Gole

Learn V_π of a given policy π in a model-free environment.

We will see two approaches:

1. Monte Carlo
2. Temporal difference

Monte Carlo Learning

For each E_n for each S_t^n.

1. $T_{n+1}(S_t^n) = T_n(S_t^n) + G_{t:T}^n$ for rest $T_{n+1}(s) = T_n(s)$
2. $N_{n+1}(S_t^n) = N_n(S_t^n) + 1$ for rest $N_{n+1}(s) = N_n(s)$
3. $V_n(S_t) = \frac{T_n(S_t)}{N_n(S_t)}$ for rest $V_{n+1}(s) = V_n(s)$

When $N(s) \to \infty, V(s) \to v_\pi(s)$.

Incremental Mean

The mean μ_1,μ_2,... of a serquence x_1, x_2, x_3,... can be camputed incrementally.

$$
\begin{aligned}
\mu_k &= \frac{1}{k}\sum_{j=1}^{k} x_j = \frac{1}{k}\left(x_k + \sum_{j=1}^{k-1} x_j\right) \\
&= \frac{1}{k}(x + k + (k-1)\mu_{k-1}) \\
&= \mu_{k-1} + \frac{1}{k}(x_k - \mu_{k-1})
\end{aligned}
$$

Incremental Monte-Carlo Updates

For each E_n for each S_t^n.

1. $N_{n+1}(S_t^n) = N_n(S_t^n) + 1$ for rest $N_{n+1}(s) = N_n(s)$
2. $V_{n+1}(S_t) = V_n(S_t) + \frac{G_{t:T}^n - V_n(S_t)}{N_n(S_t)}$ for rest $V_{n+1}(s) = V_n(s)$

or

2. $V_{n+1}(S_t) = V_n(S_t) + \alpha(G_{t:T}^n - V_n(S_t))$
   for rest $V_{n+1}(s) = V_n(s)$

Temporal-difference Learning

- Update value $V_n(S_t)$ toward actual return G_t.

$$V_{n+1}(S_t) = V_n(S_t) + \alpha(G^n_{t:T} - V_n(S_t))$$

- Temporal-difference learning algorithm: TD(0)
  - Update value $V_n(S_t)$ toward estimated return $R^n_t + \gamma V_n(S_{t+1})$

