"Where did you go to, if I may ask?" said Thorin to Gandalf as they rode along.

"To look ahead," said he.

"And what brought you back in the nick of time?"

"Looking behind," said he.

J.R.R. Tolkien, *The Hobbit*

CMPUT 628 Deep RL

Marlos C. Machado

Class 8 & 9/ 25

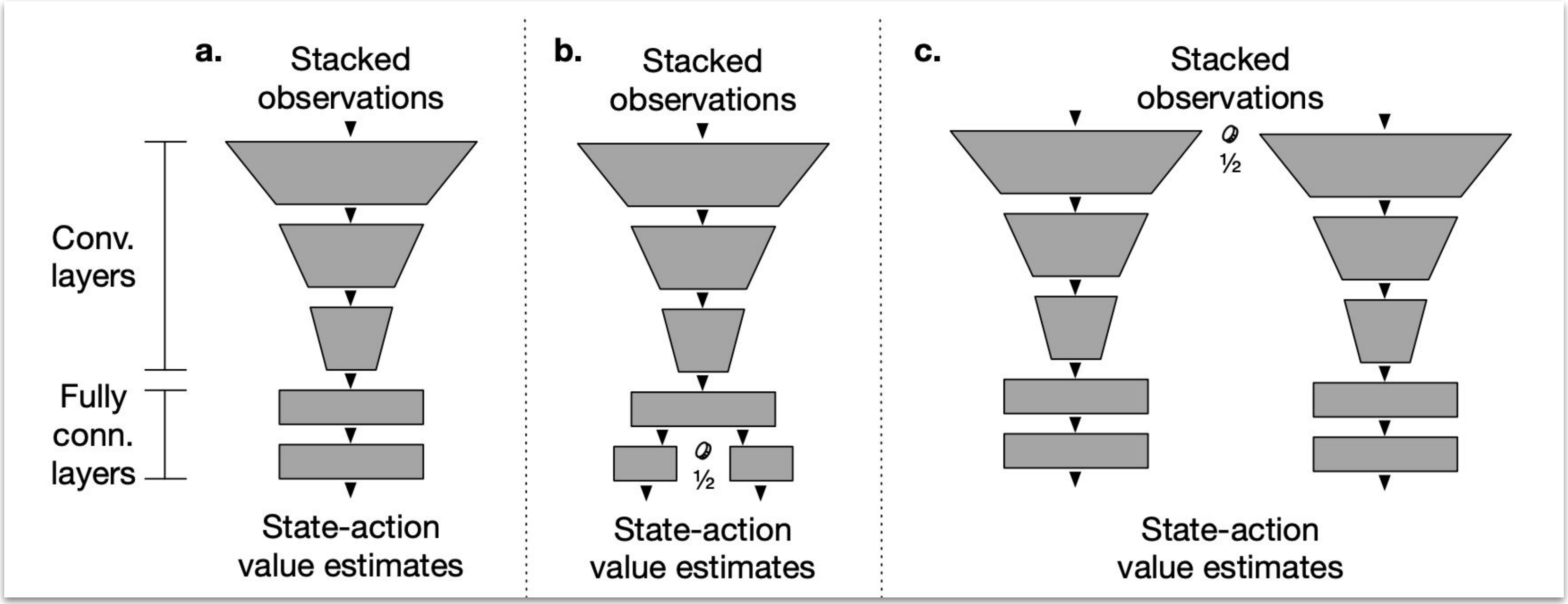
Reminders & Notes

- Assignment 2 is out, let me know how it is going next week.
- I will release instructions about seminar and paper review during the reading week (Feb 18 – Feb 21)
  You should start thinking about groups, though

Please, interrupt me at any time!

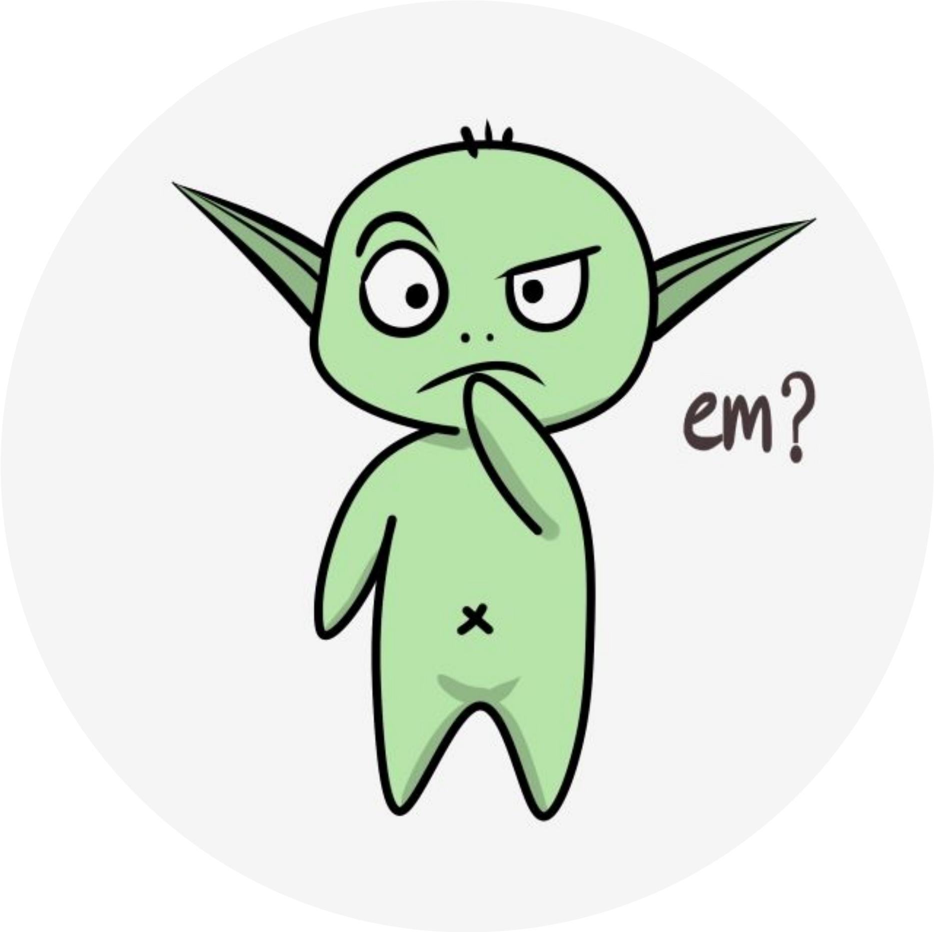
https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Last class: Double Learning in Deep RL



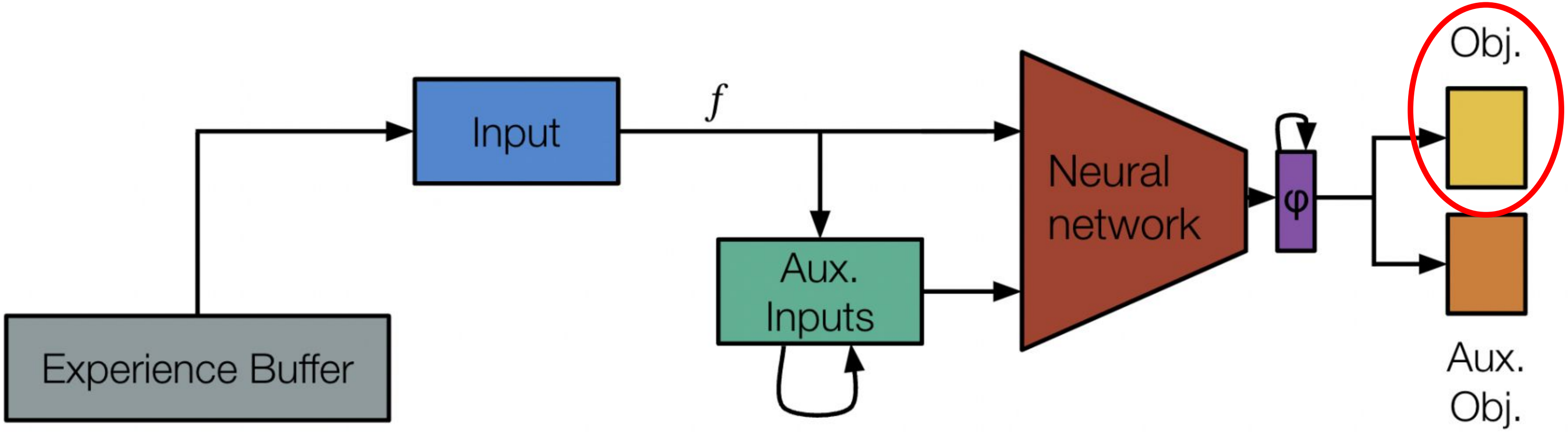
$$Y(R_{t+1}, O_{t+1}; \boldsymbol{\theta}_t) = R_{t+1} + \gamma Q\big(O_{t+1}, \arg\max_{a \in \mathcal{A}} Q(O_{t+1}, a; \boldsymbol{\theta}_t); \boldsymbol{\theta}^-\big)$$

$$\mathcal{L}^{\text{DDQN}} = \mathbb{E}_{\tau \sim U(\mathcal{D})}\Big[\big(Y(R_{t+1}, O_{t+1}; \boldsymbol{\theta}_t) - Q(O_t, A_t; \boldsymbol{\theta}_t)\big)^2\Big],$$



https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

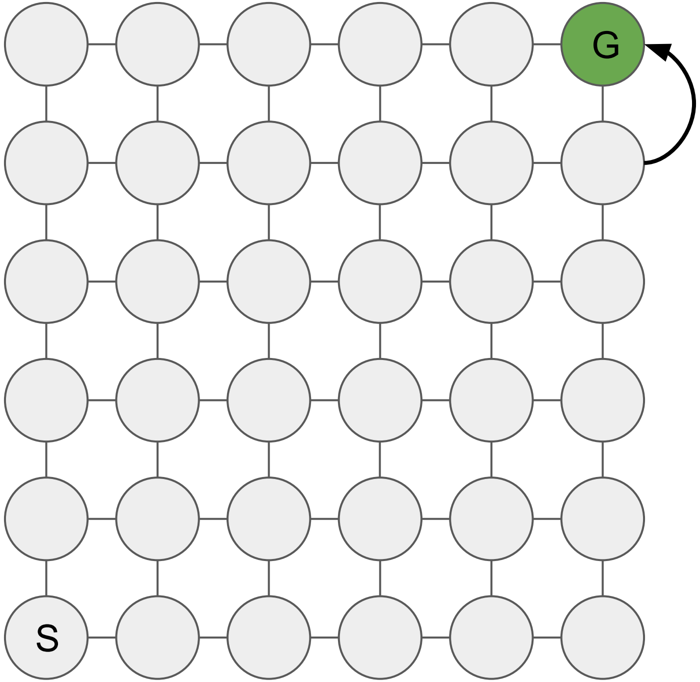
We Continue to Look at Different Objective Functions



