Course 2, Module 2
Temporal Difference Learning Methods for Prediction

CMPUT 365
Fall 2021

Admin

- Announcement about projects. Email me if you want to do one of the projects
  - compare Monte Carlo and Sarsa, in Mountain Car

Review of Course 2, Module 2
TD Learning

Video 1: What is Temporal Difference Learning?

- 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.
- *What is weird or at least unique about temporal difference learning compared with other ML methods?*

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 $V(\mathit{terminal}) = 0$

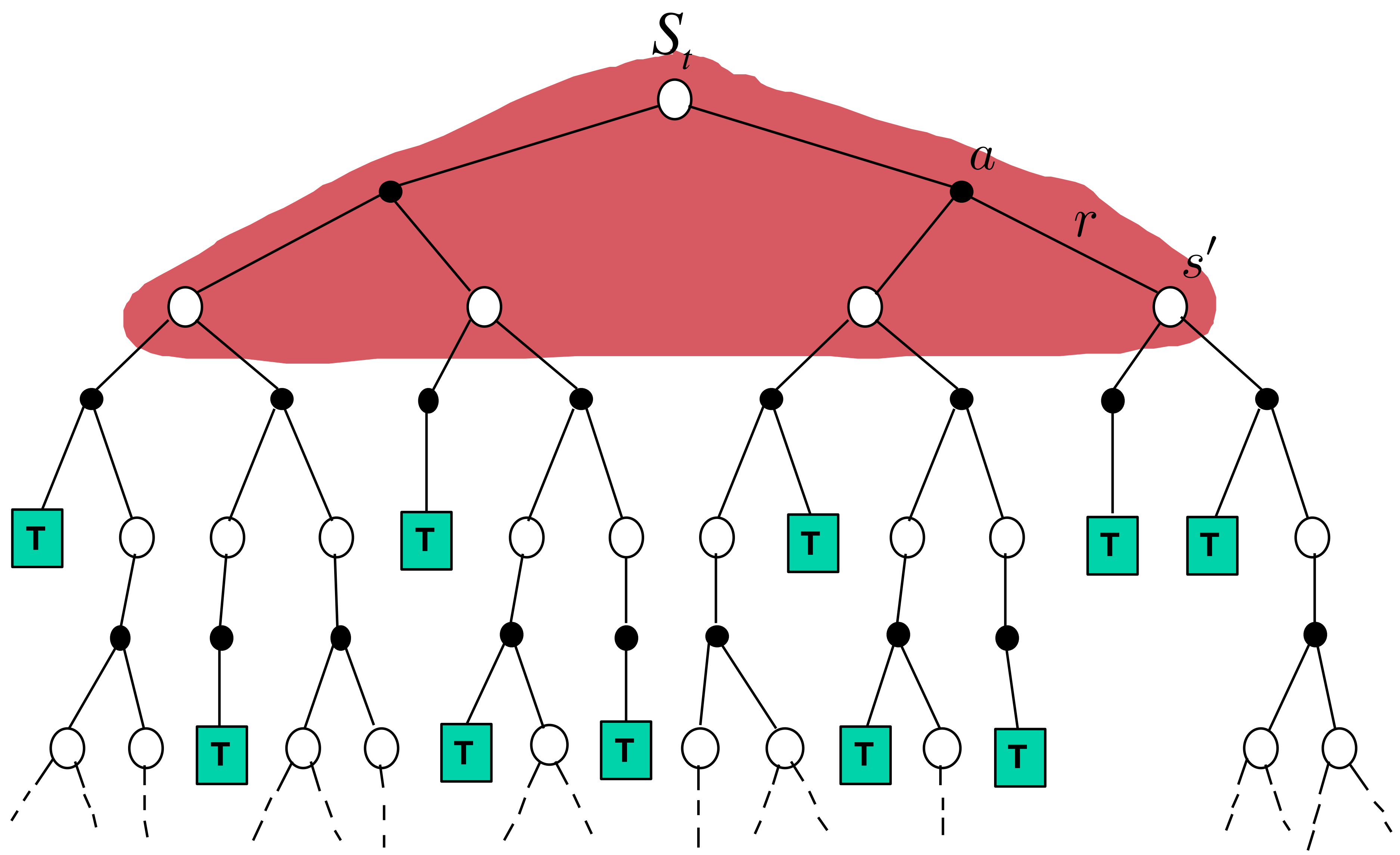
Loop for each episode:
    Initialize S
    Loop for each step of episode:
        $A \leftarrow$ action given by π for S
        Take action A, observe R, S'
        $V(S) \leftarrow V(S) + \alpha\big[R + \gamma V(S') - V(S)\big]$
        $S \leftarrow S'$
    until S is terminal

