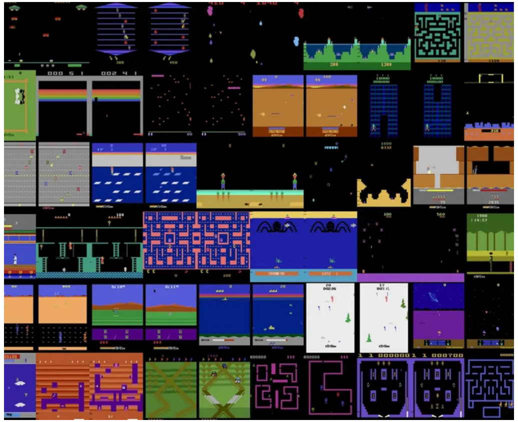
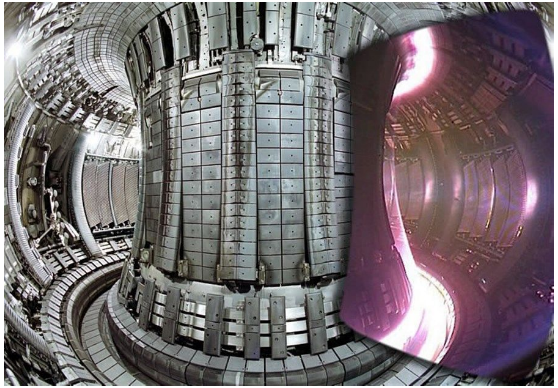
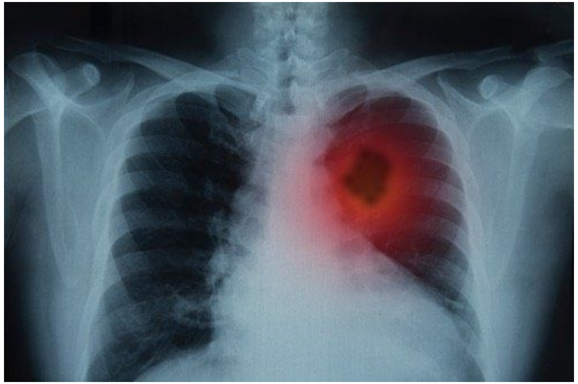
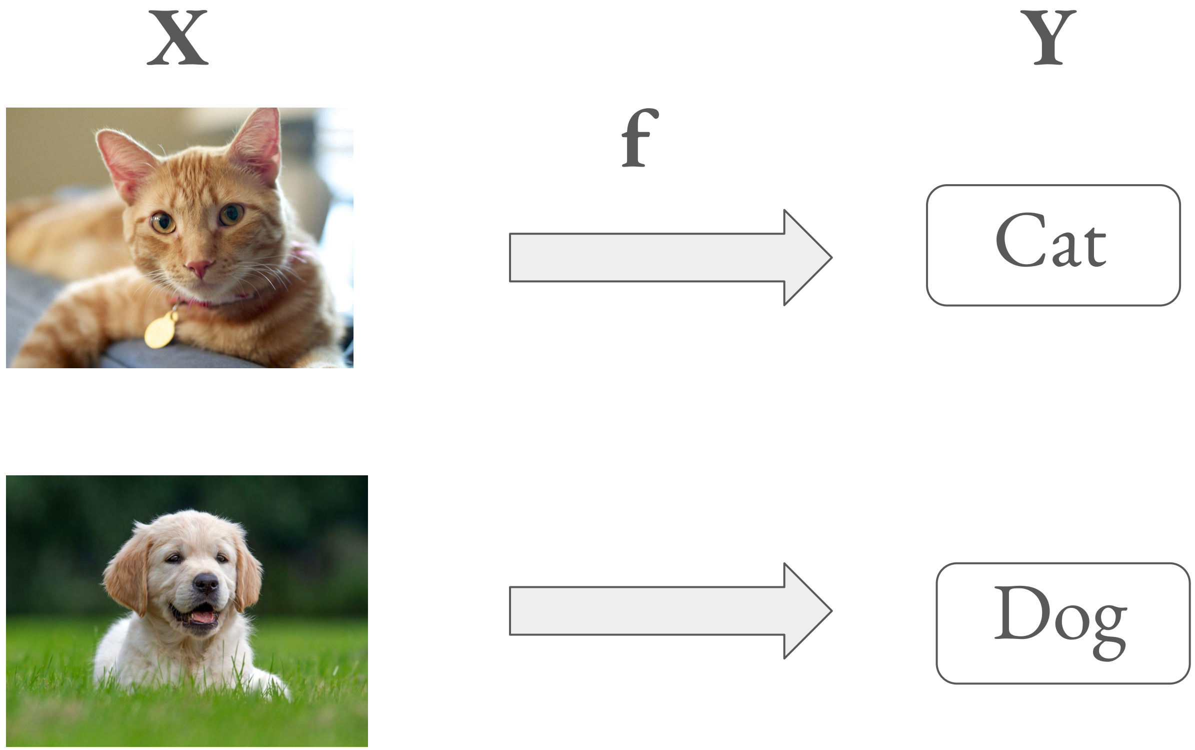
Continual Meta RL

