Proximal Policy Optimization

Reinforcement Learning

November 09, 2021

MSU

Recap: TRPO

Reminder: Policy Gradient

$$J(\pi_\theta) := \mathbb{E}_{\mathcal{T} \sim \pi_\theta} \sum_{t \geqslant 0} \gamma^t r_t$$

Reminder: Policy Gradient

$$J(\pi_\theta) := \mathbb{E}_{\mathcal{T}\sim\pi_\theta} \sum_{t\geqslant 0} \gamma^t r_t$$

$$\nabla_\theta J(\pi_\theta) \approx \underbrace{\mathbb{E}_{\mathcal{T}\sim\pi_\theta} \sum_{t\geqslant 0}}_{\substack{\text{data generated by } \pi_\theta \\ \text{(sample pairs } s, a \text{ from trajectories } \mathcal{T} \sim \pi_\theta)}} \overbrace{\nabla_\theta \log \pi_\theta(a_t \mid s_t)}^{\text{log-likelihood}} \underbrace{\left(Q^{\pi_\theta}(s_t, a_t) - \overbrace{V^{\pi_\theta}(s_t)}^{\text{baseline}}\right)}_{\substack{A^{\pi_\theta}(s_t, a_t) \\ \text{critic estimation}}}$$

Reminder: Policy Gradient

$$J(\pi_\theta) := \mathbb{E}_{\mathcal{T}\sim\pi_\theta} \sum_{t\geqslant 0} \gamma^t r_t$$

$$\nabla_\theta J(\pi_\theta) \approx \underbrace{\mathbb{E}_{\mathcal{T}\sim\pi_\theta} \sum_{t\geqslant 0}}_{\substack{\text{data generated by } \pi_\theta \\ \text{(sample pairs } s, a \text{ from trajectories } \mathcal{T}\sim\pi_\theta)}} \overbrace{\nabla_\theta \log \pi_\theta(a_t \mid s_t)}^{\text{log-likelihood}} \underbrace{\left(Q^{\pi_\theta}(s_t, a_t) - \overbrace{V^{\pi_\theta}(s_t)}^{\text{baseline}}\right)}_{\substack{A^{\pi_\theta}(s_t, a_t) \\ \text{critic estimation}}}$$

Everything is great except it is on-policy!

More efficient Policy Gradient?

$$\nabla_\theta J(\pi_\theta) = \frac{1}{1-\gamma} \underbrace{\mathbb{E}_{s \sim d_{\pi_\theta}(s)} \mathbb{E}_{a \sim \pi_\theta(a|s)}}_{\text{data generated by } \pi_\theta} \nabla_\theta \log \pi_\theta(a \mid s) A^{\pi_\theta}(s, a)$$

Suppose we:

- want to optimize π_θ (compute gradient for current θ);

More efficient Policy Gradient?

$$\nabla_\theta J(\pi_\theta) = \frac{1}{1-\gamma} \underbrace{\mathbb{E}_{s \sim d_{\pi_\theta}(s)} \mathbb{E}_{a \sim \pi_\theta(a|s)}}_{\text{data generated by } \pi_\theta} \nabla_\theta \log \pi_\theta(a \mid s) A^{\pi_\theta}(s, a)$$

Suppose we:

- want to optimize π_θ (compute gradient for current θ);
- have data (trajectory samples) from policy π^{old};

More efficient Policy Gradient?

$$\nabla_\theta J(\pi_\theta) = \frac{1}{1-\gamma} \underbrace{\mathbb{E}_{s \sim d_{\pi_\theta}(s)} \mathbb{E}_{a \sim \pi_\theta(a|s)}}_{\text{data generated by } \pi_\theta} \nabla_\theta \log \pi_\theta(a \mid s) A^{\pi_\theta}(s, a)$$

Suppose we:

- want to optimize π_θ (compute gradient for current θ);
- have data (trajectory samples) from policy π^{old};
  - i.e. we can estimate $\mathbb{E}_{s \sim d_{\pi^{\text{old}}}(s)}$ and $\mathbb{E}_{a \sim \pi^{\text{old}}(a|s)}$;

More efficient Policy Gradient?

$$\nabla_\theta J(\pi_\theta) = \frac{1}{1-\gamma} \underbrace{\mathbb{E}_{s\sim d_{\pi_\theta}(s)} \mathbb{E}_{a\sim\pi_\theta(a|s)}}_{\text{data generated by } \pi_\theta} \nabla_\theta \log \pi_\theta(a \mid s) A^{\pi_\theta}(s,a)$$

Suppose we:

- want to optimize π_θ (compute gradient for current θ);
- have data (trajectory samples) from policy π^{old};
  - i.e. we can estimate $\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)}$ and $\mathbb{E}_{a\sim\pi^{\text{old}}(a|s)}$;
  - i.e. we can train $V^{\pi^{\text{old}}}(s)$ and thus estimate $A^{\pi^{\text{old}}}(s,a)$;

More efficient Policy Gradient?

$$\nabla_\theta J(\pi_\theta) = \frac{1}{1-\gamma} \underbrace{\mathbb{E}_{s\sim d_{\pi_\theta}(s)} \mathbb{E}_{a\sim\pi_\theta(a|s)}}_{\text{data generated by } \pi_\theta} \nabla_\theta \log \pi_\theta(a \mid s) A^{\pi_\theta}(s,a)$$

Suppose we:

- want to optimize π_θ (compute gradient for current θ);
- have data (trajectory samples) from policy π^{old};
  - i.e. we can estimate $\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)}$ and $\mathbb{E}_{a\sim\pi^{\text{old}}(a|s)}$;
  - i.e. we can train $V^{\pi^{\text{old}}}(s)$ and thus estimate $A^{\pi^{\text{old}}}(s,a)$;

TRPO: use more efficient optimization procedure than SGD!

Relative Performance Identity

$$J(\pi_\theta) - J(\pi^{\mathrm{old}}) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_{\pi_\theta}(s)}\mathbb{E}_{a\sim\pi_\theta(a|s)} A^{\pi^{\mathrm{old}}}(s, a)$$

Relative Performance Identity

$$J(\pi_\theta) - J(\pi^{\text{old}}) = \frac{1}{1-\gamma} \underbrace{\mathbb{E}_{s\sim d_{\pi_\theta}(s)}\mathbb{E}_{a\sim\pi_\theta(a|s)}}_{\substack{\text{collect data with } \pi_\theta \\ \text{(wrong! we don't have it!)}}} \overbrace{A^{\pi^{\text{old}}}(s,a)}^{\substack{\text{old critic!} \\ \text{(good: can train it!)}}}$$

Relative Performance Identity

$$J(\pi_\theta) - J(\pi^{\text{old}}) = \frac{1}{1-\gamma} \underbrace{\mathbb{E}_{s\sim d_{\pi_\theta}(s)}\mathbb{E}_{a\sim\pi_\theta(a|s)}}_{\substack{\text{collect data with } \pi_\theta \\ \text{(wrong! we don't have it!)}}} \overbrace{A^{\pi^{\text{old}}}}^{\substack{\text{old critic!} \\ \text{(good: can train it!)}}}(s, a)$$

We performed **reward shaping** using another policy's value function!

Surrogate objective

Introduce **surrogate objective**:

$$J(\pi_\theta) - J(\pi^{\text{old}}) \approx L_{\pi^{\text{old}}}(\theta) :=$$

Surrogate objective

Introduce **surrogate objective**:

$$J(\pi_\theta) - J(\pi^{\text{old}}) \approx L_{\pi^{\text{old}}}(\theta) := \frac{1}{1-\gamma} \underbrace{\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)} \mathbb{E}_{a\sim\pi^{\text{old}}(a|s)}}_{\text{data generated by } \pi^{\text{old}}} \overbrace{\frac{\pi_\theta(a \mid s)}{\pi^{\text{old}}(a \mid s)}}^{\text{importance sampling correction}} \underbrace{A^{\pi^{\text{old}}}(s, a)}_{\text{do not require fresh critic}}$$

Surrogate objective

Introduce **surrogate objective**:

$$J(\pi_\theta) - J(\pi^{\text{old}}) \approx L_{\pi^{\text{old}}}(\theta) := \frac{1}{1-\gamma} \underbrace{\mathbb{E}_{s \sim d_{\pi^{\text{old}}}(s)} \mathbb{E}_{a \sim \pi^{\text{old}}(a|s)}}_{\text{data generated by } \pi^{\text{old}}} \overbrace{\frac{\pi_\theta(a \mid s)}{\pi^{\text{old}}(a \mid s)}}^{\text{importance sampling correction}} \underbrace{A^{\pi^{\text{old}}}(s, a)}_{\text{do not require fresh critic}}$$

- we can work with it;

Surrogate objective

Introduce **surrogate objective**:

$$J(\pi_\theta) - J(\pi^{\text{old}}) \approx L_{\pi^{\text{old}}}(\theta) := \frac{1}{1-\gamma} \underbrace{\mathbb{E}_{s \sim d_{\pi^{\text{old}}}(s)} \mathbb{E}_{a \sim \pi^{\text{old}}(a|s)}}_{\text{data generated by } \pi^{\text{old}}} \overbrace{\frac{\pi_\theta(a \mid s)}{\pi^{\text{old}}(a \mid s)}}^{\text{importance sampling correction}} \underbrace{A^{\pi^{\text{old}}}(s, a)}_{\text{do not require fresh critic}}$$

- we can work with it;
- directs to policy improvement of π^{old}:
    - optimizing θ with fixed π^{old} will learn $\operatorname*{argmax}_a A^{\pi^{\text{old}}}(s, a)$

Introduce **surrogate objective**:

$$J(\pi_\theta) - J(\pi^{\text{old}}) \approx L_{\pi^{\text{old}}}(\theta) := \frac{1}{1-\gamma} \underbrace{\mathbb{E}_{s \sim d_{\pi^{\text{old}}}(s)} \mathbb{E}_{a \sim \pi^{\text{old}}(a|s)}}_{\text{data generated by } \pi^{\text{old}}} \overbrace{\frac{\pi_\theta(a \mid s)}{\pi^{\text{old}}(a \mid s)}}^{\text{importance sampling correction}} \underbrace{A^{\pi^{\text{old}}}(s, a)}_{\text{do not require fresh critic}}$$

- we can work with it;
- directs to policy improvement of π^{old}:
  - optimizing θ with fixed π^{old} will learn $\operatorname{argmax}_a A^{\pi^{\text{old}}}(s, a)$
  - optimizing θ with fixed *data* will learn $\pi_\theta(a \mid s) = 1$ if $A(s, a) > 0$, $\pi_\theta(a \mid s) = 0$ otherwise.

Minorization-maximization algorithm

We discovered a **variational lower bound** for our objective:

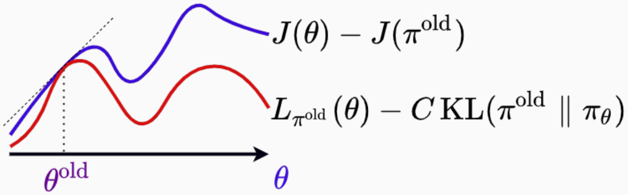
$$J(\pi_\theta) - J(\pi^{\mathrm{old}}) \geqslant L_{\pi^{\mathrm{old}}}(\theta) - C\,\mathsf{KL}^{\mathsf{max}}(\pi^{\mathrm{old}} \parallel \pi_\theta)$$

Minorization-maximization algorithm

We discovered a **variational lower bound** for our objective:

$$J(\pi_\theta) - J(\pi^{\text{old}}) \geqslant L_{\pi^{\text{old}}}(\theta) - C\,\mathsf{KL}^{\max}(\pi^{\text{old}} \parallel \pi_\theta)$$

- **Minorization**: construct a new lower bound; in our case simply use $\pi^{\text{old}} \leftarrow \pi_\theta$.

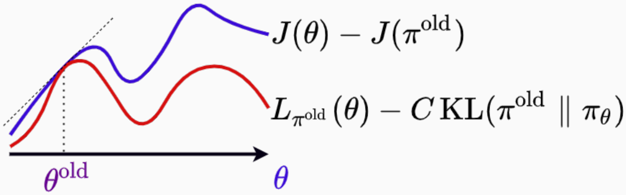


Minorization-maximization algorithm

We discovered a **variational lower bound** for our objective:

$$J(\pi_\theta) - J(\pi^{\text{old}}) \geqslant L_{\pi^{\text{old}}}(\theta) - C\,\mathrm{KL}^{\max}(\pi^{\text{old}} \parallel \pi_\theta)$$

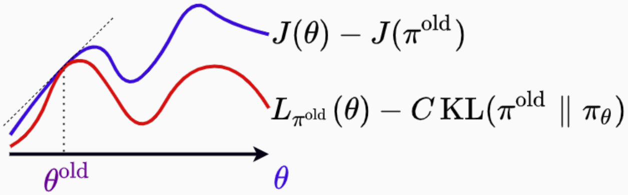
- **Minorization**: construct a new lower bound; in our case simply use $\pi^{\text{old}} \leftarrow \pi_\theta$.
- **Maximization**: optimize lower bound (as long as you want).