Video 2: The Advantages of TD Learning

- 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
  - *Where did TD come from? Is there a connection to neuroscience or animal learning?*

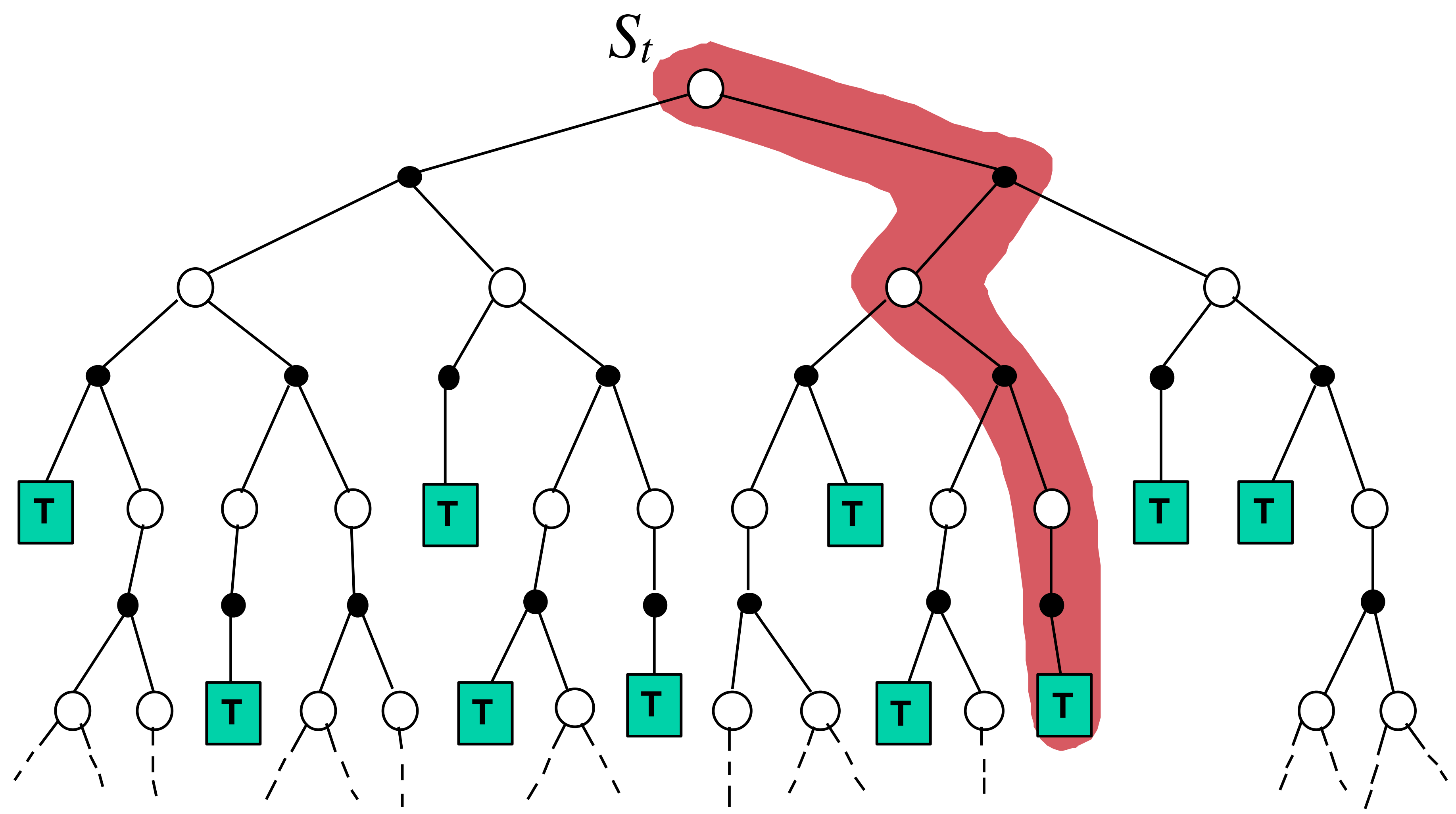
Dynamic programming

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



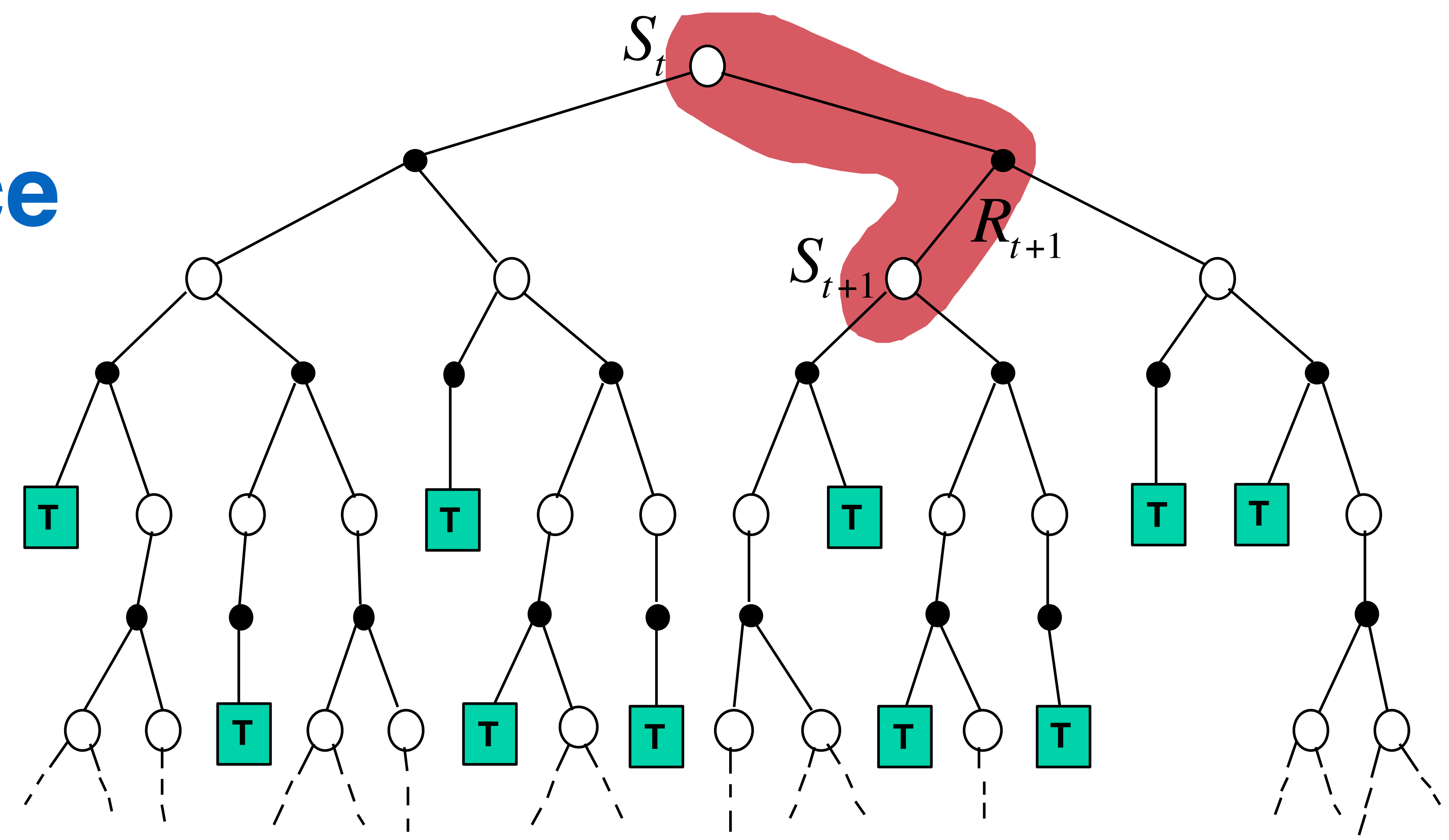
Simple Monte Carlo

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