By Kellen Kanarios

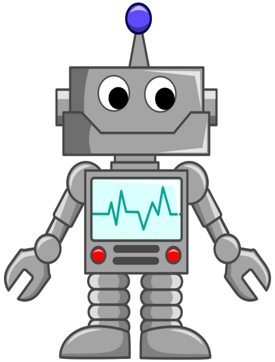
Reinforcement Learning

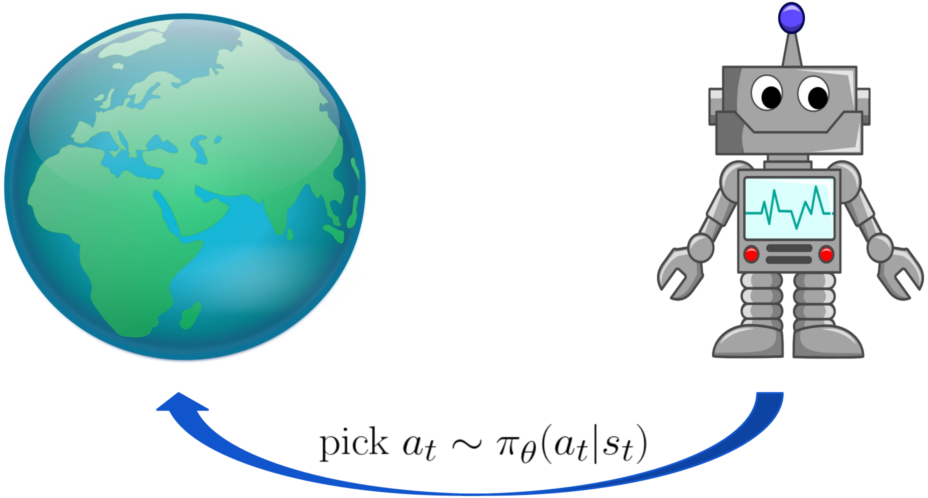

Supervised Learning



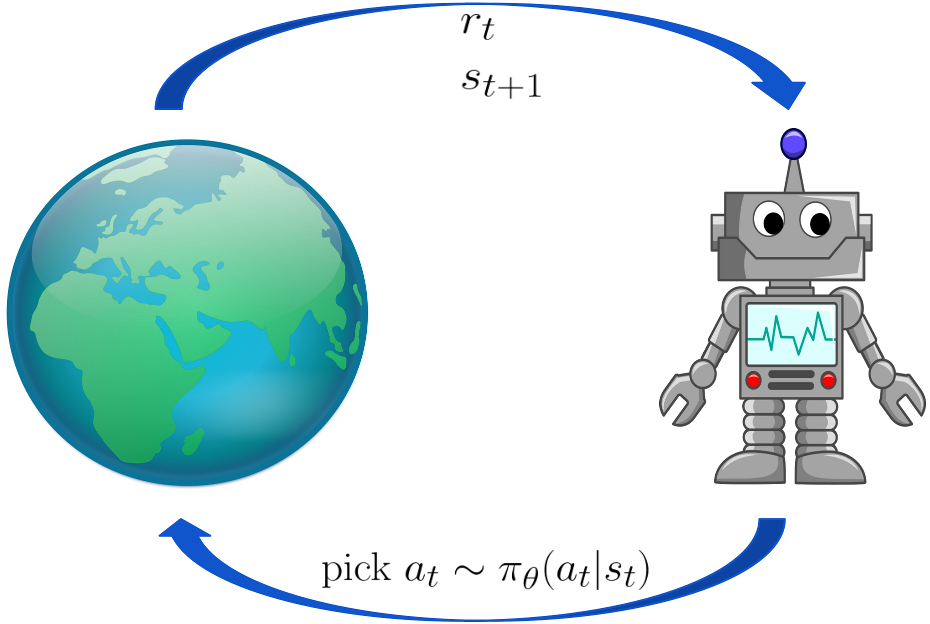
Reinforcement Learning



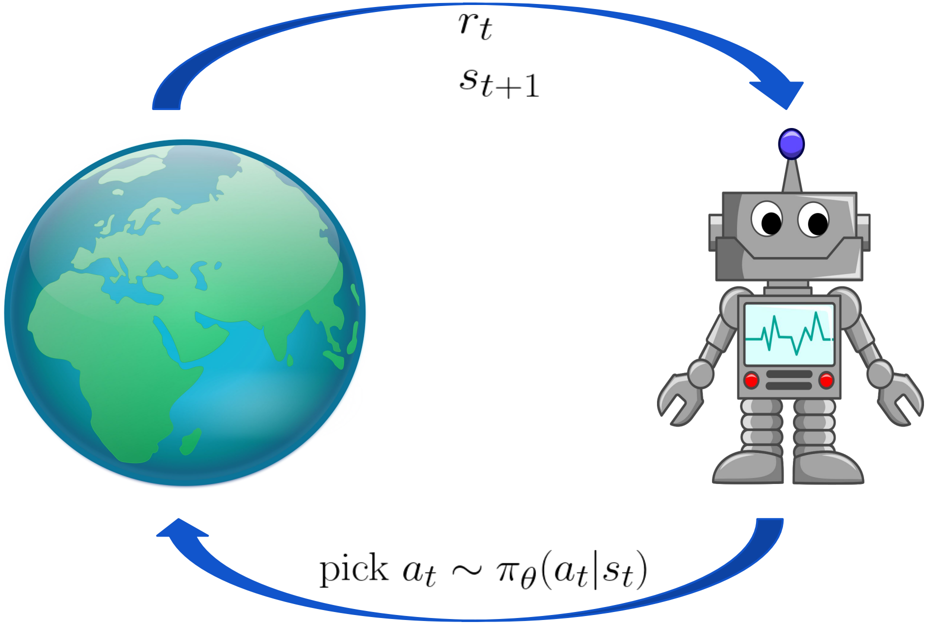
Reinforcement Learning



Reinforcement Learning



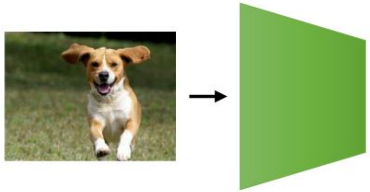
Reinforcement Learning



use (s_t, a_t, s_{t+1}, r_t) to improve π_θ

Meta Learning

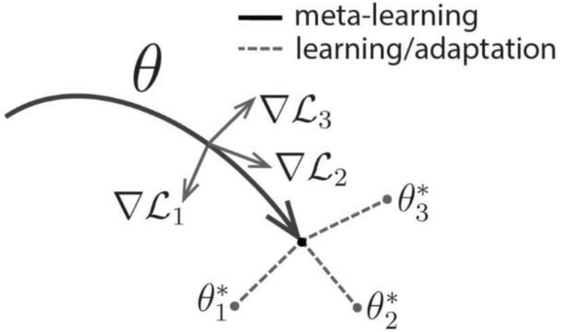
Meta Learning



θ

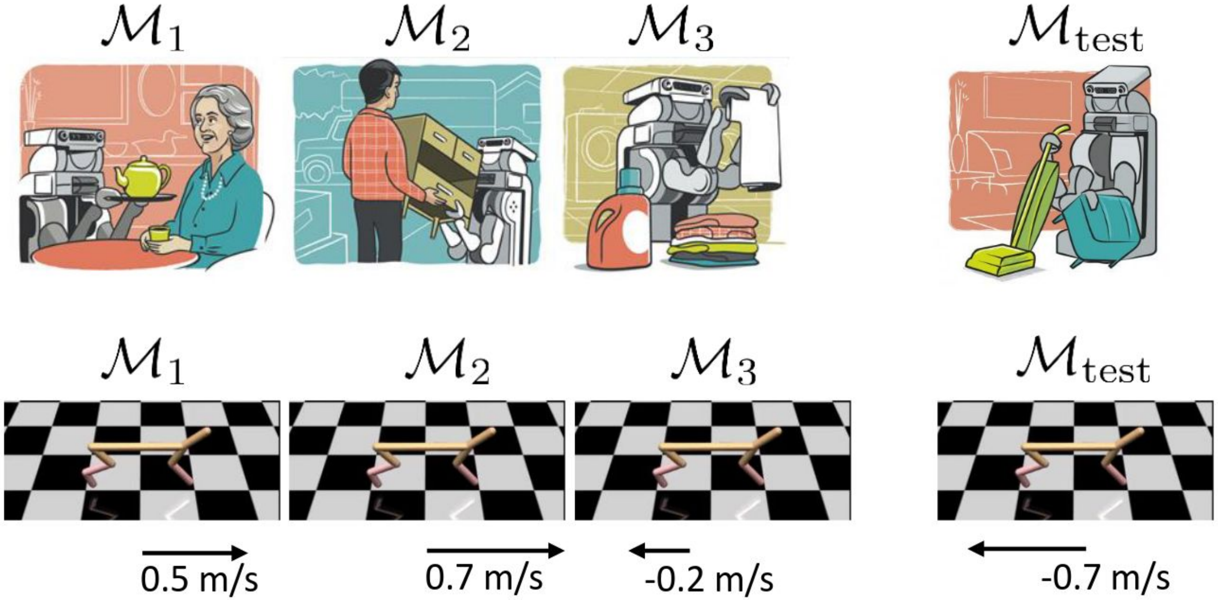
$$\cancel{\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}^{\mathrm{tr}})}$$

$$\theta \leftarrow \theta - \beta \sum_i \nabla_\theta \mathcal{L}(\theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}_i^{\mathrm{tr}}), \mathcal{D}_i^{\mathrm{ts}})$$



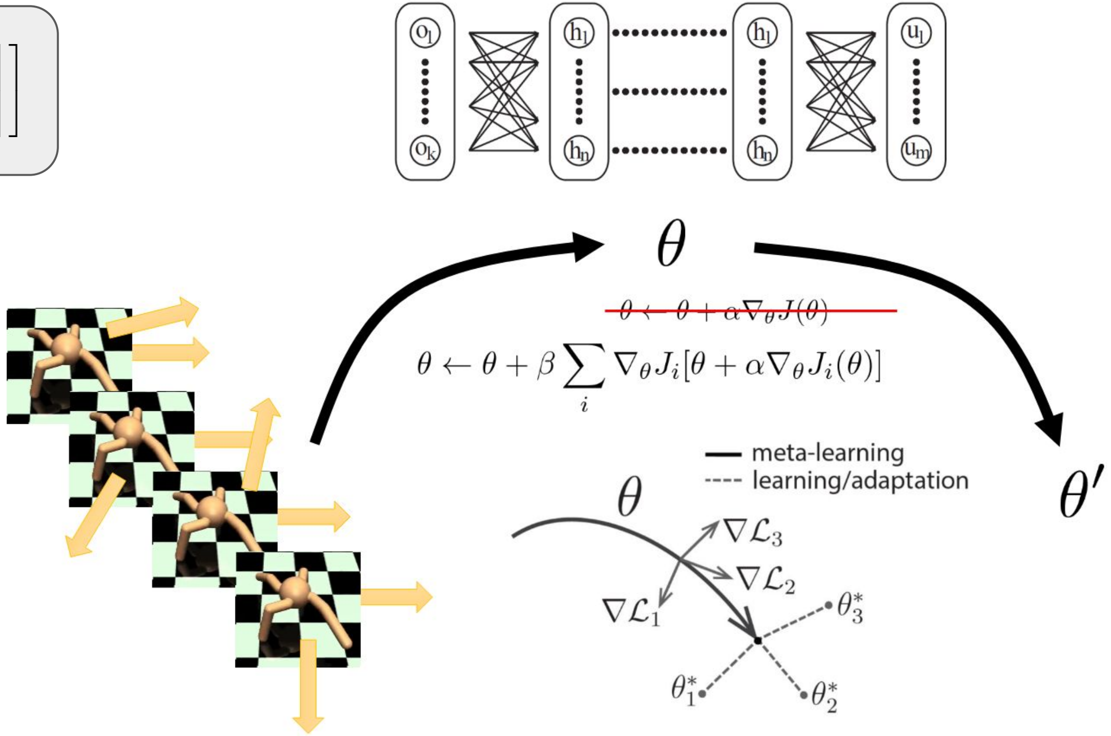
θ'

Meta Reinforcement Learning

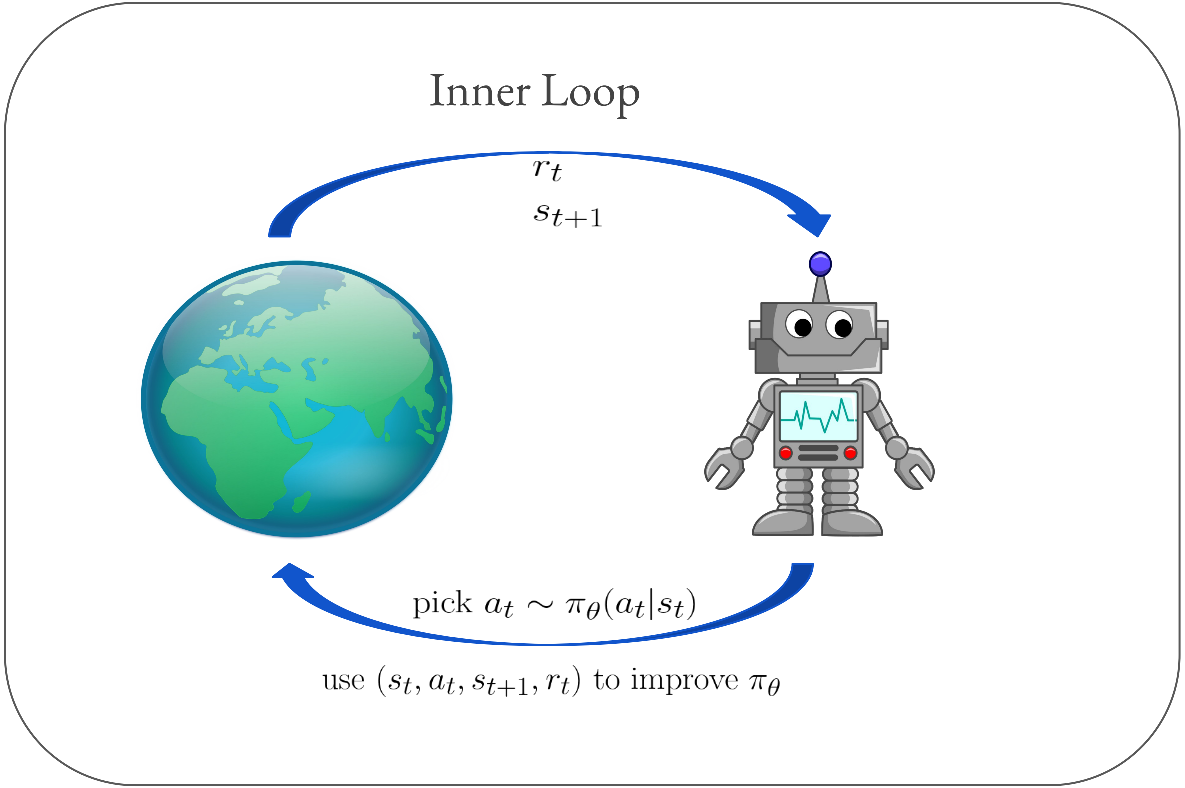


Meta Reinforcement Learning

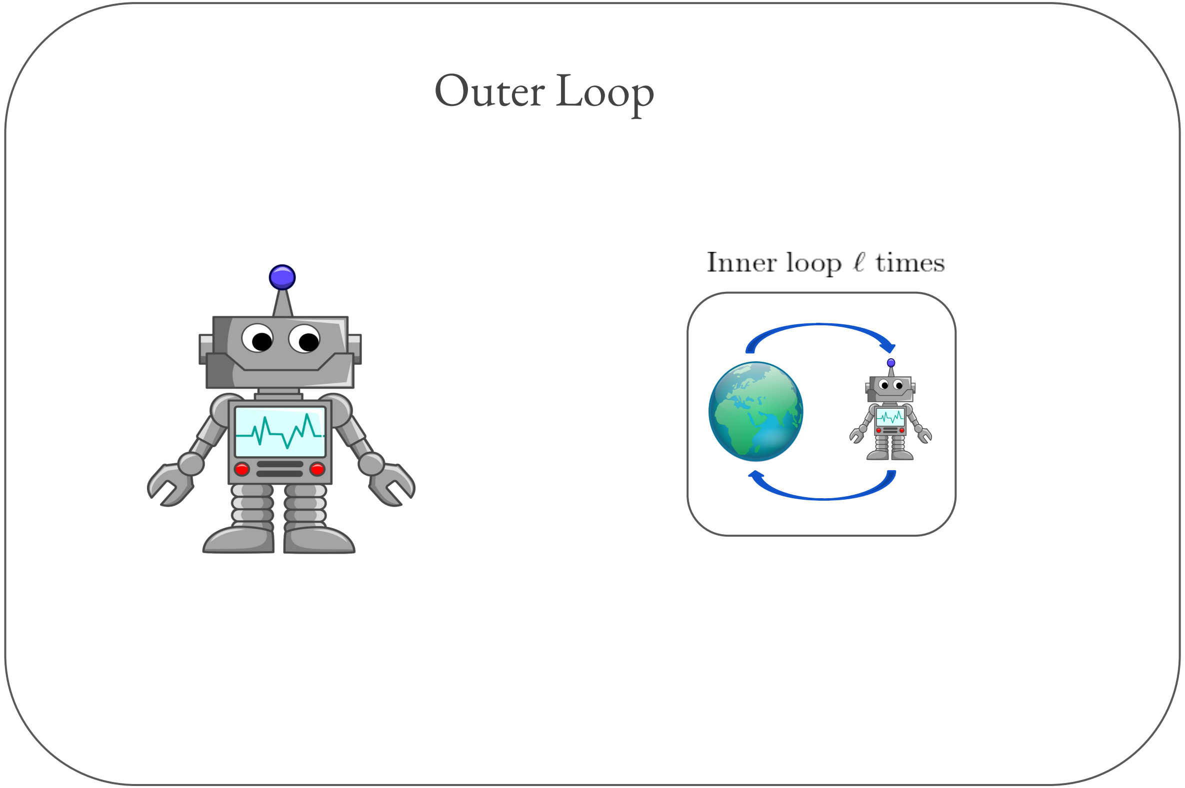
$$\mathcal{J}(\theta) = \mathbb{E}_{M^i \sim p(\mathcal{M})}\left[\mathbb{E}_{\mathcal{D}}\left[\sum_{\tau \in \mathcal{D}_{K:H}} G(\tau) \middle| f_\theta, \mathcal{M}^i\right]\right]$$