We discovered a **variational lower bound** for our objective:

$$J(\pi_\theta) - J(\pi^{\text{old}}) \geqslant L_{\pi^{\text{old}}}(\theta) - C\,\text{KL}^{\max}(\pi^{\text{old}} \parallel \pi_\theta)$$

- **Minorization**: construct a new lower bound; in our case simply use $\pi^{\text{old}} \leftarrow \pi_\theta$.
- **Maximization**: optimize lower bound (as long as you want).



✓ guarantees monotonic improvement!

$$L_{\pi^{\text{old}}}(\theta) - C\,\text{KL}^{\max}(\pi^{\text{old}} \parallel \pi_\theta) \to \max_\theta$$

Issues:

- critic is imperfect :(

Glue and Tape to the Rescue!

$$L_{\pi^{\text{old}}}(\theta) - C\,\text{KL}^{\max}(\pi^{\text{old}} \parallel \pi_\theta) \to \max_\theta$$

Issues:

- critic is imperfect :(
  - well, use what you have...

Glue and Tape to the Rescue!

$$L_{\pi^{\text{old}}}(\theta) - C\,\text{KL}^{\max}(\pi^{\text{old}} \parallel \pi_\theta) \to \max_\theta$$

Issues:

- critic is imperfect :(
  - well, use what you have...
- can't work with $\text{KL}^{\max}$:(

Glue and Tape to the Rescue!

$$L_{\pi^{\text{old}}}(\theta) - C\,\mathrm{KL}^{\max}(\pi^{\text{old}} \parallel \pi_\theta) \to \max_\theta$$

Issues:

- critic is imperfect :(
  - well, use what you have...
- can't work with $\mathrm{KL}^{\max}$:(
  - well, change to $\mathbb{E}_s \mathrm{KL}(\pi^{\text{old}}(\cdot \mid s) \parallel \pi_\theta(\cdot \mid s)))$...

Glue and Tape to the Rescue!

$$L_{\pi^{\text{old}}}(\theta) - C\,\text{KL}^{\max}(\pi^{\text{old}} \parallel \pi_\theta) \to \max_\theta$$

Issues:

- critic is imperfect :(
  - well, use what you have...
- can't work with $\text{KL}^{\max}$:(
  - well, change to $\mathbb{E}_s \text{KL}(\pi^{\text{old}}(\cdot \mid s) \parallel \pi_\theta(\cdot \mid s)))$...
- we do not know constant C :(

Glue and Tape to the Rescue!

$$L_{\pi^{\text{old}}}(\theta) - C\,\text{KL}^{\max}(\pi^{\text{old}} \parallel \pi_\theta) \to \max_\theta$$

Issues:

- critic is imperfect :(
  - well, use what you have...
- can't work with $\text{KL}^{\max}$:(
  - well, change to $\mathbb{E}_s \text{KL}(\pi^{\text{old}}(\cdot \mid s) \parallel \pi_\theta(\cdot \mid s)))$...
- we do not know constant C :(
  - well, it is some hyperparameter...

Glue and Tape to the Rescue!

$$L_{\pi^{\text{old}}}(\theta) - C\,\text{KL}^{\max}(\pi^{\text{old}} \parallel \pi_\theta) \to \max_\theta$$

Issues:

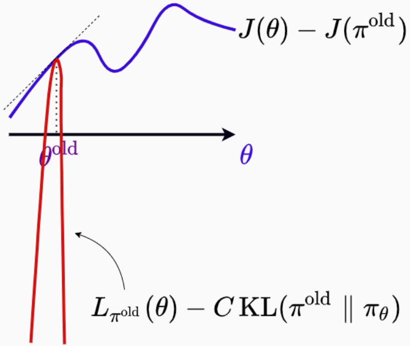
- critic is imperfect :(
  - well, use what you have...
- can't work with $\text{KL}^{\max}$:(
  - well, change to $\mathbb{E}_s \, \text{KL}(\pi^{\text{old}}(\cdot \mid s) \parallel \pi_\theta(\cdot \mid s)))$...
- we do not know constant C :(
  - well, it is some hyperparameter...
- and, actually, it is extremely huge :(
  - Hmmm...

Glue and Tape to the Rescue!

$$L_{\pi^{\text{old}}}(\theta) - C\,\text{KL}^{\max}(\pi^{\text{old}} \parallel \pi_\theta) \to \max_\theta$$

Issues:

- critic is imperfect :(
  - well, use what you have...
- can't work with $\text{KL}^{\max}$:(
  - well, change to $\mathbb{E}_s \text{KL}(\pi^{\text{old}}(\cdot \mid s) \parallel \pi_\theta(\cdot \mid s)))$...
- we do not know constant C :(
  - well, it is some hyperparameter...
- and, actually, it is extremely huge :(
  - Hmmm...

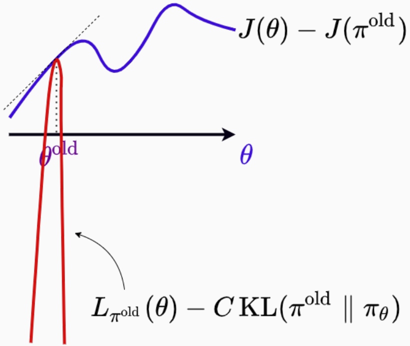


Glue and Tape to the Rescue!

$$L_{\pi^{\text{old}}}(\theta) - C\,\text{KL}^{\max}(\pi^{\text{old}} \parallel \pi_\theta) \to \max_\theta$$

Issues:

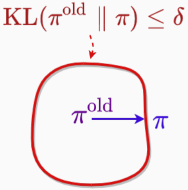
- critic is imperfect :(
  - well, use what you have...
- can't work with $\text{KL}^{\max}$:(
  - well, change to $\mathbb{E}_s\,\text{KL}(\pi^{\text{old}}(\cdot \mid s) \parallel \pi_\theta(\cdot \mid s)))$...
- we do not know constant C :(
  - well, it is some hyperparameter...
- and, actually, it is extremely huge :(
  - Hmmm...



Directed by
ROBERT B. WEIDE

Trust Region Policy Optimization (TRPO)

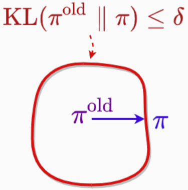
$$\begin{cases} L_{\pi^{\text{old}}}(\theta) \to \max\limits_{\theta} \\ \mathrm{KL}(\pi^{\text{old}} \parallel \pi_\theta) \leqslant \delta \end{cases}$$



✓ robust: prevents large changes;

Trust Region Policy Optimization (TRPO)

$$\begin{cases} L_{\pi^{\text{old}}}(\theta) \to \max\limits_{\theta} \\ \text{KL}(\pi^{\text{old}} \parallel \pi_\theta) \leqslant \delta \end{cases}$$

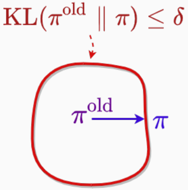


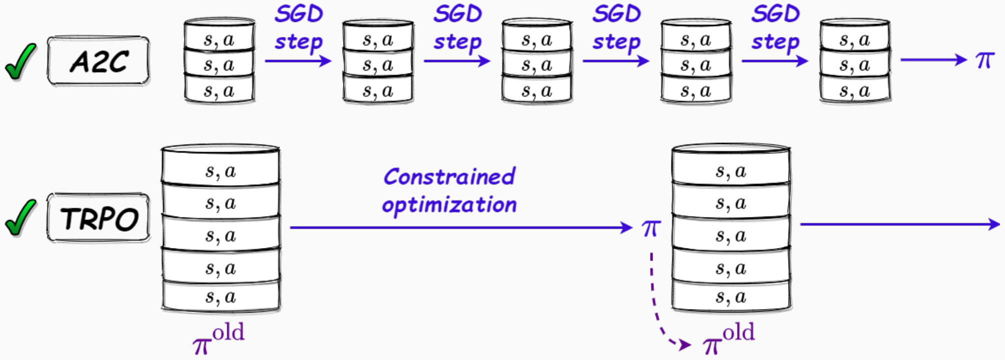
✓ robust: prevents large changes;

Trust Region Policy Optimization (TRPO)

$$\begin{cases} L_{\pi^{\text{old}}}(\theta) \to \max\limits_{\theta} \\ \text{KL}(\pi^{\text{old}} \parallel \pi_\theta) \leqslant \delta \end{cases}$$

✓ robust: prevents large changes;

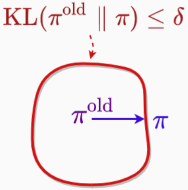




Trust Region Policy Optimization (TRPO)

$$\begin{cases} L_{\pi^{\text{old}}}(\theta) \to \max\limits_{\theta} \\ \text{KL}(\pi^{\text{old}} \parallel \pi_\theta) \leqslant \delta \end{cases}$$

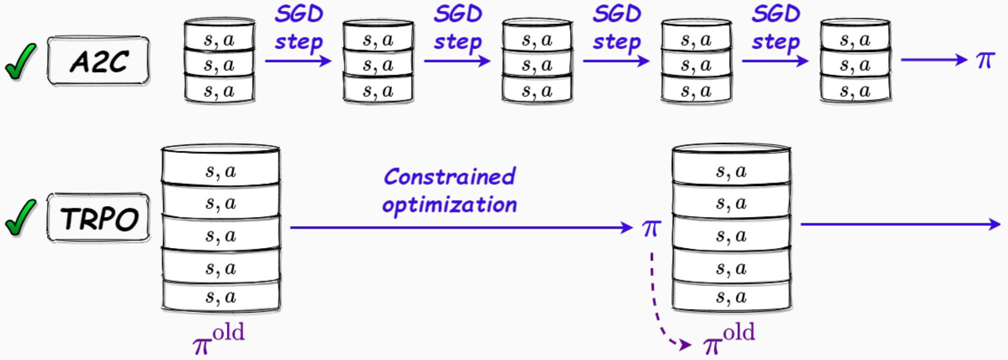
✓ robust: prevents large changes;



× critic and actor can't share backbone;

× computationally costly;

× complicated :(



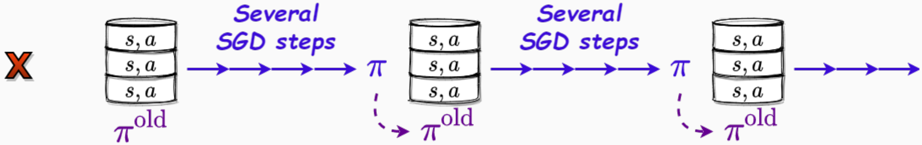
PPO Objective

Proximal Policy Optimization (PPO): Pipeline

$$\mathbb{E}_{s \sim d_{\pi^{\text{old}}}(s)} \mathbb{E}_{a \sim \pi^{\text{old}}(a|s)} \frac{\pi_\theta(a \mid s)}{\pi^{\text{old}}(a \mid s)} A^{\pi^{\text{old}}}(s, a) - C\,\text{KL}(\pi^{\text{old}} \parallel \pi_\theta) \to \max_\theta$$

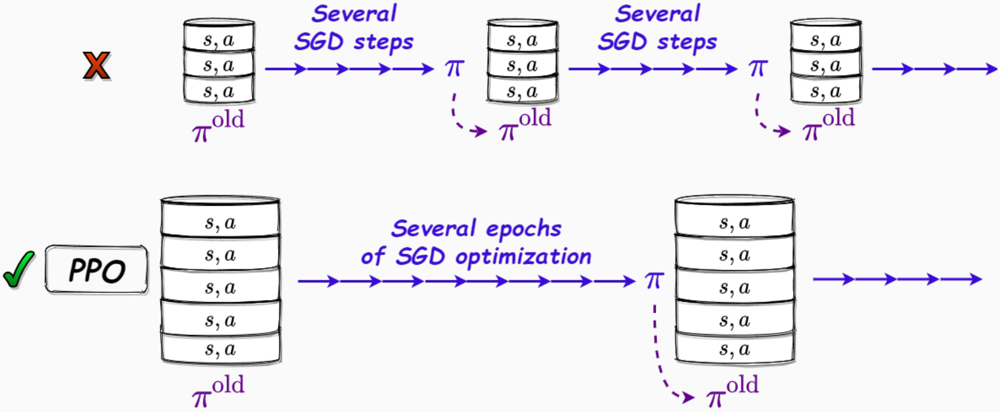
Proximal Policy Optimization (PPO): Pipeline

$$\mathbb{E}_{s \sim d_{\pi^{\text{old}}}(s)} \mathbb{E}_{a \sim \pi^{\text{old}}(a|s)} \frac{\pi_\theta(a \mid s)}{\pi^{\text{old}}(a \mid s)} A^{\pi^{\text{old}}}(s, a) - C \, \text{KL}(\pi^{\text{old}} \parallel \pi_\theta) \to \max_\theta$$