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

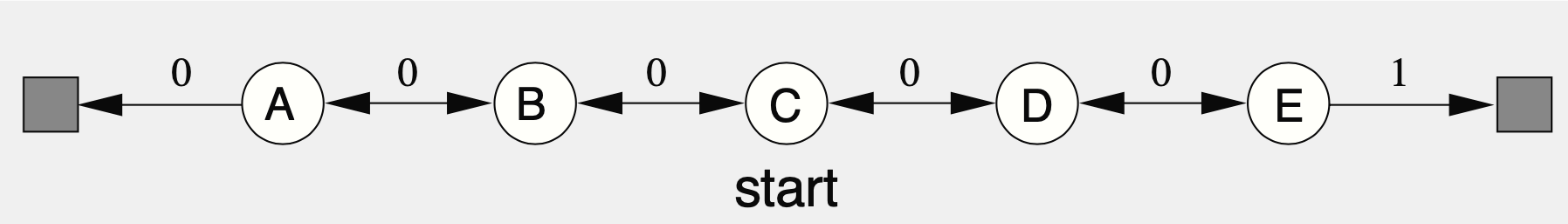
Temporal Difference Learning



Video 3: Comparing TD and Monte Carlo

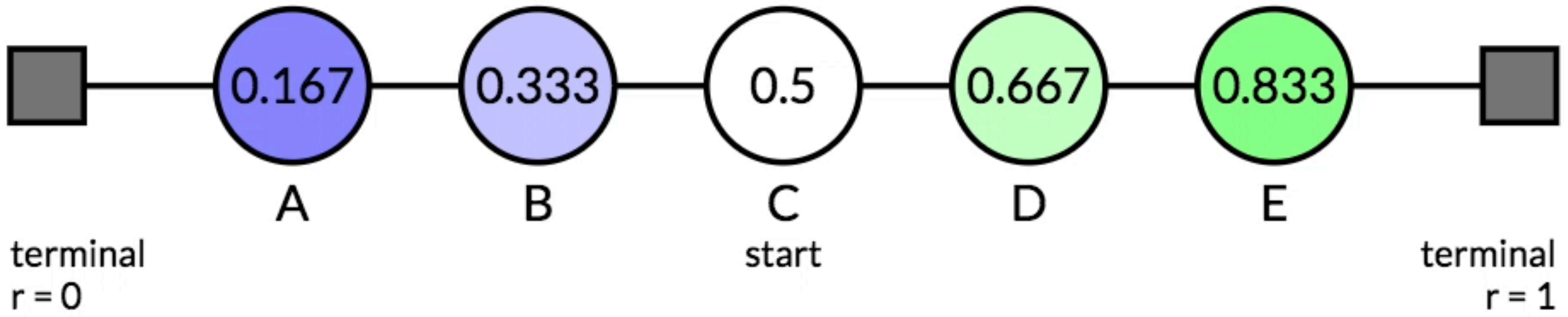
- Worked through an example using TD and Monte Carlo to learn v_π. We looked at how the updates happened on each step. And final performance via learning curves
- Goals:
  - Identify the empirical benefits of TD learning.
- *How can we understand the empirical advantages of TD over MC empirically? Let's look at some experimental results to better understand …*

A Random Walk problem

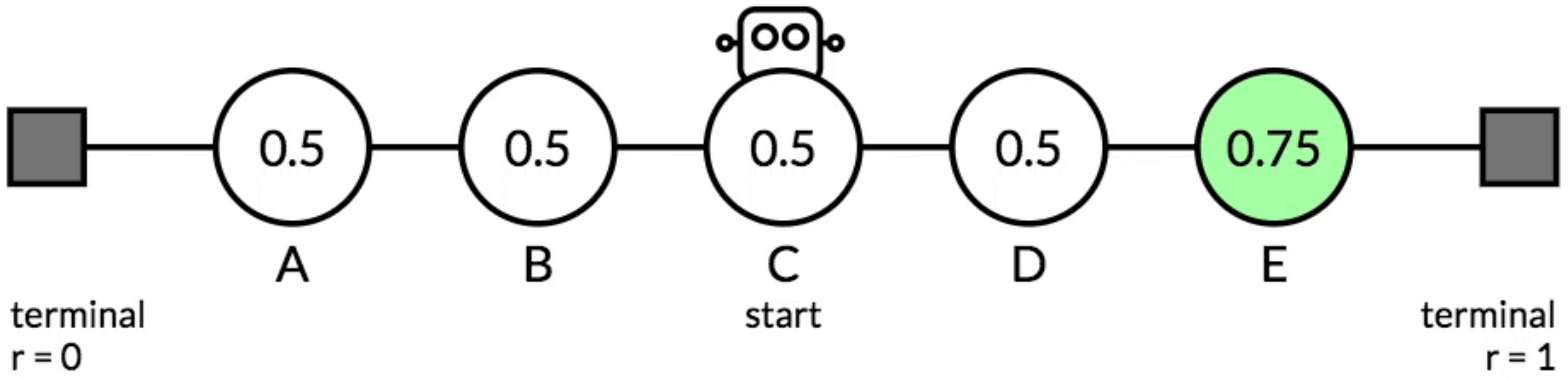


- Episodic; \gamma = 1.0
- Start in the centre
- Reward = 1 only on EXIT RIGHT
- What is the policy \pi?
- Goal: estimate v_\pi
    - What does v_\pi encode in this problem?

