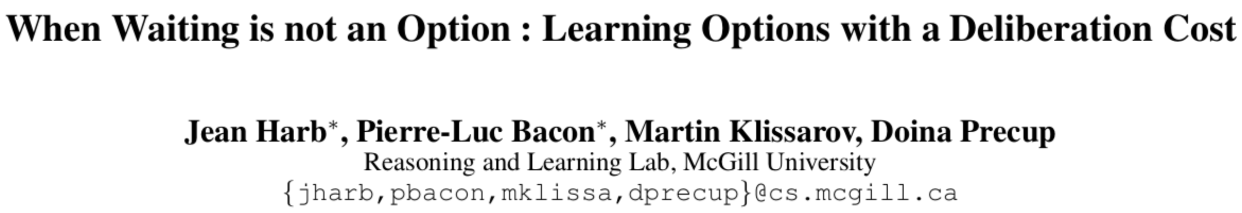
1709.04571 - When Waiting is not an Option : Learning Options with a Deliberation Cost



When Waiting is not an Option : Learning Options with a Deliberation Cost

Jean Harb*, Pierre-Luc Bacon*, Martin Klissarov, Doina Precup
Reasoning and Learning Lab, McGill University
{jharb,pbacon,mklissa,dprecup}@cs.mcgill.ca

- **Yunqiu Xu**
- Related reading:
    - Original OC: 1609.05140 - The Option-Critic Architecture
    - CASL, integration of this work: 1711.10314 - Crossmodal Attentive Skill Learner
- Implementation: https://github.com/jeanharb/a2oc_delib

---

1. Introduction

- Challenges:
    - What have been done: how to learn
    - What we need to tackle: **what good options should be**
    - OC存在终止过于频繁的问题 → 这样option起不到什么作用, 跟action没什么差别
- 摘抄下我原来对OC的总结:
    - Option-critic 可以实现end-to-end HRL, 仅仅需要预先指定option的数量
    - 我理解为每个option都是一个独立的policy, 然后对于π_Ω根据 state 选 option 的过程其实就类似于 π_ω根据state选action的过程
    - 不需要额外的reward和sub-goal, 也可以稍加修改来增加additional reward
    - 需要额外假定: 对于任何状态, 可以使用任何option
- Our work: extend OC to A2OC
    - 利用 **bounded rationality framework** 解释什么样的 temporal abstractions 是有益的

    - 尝试提升学习效率

2. Preliminaries

- An option $\omega \in \Omega$ consists of:
    - Initiation set $I \subseteq S$
    - Intra-option policy $\pi_\omega(s) : S \rightarrow A$, this is **sub-policy**
    - Termination function $\beta_\omega(s) : S \rightarrow [0, 1]$
    - Given a state, master policy π select an option ω, then intra-option policy will be executed to reach terminate state → 到达terminate state之前一直使用这个option, 停止后则选择切换option
- Option-critic : 这里结合了原始OC和CASL的总结
    - $Q_\Omega(s, \omega)$: Option value function for option $\omega \in \Omega$, 这个函数类似Q函数, 但是选择的不是action而是option

    $$Q_\Omega(s,\omega) = \sum_a \pi_\omega(a|s) \left((r(s,a) + \gamma \sum_{s'} T(s'|s,a) U(s',\omega) \right)$$

    - $U(s', \omega)$: option utality function, 这里s'为某个状态s下选用动作a到达的下一个状态

    $$U(s',\omega) = (1 - \beta_\omega(s'))Q_\Omega(\omega, s') + \beta_\omega(s')(V_\Omega(s') - c)$$