Proximal Policy Optimization (PPO): Pipeline

$$\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)}\mathbb{E}_{a\sim\pi^{\text{old}}(a|s)}\frac{\pi_\theta(a \mid s)}{\pi^{\text{old}}(a \mid s)}A^{\pi^{\text{old}}}(s,a) - C\,\mathrm{KL}(\pi^{\text{old}} \parallel \pi_\theta) \to \max_\theta$$



Clipping Objective

$$L_{\pi^{\text{old}}}(\theta) - C\,\mathrm{KL}(\pi^{\text{old}} \parallel \pi_\theta) \to \max_\theta$$

Default surrogate function:

$$\rho(\theta) := \frac{\pi_\theta(a \mid s)}{\pi^{\text{old}}(a \mid s)}$$

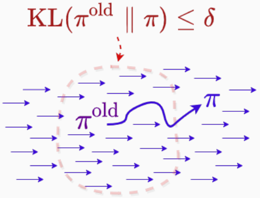
$$L_{\pi^{\text{old}}}(\theta) := \mathbb{E}_{s,a}\rho(\theta)A^{\pi^{\text{old}}}(s, a)$$

$$L_{\pi^{\text{old}}}(\theta) - C\,\mathrm{KL}(\pi^{\text{old}} \parallel \pi_\theta) \to \max_\theta$$

Default surrogate function:

$$\rho(\theta) := \frac{\pi_\theta(a \mid s)}{\pi^{\text{old}}(a \mid s)}$$

$$L_{\pi^{\text{old}}}(\theta) := \mathbb{E}_{s,a}\rho(\theta)A^{\pi^{\text{old}}}(s, a)$$



Clipping Objective

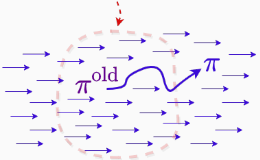
$$L_{\pi^{\text{old}}}(\theta) - C\,\text{KL}(\pi^{\text{old}} \parallel \pi_\theta) \to \max_\theta$$

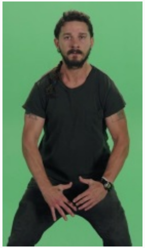
Default surrogate function:

$$\rho(\theta) := \frac{\pi_\theta(a \mid s)}{\pi^{\text{old}}(a \mid s)}$$

$$L_{\pi^{\text{old}}}(\theta) := \mathbb{E}_{s,a}\rho(\theta)A^{\pi^{\text{old}}}(s, a)$$

$$\text{KL}(\pi^{\text{old}} \parallel \pi) \leq \delta$$





Clipped surrogate function:

$$\rho^{\text{clip}}(\theta) := \text{clip}(\rho(\theta), 1 - \epsilon, 1 + \epsilon)$$

$$L^{\text{clip}}_{\pi^{\text{old}}}(\theta) := \mathbb{E}_{s,a}\rho^{\text{clip}}(\theta)A^{\pi^{\text{old}}}(s, a)$$

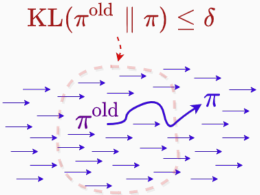
Clipping Objective

$$L_{\pi^{\text{old}}}(\theta) - C\,\mathrm{KL}(\pi^{\text{old}} \parallel \pi_\theta) \to \max_\theta$$

Default surrogate function:

$$\rho(\theta) := \frac{\pi_\theta(a \mid s)}{\pi^{\text{old}}(a \mid s)}$$

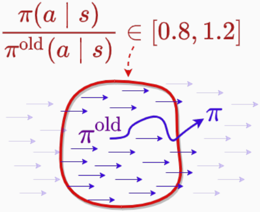
$$L_{\pi^{\text{old}}}(\theta) := \mathbb{E}_{s,a}\rho(\theta)A^{\pi^{\text{old}}}(s,a)$$



Clipped surrogate function:

$$\rho^{\text{clip}}(\theta) := \text{clip}(\rho(\theta), 1-\epsilon, 1+\epsilon)$$

$$L^{\text{clip}}_{\pi^{\text{old}}}(\theta) := \mathbb{E}_{s,a}\rho^{\text{clip}}(\theta)A^{\pi^{\text{old}}}(s,a)$$



Recalling lower bound intuition

$$\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)}\mathbb{E}_{a\sim\pi^{\text{old}}(a|s)} \min(\overbrace{\rho(\theta)A^{\pi^{\text{old}}}(s,a)}^{\text{original term}}, \overbrace{\rho^{\text{clip}}(\theta)A^{\pi^{\text{old}}}(s,a)}^{\substack{\text{term with clipped}\\ \text{importance sampling weight}}}) - \overbrace{C\,\mathrm{KL}(\pi^{\text{old}} \parallel \pi_\theta)}^{\text{«regularization»}} \to \max_\theta$$

Recalling lower bound intuition

$$\mathbb{E}_{s \sim d_{\pi^{\text{old}}}(s)} \mathbb{E}_{a \sim \pi^{\text{old}}(a|s)} \min\Big(\overbrace{\rho(\theta) A^{\pi^{\text{old}}}(s, a)}^{\text{original term}}, \; \overbrace{\rho^{\text{clip}}(\theta) A^{\pi^{\text{old}}}(s, a)}^{\substack{\text{term with clipped} \\ \text{importance sampling weight}}} \Big) - \overbrace{C \,\mathrm{KL}(\pi^{\text{old}} \parallel \pi_\theta)}^{\text{«regularization»}} \to \max_\theta$$

Advantage Sign	**Direction**	**Bad ratio case**	**Gradient**
$A^{\pi^{\text{old}}}(s, a) \geqslant 0$			

Recalling lower bound intuition

$$\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)}\mathbb{E}_{a\sim\pi^{\text{old}}(a|s)} \min\Big(\underbrace{\rho(\theta)A^{\pi^{\text{old}}}(s,a)}_{\text{original term}},\ \underbrace{\rho^{\text{clip}}(\theta)A^{\pi^{\text{old}}}(s,a)}_{\text{term with clipped importance sampling weight}}\Big) - \underbrace{C\,\mathrm{KL}(\pi^{\text{old}} \parallel \pi_\theta)}_{\text{«regularization»}} \to \max_\theta$$

Advantage Sign	Direction	Bad ratio case	Gradient
$A^{\pi^{\text{old}}}(s,a) \geqslant 0$	$\pi_\theta(a \mid s) \Uparrow$		

Recalling lower bound intuition

$$\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)}\mathbb{E}_{a\sim\pi^{\text{old}}(a|s)} \min\big(\overbrace{\rho(\theta)A^{\pi^{\text{old}}}(s,a)}^{\text{original term}},\ \overbrace{\rho^{\text{clip}}(\theta)A^{\pi^{\text{old}}}(s,a)}^{\substack{\text{term with clipped}\\ \text{importance sampling weight}}}\big) - \overbrace{C\,\mathrm{KL}(\pi^{\text{old}} \parallel \pi_\theta)}^{\text{«regularization»}} \to \max_\theta$$

Advantage Sign	Direction	Bad ratio case	Gradient
$A^{\pi^{\text{old}}}(s,a) \geqslant 0$	$\pi_\theta(a \mid s) \Uparrow$	$\rho(\theta) > 1.2$	

Recalling lower bound intuition

$$\mathbb{E}_{s \sim d_{\pi^{\text{old}}}(s)} \mathbb{E}_{a \sim \pi^{\text{old}}(a|s)} \min\Big(\overbrace{\rho(\theta) A^{\pi^{\text{old}}}(s,a)}^{\text{original term}}, \overbrace{\rho^{\text{clip}}(\theta) A^{\pi^{\text{old}}}(s,a)}^{\substack{\text{term with clipped} \\ \text{importance sampling weight}}} \Big) - \overbrace{C\, \mathrm{KL}(\pi^{\text{old}} \parallel \pi_\theta)}^{\text{«regularization»}} \to \max_\theta$$

Advantage Sign	Direction	Bad ratio case	Gradient
$A^{\pi^{\text{old}}}(s,a) \geqslant 0$	$\pi_\theta(a \mid s) \Uparrow$	$\rho(\theta) > 1.2$	0

Recalling lower bound intuition

$$\mathbb{E}_{s \sim d_{\pi^{\text{old}}}(s)} \mathbb{E}_{a \sim \pi^{\text{old}}(a|s)} \min\Big(\overbrace{\rho(\theta) A^{\pi^{\text{old}}}(s,a)}^{\text{original term}},\ \overbrace{\rho^{\text{clip}}(\theta) A^{\pi^{\text{old}}}(s,a)}^{\text{term with clipped importance sampling weight}}\Big) - \overbrace{C\,\mathrm{KL}(\pi^{\text{old}} \parallel \pi_\theta)}^{\text{«regularization»}} \to \max_\theta$$

Advantage Sign	Direction	Bad ratio case	Gradient
$A^{\pi^{\text{old}}}(s,a) \geqslant 0$	$\pi_\theta(a \mid s) \Uparrow$	$\rho(\theta) > 1.2$ $\rho(\theta) < 0.8$	0

Recalling lower bound intuition

$$\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)}\mathbb{E}_{a\sim\pi^{\text{old}}(a|s)}\min\Big(\overbrace{\rho(\theta)A^{\pi^{\text{old}}}(s,a)}^{\text{original term}},\ \overbrace{\rho^{\text{clip}}(\theta)A^{\pi^{\text{old}}}(s,a)}^{\substack{\text{term with clipped}\\ \text{importance sampling weight}}}\Big) - \overbrace{C\,\text{KL}(\pi^{\text{old}}\parallel\pi_\theta)}^{\text{«regularization»}} \to \max_\theta$$

Advantage Sign	Direction	Bad ratio case	Gradient
$A^{\pi^{\text{old}}}(s,a) \geqslant 0$	$\pi_\theta(a \mid s) \Uparrow$	$\rho(\theta) > 1.2$	0
		$\rho(\theta) < 0.8$	same

Recalling lower bound intuition

$$\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)}\mathbb{E}_{a\sim\pi^{\text{old}}(a|s)} \min(\underbrace{\rho(\theta)A^{\pi^{\text{old}}}(s,a)}_{\text{original term}},\ \underbrace{\rho^{\text{clip}}(\theta)A^{\pi^{\text{old}}}(s,a)}_{\text{term with clipped importance sampling weight}}) - \underbrace{C\,\text{KL}(\pi^{\text{old}} \parallel \pi_\theta)}_{\text{«regularization»}} \to \max_\theta$$

Advantage Sign	Direction	Bad ratio case	Gradient
$A^{\pi^{\text{old}}}(s,a) \geqslant 0$	$\pi_\theta(a \mid s) \Uparrow$	$\rho(\theta) > 1.2$	0
		$\rho(\theta) < 0.8$	same
$A^{\pi^{\text{old}}}(s,a) < 0$	$\pi_\theta(a \mid s) \Downarrow$	$\rho(\theta) > 1.2$	
		$\rho(\theta) < 0.8$	

Recalling lower bound intuition

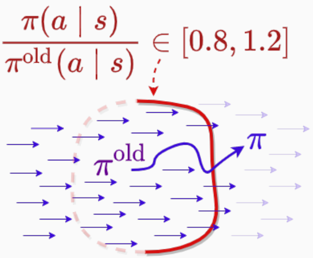
$$\mathbb{E}_{s\sim d_{\pi^{\text{old}}}(s)}\mathbb{E}_{a\sim\pi^{\text{old}}(a|s)} \min\Big(\overbrace{\rho(\theta)A^{\pi^{\text{old}}}(s,a)}^{\text{original term}},\ \overbrace{\rho^{\text{clip}}(\theta)A^{\pi^{\text{old}}}(s,a)}^{\substack{\text{term with clipped}\\ \text{importance sampling weight}}}\Big) - \overbrace{C\,\mathrm{KL}(\pi^{\text{old}}\parallel\pi_\theta)}^{\text{«regularization»}} \to \max_\theta$$

Advantage Sign	Direction	Bad ratio case	Gradient
$A^{\pi^{\text{old}}}(s,a) \geqslant 0$	$\pi_\theta(a \mid s) \Uparrow$	$\rho(\theta) > 1.2$	0
		$\rho(\theta) < 0.8$	same
$A^{\pi^{\text{old}}}(s,a) < 0$	$\pi_\theta(a \mid s) \Downarrow$	$\rho(\theta) > 1.2$	same
		$\rho(\theta) < 0.8$	0