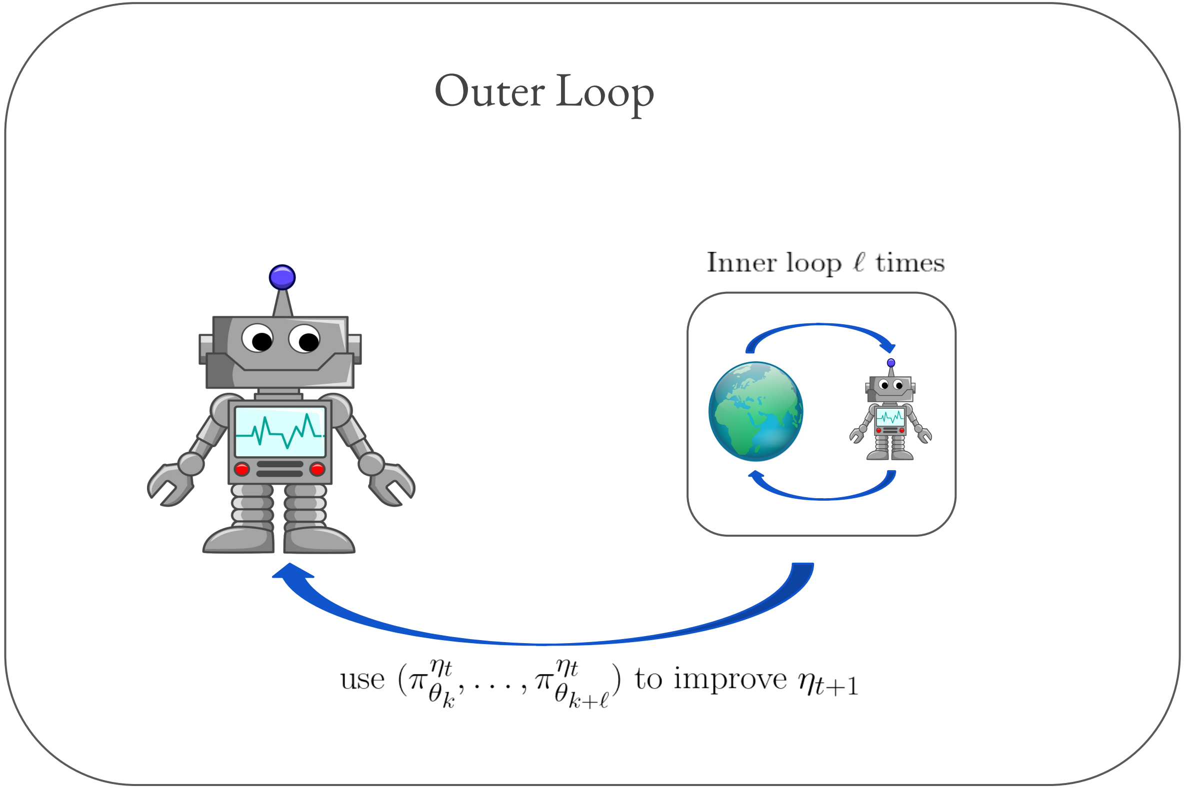
Online Meta Reinforcement Learning



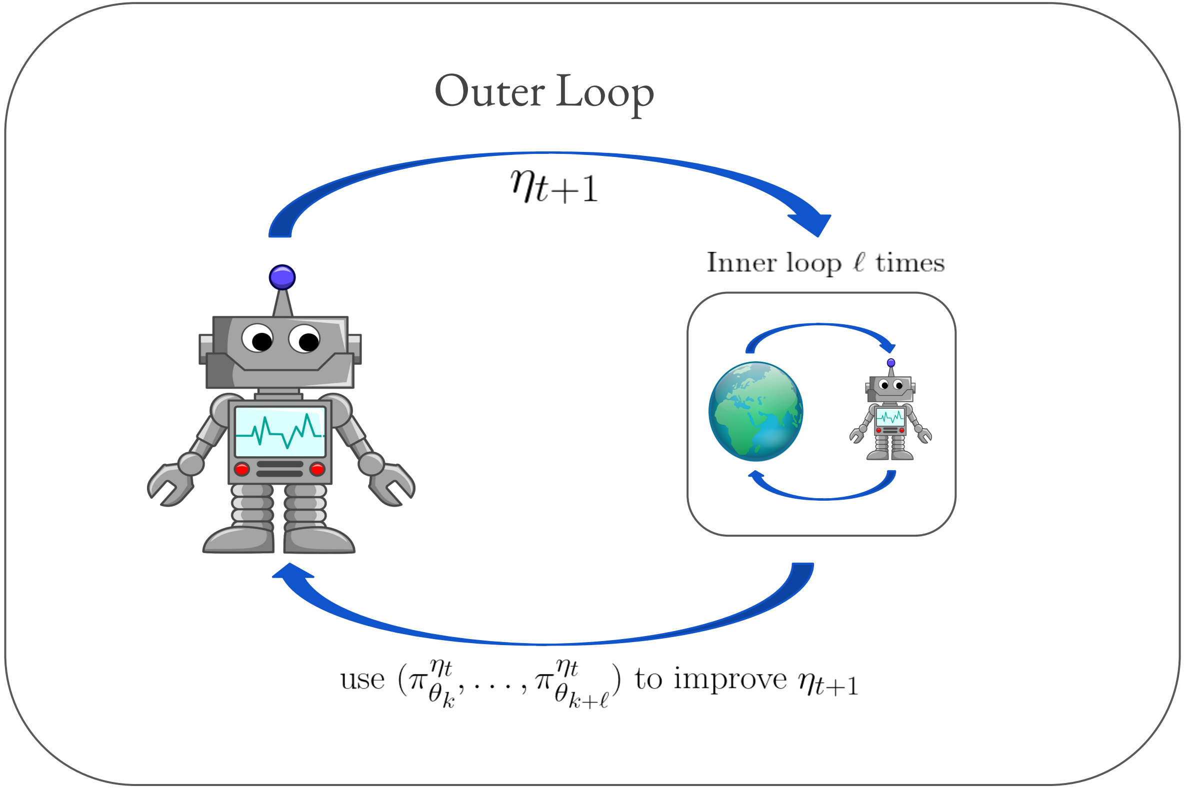
Online Meta Reinforcement Learning



Online Meta Reinforcement Learning

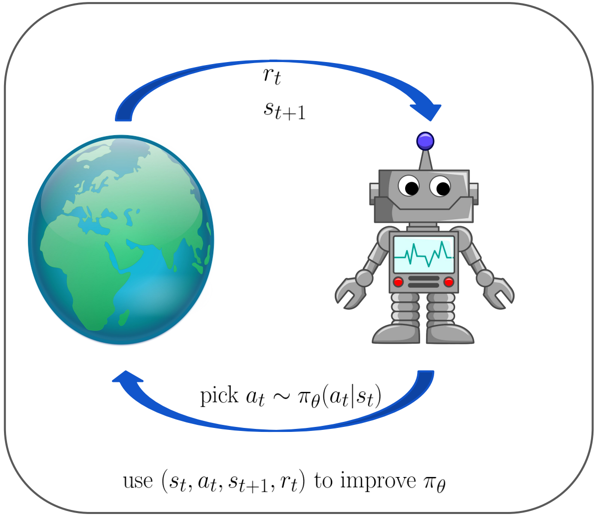


Online Meta Reinforcement Learning

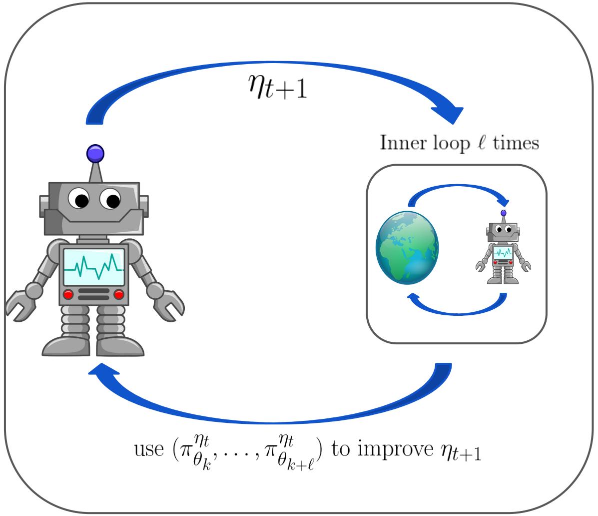


Online Meta Reinforcement Learning

Inner Loop



Outer Loop



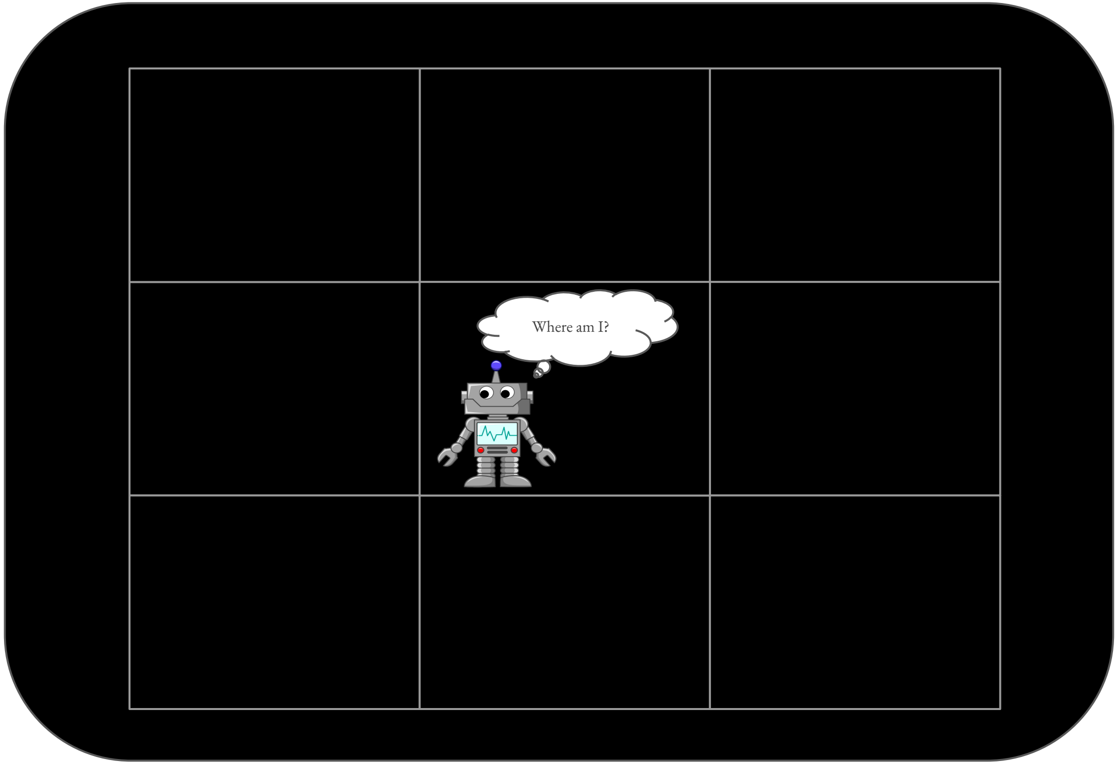
Online Meta Reinforcement Learning

Goal: meta-learn generally useful knowledge

- Learned online as we adapt to the current task.
- Transfers across similar tasks.

