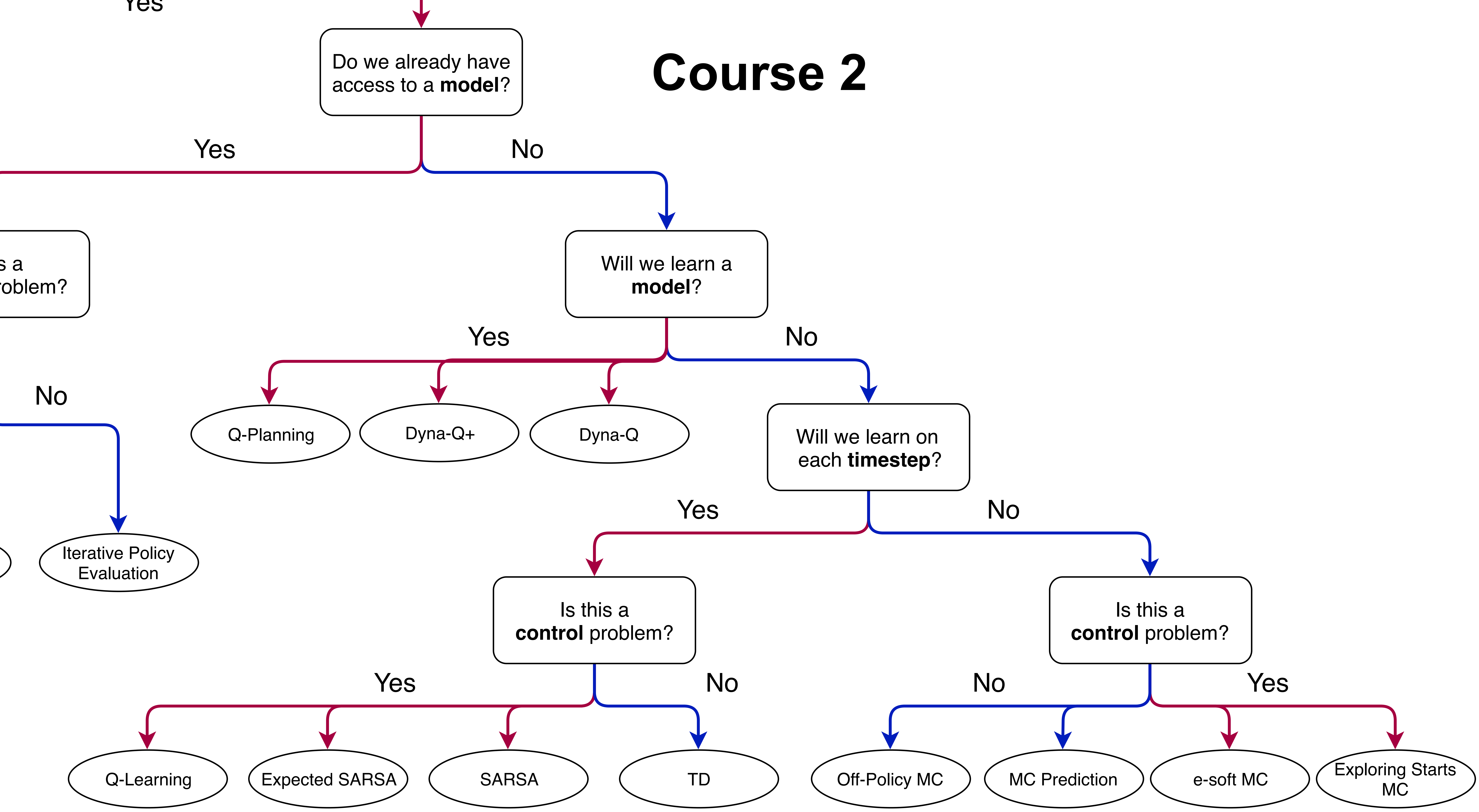


Figure 3.3: A golf example: the state-value function for putting (upper) and the optimal action-value function for using the driver (lower). ■

Course 2



Yes
Do we already have access to a **model**?
Yes
No
s a
roblem?
Will we learn a **model**?
Yes
No
No
Q-Planning
Dyna-Q+
Dyna-Q
Will we learn on each **timestep**?
Iterative Policy Evaluation
Yes
No
Is this a **control** problem?
Is this a **control** problem?
Yes
No
No
Yes
Q-Learning
Expected SARSA
SARSA
TD
Off-Policy MC
MC Prediction
e-soft MC
Exploring Starts MC

C2M1: Monte-Carlo for Prediction and Control

- Video 1: What is Monte Carlo?
- Video 2: Using Monte Carlo for Prediction
- Video 3: Using Monte Carlo to Estimate Action-Values
    - discussed importance of maintaining exploration
- Video 4: Using Monte Carlo Methods for Generalized Policy Iteration
- Video 5: Solving the Blackjack Example
- Video 6: Epsilon-Soft Policies (alternative to exploring starts)