This is where we can more easily incorporate RL knowledge into deep RL

Multi-step methods

- Pretty much everything we discussed so far are variations of 1-step TD learning

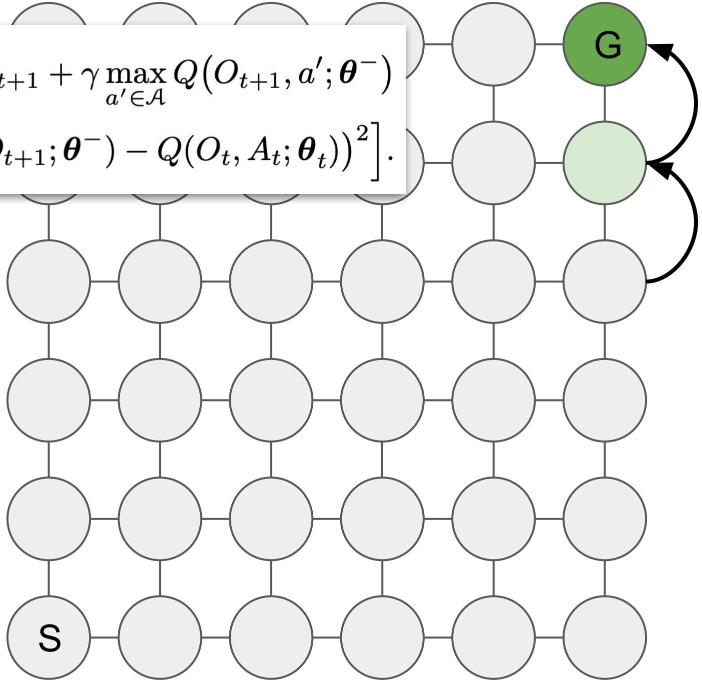


Multi-step methods

- Pretty much everything we discussed so far are variations of 1-step TD learning

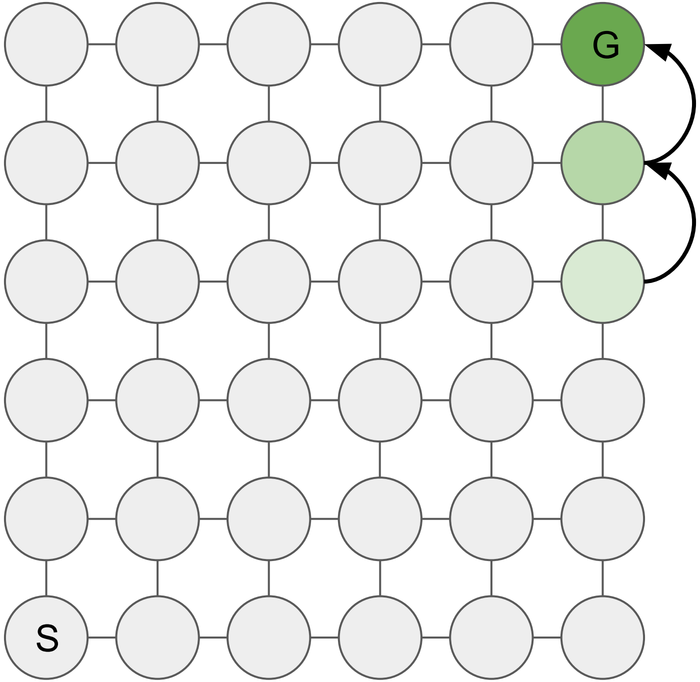
$$Y(R_{t+1}, O_{t+1}; \boldsymbol{\theta}^-) = R_{t+1} + \gamma \max_{a' \in \mathcal{A}} Q\big(O_{t+1}, a'; \boldsymbol{\theta}^-\big)$$

$$\mathcal{L}^{\text{DQN}} = \mathbb{E}_{(o,a,r,o') \sim U(\mathcal{D})}\Big[\big(Y(R_{t+1}, O_{t+1}; \boldsymbol{\theta}^-) - Q(O_t, A_t; \boldsymbol{\theta}_t)\big)^2\Big].$$



Multi-step methods

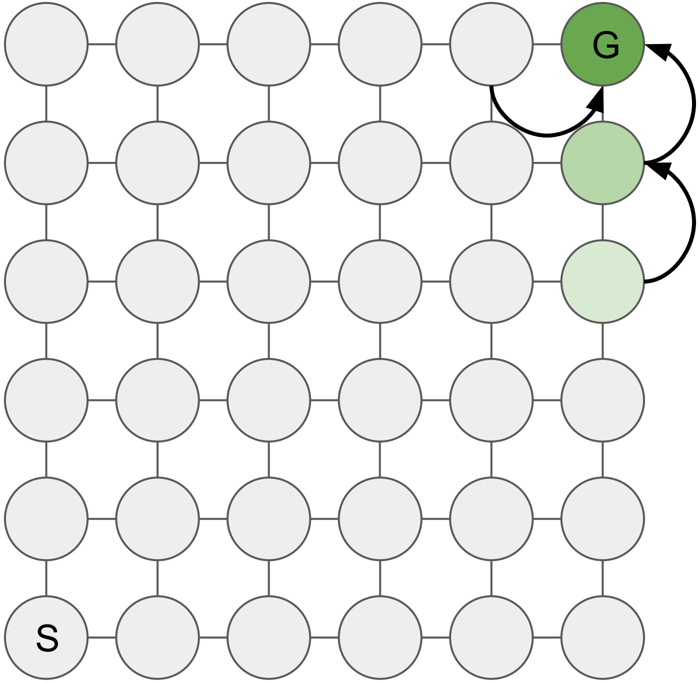
- Pretty much everything we discussed so far are variations of 1-step TD learning



But we have to always “connect” to the existing “chain”, otherwise we start from scratch

Multi-step methods

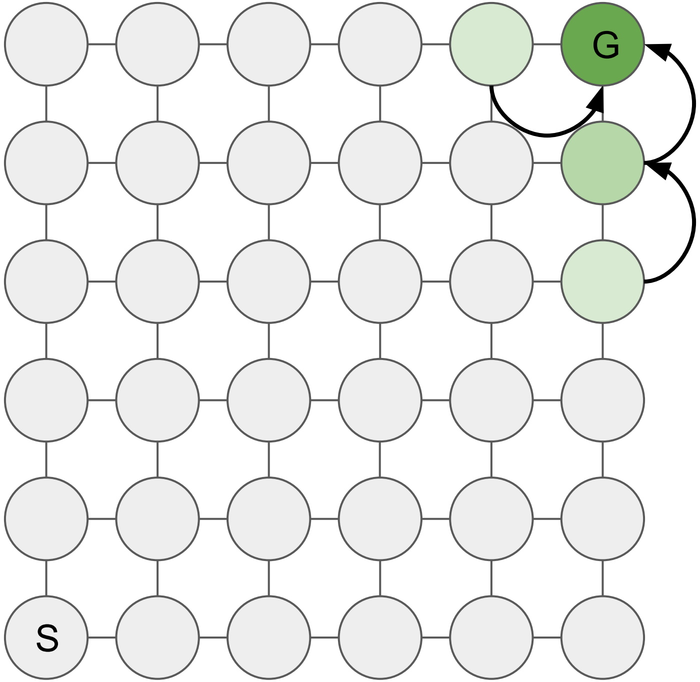
- Pretty much everything we discussed so far are variations of 1-step TD learning



But we have to always “connect” to the existing “chain”, otherwise we start from scratch

Multi-step methods

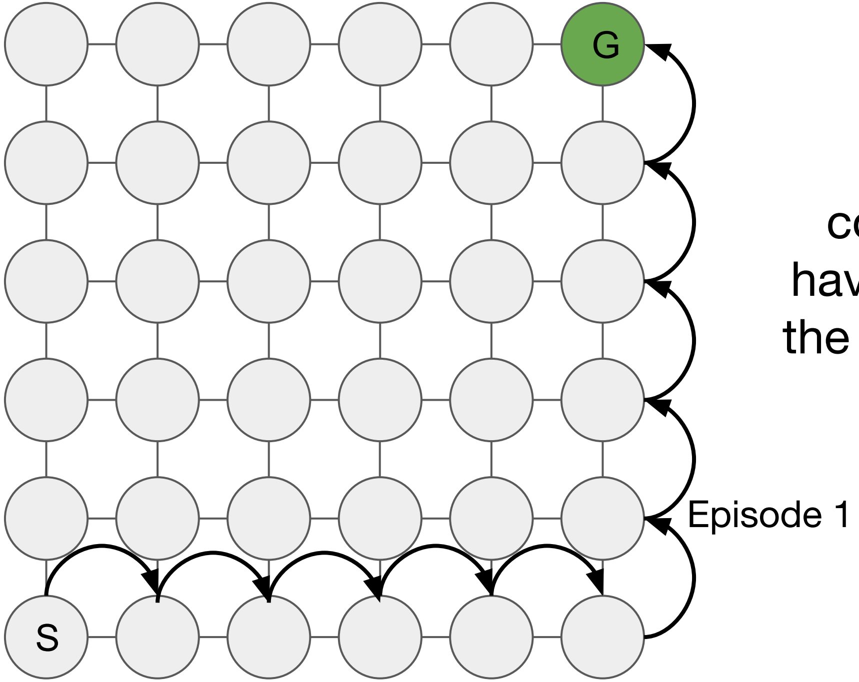
- Pretty much everything we discussed so far are variations of 1-step TD learning