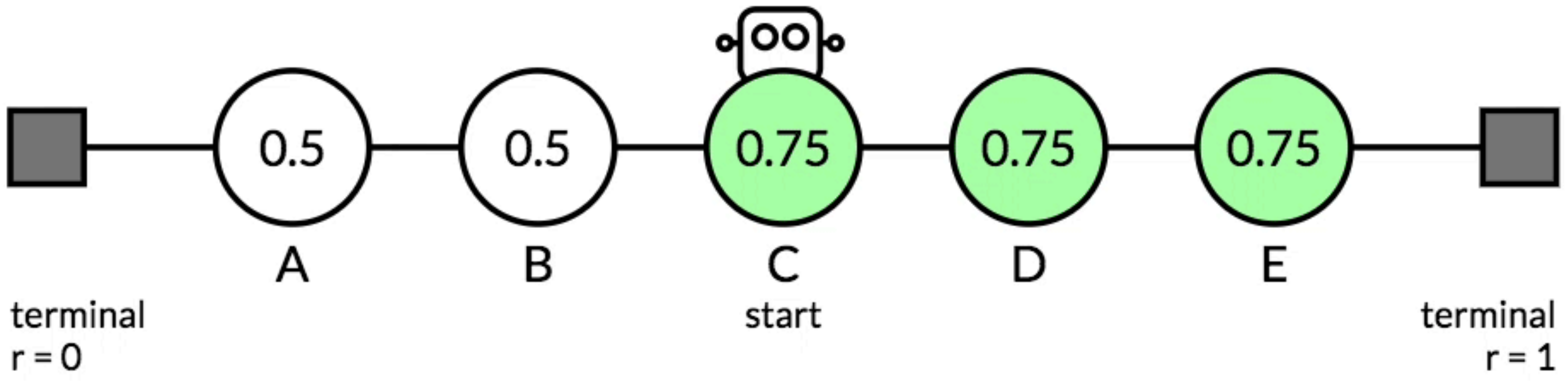
Target / Exact Values



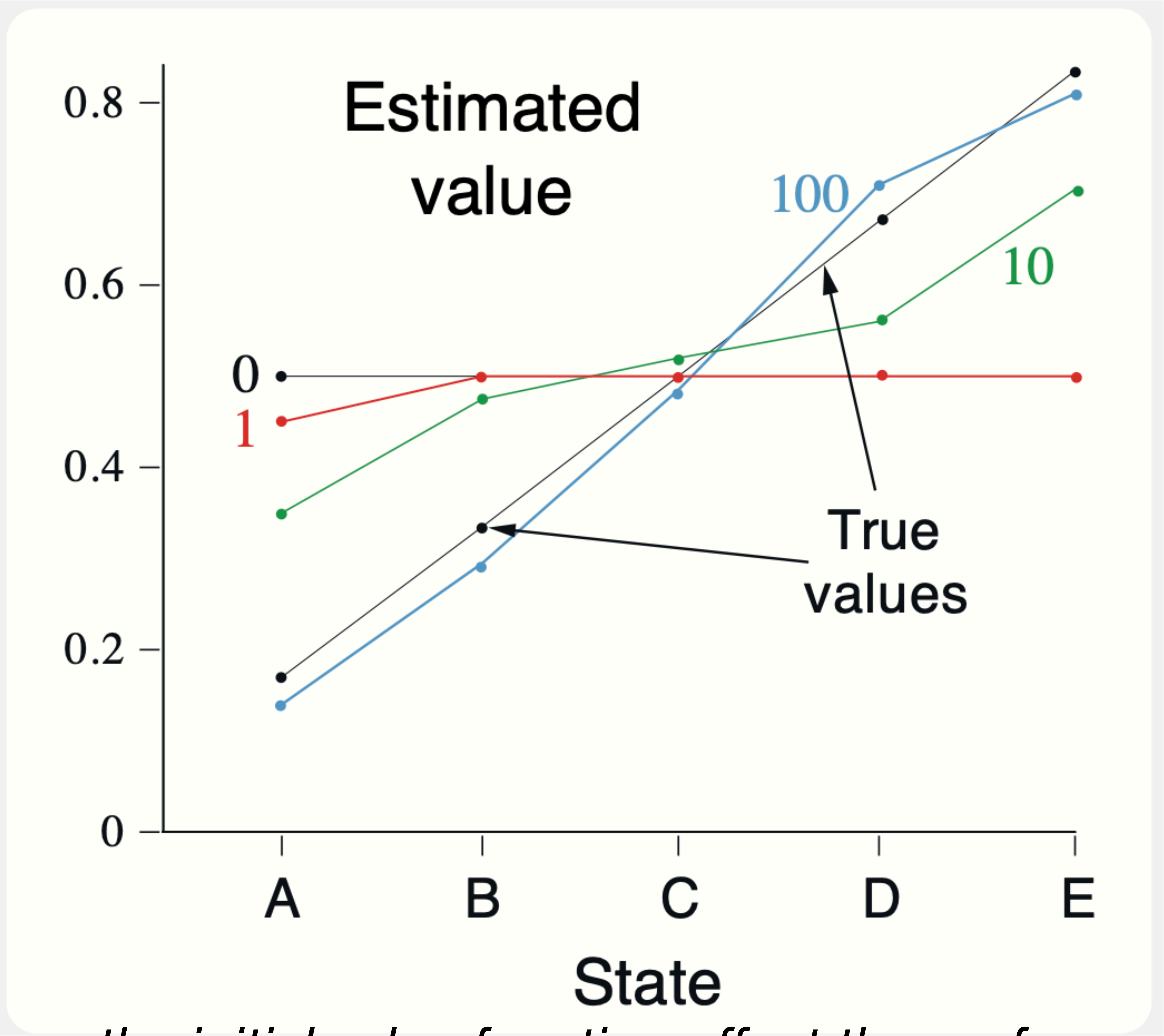
Updates using TD Learning



Updates using Monte Carlo

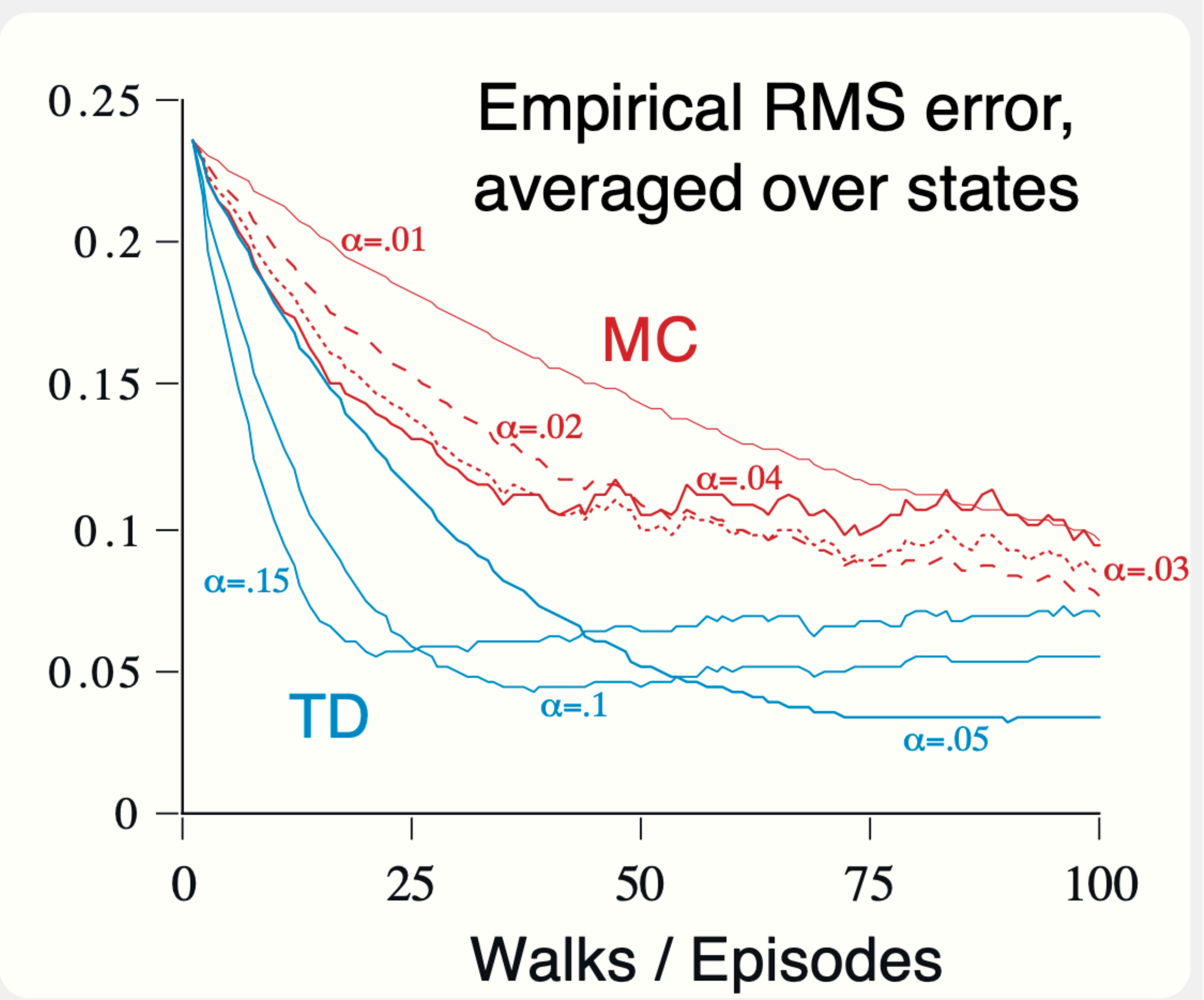


A Random Walk problem



- *In TD learning, does the initial value function effect the performance of the algorithm? Hint: look at the black line labelled '0'*

A Random Walk problem

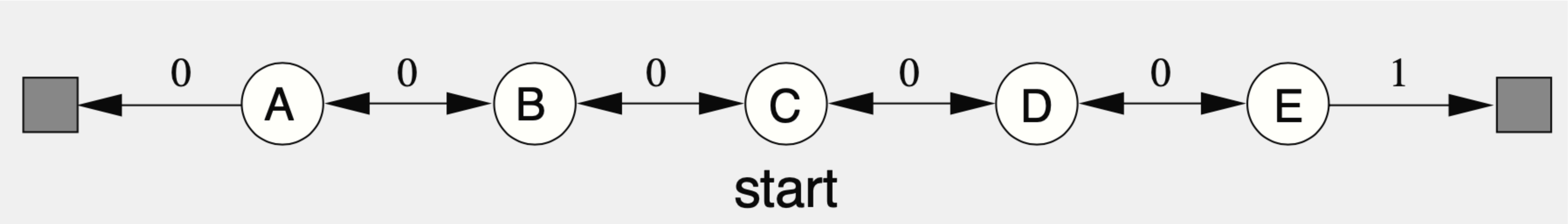


- *Why does the blue alpha=0.15 line go down fastest, but level off at a higher error?*

Back to our question

- *How can we understand the empirical advantages of TD over MC empirically?*