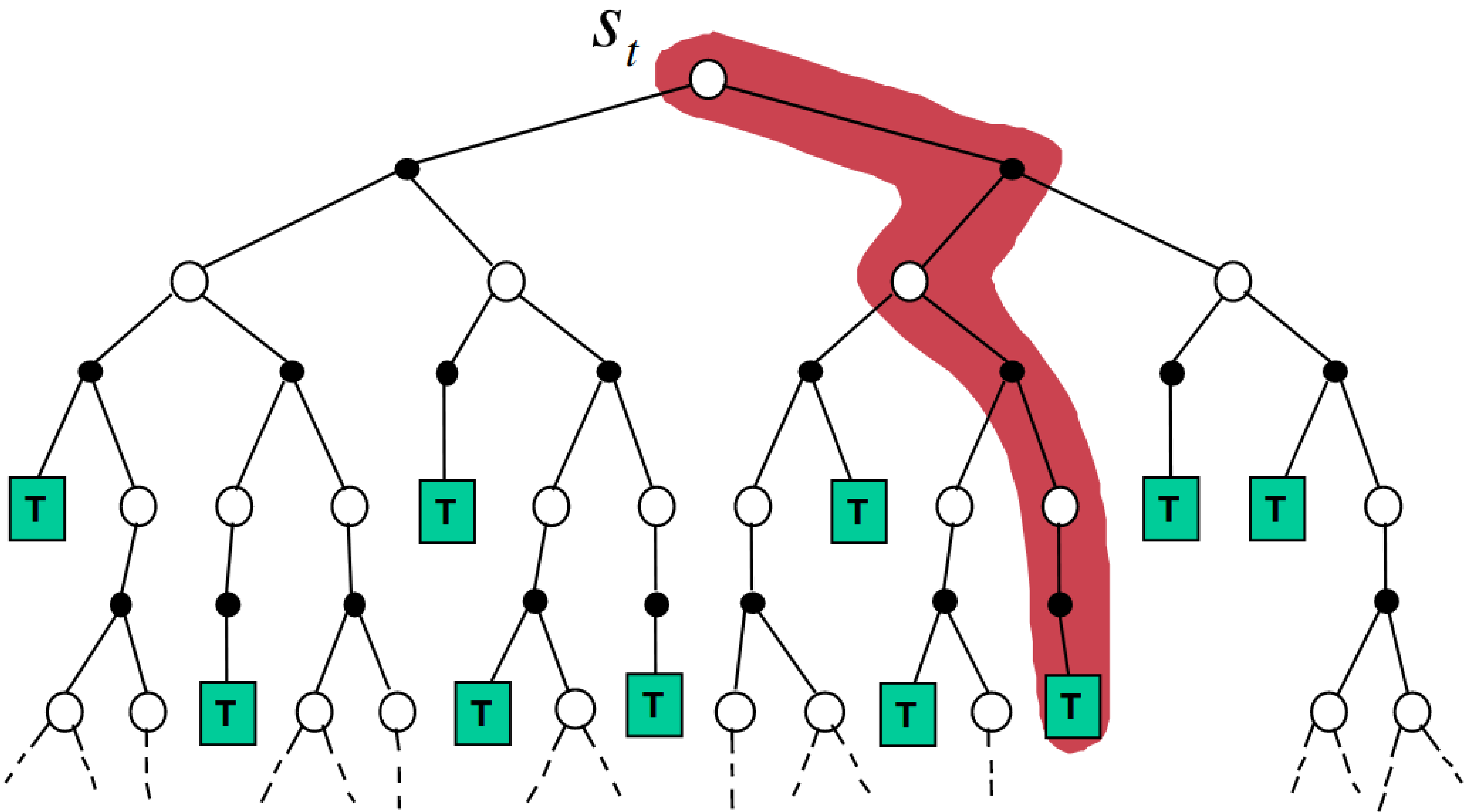
$$V_{n+1}(S_t) = V_n(S_t) + \alpha(\overbrace{\underbrace{R^n_t + \gamma V_n(S_{t+1})}_{\text{TD target}} - V_n(S_t)}^{\text{TD error}})$$

Advantages and Disadvantages of MC vs. TD

- TD can learn before knowing the final outcome.
  - TD can learn online after every step.
  - MC must wait until end of episode before return is known.
- TD can learn without the final outcome
  - TD can learn from incomplete sequences
  - MC can only learn from complete sequences
  - TD works in continuing (non-terminating) environments
  - MC only works for episodic (terminating) environments