Recalling lower bound intuition

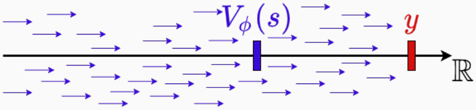
$$\mathbb{E}_{s \sim d_{\pi^{\text{old}}}(s)} \mathbb{E}_{a \sim \pi^{\text{old}}(a|s)} \min\Big(\overbrace{\rho(\theta) A^{\pi^{\text{old}}}(s,a)}^{\text{original term}},\ \overbrace{\rho^{\text{clip}}(\theta) A^{\pi^{\text{old}}}(s,a)}^{\text{term with clipped importance sampling weight}}\Big) - \overbrace{C\, \mathrm{KL}(\pi^{\text{old}} \parallel \pi_\theta)}^{\text{«regularization»}} \to \max_\theta$$

Advantage Sign	Direction	Bad ratio case	Gradient
$A^{\pi^{\text{old}}}(s,a) \geqslant 0$	$\pi_\theta(a \mid s) \Uparrow$	$\rho(\theta) > 1.2$	0
		$\rho(\theta) < 0.8$	same
$A^{\pi^{\text{old}}}(s,a) < 0$	$\pi_\theta(a \mid s) \Downarrow$	$\rho(\theta) > 1.2$	same
		$\rho(\theta) < 0.8$	0



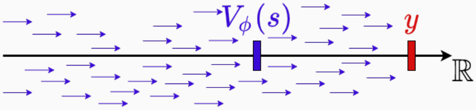
Clipped Critic Loss

$\text{Loss}(\phi) := (y - V^{\pi}(\phi))^2 =$



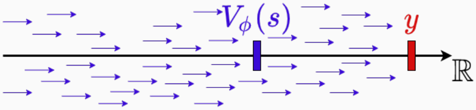
Clipped Critic Loss

$$\begin{aligned}\text{Loss}(\phi) &:= (y - V^{\pi}(\phi))^2 = \\ &= (y - V^{\text{old}} + V^{\text{old}} - V^{\pi}(\phi))^2\end{aligned}$$

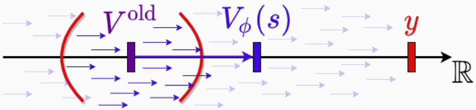


Clipped Critic Loss

$$\text{Loss}(\phi) := (y - V^{\pi}(\phi))^2 =$$
$$= (y - V^{\text{old}} + V^{\text{old}} - V^{\pi}(\phi))^2$$

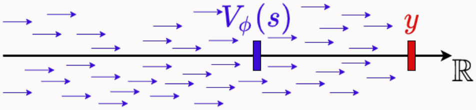


$$\text{Loss}^{\text{clip}}(\phi) := (y - V^{\text{old}} + \text{clip}(V^{\text{old}} - V^{\pi}(\phi), -\hat{\epsilon}, \hat{\epsilon}))^2$$

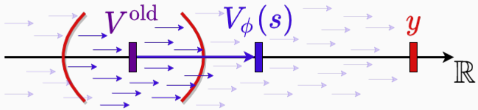


Clipped Critic Loss

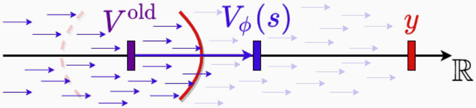
$$\text{Loss}(\phi) := (y - V^{\pi}(\phi))^2 = $$
$$= (y - V^{\text{old}} + V^{\text{old}} - V^{\pi}(\phi))^2$$



$$\text{Loss}^{\text{clip}}(\phi) := (y - V^{\text{old}} + \text{clip}(V^{\text{old}} - V^{\pi}(\phi), -\hat{\epsilon}, \hat{\epsilon}))^2$$



$$\max(\text{Loss}(\phi), \text{Loss}^{\text{clip}}(\phi))$$



Bias-Variance trade-off

Bias-Variance trade-off

Given rollout $s, r, s', r', s'', r'' \dots s^{(M)}$ from policy π and approximation of $V^{\pi}(s)$

Bias-Variance trade-off

Given rollout $s, r, s', r', s'', r'' \dots s^{(M)}$ from policy π and approximation of $V^{\pi}(s)$ perform **credit assignment** for state-action pair s, a (was this decision good or bad?)

Bias-Variance trade-off

Given rollout $s, r, s', r', s'', r'' \dots s^{(M)}$ from policy π and approximation of $V^\pi(s)$ perform **credit assignment** for state-action pair s, a (was this decision good or bad?)

For Actor:

$$\nabla := \rho(\theta)\nabla_\theta \log \pi_\theta(a \mid s) \underbrace{\Psi(s, a)}_{\text{advantage estimator}}$$

For Critic:

$$\underbrace{y_Q}_{\text{target for regression}} := \Psi(s, a) + V^\pi(s)$$

Bias-Variance trade-off

Given rollout $s, r, s', r', s'', r'' \ldots s^{(M)}$ from policy π and approximation of $V^\pi(s)$ perform **credit assignment** for state-action pair s, a (was this decision good or bad?)

For Actor:

$$\nabla := \rho(\theta)\nabla_\theta \log \pi_\theta(a \mid s) \underbrace{\Psi(s,a)}_{\text{advantage estimator}}$$

For Critic:

$$\underbrace{y_Q}_{\text{target for regression}} := \Psi(s,a) + V^\pi(s)$$

	$\Psi(s,a)$	**Bias**	**Variance**
Monte Carlo	$\Psi_{(\infty)}(s,a) := r + \gamma r' + \gamma^2 r'' + \cdots - V^\pi(s)$	0	high
1-step	$\Psi_{(1)}(s,a) := r + \gamma V^\pi(s') - V^\pi(s)$	high	low

Bias-Variance trade-off

Given rollout $s, r, s', r', s'', r'' \dots s^{(M)}$ from policy π and approximation of $V^\pi(s)$ perform **credit assignment** for state-action pair s, a (was this decision good or bad?)

For Actor:

$$\nabla := \rho(\theta)\nabla_\theta \log \pi_\theta(a \mid s) \underbrace{\Psi(s, a)}_{\text{advantage estimator}}$$

For Critic:

$$\underbrace{y_Q}_{\text{target for regression}} := \Psi(s, a) + V^\pi(s)$$

	$\Psi(s, a)$	**Bias**	**Variance**
Monte Carlo	$\Psi_{(\infty)}(s, a) := r + \gamma r' + \gamma^2 r'' + \dots - V^\pi(s)$	0	high
N-step	$\Psi_{(N)}(s, a) := r + \gamma r' + \dots + \gamma^N V^\pi(s^{(N)}) - V^\pi(s)$	intermediate	intermediate
1-step	$\Psi_{(1)}(s, a) := r + \gamma V^\pi(s') - V^\pi(s)$	high	low

Bias-Variance trade-off

Given rollout $s, r, s', r', s'', r'' \dots s^{(M)}$ from policy π and approximation of $V^\pi(s)$ perform **credit assignment** for state-action pair s, a (was this decision good or bad?)

For Actor:

$$\nabla := \rho(\theta)\nabla_\theta \log \pi_\theta(a \mid s) \underbrace{\Psi(s, a)}_{\substack{\text{advantage}\\\text{estimator}}}$$

For Critic:

$$\underbrace{y_Q}_{\substack{\text{target}\\\text{for regression}}} := \Psi(s, a) + V^\pi(s)$$

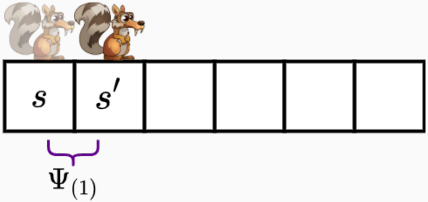
	$\Psi(s, a)$	**Bias**	**Variance**
Monte Carlo	$\Psi_{(\infty)}(s, a) := r + \gamma r' + \gamma^2 r'' + \dots - V^\pi(s)$	0	high
N-step	$\Psi_{(N)}(s, a) := r + \gamma r' + \dots + \gamma^N V^\pi(s^{(N)}) - V^\pi(s)$	intermediate	intermediate
1-step	$\Psi_{(1)}(s, a) := r + \gamma V^\pi(s') - V^\pi(s)$	high	low

Problem: hard to choose N.

Backward view: idea

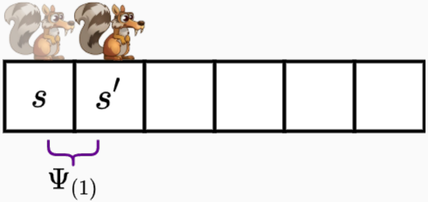
N-step update:

$$V^{\pi}(s) \leftarrow V^{\pi}(s) + \alpha \Psi_{(N)}(s, a)$$



***N*-step update:**

$$V^\pi(s) \leftarrow V^\pi(s) + \alpha \Psi_{(N)}(s, a)$$

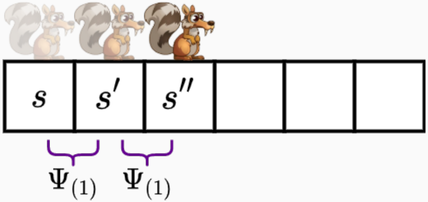


$$V^\pi(s) \leftarrow V^\pi(s) + \alpha \overbrace{\left(r + \gamma V^\pi(s') - V^\pi(s)\right)}^{\Psi_{(1)}(s,a)}$$

N-step update:

$$V^\pi(s) \leftarrow V^\pi(s) + \alpha \Psi_{(N)}(s,a)$$

How to turn 1-step update into 2-step?



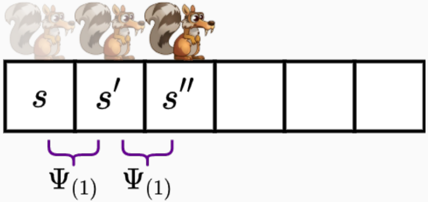
$$V^\pi(s) \leftarrow V^\pi(s) + \alpha \overbrace{\left(r + \gamma V^\pi(s') - V^\pi(s)\right)}^{\Psi_{(\mathbf{1})}(s,a)}$$

Backward view: idea

N-step update:

$$V^\pi(s) \leftarrow V^\pi(s) + \alpha\Psi_{(N)}(s,a)$$

How to turn 1-step update into 2-step?



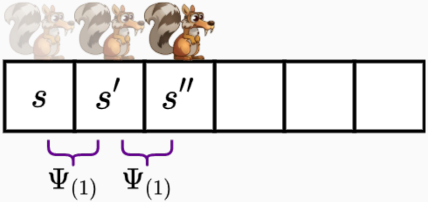
$$V^\pi(s) \leftarrow V^\pi(s) + \alpha \overbrace{\left(r + \gamma V^\pi(s') - V^\pi(s)\right)}^{\Psi_{(\mathbf{1})}(s,a)} + \alpha \overbrace{\left(\gamma r' + \gamma^2 V^\pi(s'') - \gamma V^\pi(s')\right)}^{\gamma\Psi_{(\mathbf{1})}(s',a')}$$

Backward view: idea

N-step update:

$$V^\pi(s) \leftarrow V^\pi(s) + \alpha \Psi_{(N)}(s, a)$$

How to turn 1-step update into 2-step?

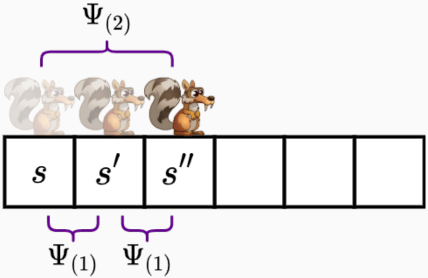


$$V^\pi(s) \leftarrow V^\pi(s) + \alpha \overbrace{\left(r + \gamma V^\pi(s') - V^\pi(s)\right)}^{\Psi_{(1)}(s,a)} + \alpha \overbrace{\left(\gamma r' + \gamma^2 V^\pi(s'') - \gamma V^\pi(s')\right)}^{\gamma \Psi_{(1)}(s',a')} =$$

N-step update:

$$V^\pi(s) \leftarrow V^\pi(s) + \alpha\Psi_{(N)}(s,a)$$

How to turn 1-step update into 2-step?



$$V^\pi(s) \leftarrow V^\pi(s) + \alpha\overbrace{\left(r + \gamma V^\pi(s') - V^\pi(s)\right)}^{\Psi_{(1)}(s,a)} + \alpha\overbrace{\left(\gamma r' + \gamma^2 V^\pi(s'') - \gamma V^\pi(s')\right)}^{\gamma\Psi_{(1)}(s',a')} =$$
$$= V^\pi(s) + \alpha\Psi_{(2)}(s,a)$$

Backward view: idea

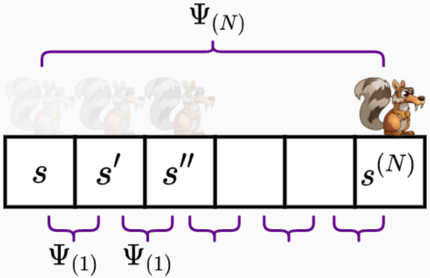
N-step update:

$$V^\pi(s) \leftarrow V^\pi(s) + \alpha \Psi_{(N)}(s, a)$$

How to turn 1-step update into 2-step?