And even if connecting, we have to connect at the right time/place

Multi-step methods

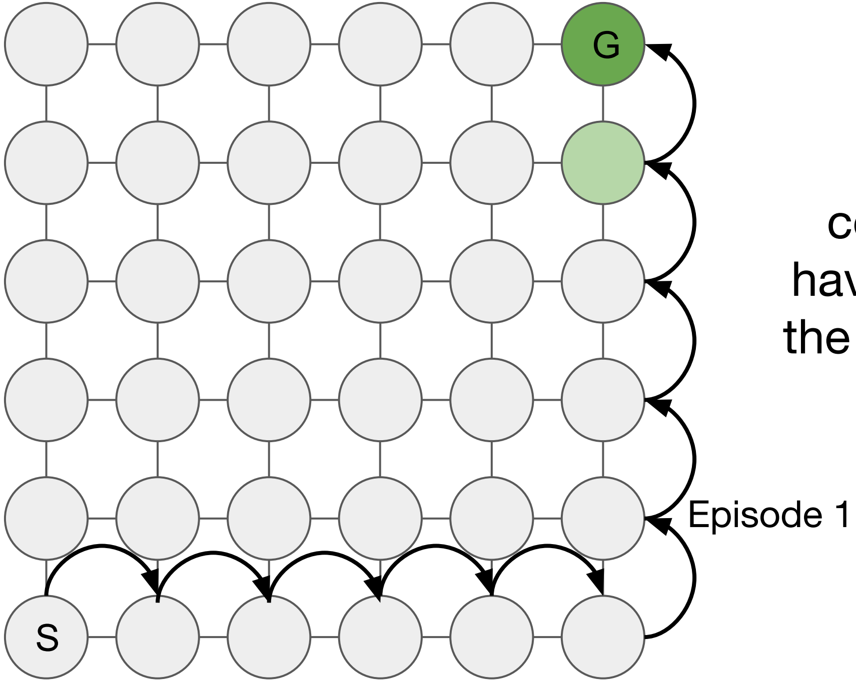
- Pretty much everything we discussed so far are variations of 1-step TD learning



And even if connecting, we have to connect at the right time/place

Multi-step methods

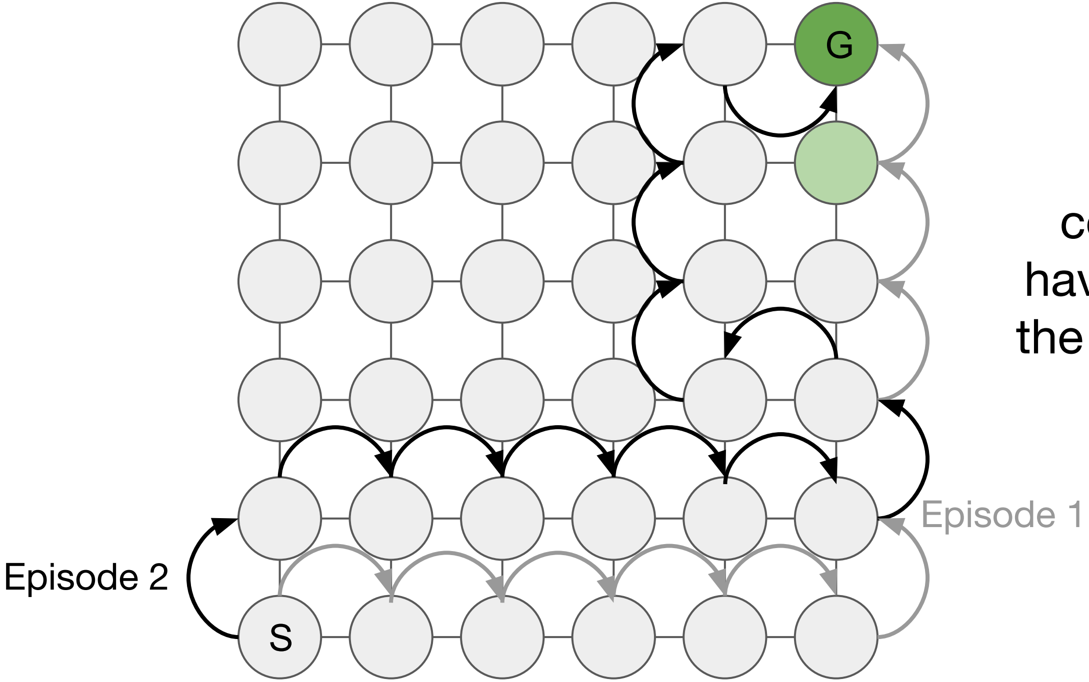
- Pretty much everything we discussed so far are variations of 1-step TD learning



And even if connecting, we have to connect at the right time/place

Multi-step methods

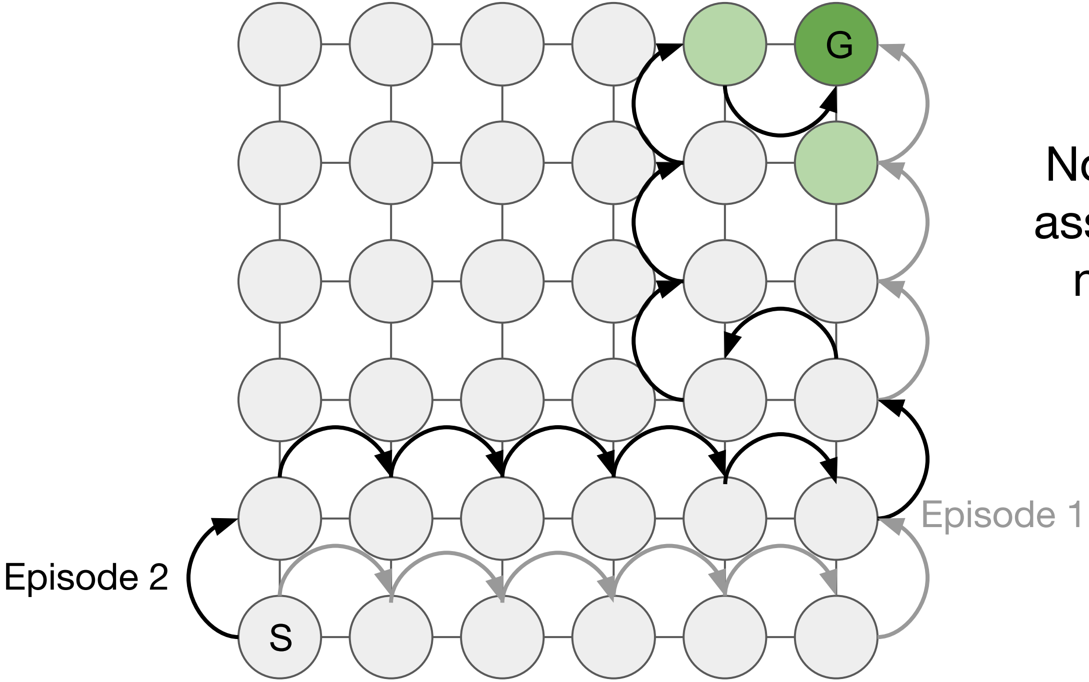
- Pretty much everything we discussed so far are variations of 1-step TD learning



And even if connecting, we have to connect at the right time/place

Multi-step methods

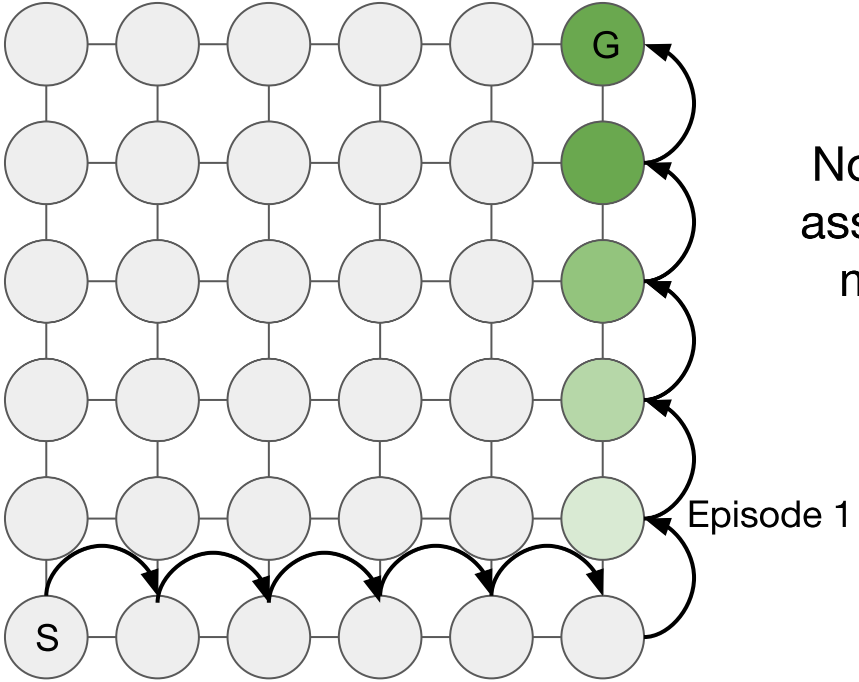
- Pretty much everything we discussed so far are variations of 1-step TD learning



