Course 2, Module 2
Temporal Difference Learning Methods for Prediction

CMPUT 397
Fall 2019

Any questions about course admin?

- Link for questions:
  - **http://www.tricider.com/brainstorming/35B8Mn3NZ5B**

Review of Course 2, Module 2
TD Learning

Video 1: What is Temporal Difference (TD)?

- One of the central ideas of Reinforcement Learning! We focus on policy evaluation first: learning v_π.
- Updating a guess from a guess: Bootstrapping. It means we can learning **during the episode. No waiting till the end of an episode!**
- Goals:
  - Define temporal-difference learning
  - Define the temporal-difference error
  - And understand the TD(0) algorithm.

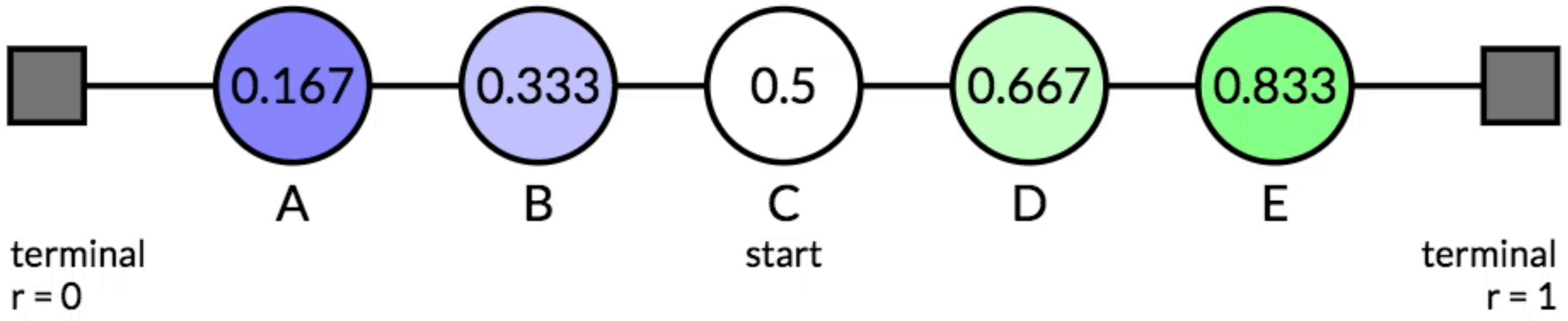
Video 2: The Advantages of Temporal Difference Learning

- How TD has some of the benefits of MC. Some of the benefits of DP. AND some benefits unique to TD
- Goals:
  - Understand the benefits of learning online with TD
  - Identify key **advantages of TD methods** over Dynamic Programming and Monte Carlo methods
    - do not need a model
    - update the value function on every time-step
    - typically learns faster than Monte Carlo methods