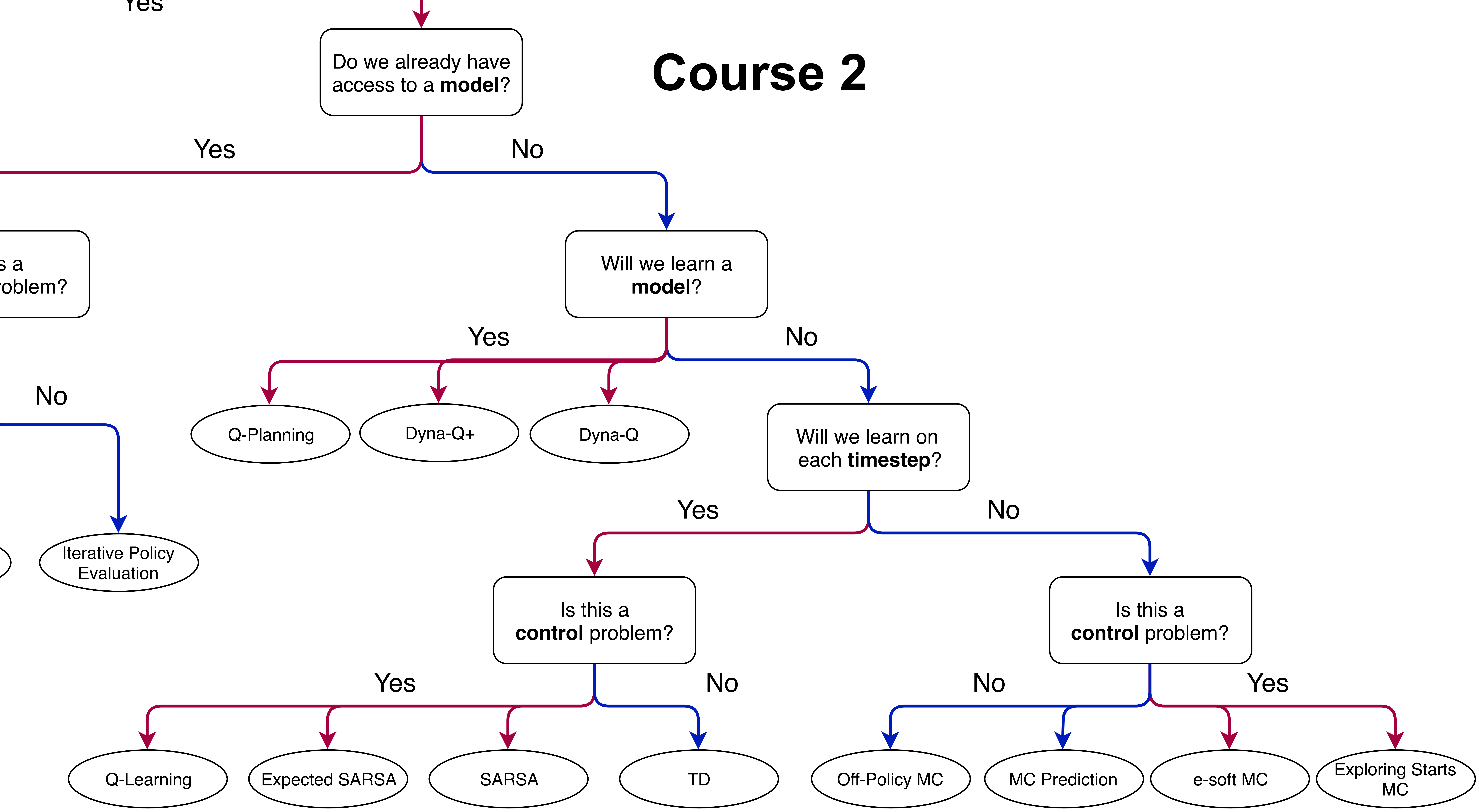
Self-test: Exploration in MC

- Why did we talk about exploring starts in MC when estimating action-values, but not when estimating state values?
- Can we use state-values for control in MC, like we did in DP?

C2M1: Monte-Carlo for Prediction and Control

- Video 4: Using Monte Carlo Methods for Generalized Policy Iteration
- Video 5: Solving the Blackjack Example
- Video 6: Epsilon-Soft Policies (alternative to exploring starts)
- Video 7: Why Does Off-Policy Learning Matter?
  - utility for exploration, discussed target policies and behavior policies
- Video 8: Importance Sampling
- Video 9: Off-Policy MC Prediction

Course 2



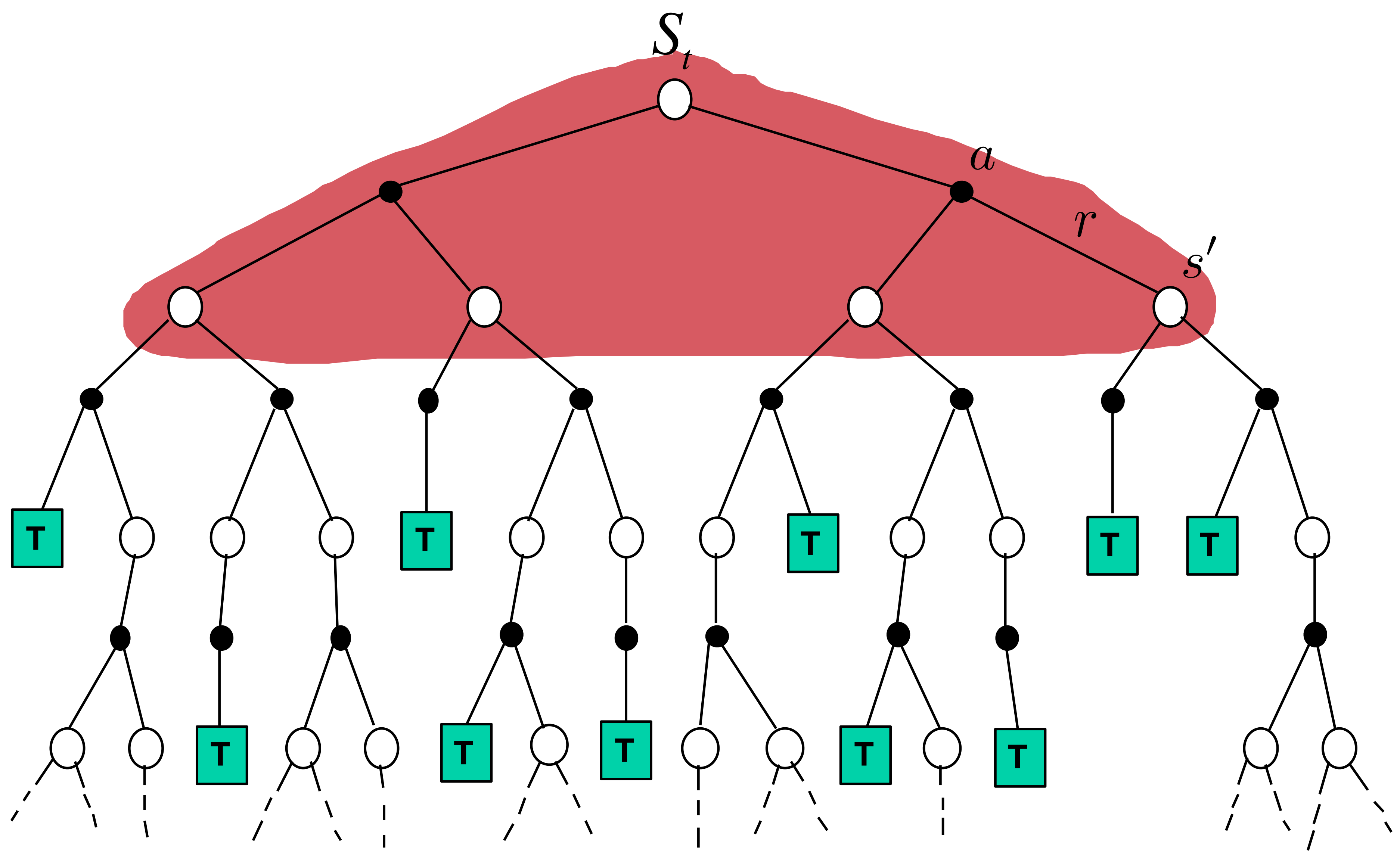
Yes
Do we already have access to a model?
Yes
No
s a
roblem?
Will we learn a model?
Yes
No
No
Q-Planning
Dyna-Q+
Dyna-Q
Will we learn on each timestep?
Iterative Policy Evaluation
Yes
No
Is this a control problem?
Is this a control problem?
Yes
No
No
Yes
Q-Learning
Expected SARSA
SARSA
TD
Off-Policy MC
MC Prediction
e-soft MC
Exploring Starts MC

