DAC: The Double Actor-Critic Architecture for Learning Options

NeurIPS 2019
Shangtong Zhang, Shimon Whiteson

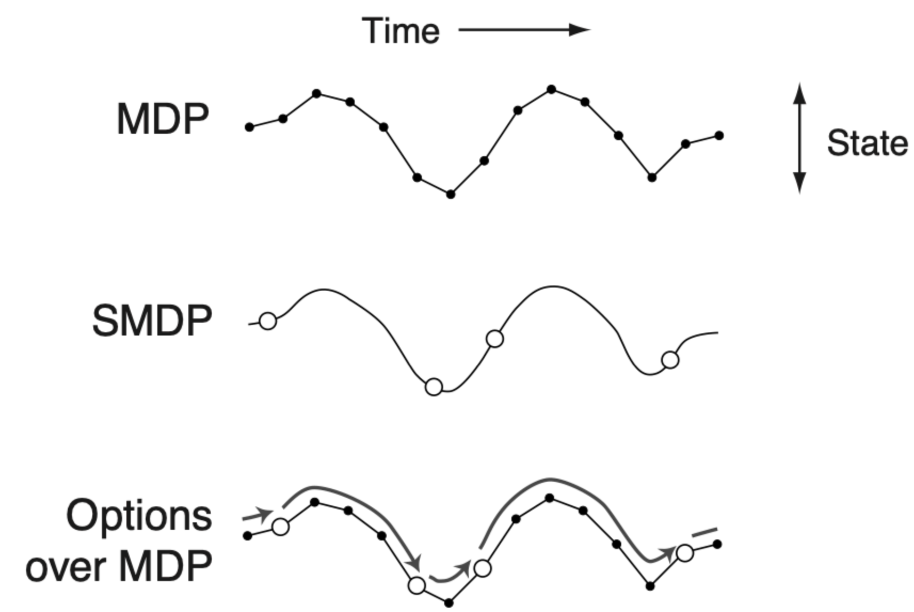
Presenter: Ehsan Mehralian
March 17, 2020

Outline

- Problem statement
- Option Critic
- Double Actor Critic

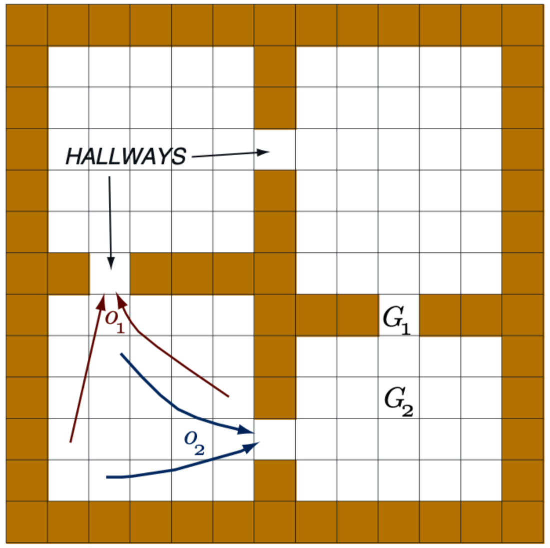
Problem statement

- Temporal abstraction is a key component in RL:
    - Better exploration
    - Faster learning
    - Better generalization
    - Transfer learning



- MDP + Temporal Abstract actions = SMDP
- SMDP algorithms are data inefficient —> The option framework (Sutton et al., 1999)
- Rises two problems:
    - Learning options
    - Learning a master policy