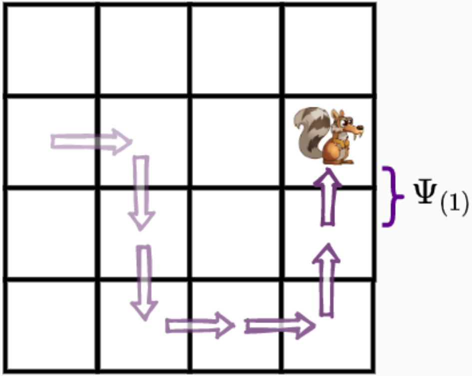
N–step error is a sum of 1–step errors

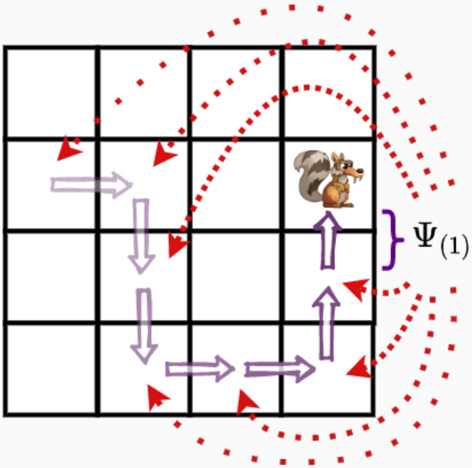
$$\Psi_{(N)}(s, a) = \sum_{t=0}^{N} \gamma^t \Psi_{(1)}(s^{(t)}, a^{(t)})$$



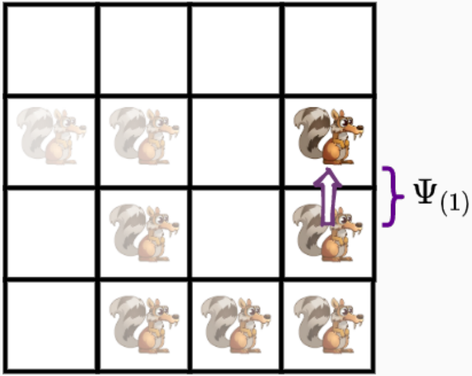
$$V^\pi(s) \leftarrow V^\pi(s) + \alpha \overbrace{\left(r + \gamma V^\pi(s') - V^\pi(s)\right)}^{\Psi_{(1)}(s,a)} + \alpha \overbrace{\left(\gamma r' + \gamma^2 V^\pi(s'') - \gamma V^\pi(s')\right)}^{\gamma \Psi_{(1)}(s',a')} =$$

$$= V^\pi(s) + \alpha \Psi_{(2)}(s, a)$$


$\Psi_{(1)}$



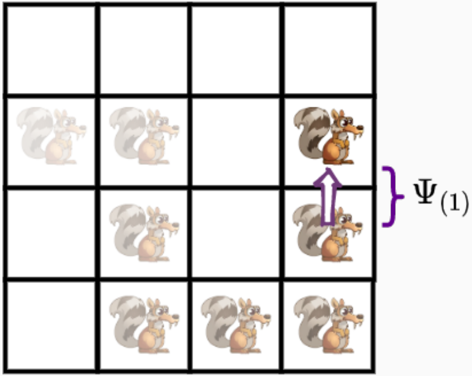
Use 1-step TD-error to update $V^{\pi}(s)$ for **all** states



Use 1-step TD-error to update $V^\pi(s)$ for **all** states

Define **eligibility trace** $e(s)$ as a coefficient of update:

$$\forall s \colon V^\pi(s) \leftarrow V^\pi(s) + \alpha e(s) \Psi_{(1)}$$



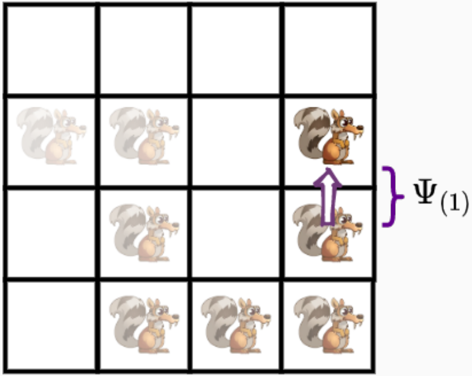
Use 1-step TD-error to update $V^{\pi}(s)$ for **all** states

Define **eligibility trace** $e(s)$ as a coefficient of update:

$$\forall s\colon V^{\pi}(s) \leftarrow V^{\pi}(s) + \alpha e(s)\Psi_{(1)}$$

Online «Monte-Carlo» updates:

- $\forall s\colon e(s) := 0$ at the start of each episode



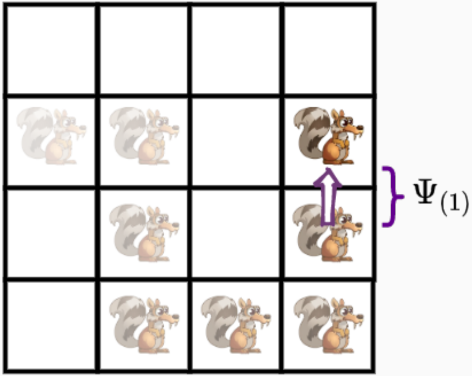
Use 1-step TD-error to update $V^\pi(s)$ for **all** states

Define **eligibility trace** $e(s)$ as a coefficient of update:

$$\forall s\colon\ V^\pi(s) \leftarrow V^\pi(s) + \alpha e(s)\Psi_{(1)}$$

Online «Monte-Carlo» updates:

- $\forall s\colon\ e(s) := 0$ at the start of each episode
- $e(s) \leftarrow e(s) + 1$ after visiting s



Use 1-step TD-error to update $V^\pi(s)$ for **all** states

Define **eligibility trace** $e(s)$ as a coefficient of update:

$$\forall s\colon V^\pi(s) \leftarrow V^\pi(s) + \alpha e(s)\Psi_{(1)}$$

Online «Monte-Carlo» updates:

- $\forall s\colon e(s) := 0$ at the start of each episode
- $e(s) \leftarrow e(s) + 1$ after visiting s
- $\forall s\colon e(s) \leftarrow \gamma e(s)$ after each step

TD(1) and TD(0)

TD(1)

Input: policy π

Initialize $V^\pi(s)$ arbitrarily

Initialize $e(s) = 0$

observe s_0

for $k = 0, 1, 2 \ldots$

- take action $a_k \sim \pi$, observe r_k, s_{k+1}

TD(1) and TD(0)

TD(1)

Input: policy π

Initialize $V^{\pi}(s)$ arbitrarily

Initialize $e(s) = 0$

observe s_0

for $k = 0, 1, 2 \ldots$

- take action $a_k \sim \pi$, observe r_k, s_{k+1}
- $\Psi_{(1)} := r_k + \gamma V^{\pi}(s_{k+1}) - V^{\pi}(s_k)$

TD(1) and TD(0)

TD(1)

Input: policy π

Initialize $V^\pi(s)$ arbitrarily

Initialize $e(s) = 0$

observe s_0

for $k = 0, 1, 2 \ldots$

- take action $a_k \sim \pi$, observe r_k, s_{k+1}
- $\Psi_{(1)} := r_k + \gamma V^\pi(s_{k+1}) - V^\pi(s_k)$
- $e(s_k) \leftarrow e(s_k) + 1$

TD(1) and TD(0)

TD(1)

Input: policy π

Initialize $V^\pi(s)$ arbitrarily

Initialize $e(s) = 0$

observe s_0

for $k = 0, 1, 2 \ldots$

- take action $a_k \sim \pi$, observe r_k, s_{k+1}
- $\Psi_{(1)} := r_k + \gamma V^\pi(s_{k+1}) - V^\pi(s_k)$
- $e(s_k) \leftarrow e(s_k) + 1$
- $\forall s\colon\ V^\pi(s) \leftarrow V^\pi(s) + \alpha e(s)\Psi_{(1)}$

TD(1)

Input: policy π

Initialize $V^\pi(s)$ arbitrarily

Initialize $e(s) = 0$

observe s_0

for $k = 0, 1, 2 \dots$

- take action $a_k \sim \pi$, observe r_k, s_{k+1}
- $\Psi_{(1)} := r_k + \gamma V^\pi(s_{k+1}) - V^\pi(s_k)$
- $e(s_k) \leftarrow e(s_k) + 1$
- $\forall s\colon V^\pi(s) \leftarrow V^\pi(s) + \alpha e(s) \Psi_{(1)}$
- $\forall s\colon e(s) \leftarrow \gamma e(s)$

TD(1) and TD(0)

TD(1)

Input: policy π

Initialize $V^\pi(s)$ arbitrarily

Initialize $e(s) = 0$

observe s_0

for $k = 0, 1, 2 \ldots$

- take action $a_k \sim \pi$, observe r_k, s_{k+1}
- $\Psi_{(1)} := r_k + \gamma V^\pi(s_{k+1}) - V^\pi(s_k)$
- $e(s_k) \leftarrow e(s_k) + 1$
- $\forall s\colon V^\pi(s) \leftarrow V^\pi(s) + \alpha e(s) \Psi_{(1)}$
- $\forall s\colon e(s) \leftarrow \gamma e(s)$

TD(0)

Input: policy π

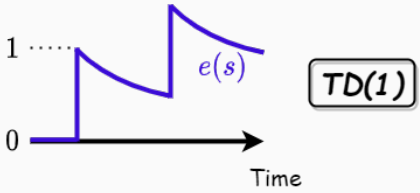
Initialize $V^\pi(s)$ arbitrarily

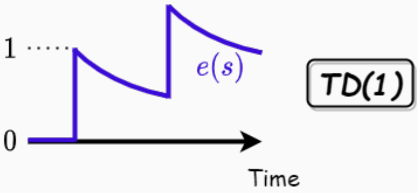
Initialize $e(s) = 0$

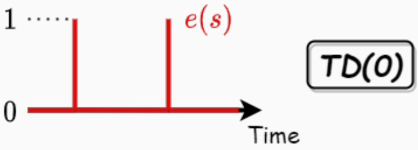
observe s_0

for $k = 0, 1, 2 \ldots$

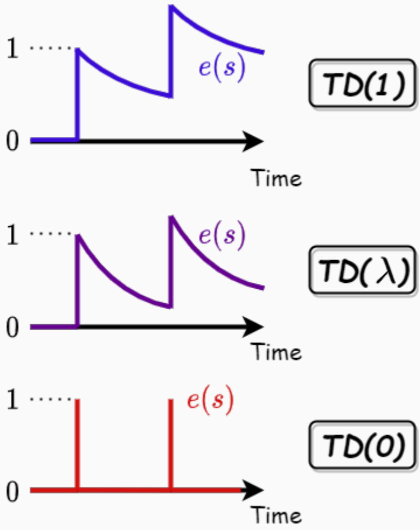
- take action $a_k \sim \pi$, observe r_k, s_{k+1}
- $\Psi_{(1)} := r_k + \gamma V^\pi(s_{k+1}) - V^\pi(s_k)$
- $e(s_k) \leftarrow e(s_k) + 1$
- $\forall s\colon V^\pi(s) \leftarrow V^\pi(s) + \alpha e(s) \Psi_{(1)}$
- $\forall s\colon e(s) \leftarrow 0 \cdot \gamma e(s)$

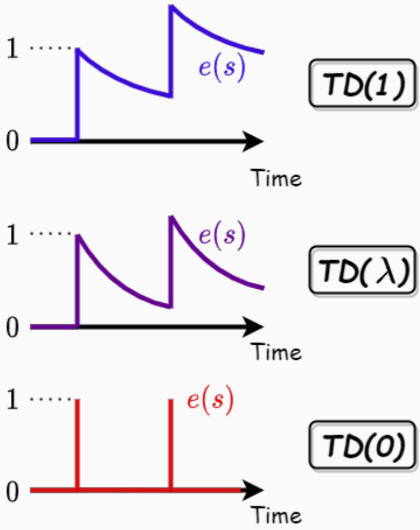

1
0
e(s)
Time
TD(1)


1
0
e(s)
TD(1)
Time


1
0
e(s)
TD(0)
Time

TD(λ)





TD (λ)

Input: policy π

Initialize $V^\pi(s)$ arbitrarily

Initialize $e(s) = 0$

observe s_0

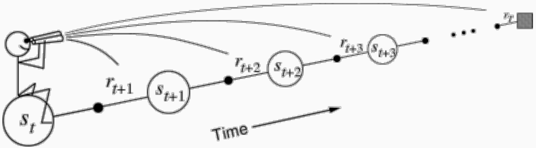
for $k = 0, 1, 2 \dots$

- take action $a_k \sim \pi$, observe r_k, s_{k+1}
- $\Psi_{(1)} := r_k + \gamma V^\pi(s_{k+1}) - V^\pi(s_k)$
- $e(s_k) \leftarrow e(s_k) + 1$
- $\forall s\colon\ V^\pi(s) \leftarrow V^\pi(s) + \alpha e(s) \Psi_{(1)}$
- $\forall s\colon\ e(s) \leftarrow \lambda \gamma e(s)$

Forward View

Give credit to **present** from known **future**

«is this decision good or bad based on the outcome?»