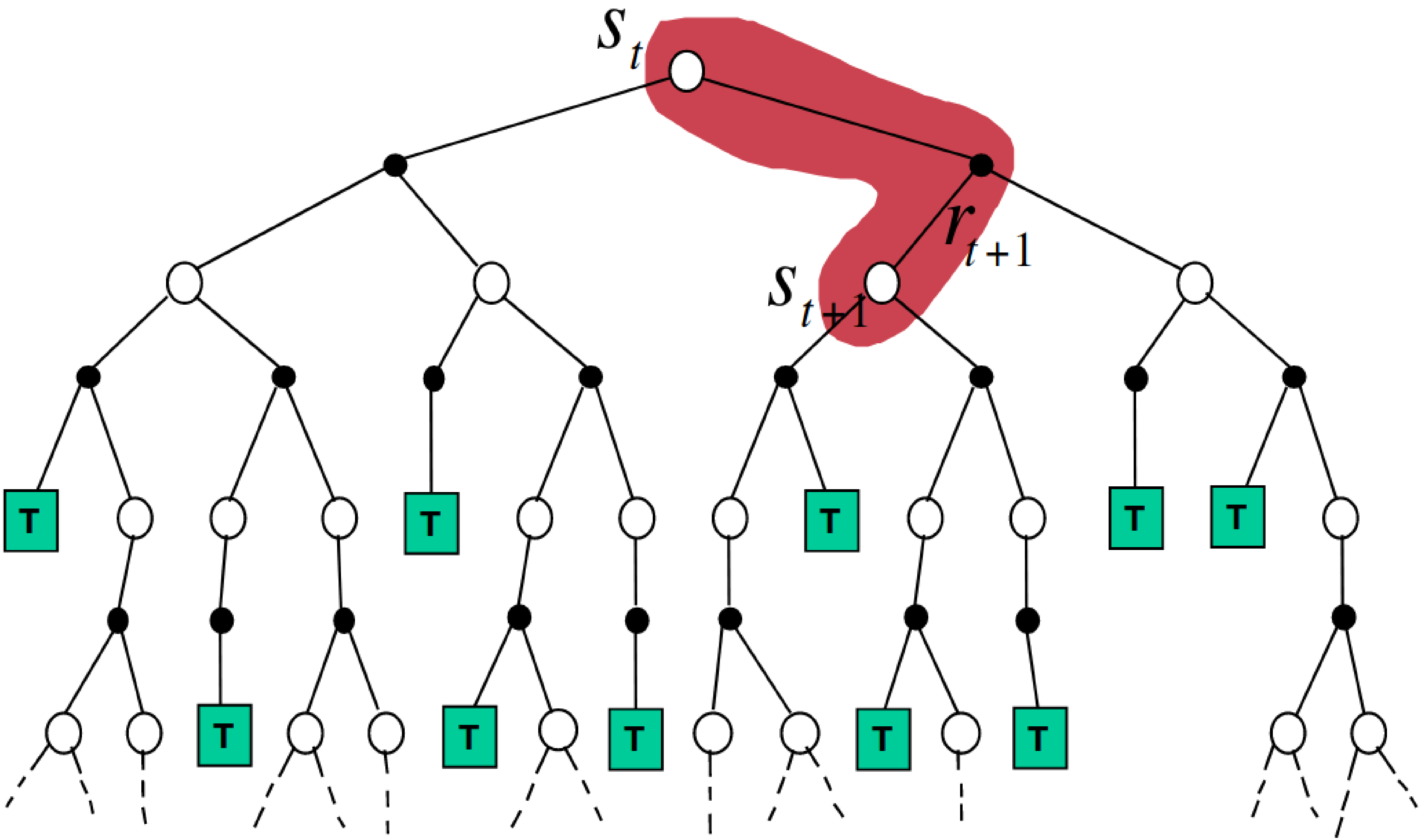
MC

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



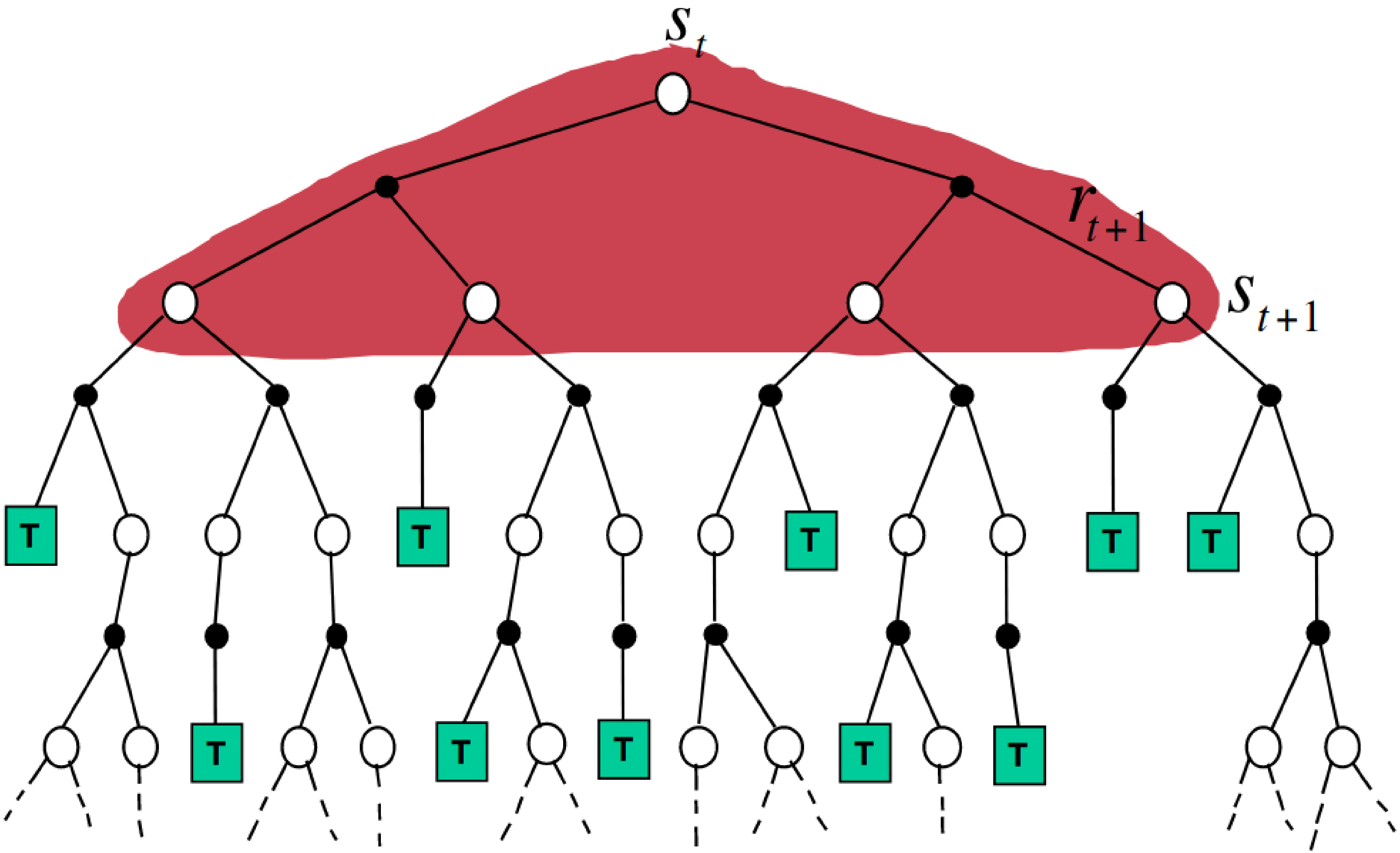
$$V(S_t) \leftarrow V(S_t) + \alpha \left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \right)$$