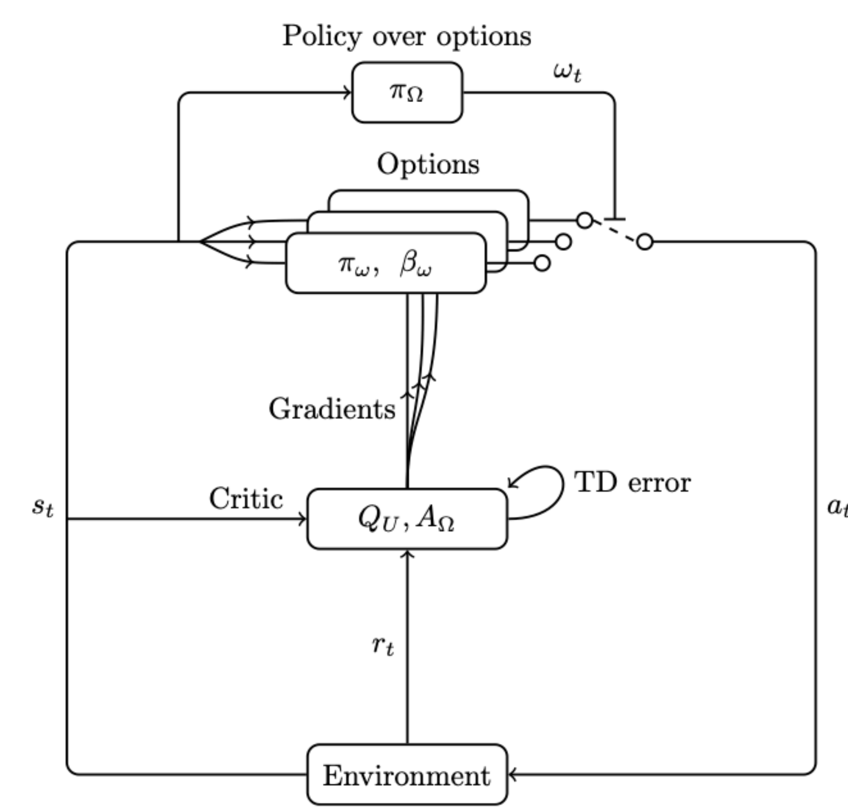
Previous works



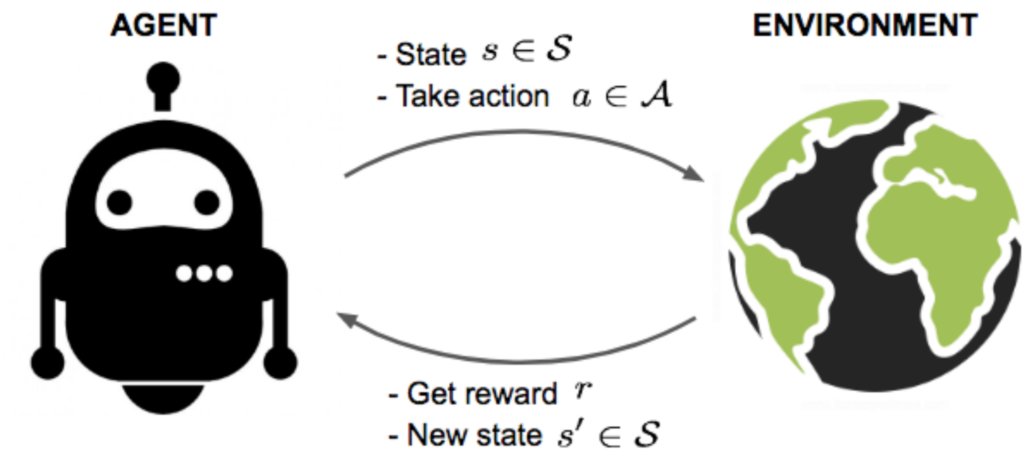
- Based on finding subgoals:
    - Difficult to scale up
    - Can be as expensive as the entire task
- Using value-based methods:
    - Can't cope with large action spaces
    - Policy based methods have better convergence properties with function approximation

The Option Critic framework (PL Bacon et.al, 2017)

- Blurs the line between *discovering* options and *learning* options
- The first scalable end-to-end approach
    - No slow down within a single task
    - Faster convergence in transfer learning



Background



- MDP

$$M \equiv \{S, A, R(s,a), P(s'|s,a), P_0(s), \gamma\}$$

- Goal:

$$\pi^* = arg\max_{\theta} \rho(\pi_\theta) = arg\max_{\theta} \mathbb{E}_{\tau \sim p_\theta(\tau)}[\sum_{t=1}^{\infty} \gamma^{t-1} r_t | s_0, \pi_\theta]$$