    - Let's think of the update targets for each:
        - MC: $V(S_t) = V(S_t) + \alpha [\mathbf{G_t} - V(S_t)]$
        - TD: $V(S_t) = V(S_t) + \alpha [\mathbf{R_{t+1} + \gamma V(S_{t+1})} - V(S_t)]$
        - $\mathrm{Var}[\mathbf{R_{t+1} + \gamma V(S_{t+1})}] < \mathrm{Var} [\mathbf{G_t}]$
- *When might MC be better empirically than TD?*

When might MC be better empirically than TD?



- Consider the Random Walk problem, estimate v_\pi, and \pi = always go right

- What is the return of the first episode? G = 1

- V(S_t) = V(S_t) + \alpha [**G_t** - V(S_t)]

  - *MC gets the value function correct after one episode! If alpha=1*

- What about TD?   V(S_t) = V(S_t) + \alpha [**R_{t+1} + \gamma V(S_{t+1})** - V(S_t)]

  - *How many episodes would it take TD to get the value function correct?*

- The variance of the **one-step TD target** is not lower than the variance of the **return**

  - In this case TD is slowed down by the initially incorrect values in the target. Bootstrapping hurts!

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: $V(S) \leftarrow V(S) + \alpha\underbrace{[R + \gamma V(S') - V(S)]}_{\text{TD-error}}$

- 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