TD



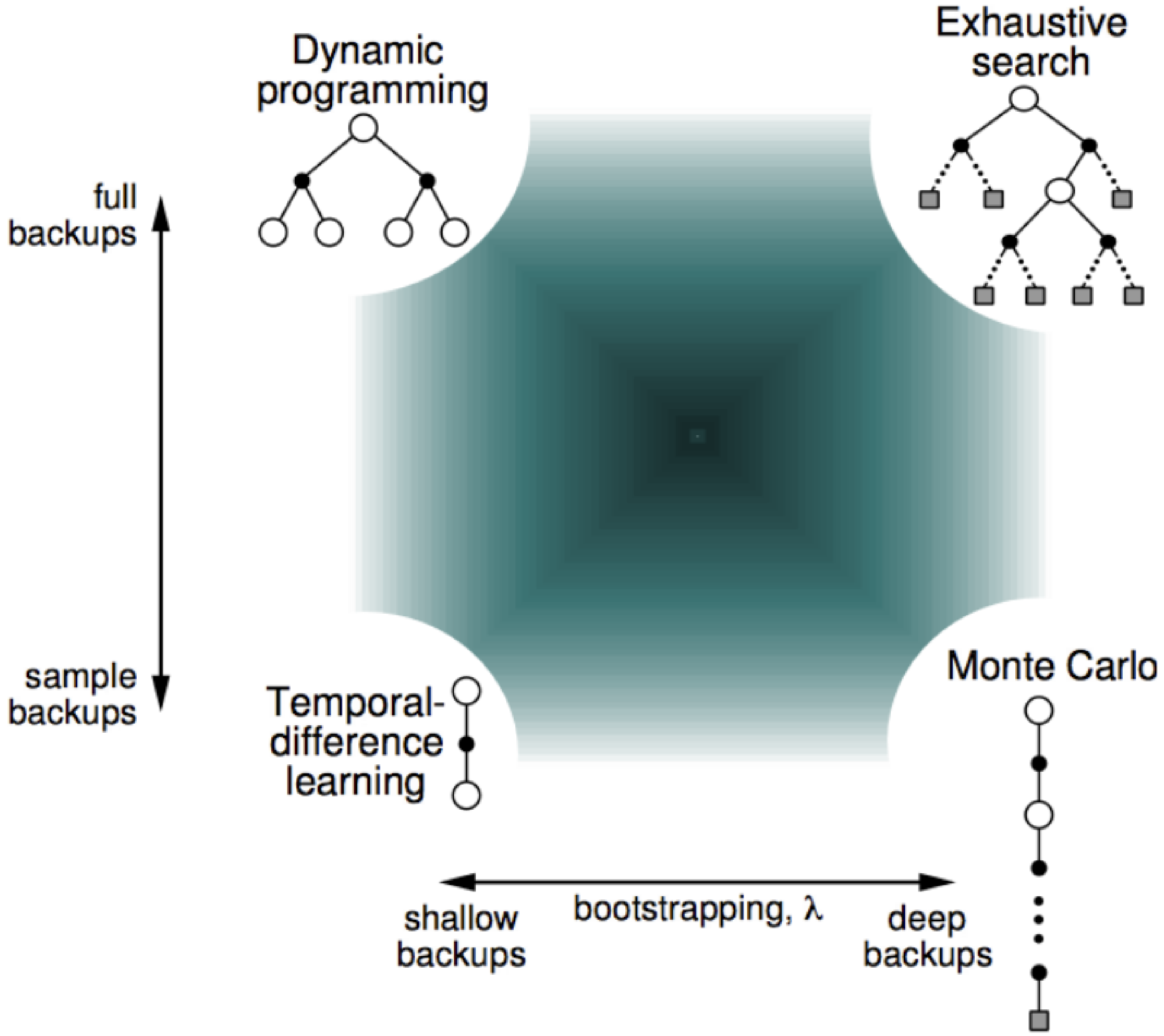
$$V(S_t) \leftarrow \mathbb{E}_\pi \left[R_{t+1} + \gamma V(S_{t+1})\right]$$

DP



Bootstrapping and Sampling

- Bootstrapping: update involves an estimate
  - MC does not bootstrap
  - DP bootstraps
  - TD bootstraps
- Sampling: update samples an expectation
  - MC samples
  - DP does not sample
    - TD samples



Dynamic programming
Exhaustive search
full backups
sample backups
Temporal-difference learning
Monte Carlo
shallow backups
bootstrapping, λ
deep backups