        - If $\beta_\omega(s') = 1$, 该option在状态s'终止 → $U(s', \omega) = V_\Omega(s') - c$
        - If $\beta_\omega(s') = 0$, 继续使用这个option → $U(s', \omega) = Q_\Omega(\omega, s')$
    - Deliberation cost c **这个是原版OC没有的** : add penalty when options terminate → **let options terminate less frequently**
    - $V_\Omega(s')$: value function over options (master policy π_Ω)

    $$V_\Omega(s') = \sum_\omega \pi_\Omega(\omega|s') Q_\Omega(\omega, s')$$

3. Deliberation Cost Model

- When to add deliberation cost

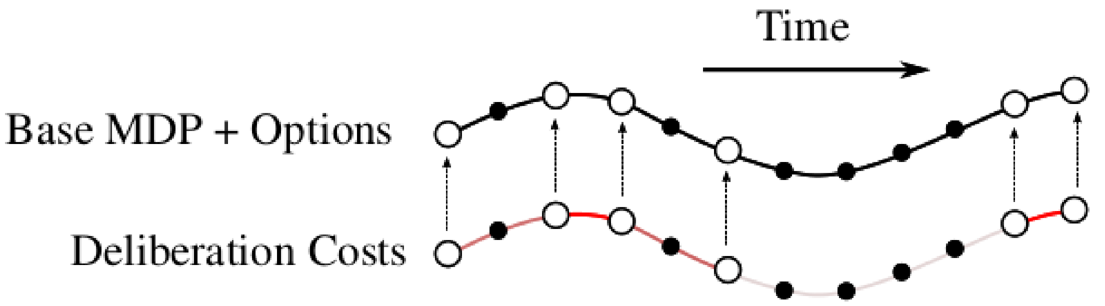
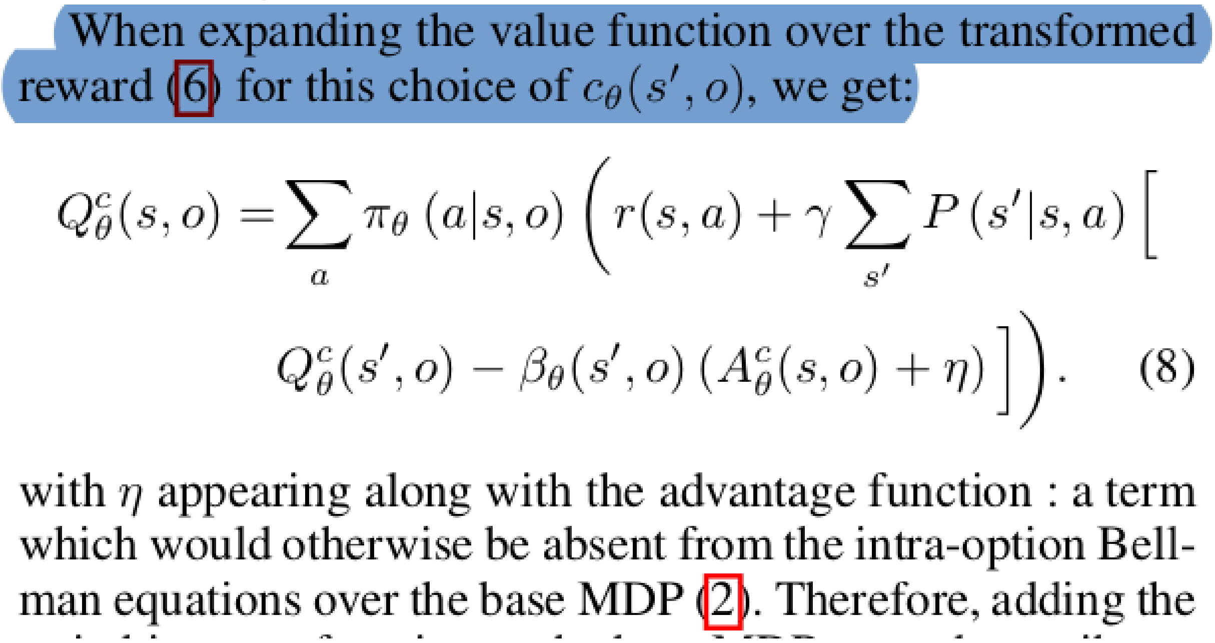


Figure 1: A deliberation cost is incurred upon switching to a new option and is subtracted from the reward of the base MDP. Open circles represent SMDP decision points while filled circles are primitive steps within an option. The cost rate for each option is represented by the intensity of the subtrajectory.