C2M1: TD for Prediction

- Video 1: What is Temporal Difference Learning?
- Video 2: The Advantages of Temporal Difference Learning
    - advantages of both DP (bootstrapping) and MC (sample-based learning)
- Video 3: Comparing TD and Monte Carlo

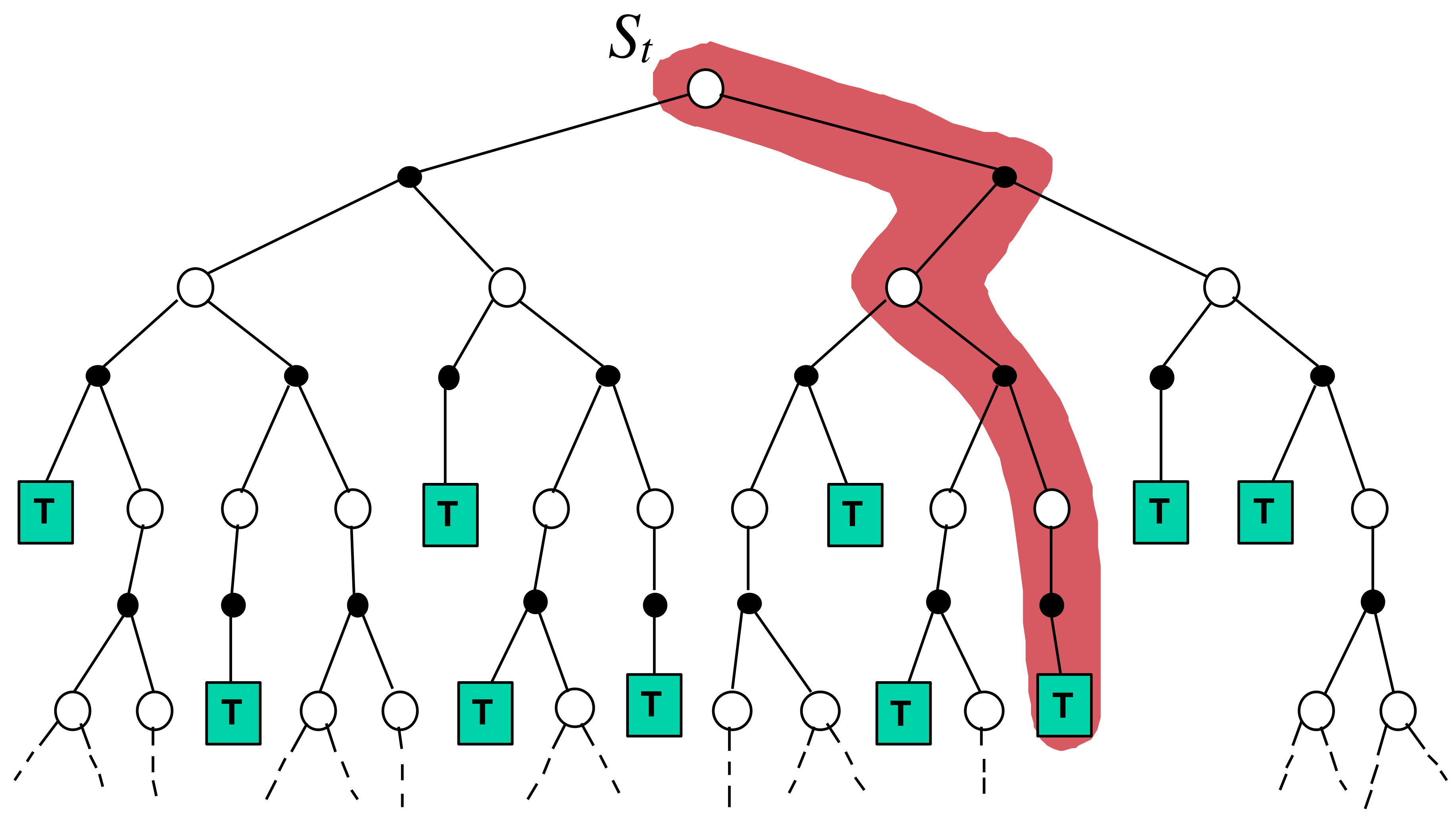
Dynamic programming

$$V(S_t) \leftarrow E_\pi\left[R_{t+1} + \gamma V(S_{t+1})\right] = \sum_a \pi(a|S_t) \sum_{s',r} p(s',r|S_t,a)[r + \gamma V(s')]$$