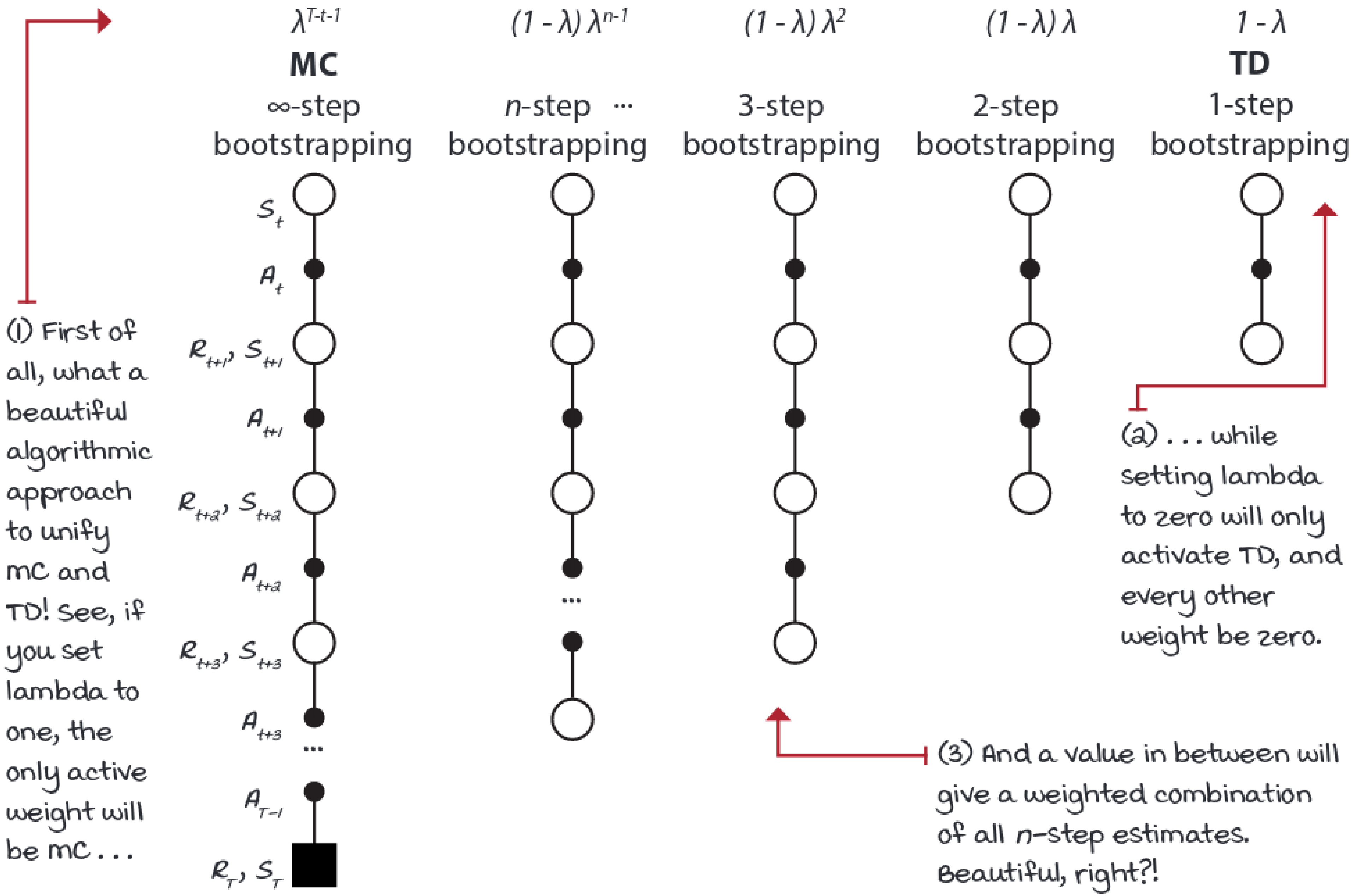
N-Step temporeal-difference learning

Given a E_n we define

$$G_{t:t+n} := R_t + \ldots + \gamma^{m-1} R_{t+n} + \gamma^n V_{t+n-1}(S_{t+n})$$

$$V_{n+1}(S_t) = V_n(S_t) + \alpha(\underbrace{\overbrace{\color{red}{G_{t:t+n}}}^{n-step\ target} - V_n(S_t)}_{n-step\ error})$$

Generalized bootstrapping



TD(λ)

$$G_{t:T}^{\lambda} := (1 - \lambda) \sum_{n=1}^{T-t} \lambda^{n-1} G_{t:t+n}$$

$$V_{n+1}(S_t) = V_n(S_t) + \alpha \underbrace{\left(\overbrace{\color{red}{G_{t:T}^{\lambda}}}^{\lambda\text{-return}} - V_n(S_t) \right)}_{\lambda\text{-error}}$$

TD(λ) Backward-view

1. Set $e_0(s) = 0 \ \forall s \epsilon \mathbb{S}$ every new episode.
2. When we encounter a state s. $e_t(s) = e_{t-1}(s) + 1$
3. $\delta^{\text{TD}}_{t:t+1} = R_{t+1} + \gamma V_t(S_{t+1}) - V_t(S_t)$
4. $V_{t+1} = V_t + \alpha \delta^{\text{TD}}_{t:t+1} e_t$
5. $e_{t+1} = e_t \gamma \lambda$

Thanks