Now, if we could assign credit over many steps at once…

Multi-step methods

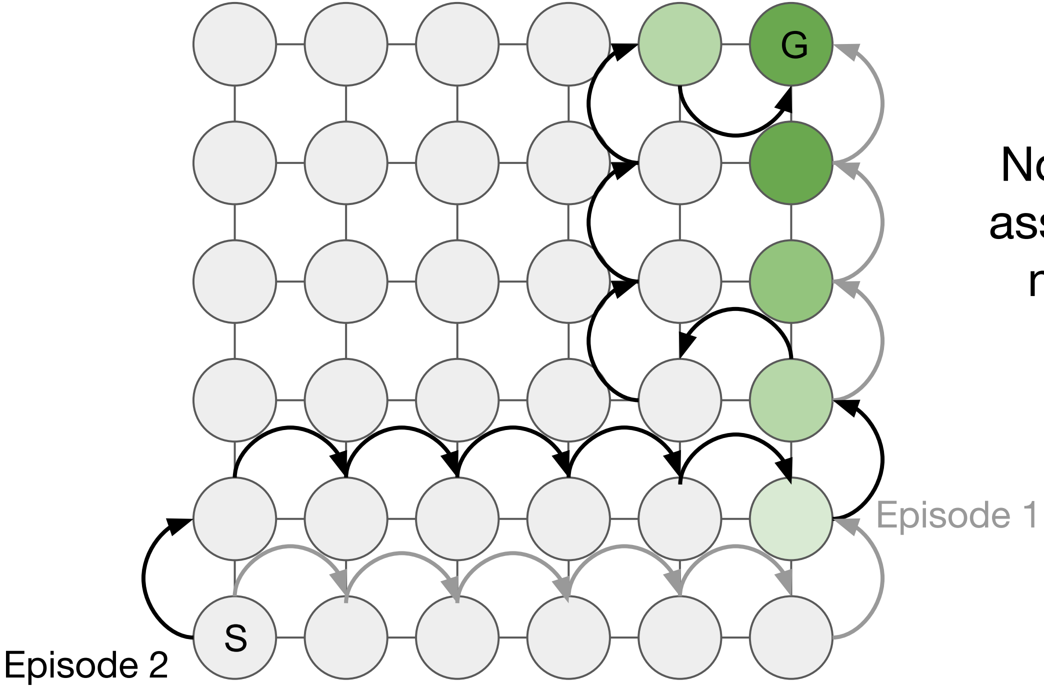
- Pretty much everything we discussed so far are variations of 1-step TD learning



Now, if we could assign credit over many steps at once…

Multi-step methods

- Pretty much everything we discussed so far are variations of 1-step TD learning

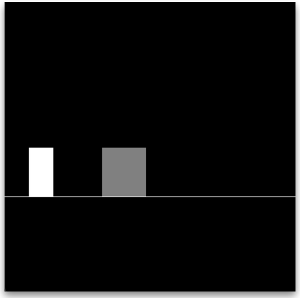


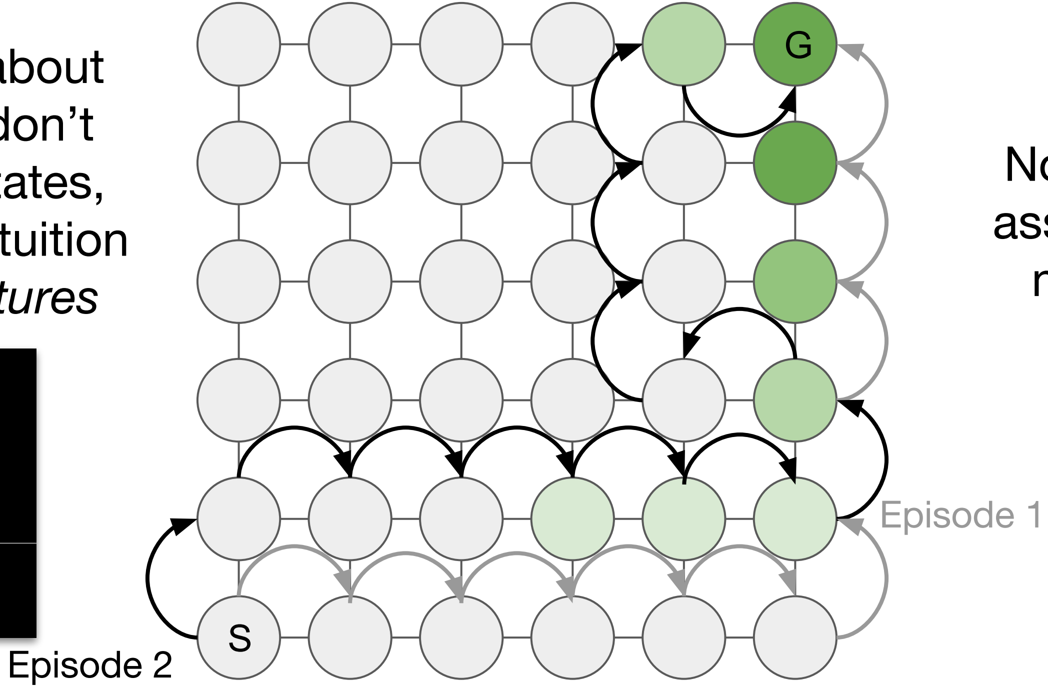
Now, if we could assign credit over many steps at once…

Multi-step methods

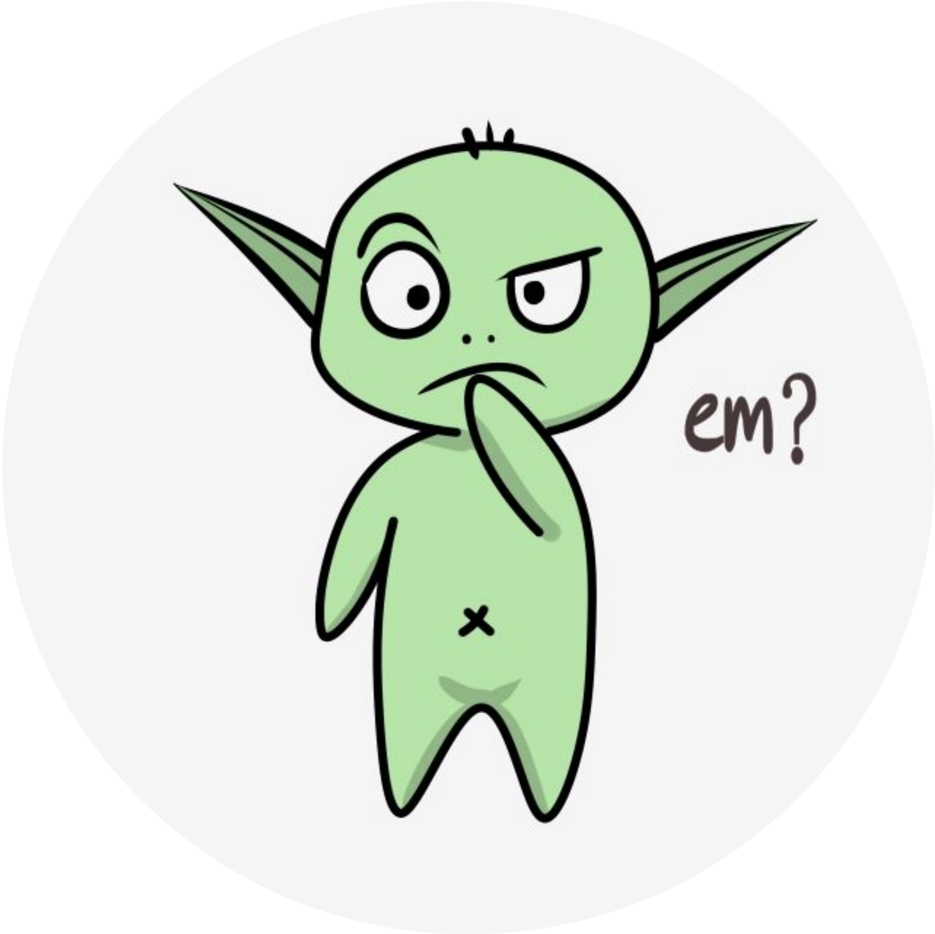
- Pretty much everything we discussed so far are variations of 1-step TD learning

When talking about deep RL, we don't have atomic states, but the same intuition holds with *features*





Now, if we could assign credit over many steps at once…



https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Reinforcement Learning 101

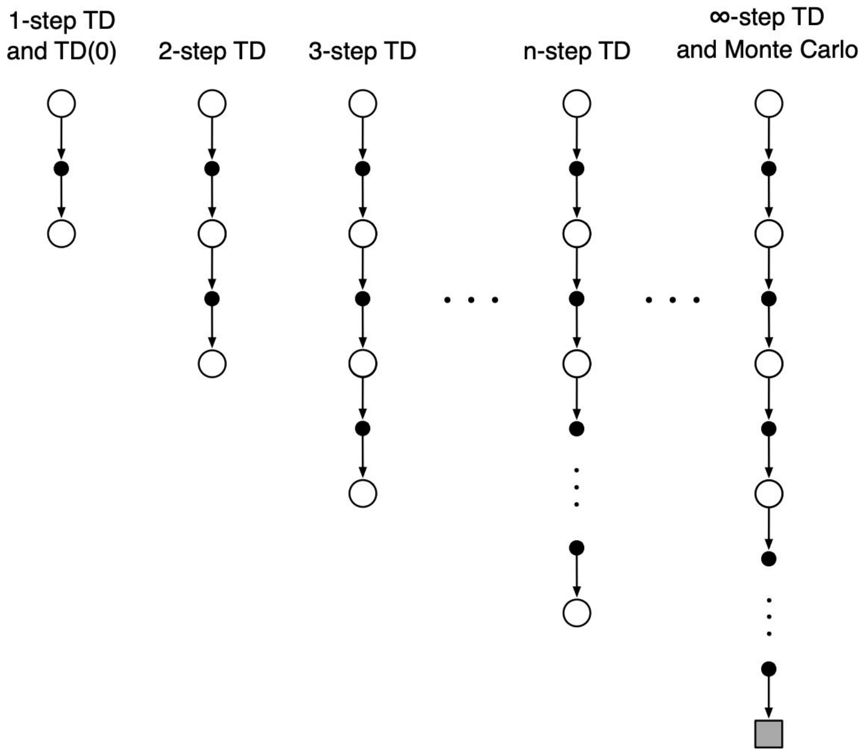


Image by Sutton & Barto (2018)