- Unconstrained optimization problem with deliberation cost (γ is the discount factor of base MDP and λ is for deliberation cost)

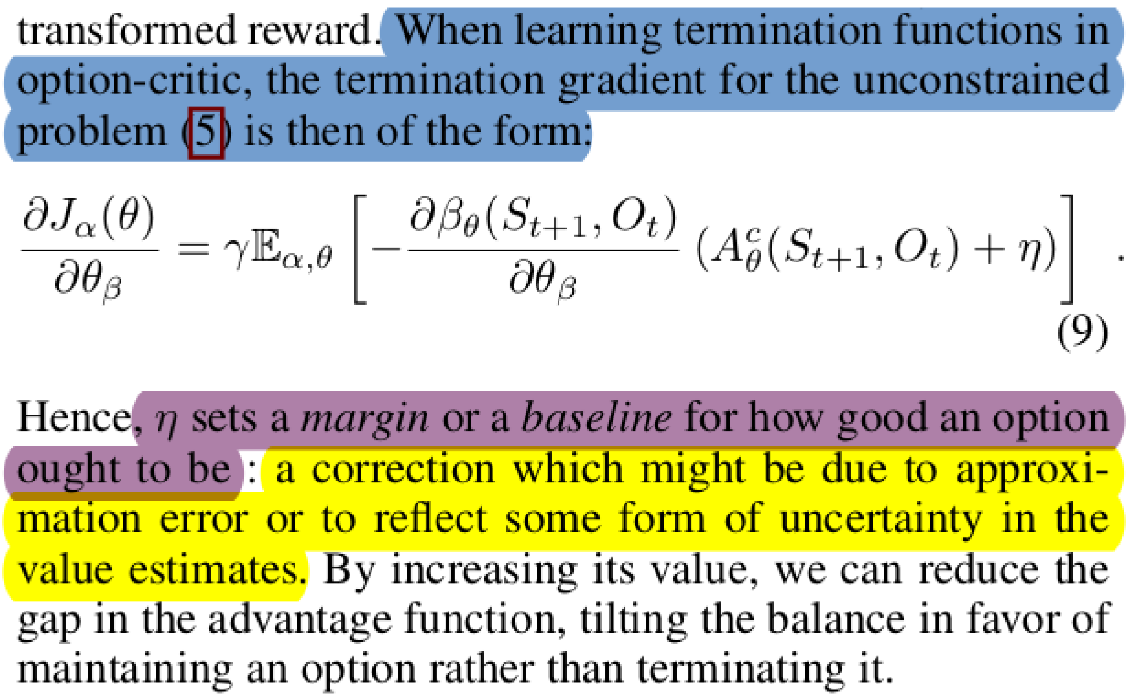
$$J_{\alpha}^{\gamma,\lambda}(\theta) = \sum_{s,o} \alpha(s,o) \left(Q_{\theta}^{\gamma}(s,o) - \eta D_{\theta}^{\lambda}(s,o) \right)$$



When expanding the value function over the transformed reward (6) for this choice of $c_\theta(s', o)$, we get:

$$Q_{\theta}^{c}(s,o) = \sum_{a} \pi_{\theta}(a|s,o) \Bigg(r(s,a) + \gamma \sum_{s'} P(s'|s,a) \Big[Q_{\theta}^{c}(s',o) - \beta_{\theta}(s',o) \left(A_{\theta}^{c}(s,o) + \eta \right) \Big] \Bigg) . \qquad (8)$$

with η appearing along with the advantage function : a term which would otherwise be absent from the intra-option Bellman equations over the base MDP (2). Therefore, adding the