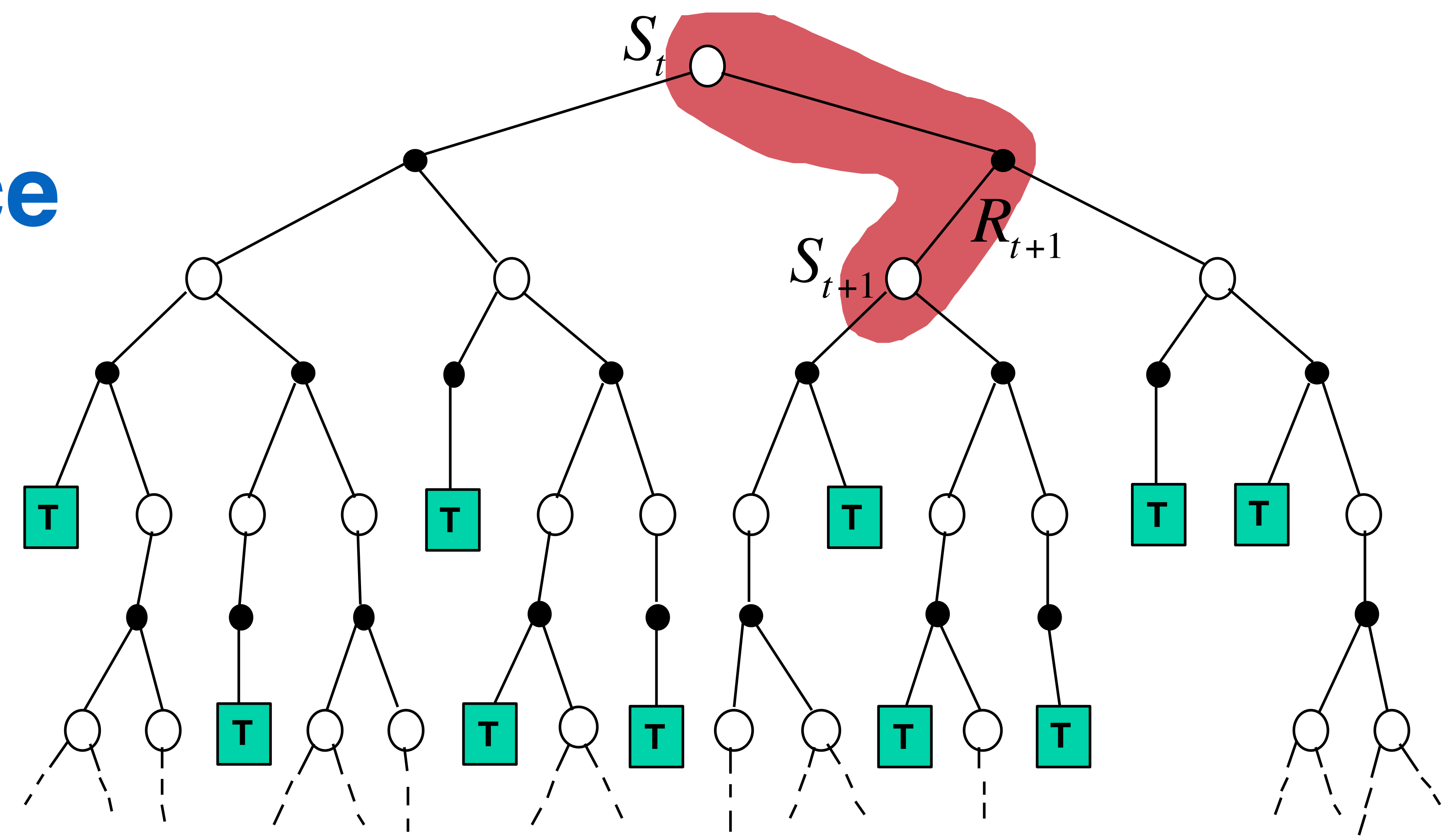
Simple Monte Carlo

$$V(S_t) \leftarrow V(S_t) + \alpha\left[G_t - V(S_t)\right]$$



$$V(S_t) \leftarrow V(S_t) + \alpha\left[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\right]$$

Temporal Difference Learning



C2M3: TD for Control

- Video 1: Sarsa: GPI with TD
  - Building an algorithm to find near optimal policies: SARSA (**S**tate, **A**ction, **R**eward, Next **S**tate, **A**ction). Combining the ideas of *policy evaluation, policy improvement, TD,* and *epsilon-soft policies*
- Video 2: Sarsa in the Windy Grid World
- Video 3: What is Q-learning
- Video 4: Q-learning in the Windy Gridworld
- Video 5: How is Q-learning Off-policy?

Self-test

- What is the target policy for Q-learning?
- What can the behavior policy be?

Self-test

- What is the target policy for Q-learning?
  - Answer: Q-learning **learns about** the greedy policy (which eventually becomes π*), while **following a different policy** (e.g., ε-greedy). That is off-policy, but there are no importance sampling corrections!
- What can the behavior policy be?

C2M3: TD for Control

- Video 3: What is Q-learning
- Video 4: Q-learning in the Windy Gridworld
- Video 5: How is Q-learning Off-policy?
- Video 6: Expected SARSA
- Video 7: Expected SARSA in the Cliff World
- Video 8: The Generality of Expected SARSA

Terminology Review

- TD methods we have learned about are **tabular, one-step, model-free** learning algorithms

- **Tabular:** we store the value function in a table. One entry in the table per value, so each value is stored independently of the others. We are implicitly assuming the state-space ($\mathcal{S}$) is small

- **One-step**: we update a single state or state-action value on each time-step. Only the value of Q(S,A) from S -- A --->S',R. We never update more than one value per learning step

- **Model-free**: we don't assume access to or make use of a model of the world. All learning is driven by sample experience. Data generated by the agent interacting with the environment


Yes
Do we already have access to a model?
Course 2
Yes
No
s a
roblem?
Module 4
Will we learn a model?
Yes
No
No
Q-Planning
Dyna-Q+
Dyna-Q
Will we learn on each timestep?
Iterative Policy Evaluation
Yes
No
Is this a control problem?
Is this a control problem?
Yes
No
No
Yes
Q-Learning
Expected SARSA
SARSA
TD
Off-Policy MC
MC Prediction
e-soft MC
Exploring Starts MC

C2M1: Planning, Learning and Acting

- Video 1: What is a Model?
- Video 2: Comparing Sample and Distribution Models
- Video 3: Random Tabular, Q-planning
- Video 4: The Dyna Architecture
- Video 5: The Dyna Algorithm

Tabular Dyna-Q

C2M4: Planning, Learning and Acting

- Video 4: The Dyna Architecture
- Video 5: The Dyna Algorithm
- Video 6: Dyna & Q-learning in a Simple Maze
- Video 7: What if the model is inaccurate?
- Video 8: In-depth with changing environments