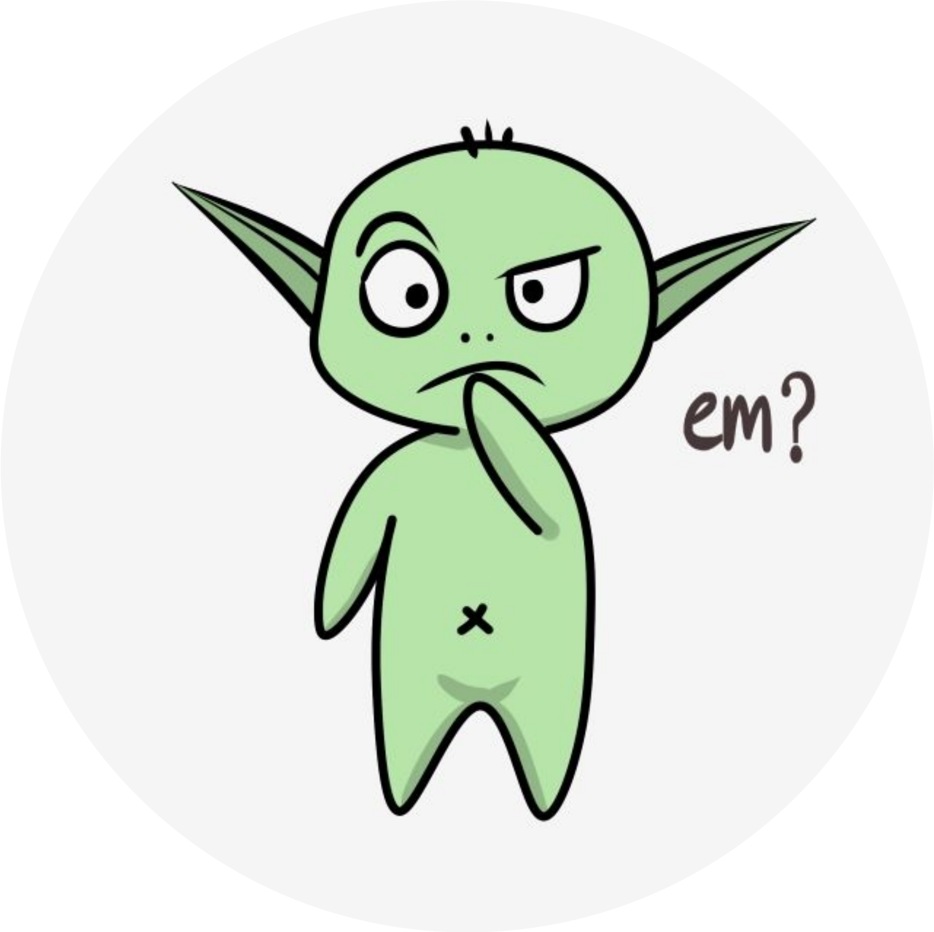
Reinforcement Learning 101

$$G_{t:t+1} \doteq R_{t+1} + \gamma V_t(S_{t+1})$$

$$G_{t:t+2} \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 V_{t+1}(S_{t+2})$$

$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n V_{t+n-1}(S_{t+n})$$

$$V_{t+n}(S_t) \doteq V_{t+n-1}(S_t) + \alpha \left[G_{t:t+n} - V_{t+n-1}(S_t) \right]$$



https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Monte Carlo returns (instead of TD error)

$$Y_{\mathrm{MC}}(R_{t+1:T}) = \sum_{t=0}^{T} \gamma^t R_{t+1}$$

$$\mathcal{L}_{\mathrm{MC}} = \mathbb{E}_{\tau^T \sim U(\mathcal{D})}\left[\left(Y_{MC}(R_{t+1:T}) - Q(O_t, A_t; \boldsymbol{\theta}_t)\right)^2\right]$$

We need some care with how we store (& sample) transitions in the exp. replay buffer

Unbiased, but high-variance

n-step returns

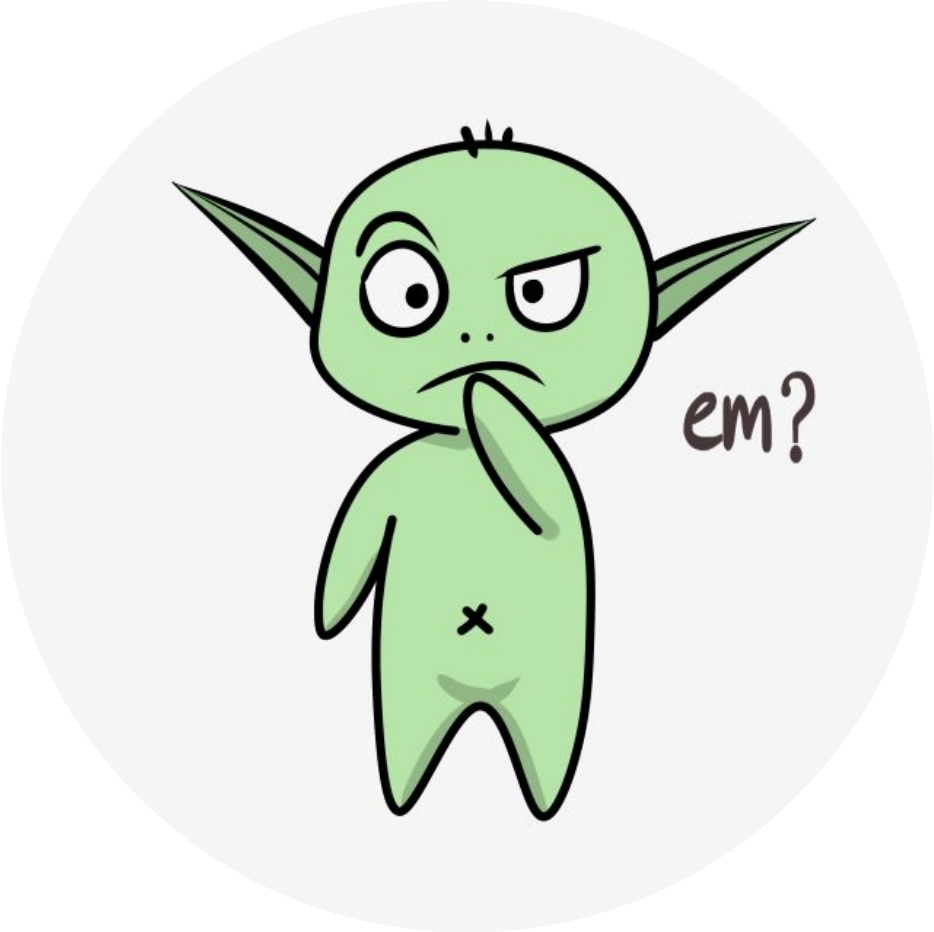
$$Y_n(R_{t+1:n}, O_{t+n}; \boldsymbol{\theta}^-) = \sum_{t=0}^{n-1} \gamma^t R_{t+1} + \gamma^n \max_{a' \in \mathcal{A}} Q\big(O_{t+n}, a'; \boldsymbol{\theta}^-\big)$$

$$\mathcal{L}^{\text{DQN}}_{\text{n-step}} = \mathbb{E}_{\tau^n \sim U(\mathcal{D})}\Big[\big(Y_n(R_{t+1:n}, O_{t+n}; \boldsymbol{\theta}^-) - Q(O_t, A_t; \boldsymbol{\theta}_t)\big)^2\Big].$$

Do we need importance sampling?

Yes, but in practice we often don't use it.

1. For DQN, it would mean simply stopping the update, because the probability of the max action is either 0 or 1. For other updates, IS corrections can lead to large variance
2. Practitioners often use 3 ≤ n ≤ 7, which might be small enough regardless



https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Mixed Monte-Carlo Returns

$$Y_{\mathrm{MC}}(R_{t+1:T}) = \sum_{t=0}^{T} \gamma^t R_{t+1}$$

$$Y_{\mathrm{TD}}(R_{t+1}, O_{t+1}; \boldsymbol{\theta}^-) = R_{t+1} + \gamma \max_{a' \in \mathcal{A}} Q\big(O_{t+1}, a'; \boldsymbol{\theta}^-\big)$$

$$\mathcal{L}_{\mathrm{MMC}}^{\mathrm{DQN}} = \mathbb{E}_{\tau^T \sim U(\mathcal{D})}\Big[\big((1-\beta)Y_{\mathrm{TD}} + \beta Y_{MC} - Q(O_t, A_t; \boldsymbol{\theta}_t)\big)^2\Big]$$

Do we need importance sampling?

Yes, but in practice we often don't use it.