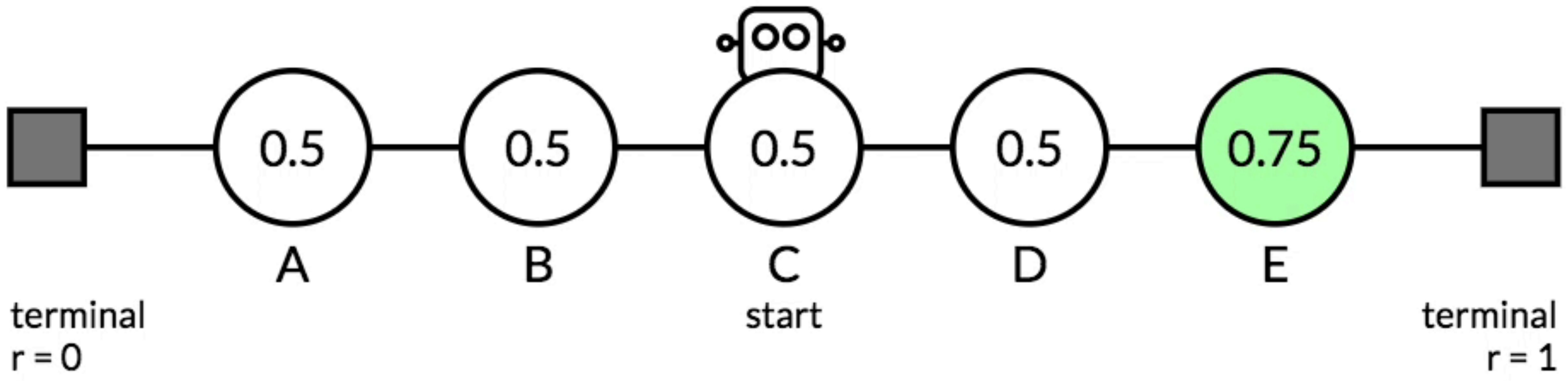
Video 3: Comparing TD and Monte Carlo

- Worked through an example using TD and Monte Carlo to learn v_{π}. We looked at how the updates happened on each step. And final performance via learning curves
- Goals:
  - Identify the empirical benefits of TD learning.

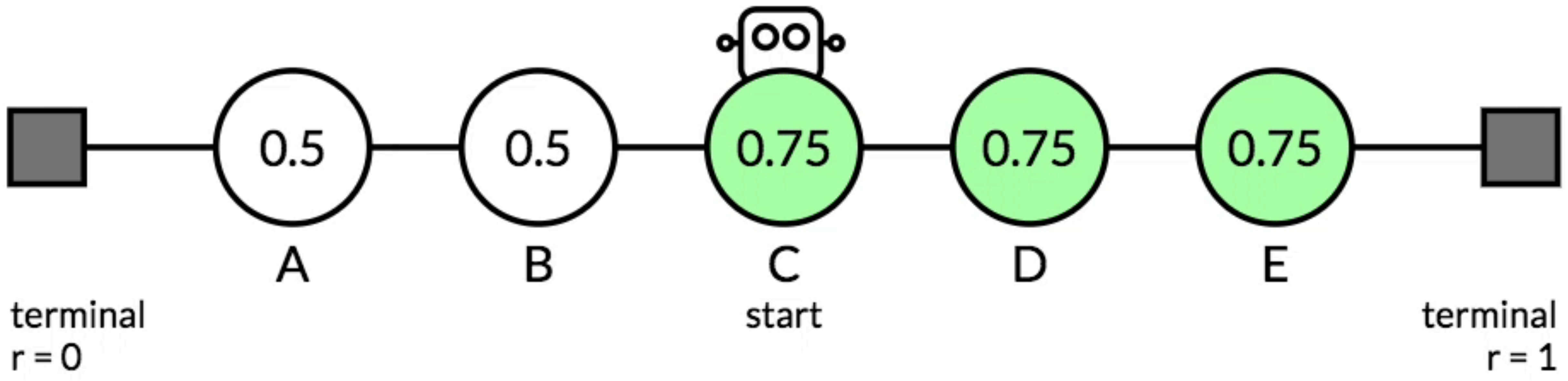
Target / Exact Values



Updates using TD Learning



Updates using Monte Carlo



Tabular TD(0) for estimating v_π

Input: the policy π to be evaluated
Algorithm parameter: step size $\alpha \in (0, 1]$
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(\mathit{terminal}) = 0$

Loop for each episode:
    Initialize S
    Loop for each step of episode:
        $A \leftarrow$ action given by π for S
        Take action A, observe R, S'
        $V(S) \leftarrow V(S) + \alpha\big[R + \gamma V(S') - V(S)\big]$
        $S \leftarrow S'$
    until S is terminal

Terminology Review

- In TD learning there are **no models, YES bootstrapping, YES learning during the episode**

- TD methods update the value estimates on a **step-by-step** basis. We **do not wait** until the end of an episode to update the values of each state.

- TD methods use **Bootstrapping**: using the estimate of the value in the next state to update the value in the current state: $V(S) \leftarrow V(S) + \alpha\underbrace{[R + \gamma V(S') - V(S)]}_{\textbf{TD-error}}$