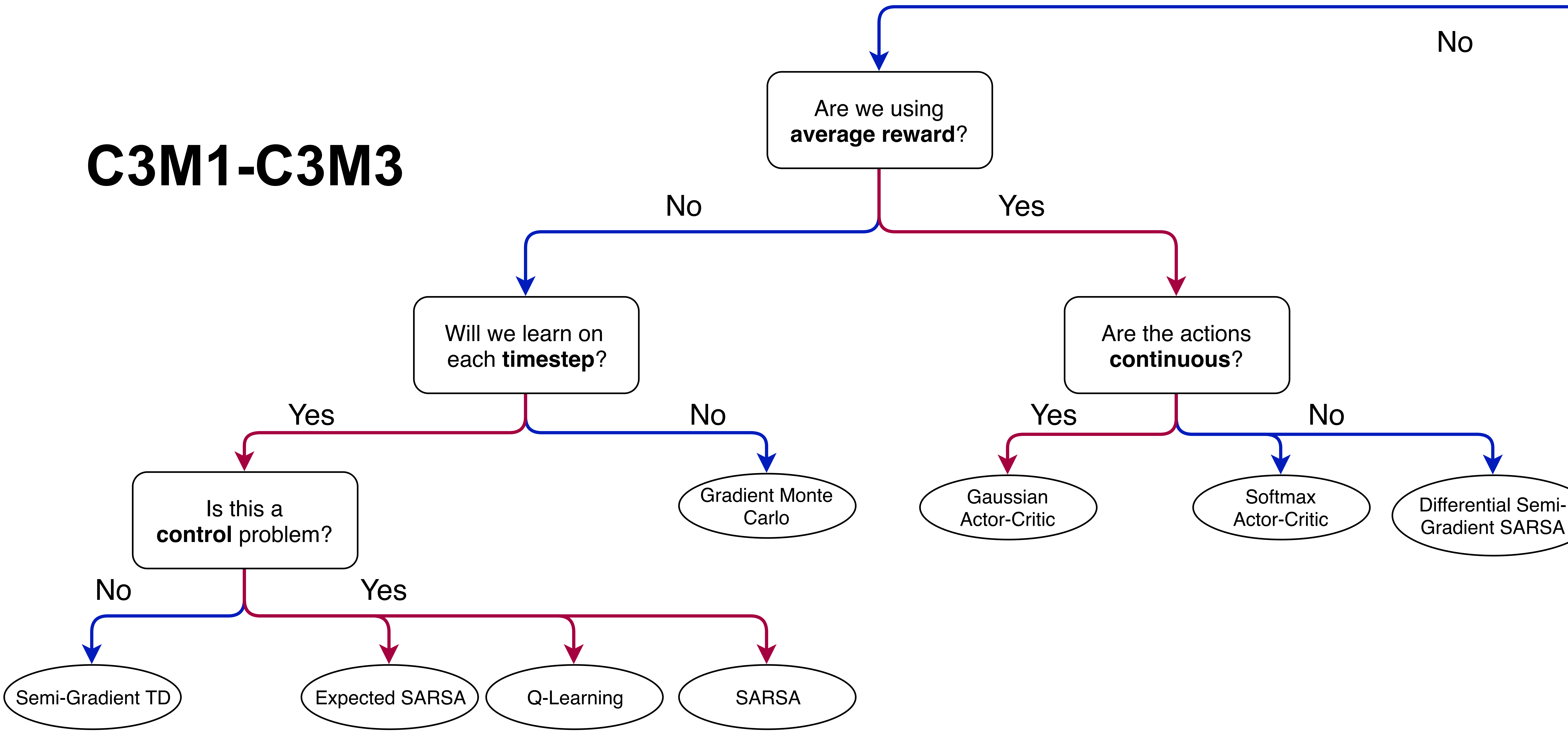
Terminology Review

- **Model:** a model of the environment. Anything that can predict how the environment will respond to the agent's actions: M(S,A) --->S',R
- **Planning**: the computational process that takes the model as input and produces or improves the policy
- **Sample Model**: a model that can produce a possible next state and reward, in agreement with the underlying transition probabilities of the world. We need not store all the probabilities to do this (think about epsilon-greedy)
- **Simulate:** sample a transition from the model. Given an S and A, ask the model for a possible next state S' and reward R
- **Simulated Experience:** samples generated by a sample model. Like dreaming or imagining things that could happen
- **Real Experience:** the states, actions, and rewards that are produced when an agent interacts with the real world.
- **Search Control:** the computational process that selects the state and action in the planning loop

Course 3

Can we represent the value function with a table?
No
Are we using average reward?
No
Yes
Will we learn on each timestep?
Yes
No
Gradient Monte Carlo
Are the actions continuous?
Yes
No
Gaussian Actor-Critic
Softmax Actor-Critic
Differential Semi-Gradient SARSA
em?
Yes
Expected SARSA
Q-Learning
SARSA
Is this a
control prob
Yes
Value Iteration
Policy Iteration