1. For DQN, we really don't want to cut the traces here, bias might not be a big deal in many of the environments in which it is used
2. We generally use a small β (e.g., β = 0.1)

Mixed Monte-Carlo Returns

$$Y_{\text{MC}}(R_{t+1:T}) = \sum_{t=0}^{T} \gamma^t R_{t+1}$$

$$Y_{\text{TD}}(R_{t+1}, O_{t+1}; \boldsymbol{\theta}^-) = R_{t+1} + \gamma \max_{a' \in \mathcal{A}} Q\big(O_{t+1}, a'; \boldsymbol{\theta}^-\big)$$

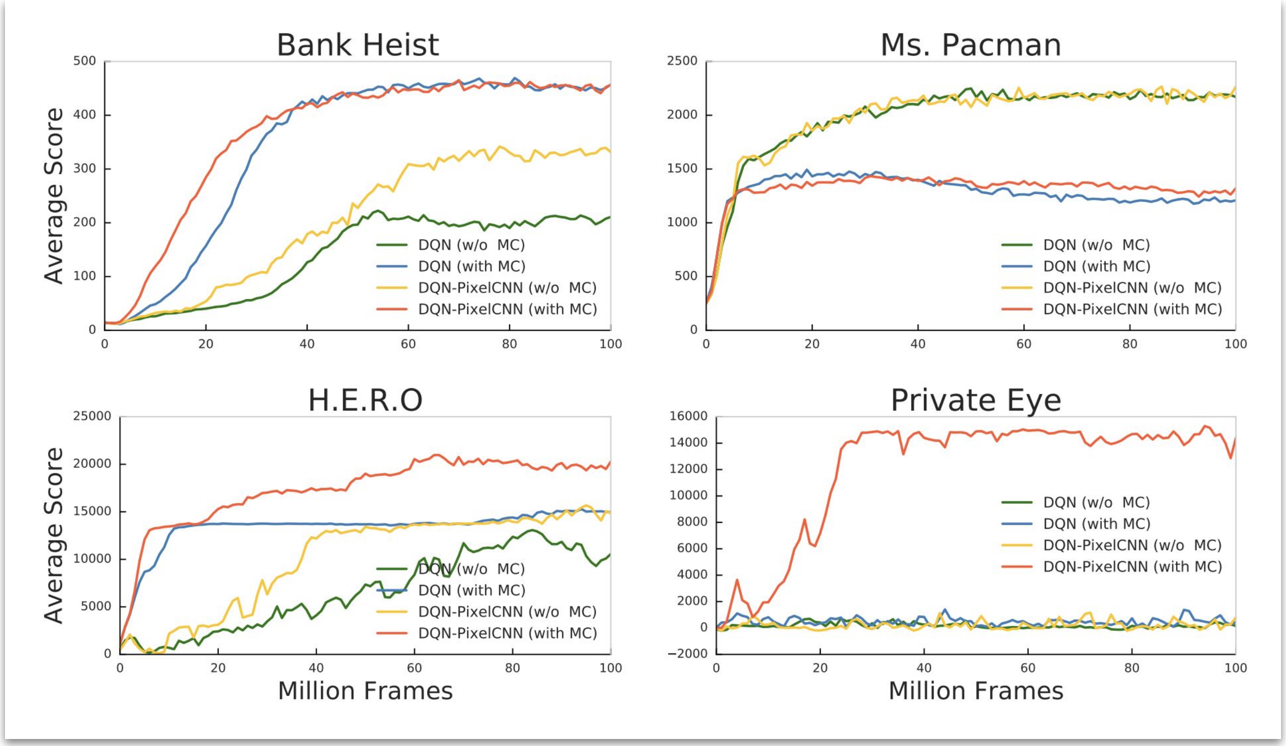
$$\mathcal{L}^{\text{DQN}}_{\text{MMC}} = \mathbb{E}_{\tau^T \sim U(\mathcal{D})}\Big[\big((1-\beta)Y_{\text{TD}} + \beta Y_{MC} - Q(O_t, A_t; \boldsymbol{\theta}_t)\big)^2\Big]$$

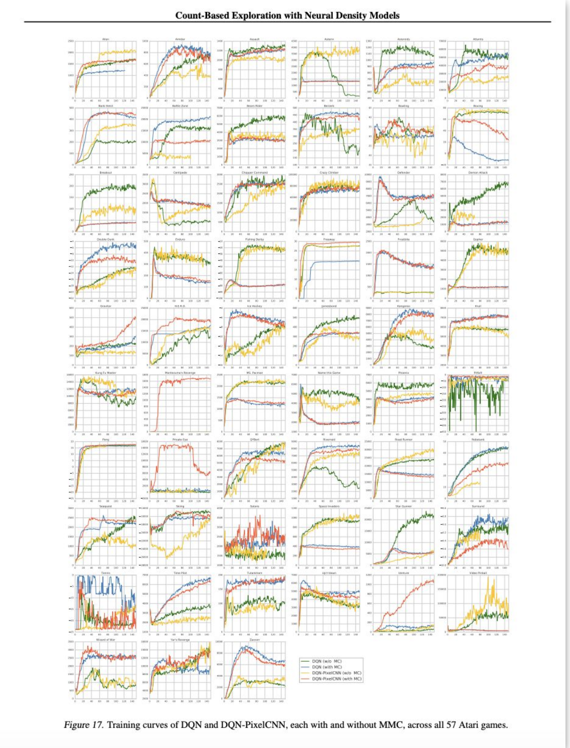
Do we need importance sampling?

Yes, but in practice we often don't use it.

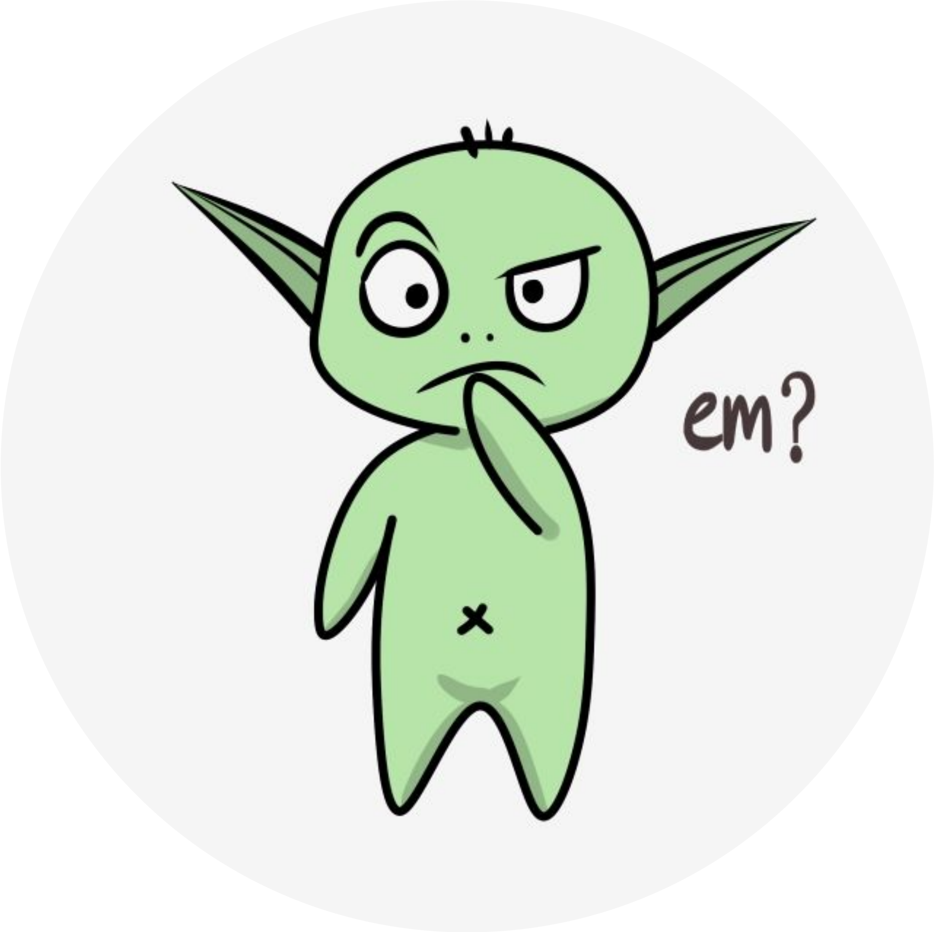
1. For DQN, we really don't want to cut the traces here, bias might not be a big deal in many of the environments in which it is used
2. We generally use a small β (e.g., $\beta = 0.1$)

Mixed Monte-Carlo Returns [Ostrovski et al., 2017]



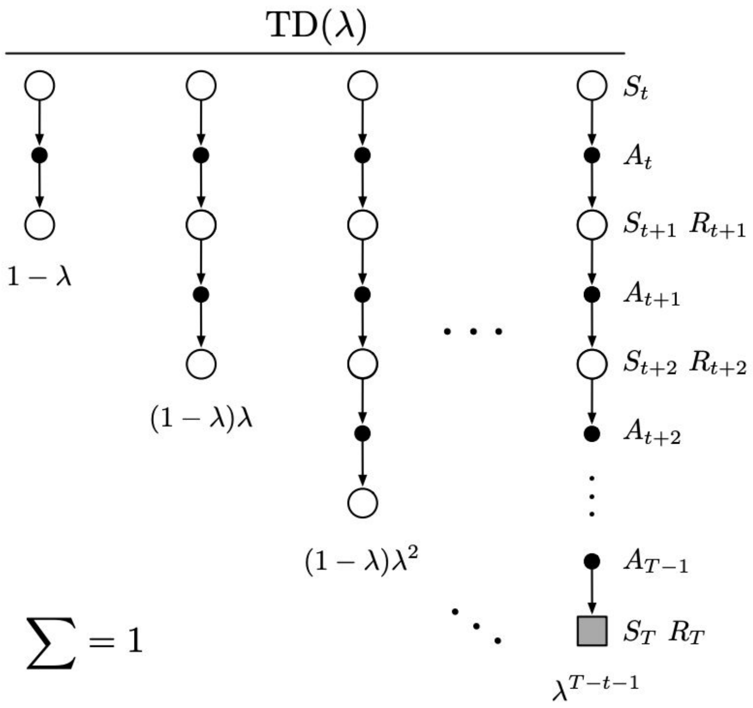


* 1 seed?