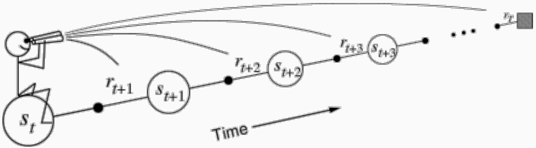


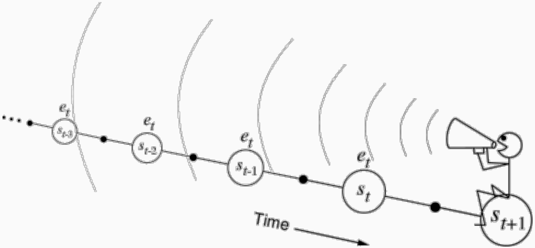
Backward view vs Forward view

Forward View

Give credit to **present** from known **future**

«is this decision good or bad based on the outcome?»





Backward View

Update **past** credits with **present** information

«which decisions in the past to blame?»

Forward view for TD(λ)

Step	**Update**	$\Psi_{(1)}(s,a)$	$\Psi_{(2)}(s,a)$	$\Psi_{(3)}(s,a)$	**...**	$\Psi_{(N)}(s,a)$
0	$\Psi_{(1)}(s,a)$	1	0	0		0

Forward view for TD(λ)

Step	Update	$\Psi_{(1)}(s,a)$	$\Psi_{(2)}(s,a)$	$\Psi_{(3)}(s,a)$	...	$\Psi_{(N)}(s,a)$
0	$\Psi_{(1)}(s,a)$	1	0	0		0
1	$\Psi_{(1)}(s,a) + \gamma\lambda\Psi_{(1)}(s',a')$	$1-\lambda$	λ	0		0

Forward view for TD(λ)

Step	Update	$\Psi_{(1)}(s,a)$	$\Psi_{(2)}(s,a)$	$\Psi_{(3)}(s,a)$	...	$\Psi_{(N)}(s,a)$
0	$\Psi_{(1)}(s,a)$	1	0	0		0
1	$\Psi_{(1)}(s,a) + \gamma\lambda\Psi_{(1)}(s',a')$	$1-\lambda$	λ	0		0
2	$\Psi_{(1)}(s,a) + \gamma\lambda\Psi_{(1)}(s',a') +$ $+ (\gamma\lambda)^2\Psi_{(1)}(s'',a'')$	$1-\lambda$	$(1-\lambda)\lambda$	λ^2		0

Forward view for TD(λ)

Step	Update	$\Psi_{(1)}(s,a)$	$\Psi_{(2)}(s,a)$	$\Psi_{(3)}(s,a)$	...	$\Psi_{(N)}(s,a)$
0	$\Psi_{(1)}(s,a)$	1	0	0		0
1	$\Psi_{(1)}(s,a) + \gamma\lambda\Psi_{(1)}(s',a')$	$1-\lambda$	λ	0		0
2	$\Psi_{(1)}(s,a) + \gamma\lambda\Psi_{(1)}(s',a') +$ $+(\gamma\lambda)^2\Psi_{(1)}(s'',a'')$	$1-\lambda$	$(1-\lambda)\lambda$	λ^2		0
$\vdots$						
N	$\sum_{t\geqslant 0}^{N}(\gamma\lambda)^t\Psi_{(1)}(s^{(t)},a^{(t)})$	$1-\lambda$	$(1-\lambda)\lambda$	$(1-\lambda)\lambda^2$		λ^N

Forward view for TD(λ)

Step	Update	$\Psi_{(1)}(s,a)$	$\Psi_{(2)}(s,a)$	$\Psi_{(3)}(s,a)$	...	$\Psi_{(N)}(s,a)$
0	$\Psi_{(1)}(s,a)$	1	0	0		0
1	$\Psi_{(1)}(s,a)+\gamma\lambda\Psi_{(1)}(s',a')$	$1-\lambda$	λ	0		0
2	$\Psi_{(1)}(s,a)+\gamma\lambda\Psi_{(1)}(s',a')+$ $+(\gamma\lambda)^2\Psi_{(1)}(s'',a'')$	$1-\lambda$	$(1-\lambda)\lambda$	λ^2		0
$\vdots$						
N	$\sum_{t\geqslant 0}^{N}(\gamma\lambda)^t\Psi_{(1)}(s^{(t)},a^{(t)})$	$1-\lambda$	$(1-\lambda)\lambda$	$(1-\lambda)\lambda^2$		λ^N

Equivalent forms of TD(λ) updates

$$\sum_{t=0}^{\infty}(\gamma\lambda)^t\Psi_{(1)}(s^{(t)},a^{(t)}) =$$

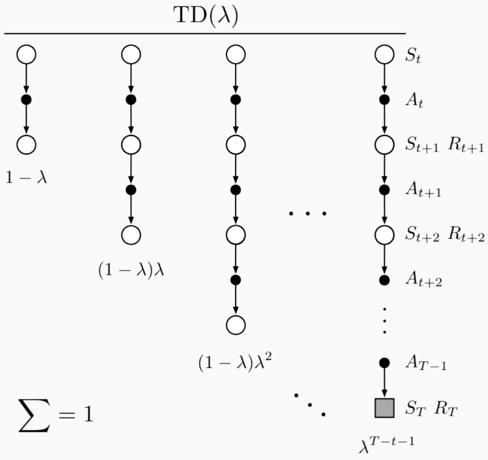
Forward view for TD(λ)

Step	Update	$\Psi_{(1)}(s,a)$	$\Psi_{(2)}(s,a)$	$\Psi_{(3)}(s,a)$	...	$\Psi_{(N)}(s,a)$
0	$\Psi_{(1)}(s,a)$	1	0	0		0
1	$\Psi_{(1)}(s,a) + \gamma\lambda\Psi_{(1)}(s',a')$	$1-\lambda$	λ	0		0
2	$\Psi_{(1)}(s,a) + \gamma\lambda\Psi_{(1)}(s',a') +$ $+ (\gamma\lambda)^2\Psi_{(1)}(s'',a'')$	$1-\lambda$	$(1-\lambda)\lambda$	λ^2		0
$\vdots$						
N	$\sum_{t\geqslant 0}^{N}(\gamma\lambda)^t\Psi_{(1)}(s^{(t)},a^{(t)})$	$1-\lambda$	$(1-\lambda)\lambda$	$(1-\lambda)\lambda^2$		λ^N

Equivalent forms of TD(λ) updates

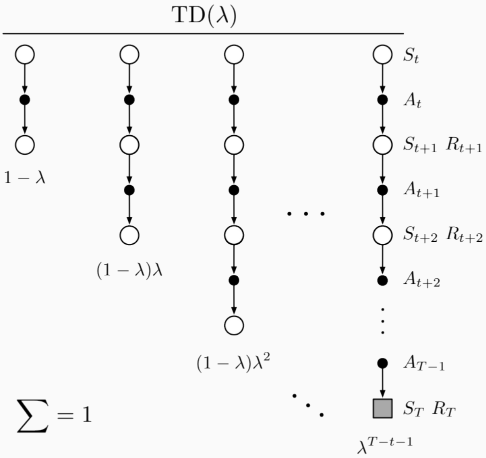
$$\sum_{t=0}^{\infty}(\gamma\lambda)^t\Psi_{(1)}(s^{(t)},a^{(t)}) = (1-\lambda)\sum_{N=1}^{\infty}\lambda^{N-1}\Psi_{(N)}(s,a)$$

Generalized Advantage Estimation (GAE)



What if for some pair s, a we do not know our future until the end of episode, but only T steps ahead?

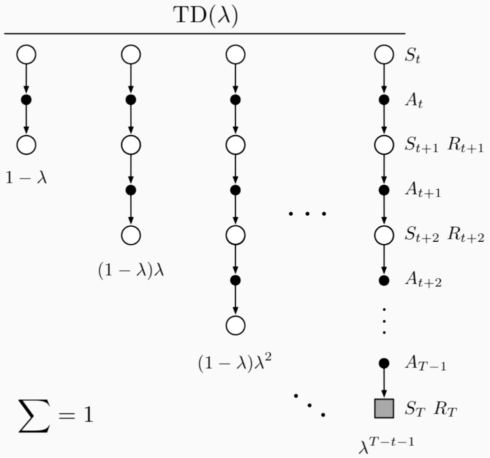
Generalized Advantage Estimation (GAE)



What if for some pair s, a we do not know our future until the end of episode, but only T steps ahead?

$$\Psi^{\mathrm{GAE}}(s, a) := \sum_{t=0}^{T} (\gamma\lambda)^t \Psi_{(1)}(s^{(t)}, a^{(t)})$$

Generalized Advantage Estimation (GAE)

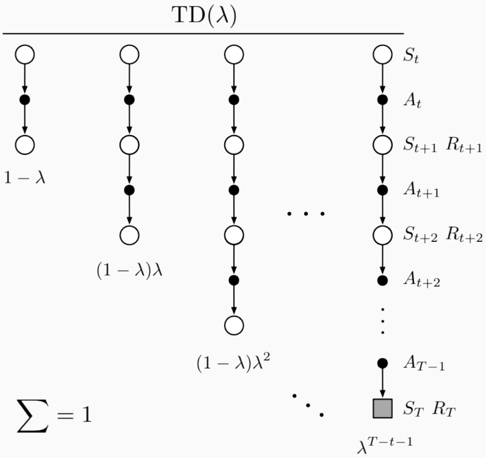


What if for some pair s, a we do not know our future until the end of episode, but only T steps ahead?

$$\Psi^{\mathrm{GAE}}(s, a) := \sum_{t=0}^{T} (\gamma\lambda)^t \Psi_{(1)}(s^{(t)}, a^{(t)})$$

Equation used in practice:

$$\Psi^{\mathrm{GAE}}(s_t, a_t) = \Psi_{(1)}(s_t, a_t) + \\ + \lambda\gamma(1 - \mathrm{done}_{t+1})\Psi^{\mathrm{GAE}}(s_{t+1}, a_{t+1})$$



What if for some pair s, a we do not know our future until the end of episode, but only T steps ahead?

$$\Psi^{\mathrm{GAE}}(s, a) := \sum_{t=0}^{T} (\gamma\lambda)^t \Psi_{(1)}(s^{(t)}, a^{(t)})$$

Equation used in practice:

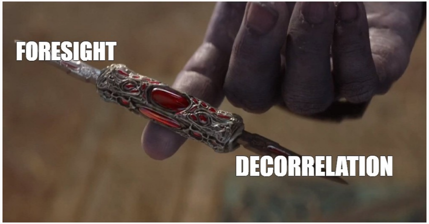
$$\Psi^{\mathrm{GAE}}(s_t, a_t) = \Psi_{(1)}(s_t, a_t) + \\ + \lambda\gamma(1 - \mathrm{done}_{t+1})\Psi^{\mathrm{GAE}}(s_{t+1}, a_{t+1})$$

GAE in Advantage Actor-Critic

Longer rollouts produce richer GAE ensemble.

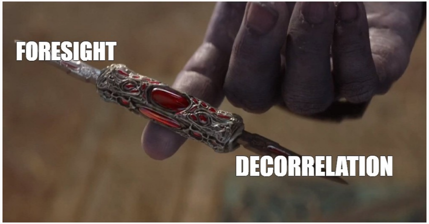
GAE in Advantage Actor-Critic

Longer rollouts produce richer GAE ensemble.



GAE in Advantage Actor-Critic

Longer rollouts produce richer GAE ensemble.