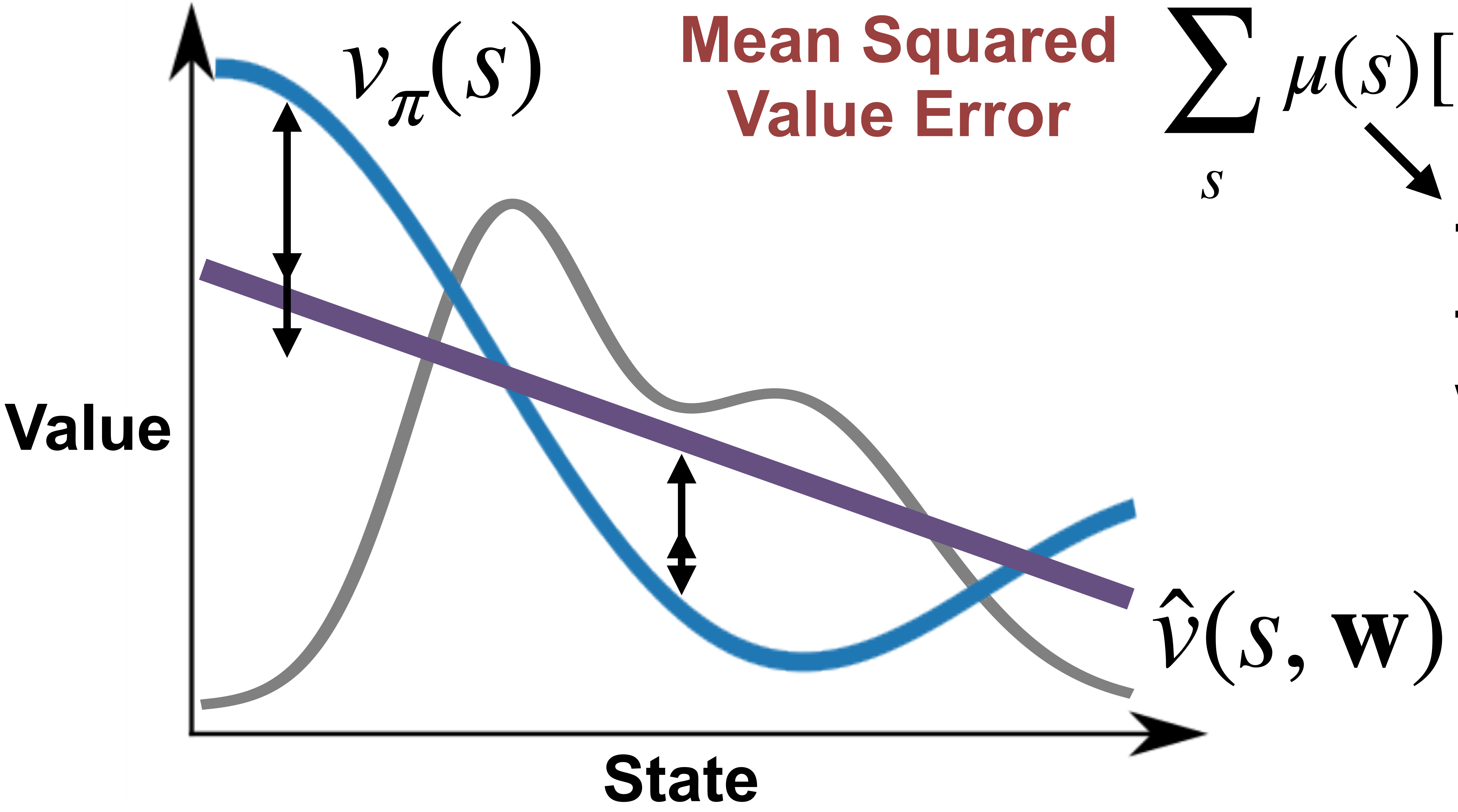
C3M1-C3M3


No
Are we using average reward?
No
Yes
Will we learn on each timestep?
Are the actions continuous?
Yes
No
Yes
No
Is this a control problem?
Gradient Monte Carlo
Gaussian Actor-Critic
Softmax Actor-Critic
Differential Semi-Gradient SARSA
No
Yes
Semi-Gradient TD
Expected SARSA
Q-Learning
SARSA

C3M1: Prediction with Approximation

- Video 1: Moving to Parameterized Functions
- Video 2: Generalization and Discrimination (and how we want both)
- Video 3: Framing Value Estimation as Supervised Learning
- Video 4: Value Error
    - role of state distribution in the objective

The Mean Squared Value Error Objective



Mean Squared Value Error

$$\sum_s \mu(s)[v_\pi(s) - \hat{v}(s, \mathbf{w})]^2$$

$\mu(s)$: **The fraction of time we spend in S when following policy π**

Question: Why didn't we use the Value Error in the tabular setting?