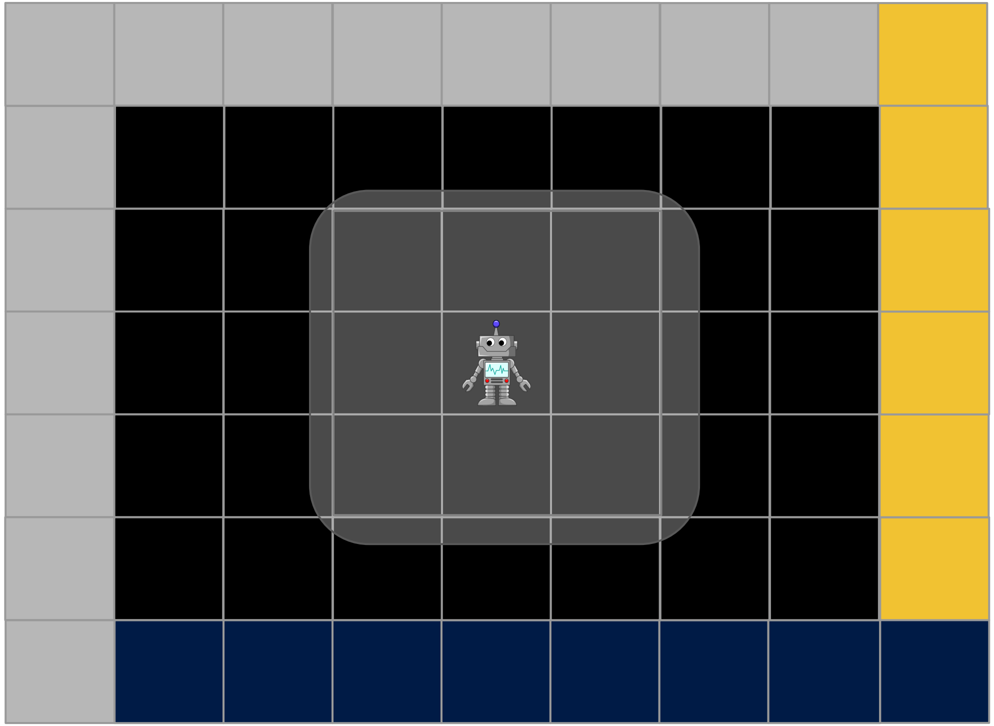
Discovery of Useful Questions (Veeriah et al.)

A Structured State Representation



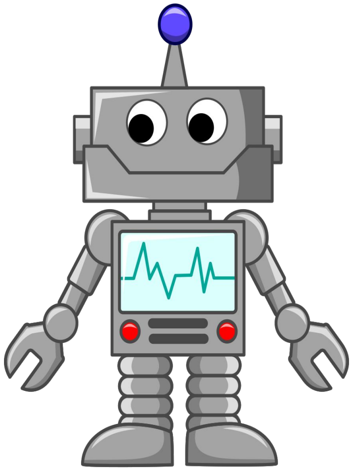
Veeriah Et al. (2019). Discovery of Useful Questions as Auxiliary Tasks

A Structured State Representation



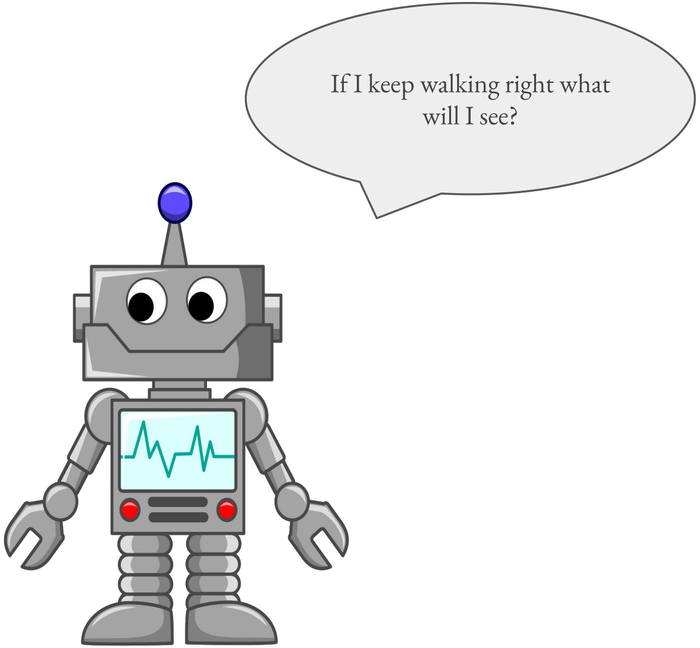
Veeriah Et al. (2019). Discovery of Useful Questions as Auxiliary Tasks

Useful Questions



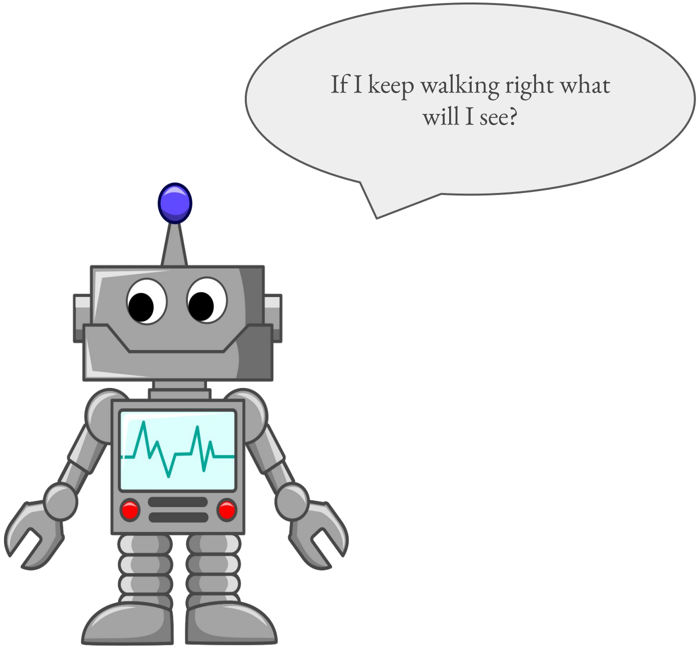
Veeriah Et al. (2019). Discovery of Useful Questions as Auxiliary Tasks

Useful Questions



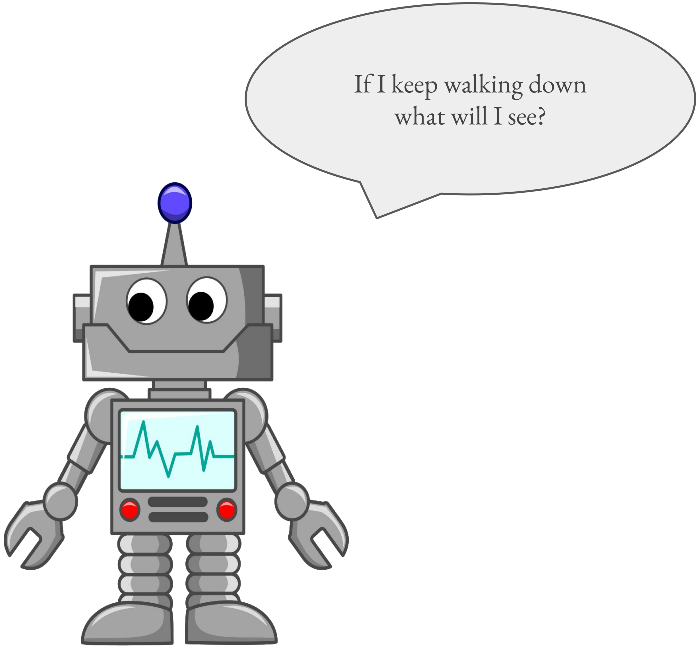
Veeriah Et al. (2019). Discovery of Useful Questions as Auxiliary Tasks

Useful Questions



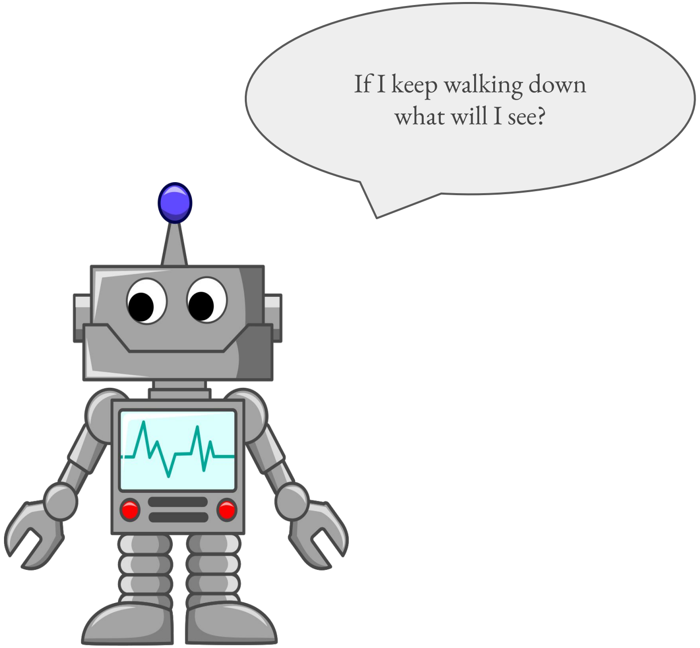
Veeriah Et al. (2019). Discovery of Useful Questions as Auxiliary Tasks