- Policy Gradient

$$\frac{\partial \rho}{\partial \theta} = \sum_s d^\pi(s) \sum_a \frac{\partial \pi(s,a)}{\partial \theta} Q^\pi(s,a)$$

$$d^\pi(s) = \sum_{t=0}^{\infty} \gamma^t Pr(s_t = s | s_0, \pi)$$

The Options Framework

- A Markovian Option $\omega \in \Omega$ is a triple: $(I_\omega, \pi_\omega, \beta_\omega)$
    - I_ω Initiation set $I_\omega \subset S$
    - π_ω Intra-option policy
    - β_ω Termination function $\beta_\omega : S \to [0, 1]$

- Let $\pi_{\omega,\theta}$ denote the intra-option policy of option ω parametrized by θ and $\beta_{\omega,\vartheta}$, the termination function of ω parameterized by ϑ

The Option Critic framework

- Discounted return

$$\rho(\Omega, \theta, \vartheta, s_0, w_0) = \mathbb{E}_{\Omega,\theta,\omega}[\sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_0, \omega_0]$$

The Option Critic framework

- Discounted return

$$\rho(\Omega, \theta, \vartheta, s_0, w_0) = \mathbb{E}_{\Omega,\theta,\omega}[\sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_0, \omega_0]$$

- Option-value function

$$Q_\Omega(s, w) = \sum_a \pi_{\omega,\theta}(a|s) Q_U(s, \omega, a)$$

The Option Critic framework

- Discounted return

$$\rho(\Omega, \theta, \vartheta, s_0, w_0) = \mathbb{E}_{\Omega,\theta,\omega}[\sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_0, \omega_0]$$

- Option-value function

$$Q_\Omega(s, w) = \sum_a \pi_{\omega,\theta}(a|s) Q_U(s, \omega, a)$$

- Value of executing an action in the context of a state-option pair

$$Q_U(s, \omega, a) = r(s, a) + \gamma \sum_{s'} P(s'|s, a) U(\omega, s')$$

The Option Critic framework

- Discounted return

$$\rho(\Omega, \theta, \vartheta, s_0, w_0) = \mathbb{E}_{\Omega,\theta,\omega}[\sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_0, \omega_0]$$

- Option-value function

$$Q_\Omega(s, w) = \sum_a \pi_{\omega,\theta}(a|s) Q_U(s, \omega, a)$$

- Value of executing an action in the context of a state-option pair

$$Q_U(s, \omega, a) = r(s, a) + \gamma \sum_{s'} P(s'|s, a) U(\omega, s')$$

- Option-value function upon arrival

$$U(\omega, s') = (1 - \beta_{\omega,\vartheta}(s')) Q_\Omega(s', w) + \beta_{\omega,\vartheta}(s') V_\Omega(s')$$

The Option Critic framework (cont.)

- (s, ω) pairs lead to an augmented state space

- One step probability transition

$$P(s_{t+1}, \omega_{t+1} | s_t, \omega_t) = \sum_a \pi_{\omega_t,\theta}(a|s_t) P(s_{t+1}|s_t, a_t)($$
$$(1 - \beta_{\omega,\vartheta}(s_{t+1}))1_{w_t = w_{t+1}} + \beta_{\omega,\vartheta}(s_{t+1})\pi_\Omega(\omega_{t+1}|s_{t+1}))$$

The option Critic framework (cont.)