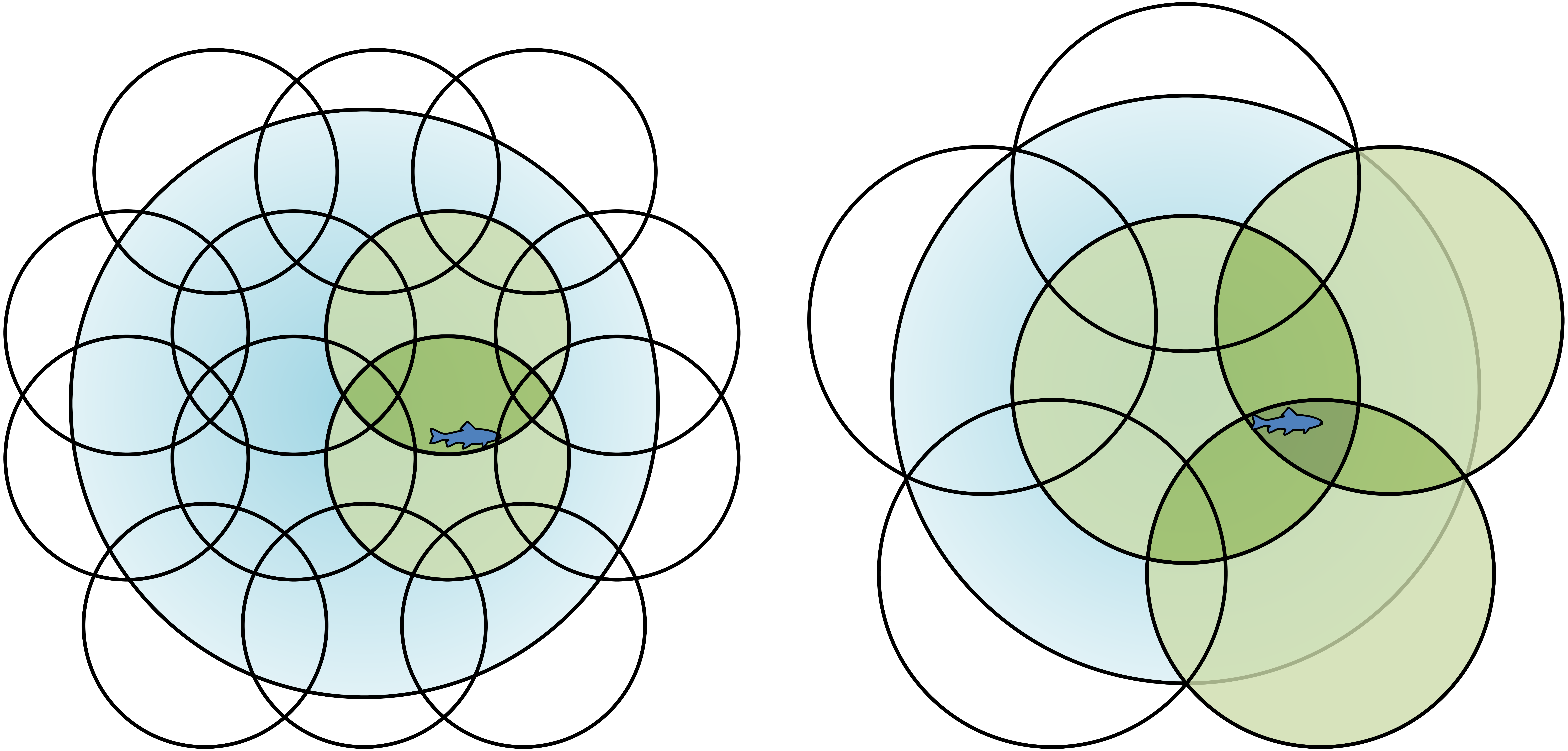
C3M1: Prediction with Approximation

- Video 5: Introducing Gradient Descent
- Video 6: Gradient Monte Carlo for Policy Evaluation
- Video 7: State Aggregation with Monte Carlo
- Video 8: Semi-gradient TD for Policy Evaluation
- Video 9: Comparing TD and MC with State Aggregation
- Video 10: The Linear TD Algorithm
- Video 11: The True Objective for TD

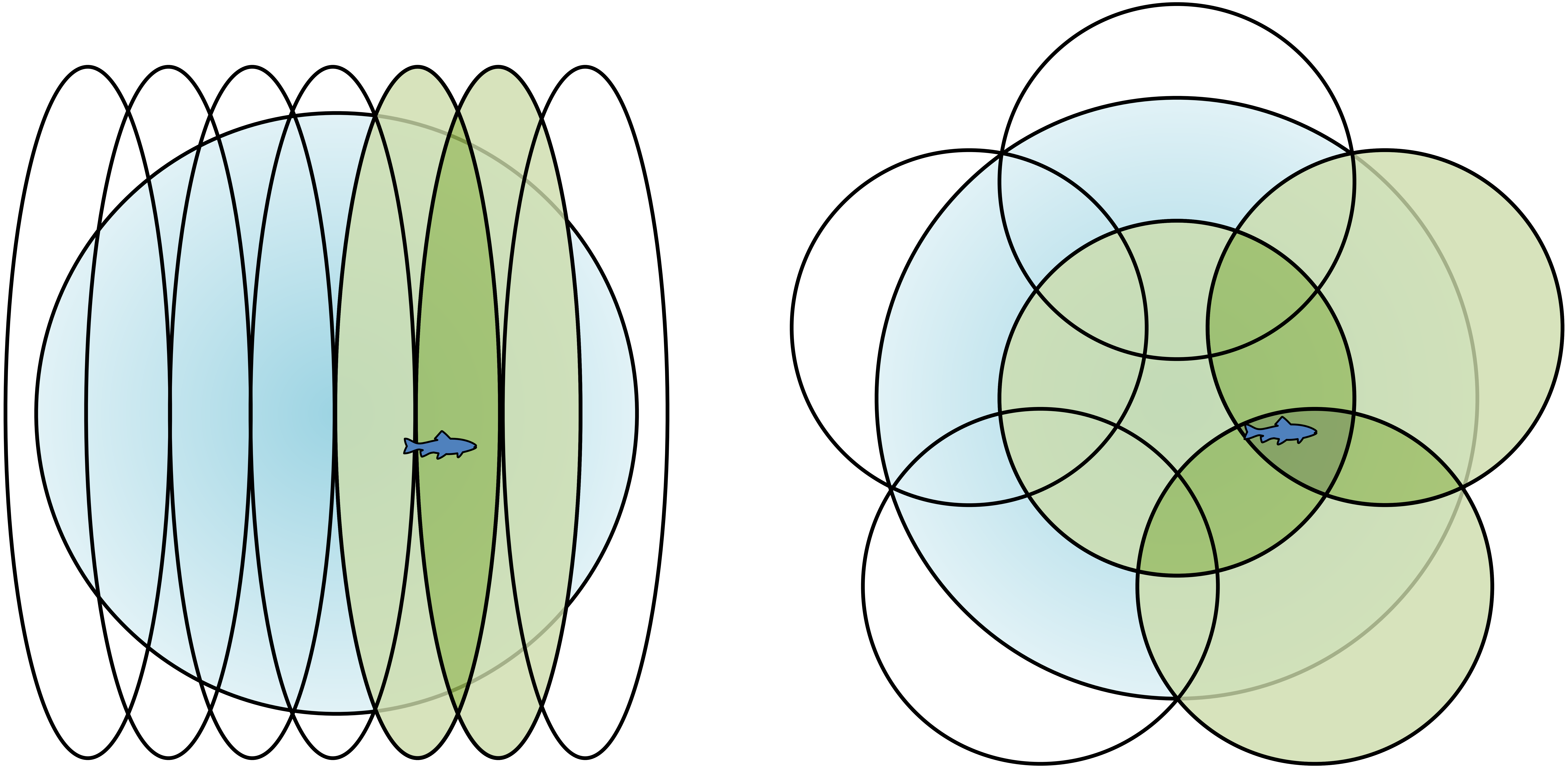
C3M2: Constructing Features

- Video 1: Coarse Coding
- Video 2: Generalization Properties of Coarse Coding
- Video 3: Tile Coding
- Video 4: Using Tile Coding for Prediction