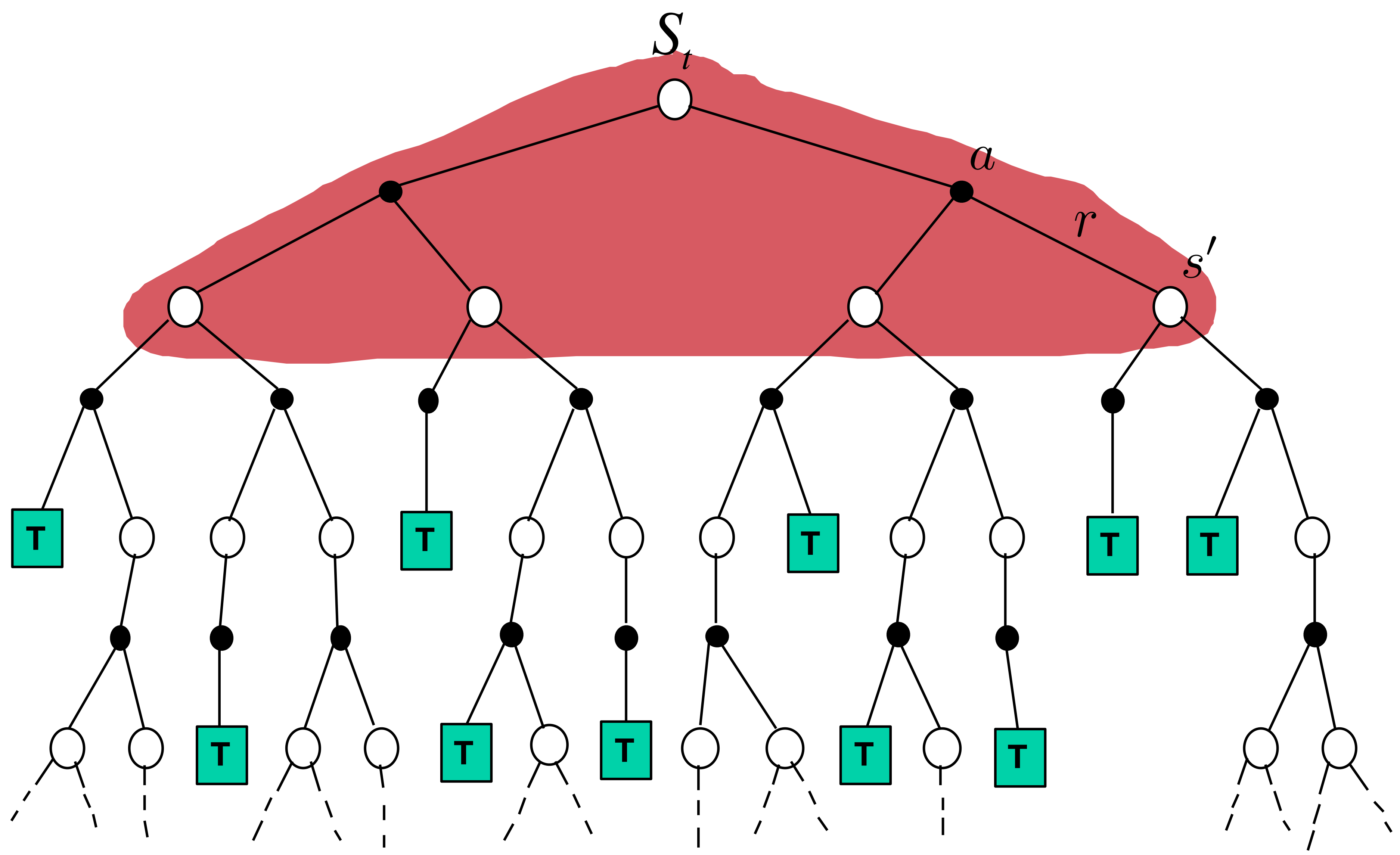
- TD is a **sample update** method: update involves the value of single sample successor state

- An **expected update** requires the complete distribution over all possible next states