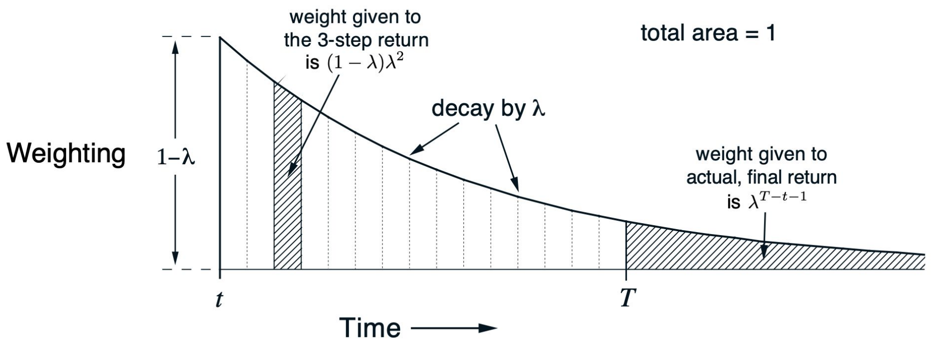


https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

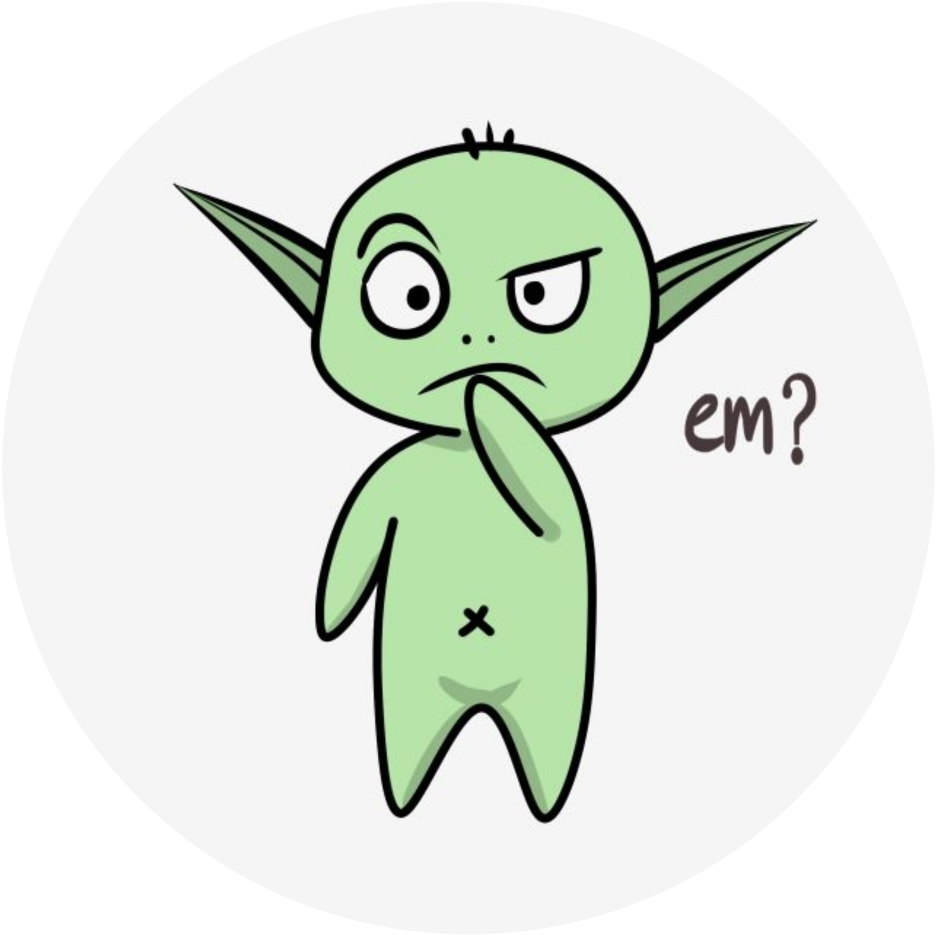
The λ-return: Considering more than 1 or 2 returns at once



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



Images by Sutton & Barto (2018)



https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

It is now much harder to justify ignoring IS ratios

$$\beta(A_k|S_k) = \frac{\pi(A_k|S_k)}{b(A_k|S_k)} \tag{4.22}$$

$$Y(R_{t+1}, O_{t+1}; \boldsymbol{\theta}^-) = R_{t+1} + \gamma \max_{a' \in \mathcal{A}} Q\big(O_{t+1}, a'; \boldsymbol{\theta}^-\big) \tag{4.23}$$

$$\delta_t^\pi = Y(R_{t+1}, O_{t+1}; \boldsymbol{\theta}^-) - Q(O_t, A_t; \boldsymbol{\theta}_t) \tag{4.24}$$

$$\mathcal{L}_{\lambda,\rho}^{\mathrm{DQN}} = \mathbb{E}_{\tau \sim U(\mathcal{D})}\left[\left(\sum_{t=0}^{\infty} (\gamma\lambda)^t \left(\prod_{k=1}^{t} \beta(A_k|S_k)\right) \delta_t^\pi\right)^2\right] \tag{4.25}$$

Retrace [Munos et al., 2016]

- Decaying and Clipping traces and importance sampling ratios

$$\beta(A_k|S_k) = \lambda \min(1, \pi(A_k|S_k)/b(A_k|S_k))$$

- Retrace is not an algorithm *per se*
  ACER (Wang et al. 2017) is an algorithm that implements Retrace
  (and many many other things)

Trajectory-aware λ-returns [Daley et al., 2023]

$$(\mathcal{M}Q)(s,a) := Q(s,a) + \mathbb{E}_\mu\left[\sum_{t=0}^{\infty} \gamma^t \beta_t \delta_t^\pi \,\middle|\, (S_0, A_0) = (s,a)\right]$$

Importance Sampling: $\beta_t = \lambda^t \Pi_t$ (Kahn & Harris, 1951).

Truncated Importance Sampling: $\beta_t = \lambda^t \min(1, \Pi_t)$

Tree Backup: $\beta_t = \prod_{k=1}^{t} \lambda\pi(A_k|S_k)$ (Precup et al., 2000)

Q$^\pi$(λ): $\beta_t = \lambda^t$ (Harutyunyan et al., 2016).

Retrace: $\beta_t = \prod_{k=1}^{t} \lambda \min(1, \rho_k)$ (Munos et al., 2016).

Recursive Retrace: $\beta_t = \lambda \min(1, \beta_{t-1}\rho_t)$ (Munos et al., 2016).

Why use λ-returns (instead of simpler n-step returns)?

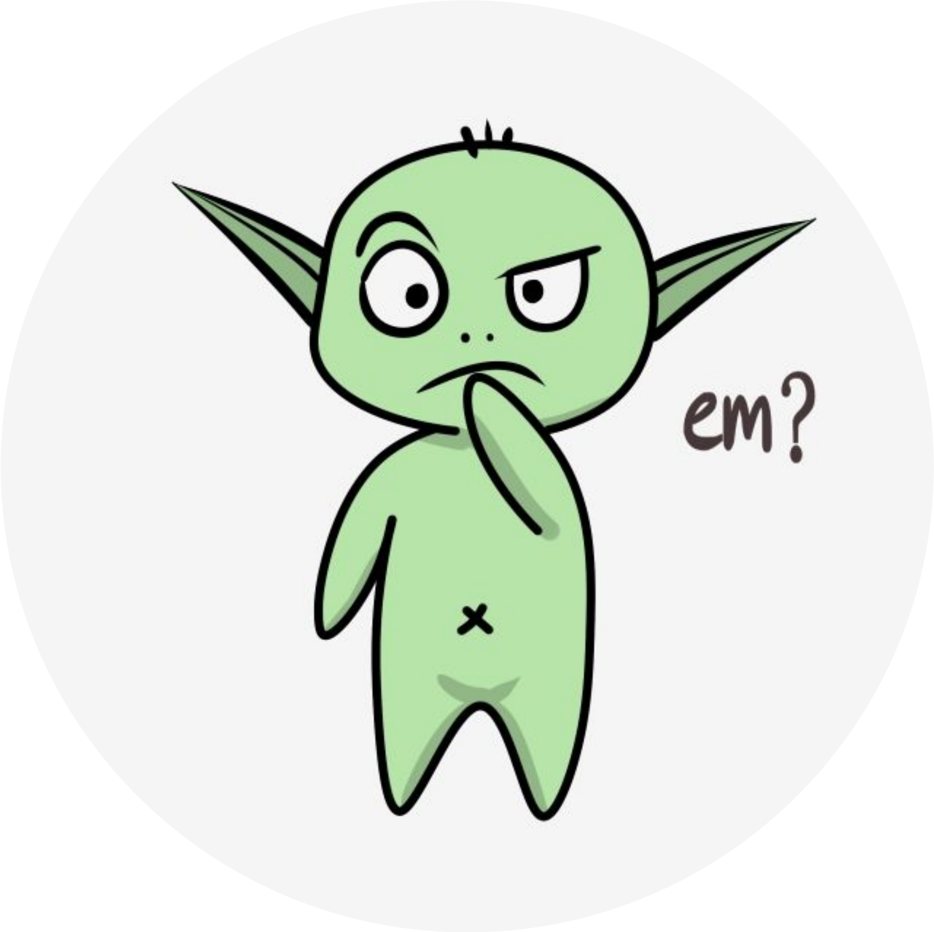
[Daley et al., 2024]

Theorem 3.7 (Variance-reduction property of compound returns). *Let $G_t^{(n)}$ be any n-step return and let $G_t^{\mathbf{c}}$ be any compound return with the same effective n-step: i.e., $\mathbf{c}$ satisfies Proposition 3.6. The inequality $\text{Var}[G_t^{\mathbf{c}} \mid S_t] \leq \text{Var}[G_t^{(n)} \mid S_t]$ always holds, and is strict whenever TD errors are not perfectly correlated ($\rho < 1$).*

Corollary 3.8 (Variance reduction of λ-return). *The magnitude of variance reduction for a λ-return is bounded by*

$$0 \leq \text{Var}[G_t^n \mid S_t] - \text{Var}[G_t^\lambda \mid S_t] \leq (1-\rho)\frac{\lambda}{1-\lambda^2}\kappa\,.$$

This magnitude is monotonic in γ and maximized at $\gamma = 1$.



https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

What about eligibility traces?