Algorithm 1: Option-critic with tabular intra-option Q-learning

$s \leftarrow s_0$
Choose ω according to an ϵ-soft policy over options $\pi_\Omega(s)$
repeat
  Choose a according to $\pi_{\omega,\theta}(a \mid s)$
  Take action a in s, observe s', r

  1. Options evaluation:
  $\delta \leftarrow r - Q_U(s, \omega, a)$
  if s' *is non-terminal* **then**
    $\delta \leftarrow \delta + \gamma(1 - \beta_{\omega,\vartheta}(s'))Q_\Omega(s', \omega) + \gamma\beta_{\omega,\vartheta}(s') \max_{\bar{\omega}} Q_\Omega(s', \bar{\omega})$
  end
  $Q_U(s, \omega, a) \leftarrow Q_U(s, \omega, a) + \alpha\delta$

  2. Options improvement:
  $\theta \leftarrow \theta + \alpha_\theta \frac{\partial \log \pi_{\omega,\theta}(a \mid s)}{\partial \theta} Q_U(s, \omega, a)$
  $\vartheta \leftarrow \vartheta - \alpha_\vartheta \frac{\partial \beta_{\omega,\vartheta}(s')}{\partial \vartheta} (Q_\Omega(s', \omega) - V_\Omega(s'))$

  if $\beta_{\omega,\vartheta}$ *terminates in* s' **then**
  choose new ω according to ϵ-soft($\pi_\Omega(s')$)
  $s \leftarrow s'$
until s' *is terminal*

- Theorem 1 (Intra-Option Policy Gradient Theorem)

$$\frac{\partial \rho}{\partial \theta} = \sum_{s,\omega} \mu_\Omega(s, \omega | s_0, \omega_0) \sum_a \frac{\partial \pi_{\omega,\theta}(a|s)}{\partial \theta} Q_U(s, w, a)$$

- Theorem 2 (Termination Gradient Theorem)

$$\frac{\partial \rho}{\partial \vartheta} = -\sum_{s',\omega} \mu_\Omega(s', \omega | s_1, \omega_0) \frac{\partial \beta_{\omega,\vartheta}(s')}{\partial \vartheta} A_\Omega(s', w)$$

$$\mu_\Omega(s, \omega | s_0, \omega_0) = \sum_{t=0}^{\infty} \gamma^t P(s_t = s, \omega_t = \omega | s_0, \omega_0)$$

Results

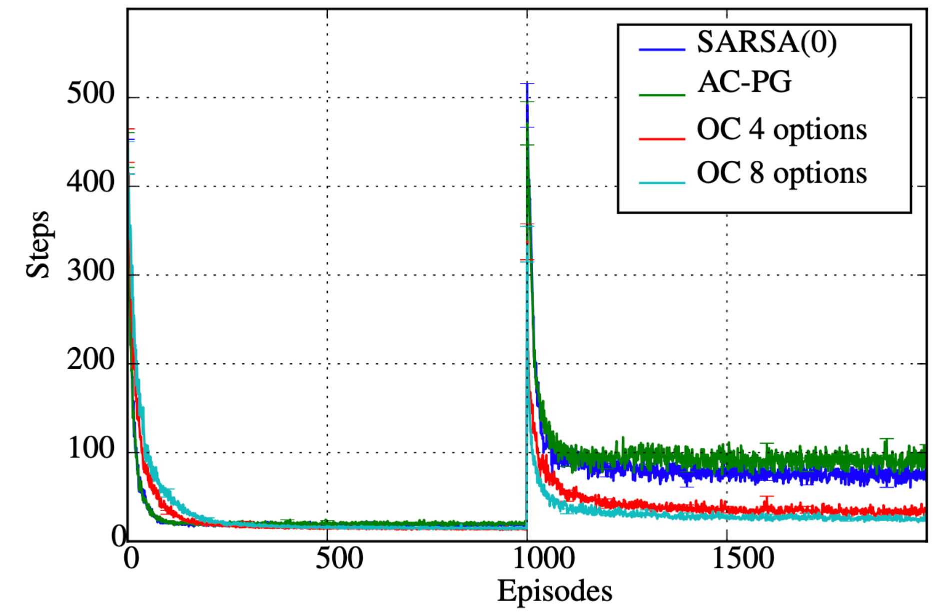


Figure 2: After a 1000 episodes, the goal location in the four-rooms domain is moved randomly. Option-critic ("OC") recovers faster than the primitive actor-critic ("AC-PG") and SARSA(0). Each line is averaged over 350 runs.