- TD and MC are sample update methods. Dynamic programming uses expected updates

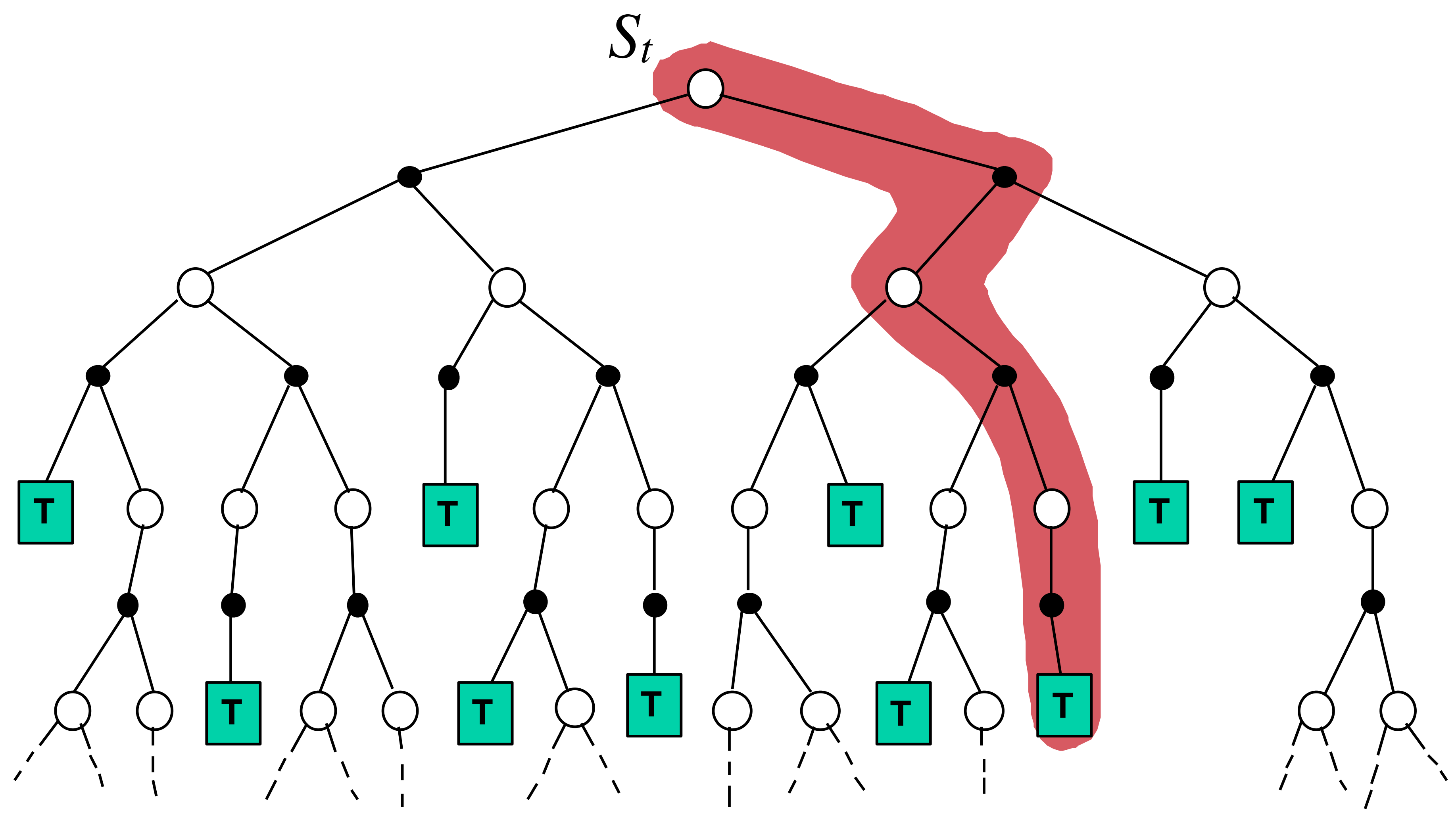
Dynamic programming

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



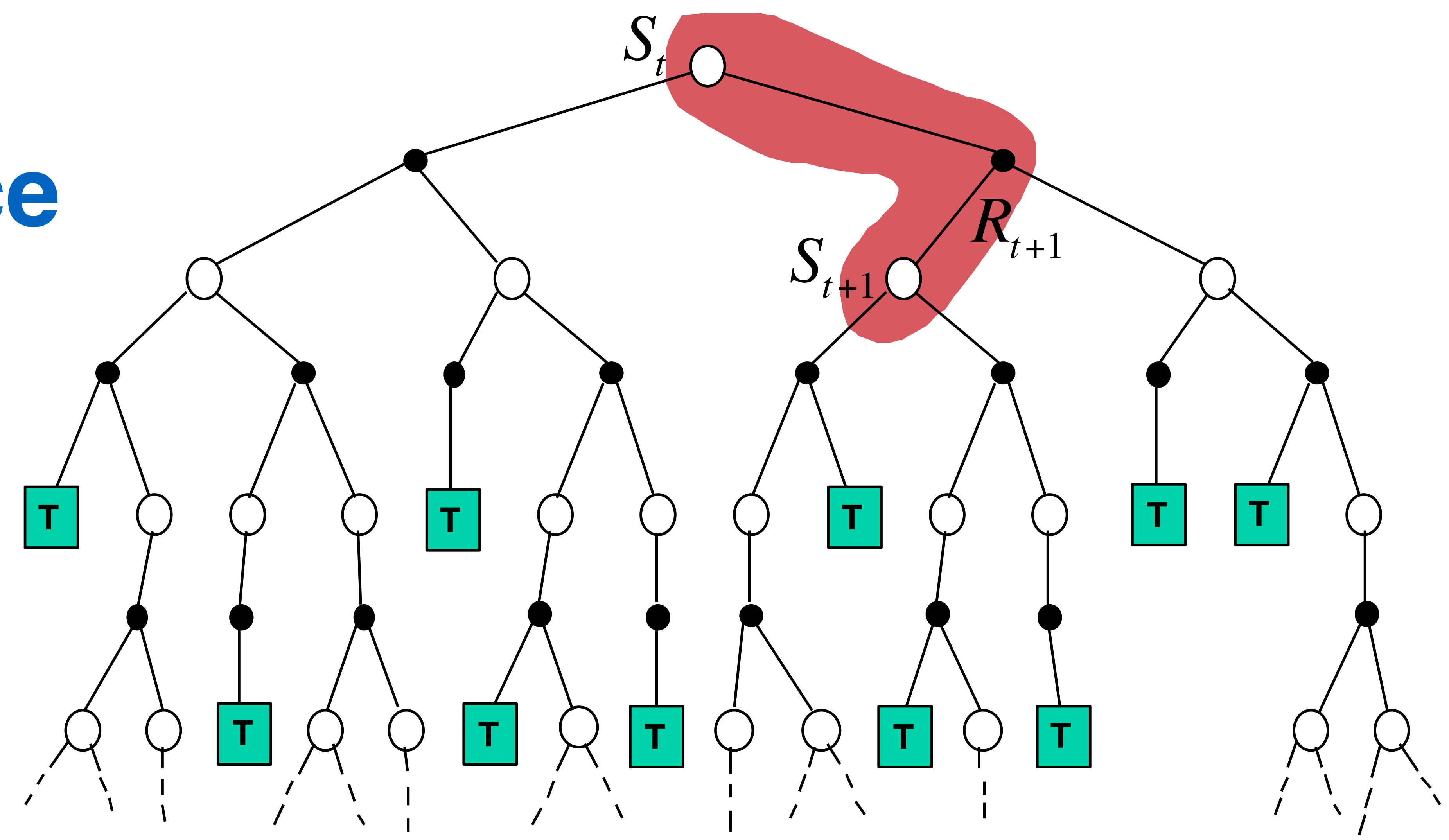
Simple Monte Carlo

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



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

Temporal Difference Learning



Worksheet Question

Modify the Tabular TD(0) algorithm for estimating v_π, to estimate q_π.

Tabular TD(0) for estimating v_π

Input: the policy π to be evaluated
Algorithm parameter: step size $\alpha \in (0, 1]$
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(\mathit{terminal}) = 0$

Loop for each episode:
    Initialize S
    Loop for each step of episode:
        $A \leftarrow$ action given by π for S
        Take action A, observe R, S'
        $V(S) \leftarrow V(S) + \alpha\big[R + \gamma V(S') - V(S)\big]$
        $S \leftarrow S'$
    until S is terminal

Challenge Question

(**Challenge Question**) In this question we consider the variance of the TD target, $R_{t+1} + \gamma V(S_{t+1})$ compared to the variance of the Monte Carlo target, G_t. Let's assume an idealized setting, where we have found a V that exactly equals v_π. We can show that, in this case, the variance of the Monte Carlo target is greater than or equal to the variance of the TD target. Note that variance of the targets is a factor in learning speed, where lower variance targets typically allow for faster learning. Show that the Monte Carlo target has at least as high of variance as the TD target, using the following decomposition, called the Law of Total Variance

$$\mathrm{Var}(G_t|S_t = s) = \mathbb{E}[\mathrm{Var}(G_t|S_t = s, S_{t+1})] + \mathrm{Var}(\mathbb{E}[G_t|S_t = s, S_{t+1}]|S_t = s).$$