Useful Questions



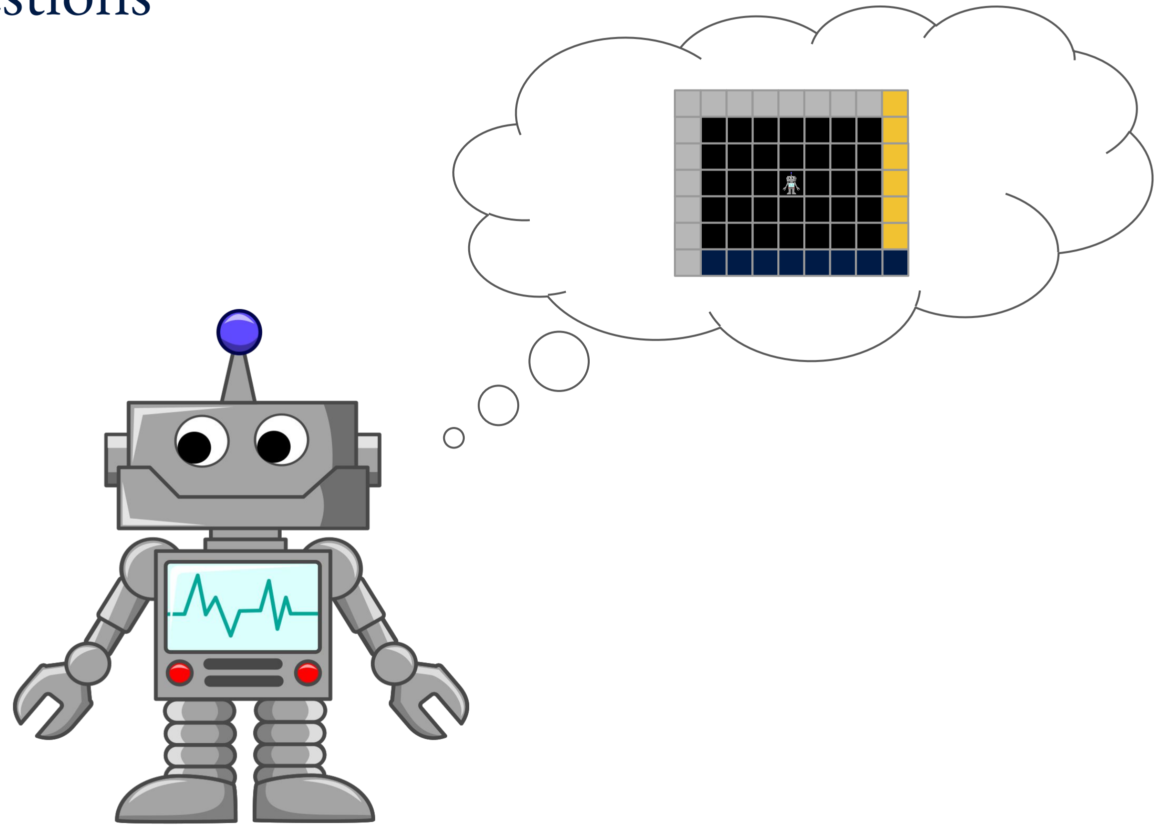
Veeriah Et al. (2019). Discovery of Useful Questions as Auxiliary Tasks

Useful Questions



Veeriah Et al. (2019). Discovery of Useful Questions as Auxiliary Tasks

Useful Questions



Veeriah Et al. (2019). Discovery of Useful Questions as Auxiliary Tasks

How do we represent predictions?

- General Value Functions (Sutton et al.)

$$V^{\pi,c,\gamma}(s) = \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t c_{t+1} | \pi, c, s_0 = s\Big]$$

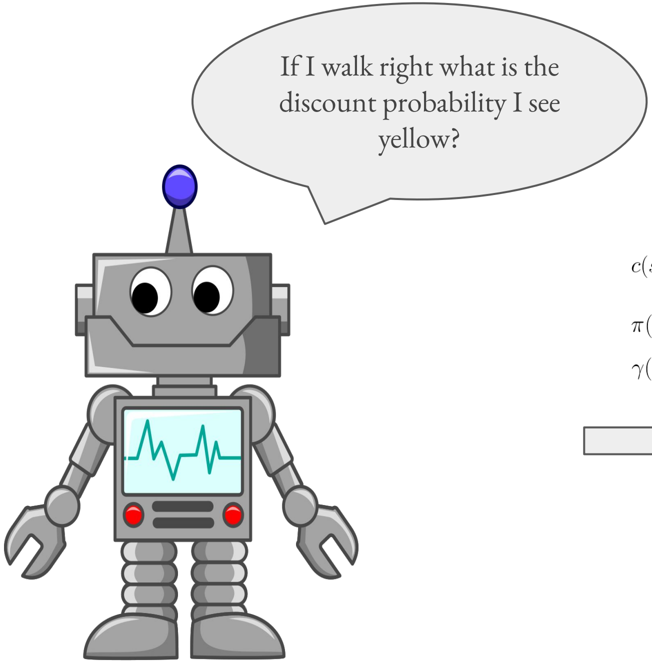
cumulant $c : \mathcal{S} \to \mathbb{R}$.

state-dependent discount factor $\gamma : \mathcal{S} \to [0, 1]$

- Can learn using traditional TD methods i.e. GAE, V-trace, etc.

Veeriah Et al. (2019). Discovery of Useful Questions as Auxiliary Tasks

How do we represent predictions?



$$c(s) = \begin{cases} 1, & \text{if yellow} \in s \\ 0, & \text{otherwise.} \end{cases}$$

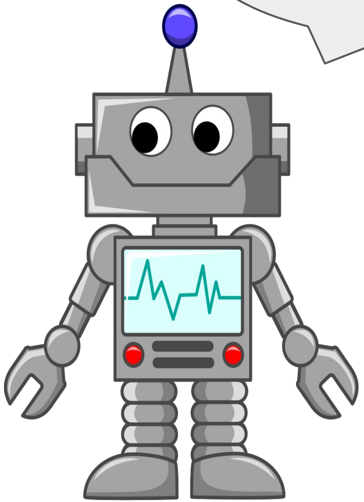
$$\pi(s) = \text{move right}$$

$$\gamma(s) = 0.9 \text{ (for example)}$$

Veeriah Et al. (2019). Discovery of Useful Questions as Auxiliary Tasks

How do we represent predictions?

If I walk right what is the discount probability I see yellow?



$$c(s) = \begin{cases} 1, & \text{if yellow} \in s \\ 0, & \text{otherwise.} \end{cases}$$

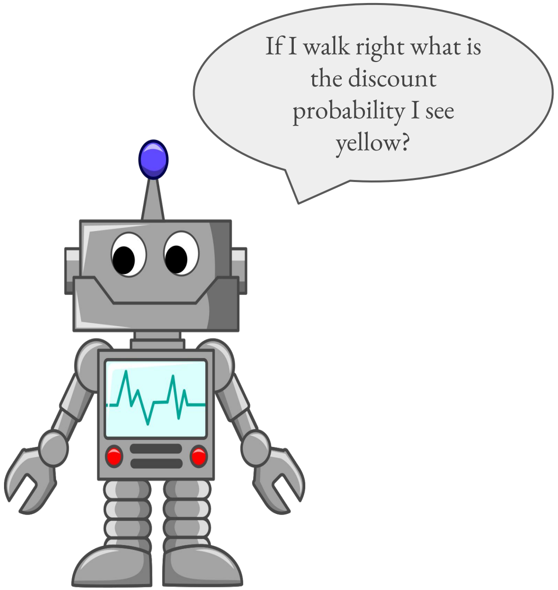
$$\pi(s) = \text{move right}$$

$$\gamma(s) = 0.9 \text{ (for example)}$$

$$V^{\pi,\gamma,c}(s)$$

Veeriah Et al. (2019). Discovery of Useful Questions as Auxiliary Tasks

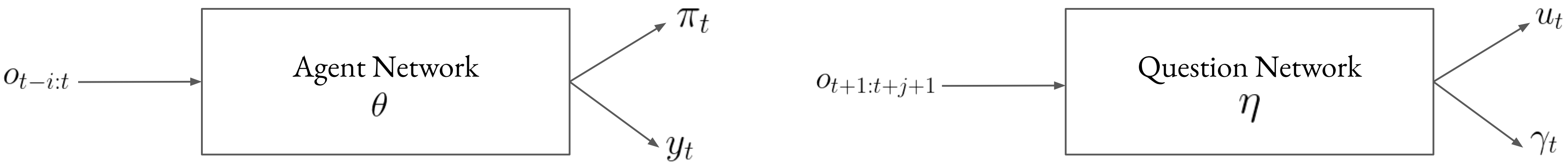
How do we represent predictions?



	0.53	0.59	0.66	0.73	0.81	0.9	1	
	0.53	0.59	0.66	0.73	0.81	0.9	1	
	0.53	0.59	0.66	0.73	0.81	0.9	1	
	0.53	0.59	0.66	0.73	0.81	0.9	1	
	0.53	0.59	0.66	0.73	0.81	0.9	1	

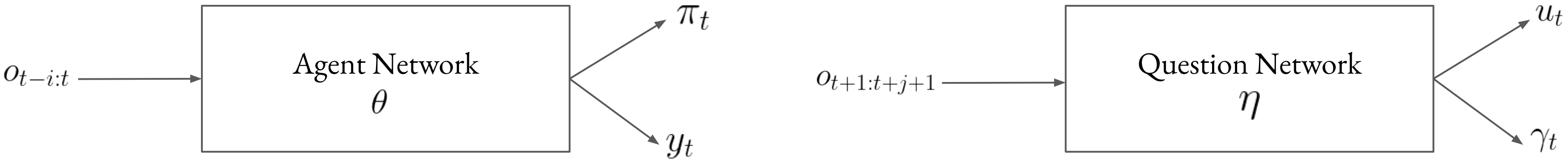
Veeriah Et al. (2019). Discovery of Useful Questions as Auxiliary Tasks

How do we learn useful questions?



Veeriah Et al. (2019). Discovery of Useful Questions as Auxiliary Tasks

How do we learn useful questions?



$$-\alpha \nabla_{\theta^{y_i}} \mathcal{L}^{ans} = \alpha\big(G_t^{y_i} - y_i(x_t)\big)\nabla_{\theta_i^y} y_i(x_t)$$

$$\theta_{t,k} \leftarrow \theta_{t,k-1} - \alpha'\nabla_{\theta_{t,k-1}}\mathcal{L}^{RL}(\theta_{t,k-1}) - \alpha'\nabla_{\theta_{t,k-1}}\mathcal{L}^{ans}(\theta_{t,k-1}).$$

$$\eta_{t+1} \leftarrow \eta_t - \alpha\nabla_\eta \sum_{k=1}^{L}\mathcal{L}^{RL}(\theta_{t,k}).$$

Veeriah Et al. (2019). Discovery of Useful Questions as Auxiliary Tasks

How do we learn useful questions?

$\theta_{t,k}$

Veeriah Et al. (2019). Discovery of Useful Questions as Auxiliary Tasks

How do we learn useful questions?

$$\theta_{t,k} \longrightarrow \theta_{t,k+1}$$

$$\theta_{t,k} \leftarrow \theta_{t,k-1} - \alpha' \nabla_{\theta_{t,k-1}} \mathcal{L}^{RL}(\theta_{t,k-1}) - \alpha' \nabla_{\theta_{t,k-1}} \mathcal{L}^{ans}(\theta_{t,k-1}).$$

Inner Update: $\longrightarrow$

Veeriah Et al. (2019). Discovery of Useful Questions as Auxiliary Tasks

How do we learn useful questions?