The Forward View

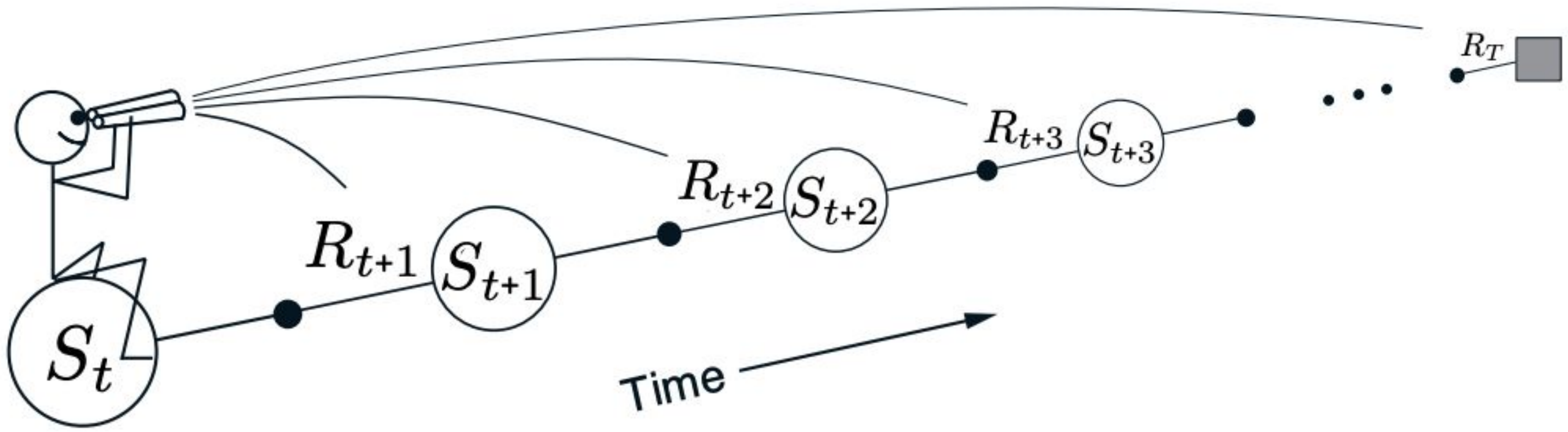


Figure 12.4: The forward view. We decide how to update each state by looking forward to future rewards and states.

Image by Sutton & Barto (2018)

The Backward View

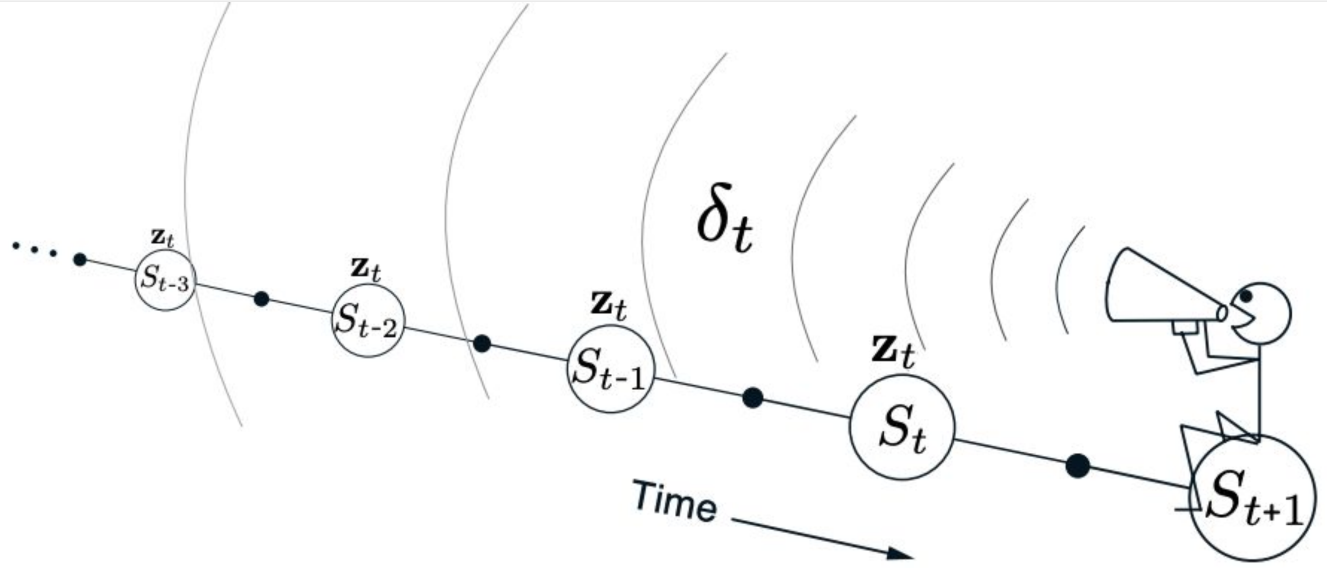
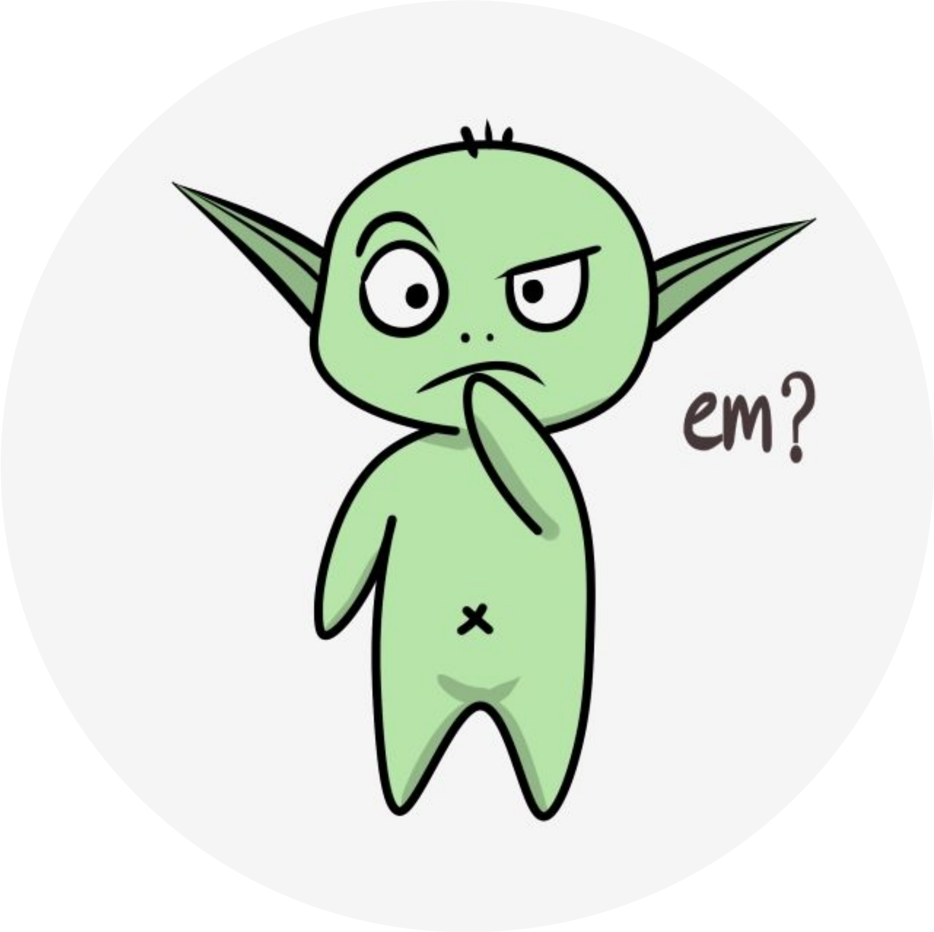


Figure 12.5: The backward or mechanistic view of TD(λ). Each update depends on the current TD error combined with the current eligibility traces of past events.

Image by Sutton & Barto (2018)

Eligibility Traces

- Many of the computational benefits of eligibility traces cannot be observed in deep RL algorithms (note I'm talking about eligibility traces, and not λ-returns)
  - The experience replay buffer requires experience to be saved anyway
  - Performing inference in a neural network can be quite expensive (vs simple dot products)
  - Existing algorithms do not perform sequential online updates
  - Eligibility traces are hard to implement when using neural networks
- We will revisit λ-returns when talking about PPO, though GAE (Generalized Advantage Estimation) is pretty much a λ-return



https://pngtree.com/freepng/question-expression-cartoon-illustration_4545209.ht

Next class

- What I plan to do:
  - Continue discussing different instantiations of deep RL algorithms through the objective function, more specifically, *distributional reinforcement learning*.
- What I recommend YOU to do for next class:
  - Read

    - Bellemare, M. G., Dabney, W., and Munos, R. (2017). A Distributional Perspective on Reinforcement Learning. In Proceedings of the International Conference on Machine Learning. Preprint made available on July 21, 2017.

    - Dabney, W., Rowland, M., Bellemare, M. G., and Munos, R. (2018). Distributional Reinforcement Learning with Quantile Regression. In Proceedings of the AAAI Conference on Artificial Intelligence. Preprint made available on October 27, 2017.

    - Farebrother, J., et al. (2024). Stop Regressing: Training Value Functions via Classification for Scalable Deep RL. In Proceedings of the International Conference on Machine Learning. Preprint made available on March 6, 2024.
  - Work on Assignment 2!