In A2C rollouts are usually short, so $\lambda = 1$ is common choice.

(sometimes called **max-trace** estimation)

Combining all together

Proximal Policy Optimization: implementation matters

Key elements:

- ✓ Clipped policy loss
- ✓ Clipped critic loss
- ✓ GAE

Pipeline details:

- ! Advantage normalization in mini-batches
- No KL regularization
- Entropy loss

Other hacks:

- ! Reward normalization[1] and clipping
- Observations normalization and clipping[2]
- Orthogonal initialization of layers
- ϵ (clipping parameter) annealing

Standard tricks:

- Adam, learning rate annealing
- Tanh activation functions
- ! Gradient clipping

[1] divided by running std of collected cumulative rewards
[2] can be critical in continuous control

Full Pipeline: pt.I

Proximal Policy Optimization (PPO)

Initialize $\pi(a \mid s, \theta), V_\phi^\pi(s)$;

Full Pipeline: pt.I

Proximal Policy Optimization (PPO)

Initialize $\pi(a \mid s, \theta), V^{\pi}_{\phi}(s)$;

for $k = 0, 1, 2 \ldots$

- collect several rollouts $s_0, a_0, r_0, s_1, \mathrm{done}_1, a_1 \ldots s_N, \mathrm{done}_N$ using $\pi(a \mid s, \theta)$;
  store probabilities of selected actions as $\pi^{\mathrm{old}}(a_t \mid s_t) := \pi(a_t \mid s_t, \theta)$
  store critic output as $V^{\mathrm{old}}(s_t) := V^{\pi}_{\phi}(s_t)$

Full Pipeline: pt.I

Proximal Policy Optimization (PPO)

Initialize $\pi(a \mid s, \theta)$, $V^{\pi}_{\phi}(s)$;

for $k = 0, 1, 2 \ldots$

- collect several rollouts $s_0, a_0, r_0, s_1, \mathrm{done}_1, a_1 \ldots s_N, \mathrm{done}_N$ using $\pi(a \mid s, \theta)$;
  store probabilities of selected actions as $\pi^{\mathrm{old}}(a_t \mid s_t) := \pi(a_t \mid s_t, \theta)$
  store critic output as $V^{\mathrm{old}}(s_t) := V^{\pi}_{\phi}(s_t)$
- compute 1-step errors: $\Psi_{(1)}(s_t, a_t) := r_t + \gamma(1 - \mathrm{done}_{t+1}) V^{\pi}_{\phi}(s_{t+1}) - V^{\pi}_{\phi}(s_t)$

Full Pipeline: pt.I

Proximal Policy Optimization (PPO)

Initialize $\pi(a \mid s, \theta), V^{\pi}_{\phi}(s)$;

for $k = 0, 1, 2 \ldots$

- collect several rollouts $s_0, a_0, r_0, s_1, \mathrm{done}_1, a_1 \ldots s_N, \mathrm{done}_N$ using $\pi(a \mid s, \theta)$;
  store probabilities of selected actions as $\pi^{\mathrm{old}}(a_t \mid s_t) := \pi(a_t \mid s_t, \theta)$
  store critic output as $V^{\mathrm{old}}(s_t) := V^{\pi}_{\phi}(s_t)$
- compute 1-step errors: $\Psi_{(1)}(s_t, a_t) := r_t + \gamma(1 - \mathrm{done}_{t+1})V^{\pi}_{\phi}(s_{t+1}) - V^{\pi}_{\phi}(s_t)$
- compute GAE advantage estimations: $\Psi^{\mathrm{GAE}}(s_{N-1}, a_{N-1}) := \Psi_{(1)}(s_{N-1}, a_{N-1})$
- for t from $N - 2$ to 0:
  - $\Psi^{\mathrm{GAE}}(s_t, a_t) :=$

Full Pipeline: pt.I

Proximal Policy Optimization (PPO)

Initialize $\pi(a \mid s, \theta)$, $V^{\pi}_{\phi}(s)$;

for $k = 0, 1, 2 \dots$

- collect several rollouts $s_0, a_0, r_0, s_1, \mathrm{done}_1, a_1 \dots s_N, \mathrm{done}_N$ using $\pi(a \mid s, \theta)$;
  store probabilities of selected actions as $\pi^{\mathrm{old}}(a_t \mid s_t) := \pi(a_t \mid s_t, \theta)$
  store critic output as $V^{\mathrm{old}}(s_t) := V^{\pi}_{\phi}(s_t)$
- compute 1-step errors: $\Psi_{(1)}(s_t, a_t) := r_t + \gamma(1 - \mathrm{done}_{t+1}) V^{\pi}_{\phi}(s_{t+1}) - V^{\pi}_{\phi}(s_t)$
- compute GAE advantage estimations: $\Psi^{\mathrm{GAE}}(s_{N-1}, a_{N-1}) := \Psi_{(1)}(s_{N-1}, a_{N-1})$
- for t from $N - 2$ to 0:
  - $\Psi^{\mathrm{GAE}}(s_t, a_t) := \Psi_{(1)}(s_t, a_t) + \lambda\gamma(1 - \mathrm{done}_{t+1})\Psi^{\mathrm{GAE}}(s_{t+1}, a_{t+1})$

Full Pipeline: pt.I

Proximal Policy Optimization (PPO)

Initialize $\pi(a \mid s, \theta)$, $V^{\pi}_{\phi}(s)$;

for $k = 0, 1, 2 \dots$

- collect several rollouts $s_0, a_0, r_0, s_1, \mathrm{done}_1, a_1 \dots s_N, \mathrm{done}_N$ using $\pi(a \mid s, \theta)$;
  store probabilities of selected actions as $\pi^{\mathrm{old}}(a_t \mid s_t) := \pi(a_t \mid s_t, \theta)$
  store critic output as $V^{\mathrm{old}}(s_t) := V^{\pi}_{\phi}(s_t)$
- compute 1-step errors: $\Psi_{(1)}(s_t, a_t) := r_t + \gamma(1 - \mathrm{done}_{t+1})V^{\pi}_{\phi}(s_{t+1}) - V^{\pi}_{\phi}(s_t)$
- compute GAE advantage estimations: $\Psi^{\mathrm{GAE}}(s_{N-1}, a_{N-1}) := \Psi_{(1)}(s_{N-1}, a_{N-1})$
- for t from $N - 2$ to 0:
  - $\Psi^{\mathrm{GAE}}(s_t, a_t) := \Psi_{(1)}(s_t, a_t) + \lambda\gamma(1 - \mathrm{done}_{t+1})\Psi^{\mathrm{GAE}}(s_{t+1}, a_{t+1})$
- compute critic targets: $y(s_t) := \Psi^{\mathrm{GAE}}(s_t, a_t) + V^{\pi}_{\phi}(s_t)$

Full Pipeline: pt.I

Proximal Policy Optimization (PPO)

Initialize $\pi(a \mid s, \theta)$, $V^{\pi}_{\phi}(s)$;

for $k = 0, 1, 2 \dots$

- collect several rollouts $s_0, a_0, r_0, s_1, \mathrm{done}_1, a_1 \dots s_N, \mathrm{done}_N$ using $\pi(a \mid s, \theta)$;
  store probabilities of selected actions as $\pi^{\mathrm{old}}(a_t \mid s_t) := \pi(a_t \mid s_t, \theta)$
  store critic output as $V^{\mathrm{old}}(s_t) := V^{\pi}_{\phi}(s_t)$
- compute 1-step errors: $\Psi_{(1)}(s_t, a_t) := r_t + \gamma(1 - \mathrm{done}_{t+1})V^{\pi}_{\phi}(s_{t+1}) - V^{\pi}_{\phi}(s_t)$
- compute GAE advantage estimations: $\Psi^{\mathrm{GAE}}(s_{N-1}, a_{N-1}) := \Psi_{(1)}(s_{N-1}, a_{N-1})$
- for t from $N-2$ to 0:
  - $\Psi^{\mathrm{GAE}}(s_t, a_t) := \Psi_{(1)}(s_t, a_t) + \lambda\gamma(1 - \mathrm{done}_{t+1})\Psi^{\mathrm{GAE}}(s_{t+1}, a_{t+1})$
- compute critic targets: $y(s_t) := \Psi^{\mathrm{GAE}}(s_t, a_t) + V^{\pi}_{\phi}(s_t)$
- construct dataset of $(s_t, a_t, \Psi^{\mathrm{GAE}}(s_t, a_t), y(s_t), \pi^{\mathrm{old}}(a_t \mid s_t), V^{\mathrm{old}}(s_t))$

Full Pipeline: pt.II

Proximal Policy Optimization (PPO) -- cont.

- go through dataset $\mathrm{n\_epochs}$ times, sampling mini-batches of size B; for each mini-batch:

Full Pipeline: pt.II

Proximal Policy Optimization (PPO) -- cont.

- go through dataset $\mathrm{n\_epochs}$ times, sampling mini-batches of size B; for each mini-batch:
  - normalize $\Psi^{\mathrm{GAE}}(s, a)$ in the batch by subtracting mean and dividing by std

Full Pipeline: pt.II

Proximal Policy Optimization (PPO) -- cont.

- go through dataset $\mathrm{n\_epochs}$ times, sampling mini-batches of size B; for each mini-batch:
  - normalize $\Psi^{\mathrm{GAE}}(s, a)$ in the batch by subtracting mean and dividing by std
  - compute importance sampling weights:

$$\rho(s, a, \theta) := \frac{\pi(a \mid s, \theta)}{\pi^{\mathrm{old}}(a \mid s)}, \qquad \rho^{\mathrm{clip}}(s, a, \theta) = \mathrm{clip}(\rho(s, a, \theta), 1 - \epsilon, 1 + \epsilon)$$

Full Pipeline: pt.II

Proximal Policy Optimization (PPO) -- cont.

- go through dataset n_epochs times, sampling mini-batches of size B; for each mini-batch:
  - normalize $\Psi^{\text{GAE}}(s, a)$ in the batch by subtracting mean and dividing by std
  - compute importance sampling weights:

$$\rho(s, a, \theta) := \frac{\pi(a \mid s, \theta)}{\pi^{\text{old}}(a \mid s)}, \qquad \rho^{\text{clip}}(s, a, \theta) = \text{clip}(\rho(s, a, \theta), 1 - \epsilon, 1 + \epsilon)$$

  - update actor:

$$L_1(s, a, \theta) := \rho(s, a, \theta)\Psi^{\text{GAE}}(s, a), \qquad L_2(s, a, \theta) := \rho^{\text{clip}}(s, a, \theta)\Psi^{\text{GAE}}(s, a)$$

$$\theta \leftarrow \theta + \alpha \nabla_\theta \frac{1}{B} \sum_{s,a} \min(L_1(s, a, \theta), L_2(s, a, \theta))$$

Full Pipeline: pt.II

Proximal Policy Optimization (PPO) -- cont.

- go through dataset n_epochs times, sampling mini-batches of size B; for each mini-batch:
  - normalize $\Psi^{\text{GAE}}(s, a)$ in the batch by subtracting mean and dividing by std
  - compute importance sampling weights:

$$\rho(s, a, \theta) := \frac{\pi(a \mid s, \theta)}{\pi^{\text{old}}(a \mid s)}, \qquad \rho^{\text{clip}}(s, a, \theta) = \text{clip}(\rho(s, a, \theta), 1 - \epsilon, 1 + \epsilon)$$

  - update actor:

$$L_1(s, a, \theta) := \rho(s, a, \theta)\Psi^{\text{GAE}}(s, a), \qquad L_2(s, a, \theta) := \rho^{\text{clip}}(s, a, \theta)\Psi^{\text{GAE}}(s, a)$$

$$\theta \leftarrow \theta + \alpha \nabla_\theta \frac{1}{B} \sum_{s,a} \min(L_1(s, a, \theta), L_2(s, a, \theta))$$

  - update critic:

$$\text{Loss}_1(s, \phi) := \left(y(s) - V^\pi_\phi(s)\right)^2$$

$$\text{Loss}_2(s, \phi) := \left(y(s) - V^{\text{old}}(s) - \text{clip}(V^\pi_\phi(s) - V^{\text{old}}(s), \hat{\epsilon}, -\hat{\epsilon})\right)^2$$

$$\phi \leftarrow \phi - \alpha \nabla_\phi \frac{1}{B} \sum_{s} \max(\text{Loss}_1(s, \phi), \text{Loss}_2(s, \phi))$$

Literature

- Proximal Policy Optimization Algorithms;
- Implementation Matters in Deep Policy Gradients: A Case Study on PPO and TRPO;
- High-Dimensional Continuous Control Using Generalized Advantage Estimation;
- Sutton, Barto — Reinforcement Learning, an Introduction, ch. 12;

Appendix: Retrace

Reminder: Policy Gradient VS Value-based

Value-based (DQN+)

✓ off-policy;
(can use experience replay)

✗ trains $Q^*(s, a)$;
(complicated intermediate stage)

✗ exploration-exploitation issues;
(since Value Iteration works with deterministic policies)

✗ 1-step targets;
(can we do anything about it?)