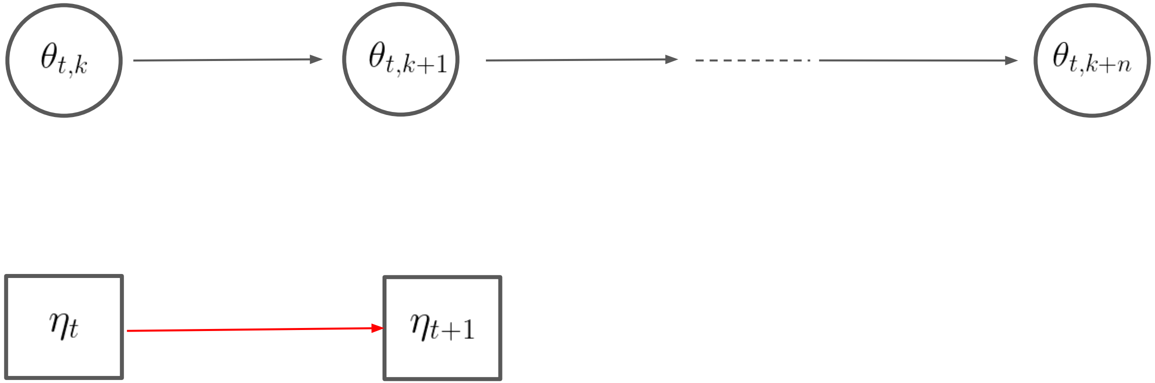
$$\theta_{t,k} \longrightarrow \theta_{t,k+1} \longrightarrow \cdots \longrightarrow \theta_{t,k+n}$$

$$\boxed{\theta_{t,k} \leftarrow \theta_{t,k-1} - \alpha' \nabla_{\theta_{t,k-1}} \mathcal{L}^{RL}(\theta_{t,k-1}) - \alpha' \nabla_{\theta_{t,k-1}} \mathcal{L}^{ans}(\theta_{t,k-1}).}$$

Inner Update: $\longrightarrow$

Veeriah Et al. (2019). Discovery of Useful Questions as Auxiliary Tasks

How do we learn useful questions?



$$\theta_{t,k} \leftarrow \theta_{t,k-1} - \alpha' \nabla_{\theta_{t,k-1}} \mathcal{L}^{RL}(\theta_{t,k-1}) - \alpha' \nabla_{\theta_{t,k-1}} \mathcal{L}^{ans}(\theta_{t,k-1}).$$

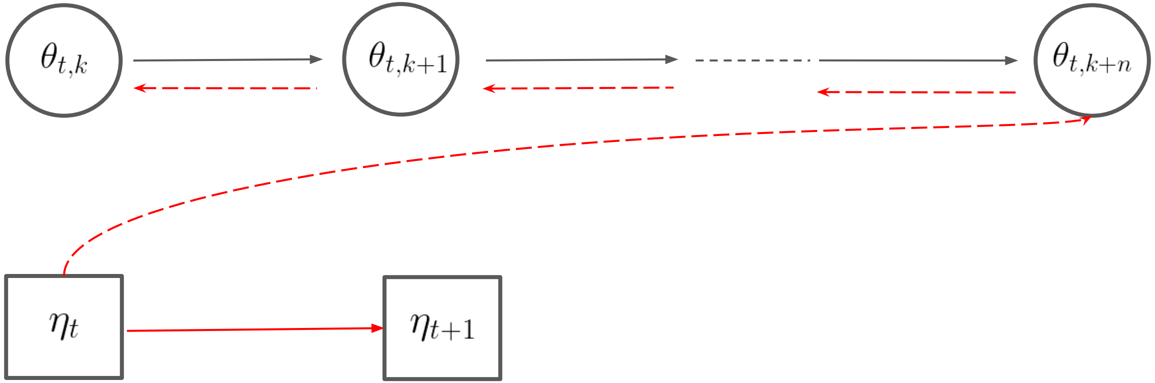
$$\eta_{t+1} \leftarrow \eta_t - \alpha \nabla_\eta \sum_{k=1}^{L} \mathcal{L}^{RL}(\theta_{t,k}).$$

Inner Update:

Outer Update:

Veeriah Et al. (2019). Discovery of Useful Questions as Auxiliary Tasks

How do we learn useful questions?



$$\theta_{t,k} \leftarrow \theta_{t,k-1} - \alpha' \nabla_{\theta_{t,k-1}} \mathcal{L}^{RL}(\theta_{t,k-1}) - \alpha' \nabla_{\theta_{t,k-1}} \mathcal{L}^{ans}(\theta_{t,k-1}).$$

$$\eta_{t+1} \leftarrow \eta_t - \alpha \nabla_\eta \sum_{k=1}^{L} \mathcal{L}^{RL}(\theta_{t,k}).$$

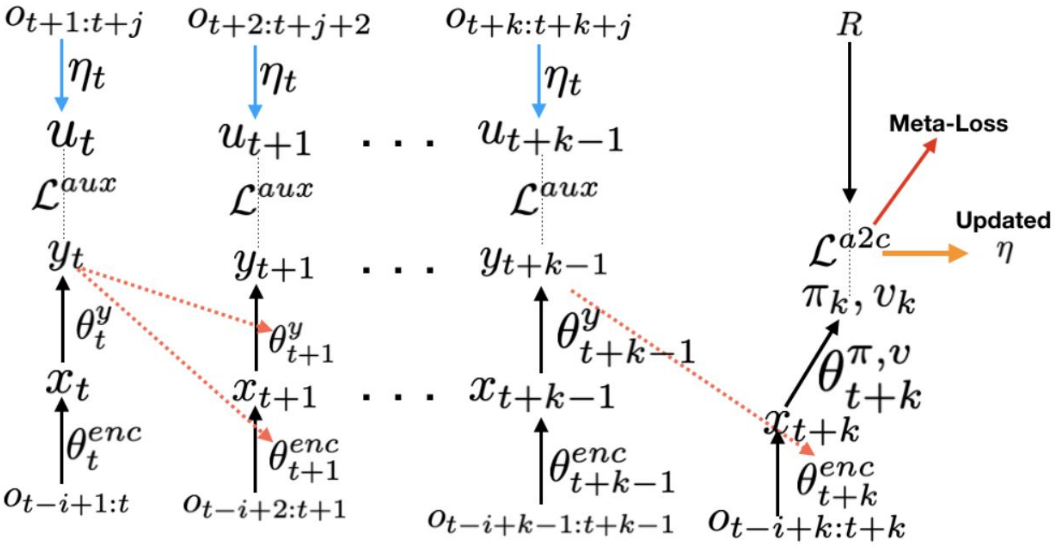
Inner Update:

Outer Update:

Gradient Flow:

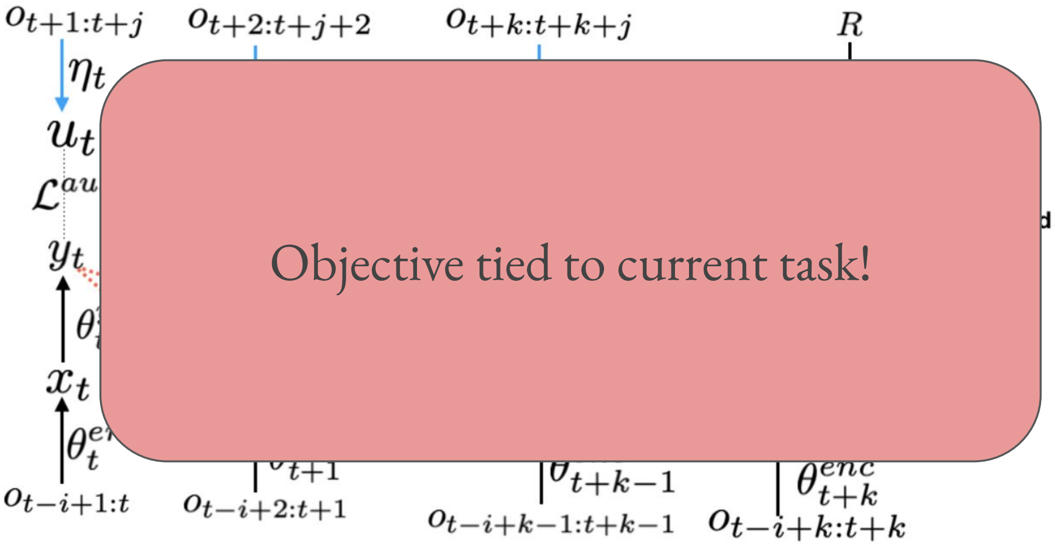
Veeriah Et al. (2019). Discovery of Useful Questions as Auxiliary Tasks

How do we learn useful questions?



Veeriah Et al. (2019). Discovery of Useful Questions as Auxiliary Tasks

How do we learn useful questions?



Objective tied to current task!

Veeriah Et al. (2019). Discovery of Useful Questions as Auxiliary Tasks

Decoupling Task Objective

- Meta objective based on learning dynamics
- **Idea**: Future policy will be better than current one

Flennerhag Et al. (2018). Bootstrapped Meta Learning

Decoupling Task Objective

- Meta objective based on learning dynamics
- **Idea**: Future policy will be better than current one

Bootstrapped **M**eta **G**radient (**BMG**):

$$\mathcal{L}_{\text{outer}} = \text{KL}(\pi_{\theta_{t+K(\eta)}} || \pi_{\theta_{t+K+L}}),$$