- Works great!
- But:
  - Cannot directly leverage recent advances in gradient-based policy optimization from MDPs
- Solution:
  - DAC (The Double Actor-Critic Architecture)

The Double Actor-Critic Architecture for Learning Options (DAC)

- Reformulate the option framework as two parallel augmented MDPs
    - A policy based method
    - All policy optimization algorithms can be used off the shelf (advantage over Option Critic)
- Apply an actor-critic algorithm on each augmented MDP (double actor critic)
- Show that one critic is enough.

Two Augmented MDPs

- The high-MDP: M^H
  - The agent makes high-level decisions (i.e., option selection) in M^H according to $\pi, \{\beta_\omega\}$ and thus optimizes $\pi, \{\beta_\omega\}$

- The low-MDP: M^L
  - The agent makes low-level decisions (i.e., action selection) in M^L according to $\{\pi_\omega\}$ and thus optimizes $\{\pi_\omega\}$

High MDP

- Define a dummy option # and $\Omega^{+} = \Omega \cup \{\#\}$
- Interpret a state-option pair as new state and an option as a new action.

$$M^{\mathcal{H}} \doteq \{\mathcal{S}^{\mathcal{H}}, \mathcal{A}^{\mathcal{H}}, p^{\mathcal{H}}, p_0^{\mathcal{H}}, r^{\mathcal{H}}, \gamma\}, \quad \mathcal{S}^{\mathcal{H}} \doteq \mathcal{O}^{+} \times \mathcal{S}, \quad \mathcal{A}^{\mathcal{H}} \doteq \mathcal{O},$$

$$p^{\mathcal{H}}(S_{t+1}^{\mathcal{H}} | S_t^{\mathcal{H}}, A_t^{\mathcal{H}}) \doteq p^{\mathcal{H}}\big((O_t, S_{t+1}) | (O_{t-1}, S_t), A_t^{\mathcal{H}})\big) \doteq \mathbb{I}_{A_t^{\mathcal{H}} = O_t} p(S_{t+1} | S_t, O_t),$$

$$p_0^{\mathcal{H}}(S_0^{\mathcal{H}}) \doteq p_0^{\mathcal{H}}\big((O_{-1}, S_0)\big) \doteq p_0(S_0) \mathbb{I}_{O_{-1} = \#},$$

$$r^{\mathcal{H}}(S_t^{\mathcal{H}}, A_t^{\mathcal{H}}) \doteq r^{\mathcal{H}}\big((O_{t-1}, S_t), O_t\big) \doteq r(S_t, O_t)$$

- Define Markov Policy

$$\pi^{\mathcal{H}}(A_t^{\mathcal{H}} | S_t^{\mathcal{H}}) \doteq \pi^{\mathcal{H}}(O_t | (O_{t-1}, S_t)) \doteq p(O_t | S_t, O_{t-1}) \mathbb{I}_{O_{t-1} \neq \#} + \pi(S_t, O_t) \mathbb{I}_{O_{t-1} = \#}$$

Low MDP

- Interpret state-option pair as state and leave actions unchanged.

$$M^{\mathcal{L}} \doteq \{\mathcal{S}^{\mathcal{L}}, \mathcal{A}^{\mathcal{L}}, p^{\mathcal{L}}, p_0^{\mathcal{L}}, r^{\mathcal{L}}, \gamma\}, \quad \mathcal{S}^{\mathcal{L}} \doteq \mathcal{S} \times \mathcal{O}, \quad \mathcal{A}^{\mathcal{L}} \doteq \mathcal{A},$$

$$p^{\mathcal{L}}(S_{t+1}^{\mathcal{L}} | S_t^{\mathcal{L}}, A_t^{\mathcal{L}}) \doteq p^{\mathcal{L}}\big((S_{t+1}, O_{t+1}) | (S_t, O_t), A_t\big) \doteq p(S_{t+1} | S_t, A_t) p(O_{t+1} | S_{t+1}, O_t),$$

$$p_0^{\mathcal{L}}(S_0^{\mathcal{L}}) \doteq p^{\mathcal{L}}\big((S_0, O_0)\big) \doteq p_0(S_0) \pi(S_0, O_0),$$

$$r^{\mathcal{L}}(S_t^{\mathcal{L}}, A_t^{\mathcal{L}}) \doteq r^{\mathcal{L}}\big((S_t, O_t), A_t\big) \doteq r(S_t, A_t)$$

- Define Markov policy:

$$\pi^{\mathcal{L}}(A_t^{\mathcal{L}} | S_t^{\mathcal{L}}) \doteq \pi^{\mathcal{L}}\big(A_t | (S_t, O_t)\big) \doteq \pi_{O_t}(A_t | S_t)$$

How to sample from these MDPs?