Policy Gradient

✗ on-policy;
(all data is useless after each SGD step)

✓ trains policy directly;
(requires only $V^\pi(s)$, which is much simpler)

✓ «natural» exploration;
(sampling from stochastic policy $\pi(a \mid s)$)

✓ ∞-step targets;
(can use GAE for both critic and actor)

Reminder: Policy Gradient VS Value-based

Value-based (DQN+)

✓ off-policy;
(can use experience replay)

✗ trains $Q^*(s, a)$;
(complicated intermediate stage)

✗ exploration-exploitation issues;
(since Value Iteration works with deterministic policies)

✗ 1-step targets;
(can we do anything about it?)

Policy Gradient

✗ on-policy;
(all data is useless after each SGD step)

✓ trains policy directly;
(requires only $V^\pi(s)$, which is much simpler)

✓ «natural» exploration;
(sampling from stochastic policy $\pi(a \mid s)$)

✓ ∞-step targets;
(can use GAE for both critic and actor)

Off-policy Credit Assignment

Given rollout $s_0, r_0, s_1, r_1, s_2, r_2 \ldots s_M$ from policy μ and approximation of $V^\pi(s)$

Off-policy Credit Assignment

Given rollout $s_0, r_0, s_1, r_1, s_2, r_2 \ldots s_M$ from policy μ and approximation of $V^\pi(s)$ perform **credit assignment** for state-action pair s_0, a_0 in **off-policy** mode: $\mu \neq \pi$

Off-policy Credit Assignment

Given rollout $s_0, r_0, s_1, r_1, s_2, r_2 \ldots s_M$ from policy μ and approximation of $V^\pi(s)$ perform **credit assignment** for state-action pair s_0, a_0 in **off-policy** mode: $\mu \neq \pi$

Would be great to use GAE:

$$\sum_{t \geqslant 0} (\gamma\lambda)^t \Psi_{(1)}(s_t, a_t),$$

but $\Psi_{(1)}(s_t, a_t)$ depends on random variables: $a_0, s_0, a_1, s_2, \ldots s_{t+1}$.

Off-policy Credit Assignment

Given rollout $s_0, r_0, s_1, r_1, s_2, r_2 \dots s_M$ from policy μ and approximation of $V^\pi(s)$ perform **credit assignment** for state-action pair s_0, a_0 in **off-policy** mode: $\mu \neq \pi$

Would be great to use GAE:

$$\sum_{t \geqslant 0} (\gamma\lambda)^t \Psi_{(1)}(s_t, a_t),$$

but $\Psi_{(1)}(s_t, a_t)$ depends on random variables: $a_0, s_0, a_1, s_2, \dots s_{t+1}$.

Danger!

If $\pi(a_0 | s_0) = 0$ than we can't do anything.

Importance Sampling Correction

Use importance sampling correction!

Importance Sampling Correction

Use importance sampling correction!

$$\Psi = \sum_{t \geqslant 0} (\gamma\lambda)^t \left(\prod_{\hat{t}=0}^{\hat{t}=t} \frac{\pi(a_{\hat{t}} \mid s_{\hat{t}}) p(s_{\hat{t}+1} \mid s_{\hat{t}}, a_{\hat{t}})}{\mu(a_{\hat{t}} \mid s_{\hat{t}}) p(s_{\hat{t}+1} \mid s_{\hat{t}}, a_{\hat{t}})} \right) \Psi_{(1)}(s_t, a_t) =$$

$$=$$

Importance Sampling Correction

Use importance sampling correction!

$$\Psi = \sum_{t \geqslant 0} (\gamma\lambda)^t \left(\prod_{\hat{t}=0}^{\hat{t}=t} \frac{\pi(a_{\hat{t}} \mid s_{\hat{t}}) \cancel{p(s_{\hat{t}+1} \mid s_{\hat{t}}, a_{\hat{t}})}}{\mu(a_{\hat{t}} \mid s_{\hat{t}}) \cancel{p(s_{\hat{t}+1} \mid s_{\hat{t}}, a_{\hat{t}})}} \right) \Psi_{(1)}(s_t, a_t) =$$

$$= \sum_{t \geqslant 0} (\gamma\lambda)^t \left(\prod_{\hat{t}=0}^{\hat{t}=t} \frac{\pi(a_{\hat{t}} \mid s_{\hat{t}})}{\mu(a_{\hat{t}} \mid s_{\hat{t}})} \right) \Psi_{(1)}(s_t, a_t)$$

Importance Sampling Correction

Use importance sampling correction!

$$\Psi = \sum_{t \geqslant 0} (\gamma\lambda)^t \left(\prod_{\hat{t}=0}^{\hat{t}=t} \frac{\pi(a_{\hat{t}} \mid s_{\hat{t}}) \cancel{p(s_{\hat{t}+1} \mid s_{\hat{t}}, a_{\hat{t}})}}{\mu(a_{\hat{t}} \mid s_{\hat{t}}) \cancel{p(s_{\hat{t}+1} \mid s_{\hat{t}}, a_{\hat{t}})}} \right) \Psi_{(1)}(s_t, a_t) =$$

$$= \sum_{t \geqslant 0} (\gamma\lambda)^t \left(\prod_{\hat{t}=0}^{\hat{t}=t} \frac{\pi(a_{\hat{t}} \mid s_{\hat{t}})}{\mu(a_{\hat{t}} \mid s_{\hat{t}})} \right) \Psi_{(1)}(s_t, a_t)$$

Impractical: extremely high variance!

Importance Sampling Correction

Use importance sampling correction!

$$\Psi = \sum_{t \geqslant 0} (\gamma\lambda)^t \left(\prod_{\hat{t}=0}^{\hat{t}=t} \frac{\pi(a_{\hat{t}} \mid s_{\hat{t}}) \cancel{p(s_{\hat{t}+1} \mid s_{\hat{t}}, a_{\hat{t}})}}{\mu(a_{\hat{t}} \mid s_{\hat{t}}) \cancel{p(s_{\hat{t}+1} \mid s_{\hat{t}}, a_{\hat{t}})}} \right) \Psi_{(1)}(s_t, a_t) =$$

$$= \sum_{t \geqslant 0} (\gamma\lambda)^t \left(\prod_{\hat{t}=0}^{\hat{t}=t} \frac{\pi(a_{\hat{t}} \mid s_{\hat{t}})}{\mu(a_{\hat{t}} \mid s_{\hat{t}})} \right) \Psi_{(1)}(s_t, a_t)$$

Impractical: extremely high variance!

- **vanishing** trace: $\mu(a|s) \gg \pi(a|s)$
  - typical situation: μ making stupid random moves that π rarely does now. No cure.

Importance Sampling Correction

Use importance sampling correction!

$$\Psi = \sum_{t \geqslant 0} (\gamma\lambda)^t \left(\prod_{\hat{t}=0}^{\hat{t}=t} \frac{\pi(a_{\hat{t}} \mid s_{\hat{t}}) \cancel{p(s_{\hat{t}+1} \mid s_{\hat{t}}, a_{\hat{t}})}}{\mu(a_{\hat{t}} \mid s_{\hat{t}}) \cancel{p(s_{\hat{t}+1} \mid s_{\hat{t}}, a_{\hat{t}})}} \right) \Psi_{(1)}(s_t, a_t) =$$

$$= \sum_{t \geqslant 0} (\gamma\lambda)^t \left(\prod_{\hat{t}=0}^{\hat{t}=t} \frac{\pi(a_{\hat{t}} \mid s_{\hat{t}})}{\mu(a_{\hat{t}} \mid s_{\hat{t}})} \right) \Psi_{(1)}(s_t, a_t)$$

Impractical: extremely high variance!

- **vanishing** trace: $\mu(a|s) \gg \pi(a|s)$
  - typical situation: μ making stupid random moves that π rarely does now. No cure.
- **exploding** trace: $\mu(a|s) \ll \pi(a|s)$
  - μ selected action with small $\mu(a|s)$, but *probable* for π. Is the reason of high variance.

Let's rewrite credit in the following way:

$$\Psi = \sum_{t \geqslant 0} \gamma^t \left(\prod_{i=0}^{i=t} c_i \right) \Psi_{(1)}(s_t, a_t),$$

where c_i are coefficients of «trace annealing»:

Name	**Coefficients c_i**	**Issue**
GAE	λ	on-policy only

Credit Assignment: General Form

Let's rewrite credit in the following way:

$$\Psi = \sum_{t \geqslant 0} \gamma^t \left(\prod_{i=0}^{i=t} c_i \right) \Psi_{(1)}(s_t, a_t),$$

where c_i are coefficients of «trace annealing»:

Name	**Coefficients c_i**	**Issue**
GAE	λ	on-policy only
One-step	0	high bias

Credit Assignment: General Form

Let's rewrite credit in the following way:

$$\Psi = \sum_{t \geqslant 0} \gamma^t \left(\prod_{i=0}^{i=t} c_i \right) \Psi_{(1)}(s_t, a_t),$$

where c_i are coefficients of «trace annealing»:

Name	**Coefficients** c_i	**Issue**
GAE	λ	on-policy only
One-step	0	high bias
Importance Sampling	$\lambda \frac{\pi(a_i \mid s_i)}{\mu(a_i \mid s_i)}$	easily explodes

Retrace: Main Theorem

$$\Psi = \sum_{t \geqslant 0} \gamma^t \left(\prod_{i=0}^{i=t} c_i \right) \Psi_{(1)}(s_t, a_t),$$

Retrace Theorem

While in on-policy mode you could select **any** coefficient $c_i \in [0, 1]$, in off-policy mode you can select **any** coefficient

$$c_i \in \left[0, \frac{\pi(a_i \mid s_i)}{\mu(a_i \mid s_i)} \right]$$

Retrace: Main Theorem

$$\Psi = \sum_{t \geqslant 0} \gamma^t \left(\prod_{i=0}^{i=t} c_i \right) \Psi_{(1)}(s_t, a_t),$$

Retrace Theorem

While in on-policy mode you could select **any** coefficient $c_i \in [0, 1]$, in off-policy mode you can select **any** coefficient

$$c_i \in \left[0, \frac{\pi(a_i \mid s_i)}{\mu(a_i \mid s_i)} \right]$$

- **vanishing trace:** can't do anything;

Retrace: Main Theorem

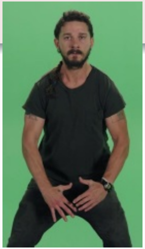
$$\Psi = \sum_{t \geqslant 0} \gamma^t \left(\prod_{i=0}^{i=t} c_i \right) \Psi_{(1)}(s_t, a_t),$$

Retrace Theorem

While in on-policy mode you could select **any** coefficient $c_i \in [0, 1]$, in off-policy mode you can select **any** coefficient

$$c_i \in \left[0, \frac{\pi(a_i \mid s_i)}{\mu(a_i \mid s_i)}\right]$$

- **vanishing trace:** can't do anything;
- **exploding trace:** if importance sampling is more than 1, JUST CLIP IT!



Retrace: final result

$$\Psi = \sum_{t \geqslant 0} \gamma^t \left(\prod_{i=0}^{i=t} c_i \right) \Psi_{(1)}(s_t, a_t),$$

where

$$c_i := \lambda \min \left(1, \frac{\pi(a_i \mid s_i)}{\mu(a_i \mid s_i)} \right)$$

$$\Psi = \sum_{t \geqslant 0} \gamma^t \left(\prod_{i=0}^{i=t} c_i \right) \Psi_{(1)}(s_t, a_t),$$

where

$$c_i := \lambda \min \left(1, \frac{\pi(a_i \mid s_i)}{\mu(a_i \mid s_i)} \right)$$

Used in:

- off-policy RL algorithms for theoretically correct multi-step targets;
    - ($\lambda = 1$ because it vanishes fast)

Retrace: final result

$$\Psi = \sum_{t \geqslant 0} \gamma^t \left(\prod_{i=0}^{i=t} c_i \right) \Psi_{(1)}(s_t, a_t),$$

where

$$c_i := \lambda \min \left(1, \frac{\pi(a_i \mid s_i)}{\mu(a_i \mid s_i)} \right)$$

Used in:

- off-policy RL algorithms for theoretically correct multi-step targets;
  - ($\lambda = 1$ because it vanishes fast)
- distributed on-policy RL systems where data about gradient from some servers can be several updates late.