Flennerhag Et al. (2018). Bootstrapped Meta Learning

Non-Stationarity

Answer targets are constantly changing

- Plasticity Loss

Non-Stationarity

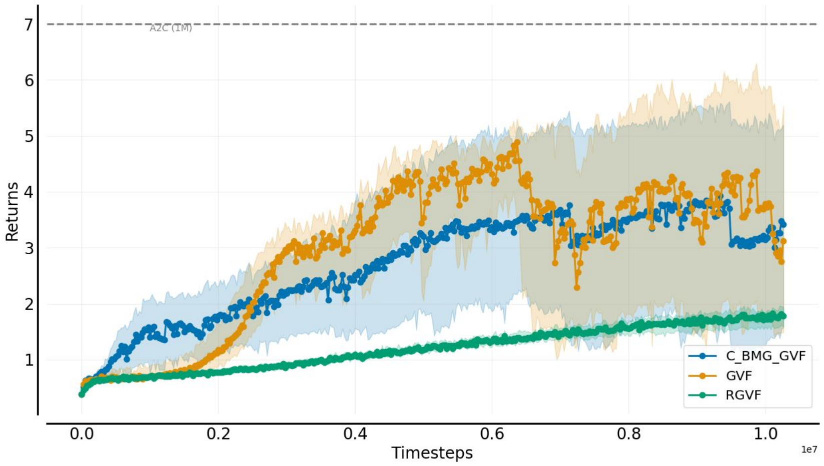
Answer targets are constantly changing

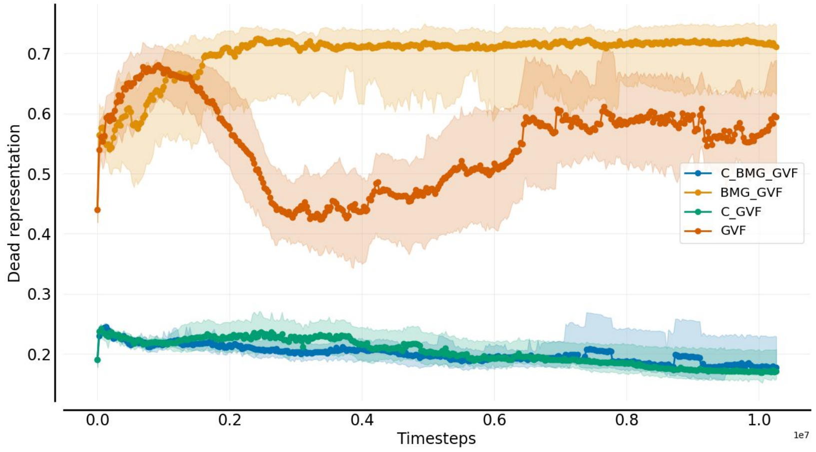
- Plasticity Loss

Concatenated **Re**ctified **L**inear **U**nit

$$\mathrm{CReLU}(x) = [\mathrm{ReLU}(x),\ \mathrm{ReLU}(-x)]$$

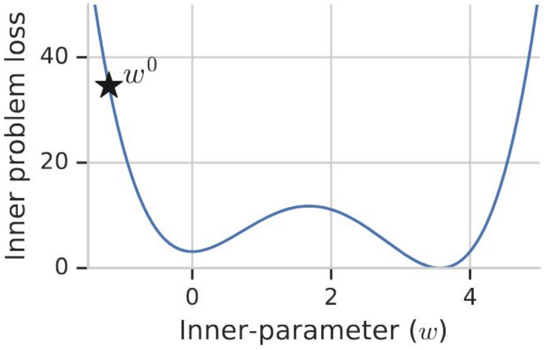
Experiments

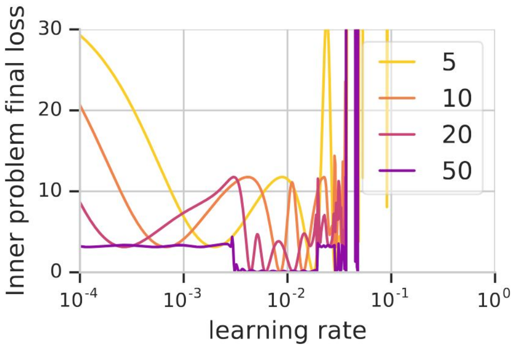


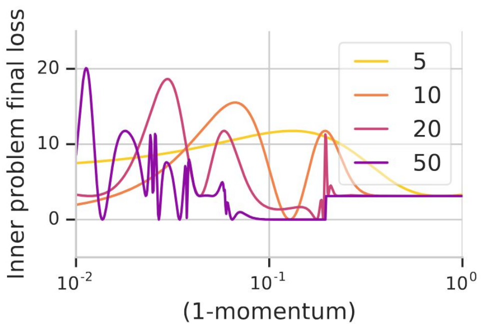


Meta RL Is HARD!

- Truncated backprop
- Evolutionary strategies







Metz et al. (2019). Understanding and correcting pathologies in the training of learned optimizers

Continual RL: Where to go from here?

What does it mean to learn continually?

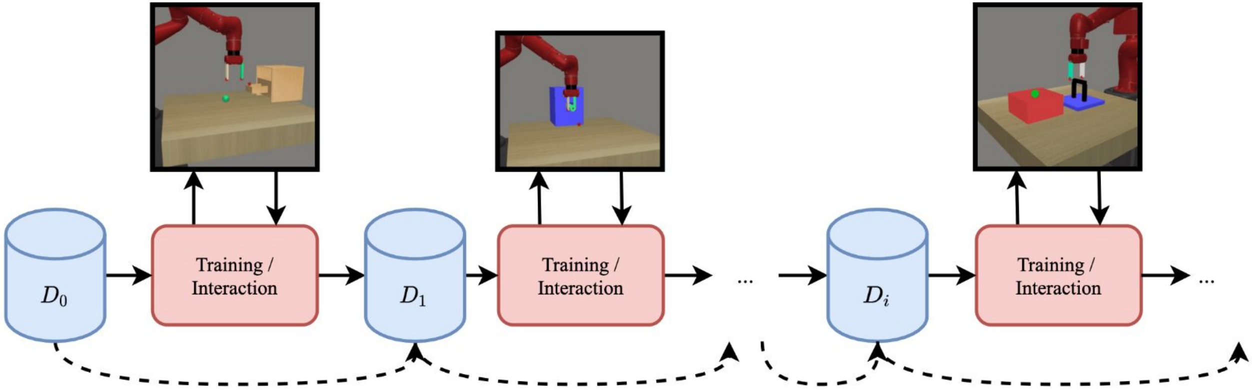
"A continual learning agent is an agent that never stops learning" - ***Abel et al.***

What does it mean to learn continually?

"A continual learning agent is an agent that never stops learning" - ***Abel et al.***

In practice:

- No known task distribution a priori.
- Tasks are encountered sequentially and may or may not be seen again.



What does it mean to learn continually?

"A continual learning agent is an agent that never stops learning" - ***Abel et al.***

In practice:

- No known task distribution a priori.
- Tasks are encountered sequentially and may or may not be seen again.

What we want:

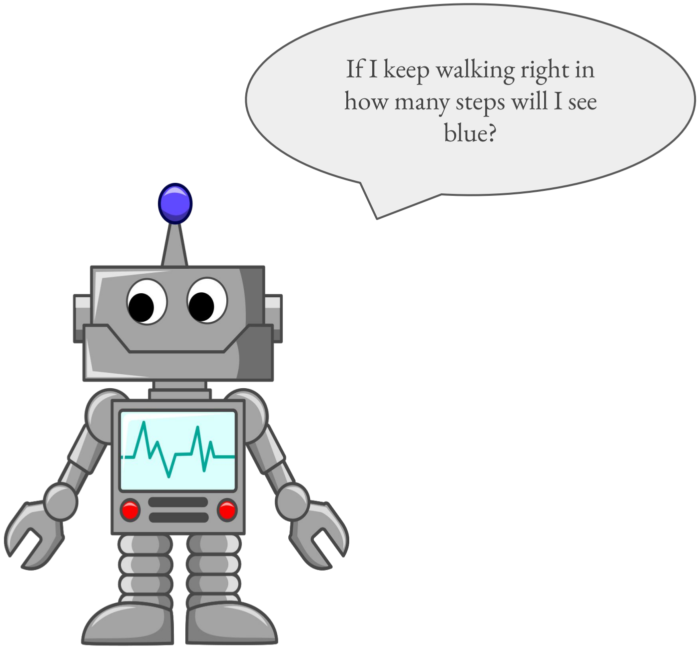
- Given a sufficient amount of samples in each task learn the optimal policy for each task -> **Plasticity.**
- Perform "better" on previously learned tasks -> **Minimizing Catastrophic Forgetting/Interference.**
- Similar "structure" to a previous task better performance -> **Forward Transfer.**

Forward Transfer: Off Policy GVF

- GVFs are learned on policy in previous method.

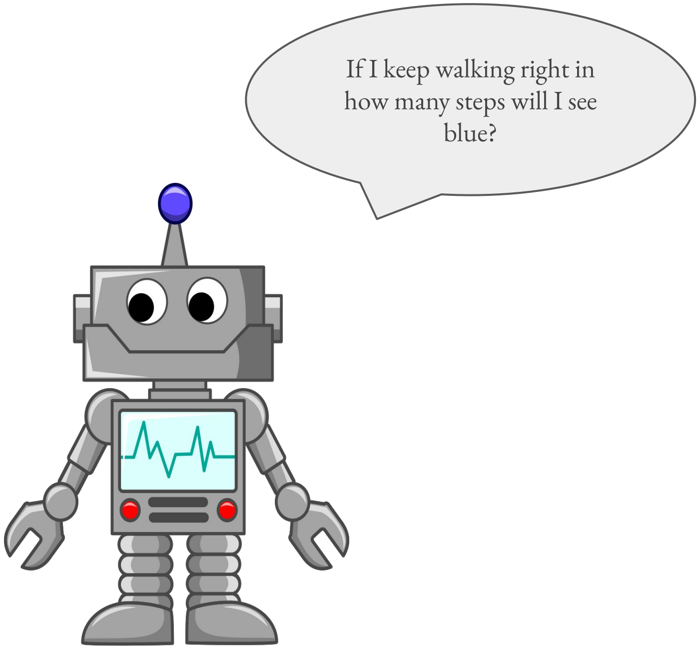
Forward Transfer: Off Policy GVF

- GVFs are learned on policy in previous method.



Forward Transfer: Off Policy GVF

- GVFs are learned on policy in previous method.



Continual Auxiliary Tasks: Successor Features

Linear MDP Assumption: Can decompose reward as $r_t = \langle \phi(s_t, a_t), w \rangle$

Barreto Et al. (2016). Successor Features for Transfer in Reinforcement Learning

Continual Auxiliary Tasks: Successor Features

Linear MDP Assumption: Can decompose reward as $r_t = \langle \phi(s_t, a_t), w \rangle$

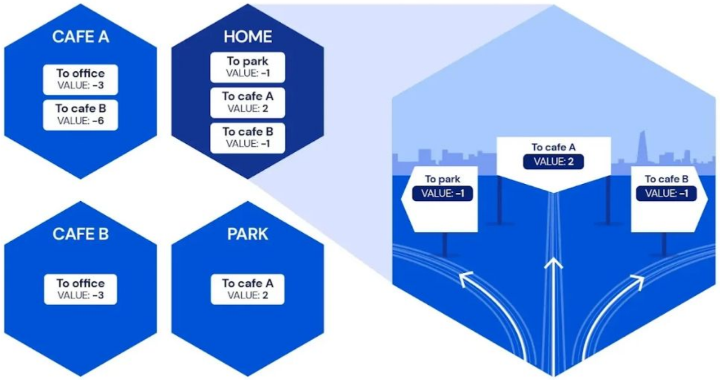
- Assumed to be given *cumulant* $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$
- $w \in \mathbb{R}^d$ encodes the task representation
- We learn a *successor feature* $\psi(s_t, a_t)$ via bootstrapping $\psi(s_t, a_t) = \phi(s_t, a_t) + \gamma \psi(s_{t+1}, a^*)$
- Then we have that $\langle \psi(s_t, a_t), w \rangle = Q(s_t, a_t)$
- Then we can quickly learn GVFs of new tasks via learning new w

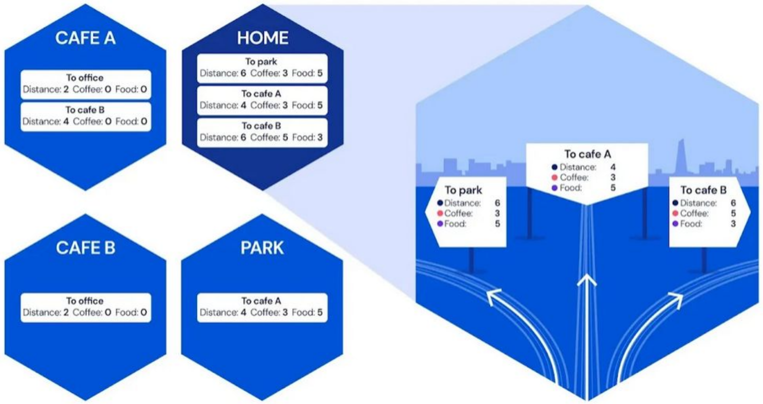
Barreto Et al. (2016). Successor Features for Transfer in Reinforcement Learning