- Consider trajectories with non zero probability:

$$\Omega = \{\tau | p(\tau | \pi, \mathcal{O}, M) > 0\}$$

$$\Omega^H = \{\tau^H | p(\tau^H | \pi^H, M^H) > 0\}$$

$$\Omega^L = \{\tau^L | p(\tau^L | \pi^L, M^L) > 0\}$$

- Where $\tau = \{S_0, O_0, S_1, O_1, ..., S_T\}$

- Define functions: $f^H : \Omega \to \Omega^H$ $f^L : \Omega \to \Omega^L$

How to sample from these MDPs?

- Lemma 1: $f^H : \Omega \to \Omega^H$ is a bijection and

$$p(\tau|\pi, \mathcal{O}, M) = p(\tau^H|\pi^H, M^H), r(\tau) = r(\tau^H)$$

- Lemma 2: $f^L : \Omega \to \Omega^L$ is a bijection and

$$p(\tau|\pi, \mathcal{O}, M) = p(\tau^L|\pi^L, M^L), r(\tau) = r(\tau^L)$$

- So, Sampling from $\{\pi, \mathcal{O}, M\}$ is equivalent to sampling from $\{\pi^H, M^H\}$ and $\{\pi^L, M^L\}$

How to optimize π^H and π^L in these MDPs?

- Proposition:

$$J = \int r(\tau)p(\tau|\pi, \mathcal{O}, M)d\tau = \int r(\tau^H)p(\tau^H|\pi^H, M^H)d\tau^H = \int r(\tau^L)p(\tau^L|\pi^L, M^L)d\tau^L$$

  - Optimizing $\pi, \mathcal{O}$ in M is equivalent to optimizing π^H in M^H and optimizing π^L in M^L

How to optimize π^H and π^L in these MDPs?

- Observation1: M^H *depends on* $\{\pi_o\}$ *while* π^H *depends on* π *and* $\{\beta_o\}$
  - When we keep the intra-option policies $\{\pi_o\}$ fixed and optimize π^H, we are implicitly optimizing π and $\{\beta_o\}$

- Observation 2: M^L *depends on* π *and* $\{\beta_o\}$ *while* π^L *depends on* $\{\pi_o\}$
  - When we keep the master policy π and the termination conditions $\{\beta_o\}$ fixed and optimize π^L, we are implicitly optimizing $\{\pi_o\}$

Algorithm 1: Pseudocode of DAC

Input:
Parameterized π, $\{\pi_o, \beta_o\}_{o\in\mathcal{O}}$
Policy optimization algorithms $\mathbb{A}_1, \mathbb{A}_2$

Get an initial state S_0
$t \leftarrow 0$
while *True* **do**
  Sample O_t from $\pi^{\mathcal{H}}\big(\cdot|(O_{t-1}, S_t)\big)$
  Sample A_t from $\pi^{\mathcal{L}}\big(\cdot|(S_t, O_t)\big)$
  Execute A_t, get R_{t+1}, S_{t+1}
  `// The two optimizations can be done in any order or alternatively`
  Optimize $\pi^{\mathcal{H}}$ with $(S_t^{\mathcal{H}}, A_t^{\mathcal{H}}, R_{t+1}, S_{t+1}^{\mathcal{H}})$ and $\mathbb{A}_1$
  Optimize $\pi^{\mathcal{L}}$ with $(S_t^{\mathcal{L}}, A_t^{\mathcal{L}}, R_{t+1}, S_{t+1}^{\mathcal{L}})$ and $\mathbb{A}_2$
  $t \leftarrow t + 1$
end

Do we need two critics?