Broadness of generalization



Direction of generalization

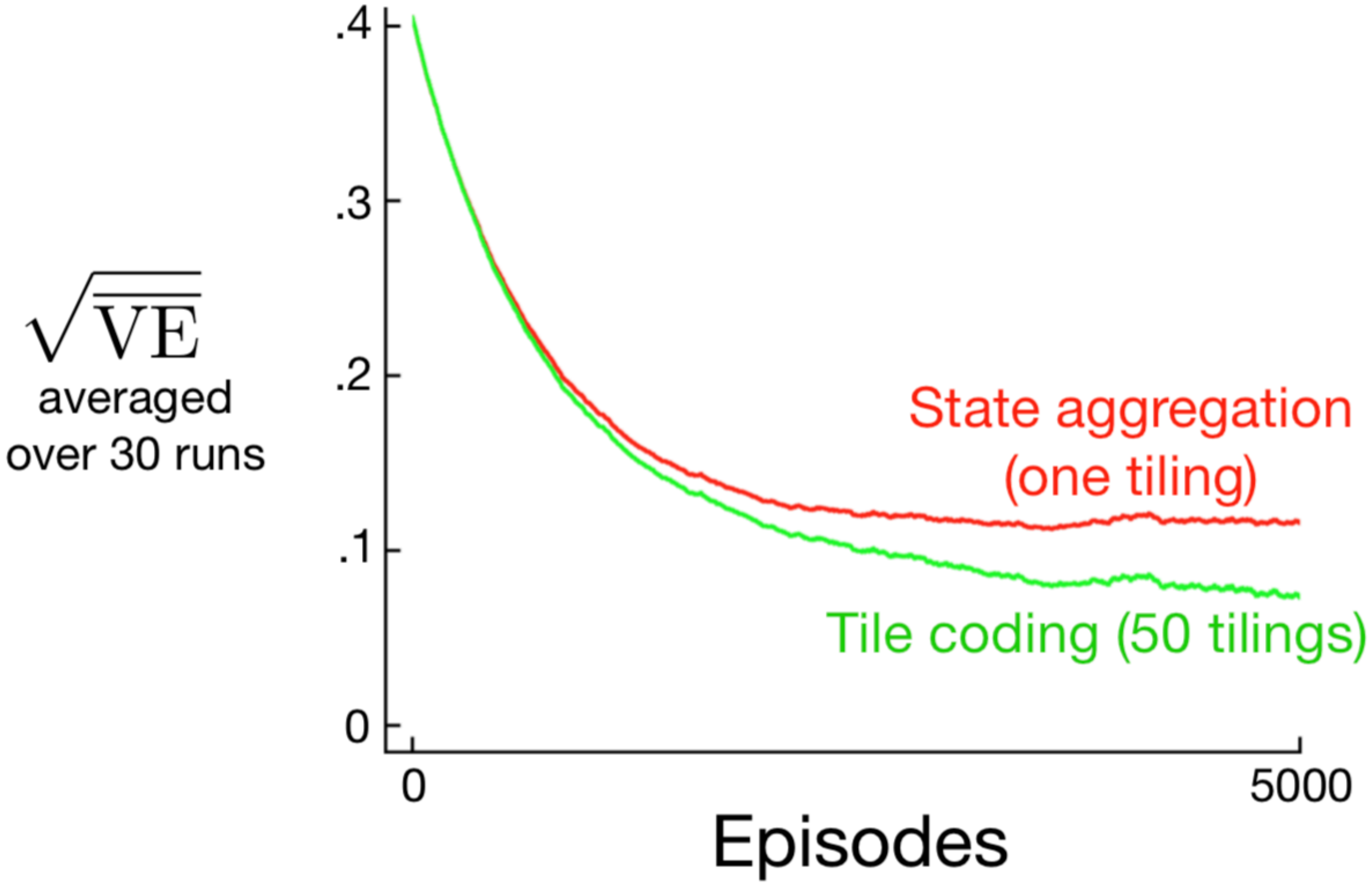


C3M2: Constructing Features

- Video 5: What is a Neural Network
- Video 6: Non-linear Approximation with Neural Networks
- Video 7: Deep Neural Networks
- Video 8: How to compute the gradient
- Video 9: Optimization Strategies for NNs
    - initialization, vector stepsizes, momentum

Which will work better on the Random Walk?

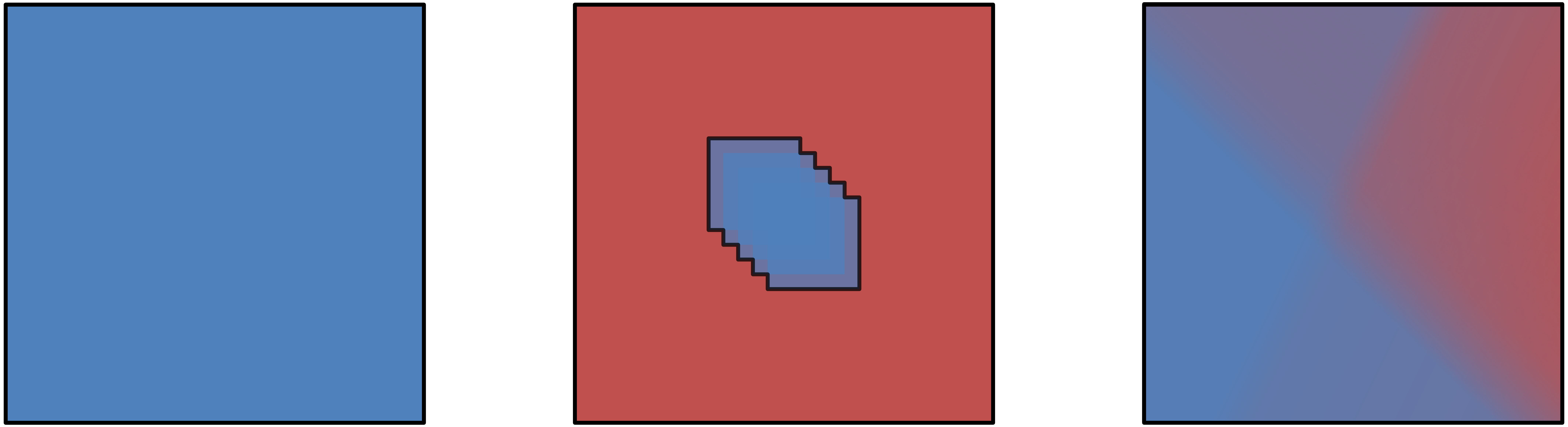
- State aggregation?
- Tile Coding?
- A NN?



C3M3: Control with Approximation

- Video 1: Episodic Sarsa with Function Approximation
- Video 2: Episodic Sarsa in Mountain Car
- Video 3: Expected Sarsa with Function Approximation
- Video 4: Exploration under Function Approximation
  - difficulties using optimisitic initial values
- Video 5: Average Reward: A New Way of Formulating Control Problems

How Optimism Interacts with Generalization



Single feature   Tile coding   Neural network

Self-test

- We discussed multiple ways to incorporate actions under FA
- Action-dependent features, where we stack state features
  - e.g., tile code input state to get d features, weights are d x |A| size and state-action features are all zero except in the a location, which has those d features
- How do we incorporate the action in a NN?

C3M4: Policy Gradient

- I will not test you on average reward, nor on policy gradient
- I am skipping this in the review

Let’s go through the practice quizzes