Continual Auxiliary Tasks: Successor Features

Linear MDP Assumption: Can decompose reward as $r_t = \langle \phi(s_t, a_t), w \rangle$

- Assumed to be given *cumulant* $\phi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}^d$
- $w \in \mathbb{R}^d$ encodes the task representation
- We learn a *successor feature* $\psi(s_t, a_t)$ via bootstrapping $\psi(s_t, a_t) = \phi(s_t, a_t) + \gamma \psi(s_{t+1}, a^*)$
- Then we have that $\langle \psi(s_t, a_t), w \rangle = Q(s_t, a_t)$
- Then we can quickly learn GVFs of new tasks via learning new w





Barreto Et al. (2016). Successor Features for Transfer in Reinforcement Learning

Continual Auxiliary Tasks: Successor Features

Universal Successor Feature Approximator:

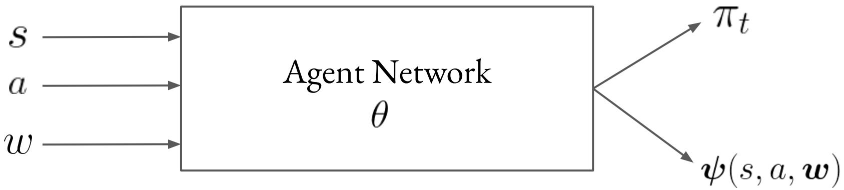
- Learn task conditioned successor feature approximator $\boldsymbol{\psi}(s, a, \boldsymbol{w})$
- Can learn optimal off policy $\boldsymbol{\psi}(s_t, a_t, \boldsymbol{w}) = \boldsymbol{\phi}(s_t) + \gamma\boldsymbol{\psi}(s_{t+1}, a^*, \boldsymbol{w})$

Borsa Et al. (2018). Universal Successor Feature Approximators

Continual Auxiliary Tasks: Successor Features

Universal Successor Feature Approximator:

- Learn task conditioned successor feature approximator $\boldsymbol{\psi}(s, a, \boldsymbol{w})$
- Can learn optimal off policy $\boldsymbol{\psi}(s_t, a_t, \boldsymbol{w}) = \boldsymbol{\phi}(s_t) + \gamma \boldsymbol{\psi}(s_{t+1}, a^*, \boldsymbol{w})$



$$\min_{w} \|r(s_t, a_t) - \langle \boldsymbol{\phi}(s_t, a_t), \boldsymbol{w} \rangle\|$$

Borsa Et al. (2018). Universal Successor Feature Approximators

Off Policy: Deadly Triad

- If target bootstrap at s_1 increases θ then bootstrap target also increases
- Causes divergence when s_2 not visited by behavior policy

Espeholt et al. (2018). IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures

Off Policy: Deadly Triad

- If target bootstrap at s_1 increases θ then bootstrap target also increases
- Causes divergence when s_2 not visited by behavior policy

V-Trace:

$$\delta_t V = \rho_t(r_t + \gamma V(x_{t+1}) - V(x_t))$$

$$\rho_t = \min\left(\bar{\rho}, \frac{\pi(a_t|x_t)}{\mu(a_t|x_t)}\right)$$

Espeholt et al. (2018). IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures

Off Policy: Deadly Triad

- If target bootstrap at s_1 increases θ then bootstrap target also increases
- Causes divergence when s_2 not visited by behavior policy

V-Trace:

$$\delta_t V = \rho_t(r_t + \gamma V(x_{t+1}) - V(x_t)) \qquad \rho_t = \min\left(\bar{\rho}, \frac{\pi(a_t|x_t)}{\mu(a_t|x_t)}\right)$$

$$v_s \stackrel{\text{def}}{=} V(x_s) + \sum_{t=s}^{s+n-1} \gamma^{t-s} \left(\prod_{i=s}^{t-1} c_i\right) \delta_t V \qquad c_i = \min\left(\bar{c}, \frac{\pi(a_i|x_i)}{\mu(a_i|x_i)}\right)$$

Espeholt et al. (2018). IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures

Catastrophic Forgetting

$$\min_{\theta_0^j} \left(\sum_{i=1}^{t} \left(\ell_i \left(\theta_0^j \right) \right) - \alpha \sum_{p,q \leq t} \left(\frac{\partial \ell_p \left(\theta_0^j \right)}{\partial \theta_0^j} \cdot \frac{\partial \ell_q \left(\theta_0^j \right)}{\partial \theta_0^j} \right) \right)$$

Matthew Riemer et al. (2019) Learning to Learn without Forgetting by Maximizing Transfer and Minimizing Interference.

Catastrophic Forgetting

$$\min_{\theta_0^j}\left(\sum_{i=1}^{t}\left(\ell_i\left(\theta_0^j\right)\right)-\alpha\sum_{p,q\leq t}\left(\frac{\partial\ell_p\left(\theta_0^j\right)}{\partial\theta_0^j}\cdot\frac{\partial\ell_q\left(\theta_0^j\right)}{\partial\theta_0^j}\right)\right)$$

Forward Transfer

$$\frac{\partial\ell_p\left(\theta_0^j\right)}{\partial\theta_0^j}\cdot\frac{\partial\ell_q\left(\theta_0^j\right)}{\partial\theta_0^j}>0$$

Catastrophic Interference

$$\frac{\partial\ell_p\left(\theta_0^j\right)}{\partial\theta_0^j}\cdot\frac{\partial\ell_q\left(\theta_0^j\right)}{\partial\theta_0^j}<0$$

Matthew Riemer et al. (2019) Learning to Learn without Forgetting by Maximizing Transfer and Minimizing Interference.

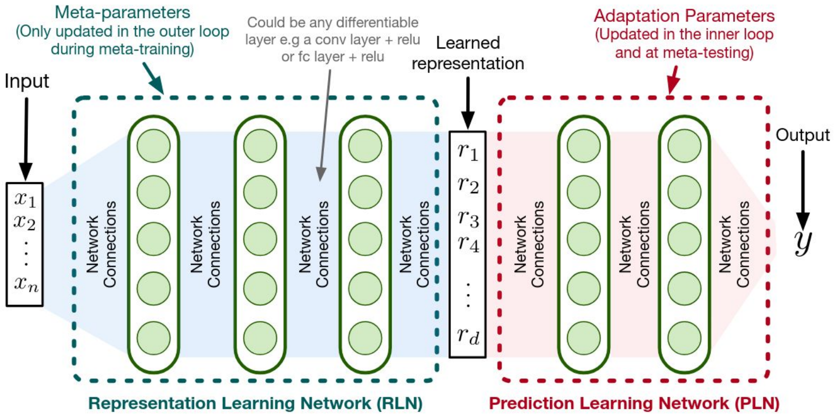
Catastrophic Forgetting

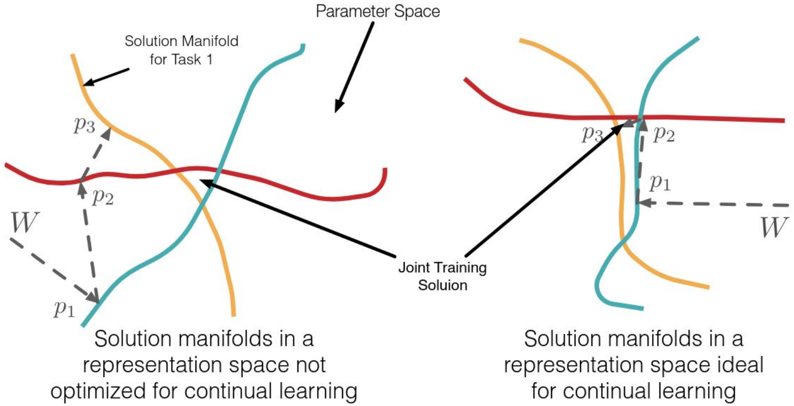
$$\min_{\theta_0^j} \left(\sum_{i=1}^{t} \left(\ell_i \left(\theta_0^j \right) \right) - \alpha \sum_{p,q \leq t} \left(\frac{\partial \ell_p \left(\theta_0^j \right)}{\partial \theta_0^j} \cdot \frac{\partial \ell_q \left(\theta_0^j \right)}{\partial \theta_0^j} \right) \right)$$

Problem: No access to ℓ_i in continual RL

Matthew Riemer et al. (2019) Learning to Learn without Forgetting by Maximizing Transfer and Minimizing Interference.

Catastrophic Forgetting: Intuition

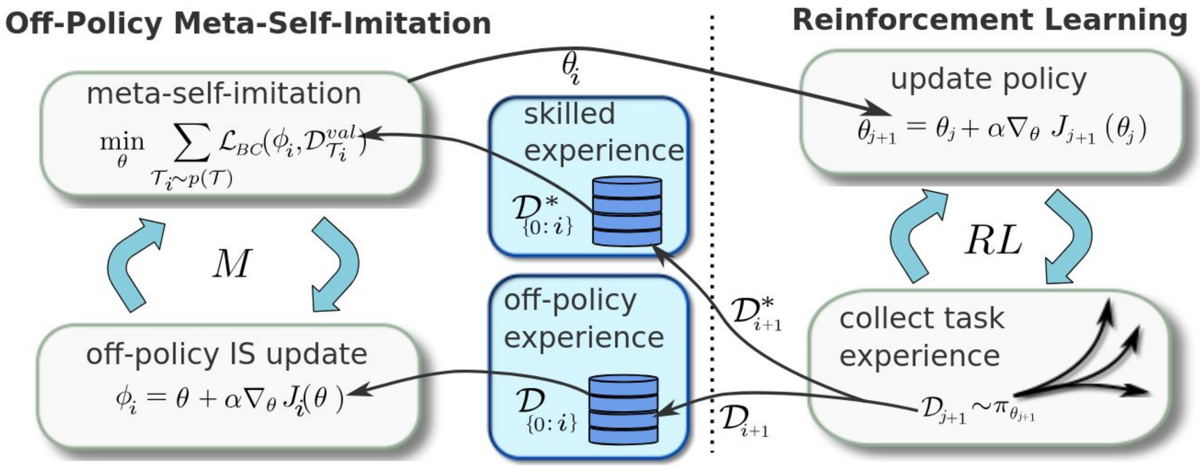




Javed, Khurram; White, Martha. (2019). Meta Learning Representations for Continual Learning

Catastrophic Forgetting: An Idea

Use representative task trajectory.



Berseth et al. (2019) Continual Meta Policy Search.

Rough Plans for Future Work

- Investigate plasticity of meta-gradients / viability for continual learning.
- Improving stability of online meta RL.
- Mitigating interference / improving transfer in continual reinforcement learning.
- How to co-learn tasks and successor features for structured representation in CRL.

If any of this sounds interesting reach out!