- **Proposition:** When state-value functions are used as critics, one critic can be expressed in terms of the other, and hence only one critic is necessary

$$v_{\pi^{\mathcal{H}}}\big((o, s')\big) = \sum_{o'} \pi^{\mathcal{H}}\big(o'|(o, s')\big) v_{\pi^{\mathcal{L}}}\big((s', o')\big), \textit{ where}$$

$$v_{\pi^{\mathcal{H}}}\big((o, s')\big) \doteq \mathbb{E}_{\pi^{\mathcal{H}}, M^{\mathcal{H}}}[\textstyle\sum_{i=1}^{\infty} \gamma^{i-1} R^{\mathcal{H}}_{t+i} \mid S^{\mathcal{H}}_t = (o, s')],$$

$$v_{\pi^{\mathcal{L}}}\big((s', o')\big) \doteq \mathbb{E}_{\pi^{\mathcal{L}}, M^{\mathcal{L}}}[\textstyle\sum_{i=1}^{\infty} \gamma^{i-1} R^{\mathcal{L}}_{t+i} \mid S^{\mathcal{L}}_t = (s', o')],$$

Experiments setup

- Want to see
  - DAC vs. existing gradient based option learnings
  - DAC vs. hierarchy-free methods
- DAC can use any policy optimization (AC, SAC, NAC, PPO, …).
- Here focus on DAC + PPO.

Results

- Single task

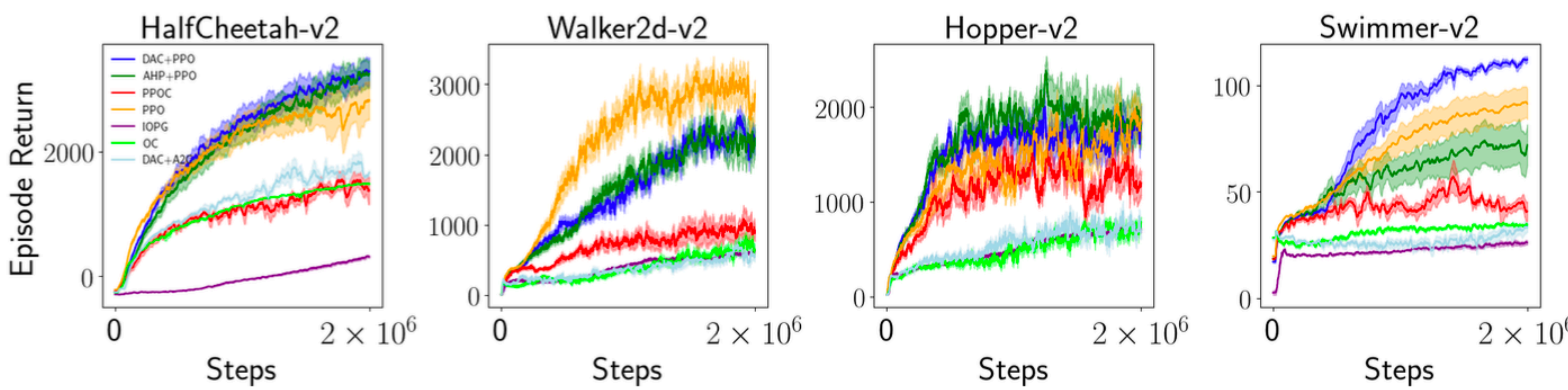


Figure 1: Online performance on a single task

- DAC + A2C similar to OC

- DAC + PPO similar to PPO

Results

- Transfer Learning

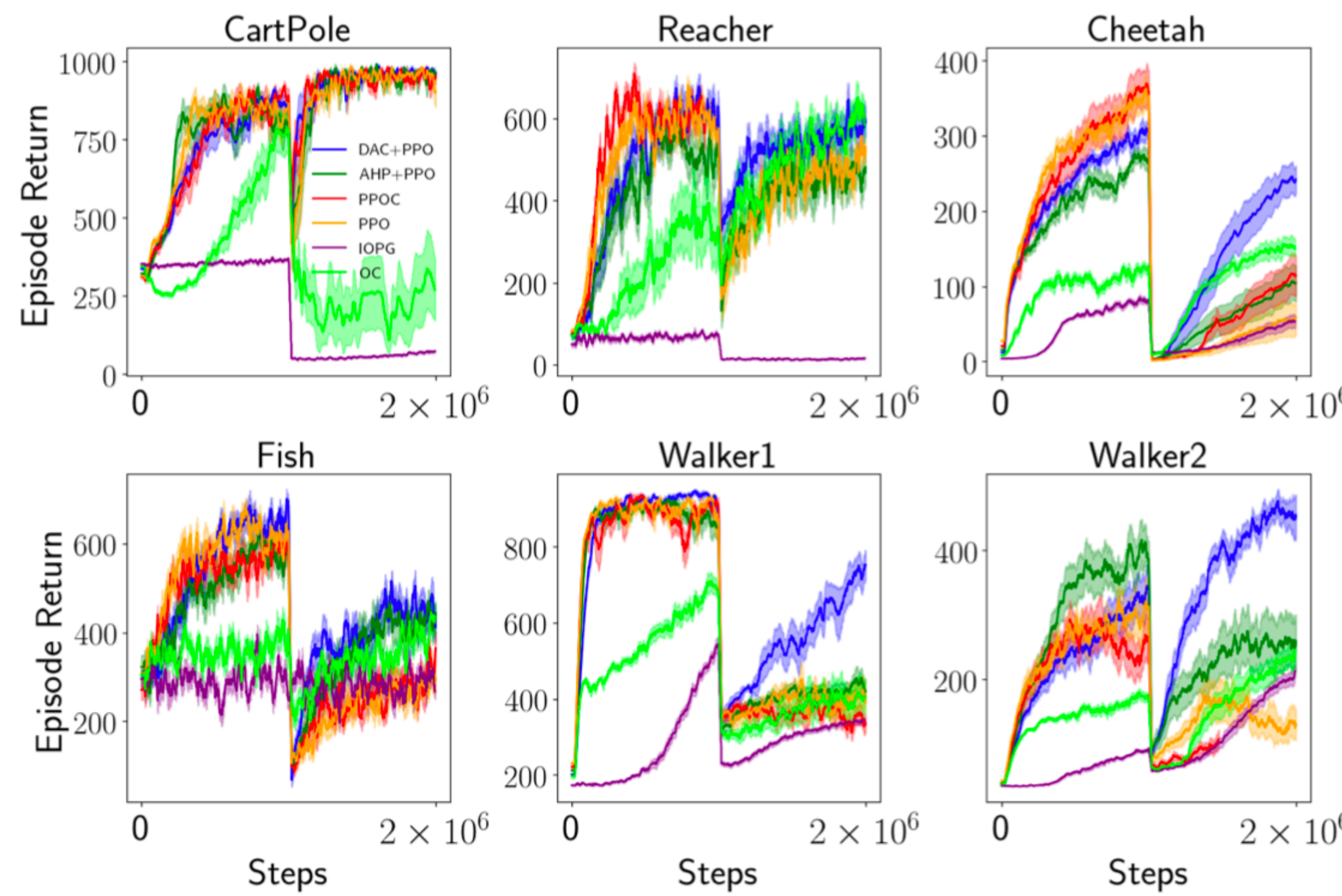


Figure 2: Online performance for transfer learning

CartPole = (balance, balance_sparse)

Reacher = (easy, hard)

Cheetah = (run, backward)

Fish = (upright, downleft)

Walker1 = (squat, stand)

Walker2 = (walk, backward)

Recap

- **Problem:** An end to end, policy based method to learn options and policy over them
- **Solution:** Option Critic
- **Limitation:** Can't use other policy optimization algorithms off-the-shelf
  - **Solution:** Reformulate the SMDP of the option framework as two augmented MDPs (Double Actor Critic)

Thank you