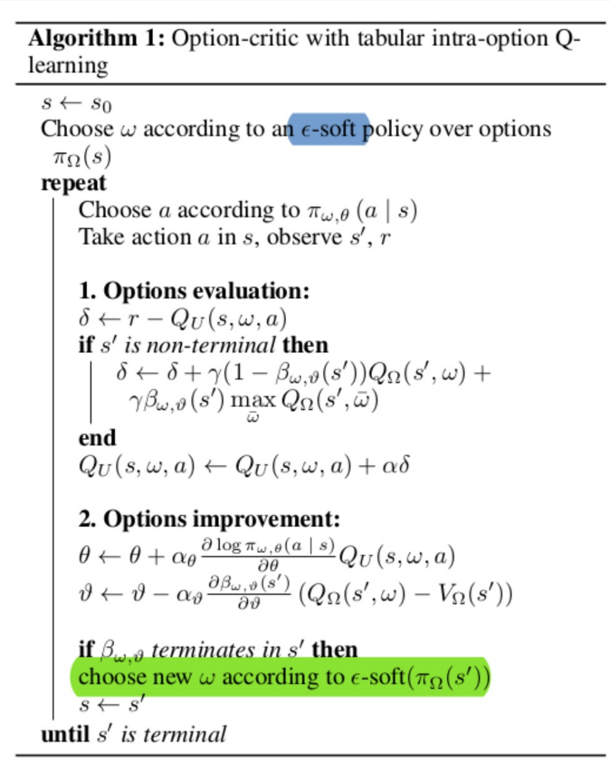
transformed reward. When learning termination functions in option-critic, the termination gradient for the unconstrained problem (5) is then of the form:

$$\frac{\partial J_\alpha(\theta)}{\partial \theta_\beta} = \gamma \mathbb{E}_{\alpha,\theta}\left[-\frac{\partial \beta_\theta(S_{t+1}, O_t)}{\partial \theta_\beta}\left(A^c_\theta(S_{t+1}, O_t) + \eta\right)\right] . \tag{9}$$

Hence, η sets a *margin* or a *baseline* for how good an option ought to be : a correction which might be due to approximation error or to reflect some form of uncertainty in the value estimates. By increasing its value, we can reduce the gap in the advantage function, tilting the balance in favor of maintaining an option rather than terminating it.

4. Algorithm

这里可以对比下A2OC和原版OC的区别



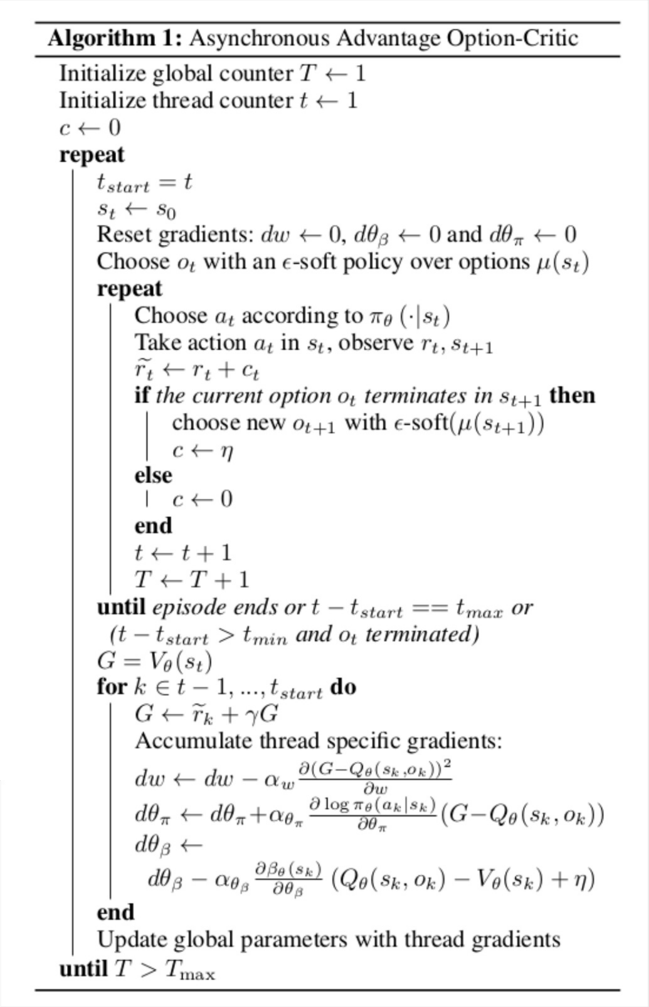
Algorithm 1: Option-critic with tabular intra-option Q-learning

```
$s \leftarrow s_0$
Choose $\omega$ according to an $\epsilon$-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 $\omega$ according to $\epsilon$-soft($\pi_\Omega(s')$)
    $s \leftarrow s'$
until $s'$ is terminal
```

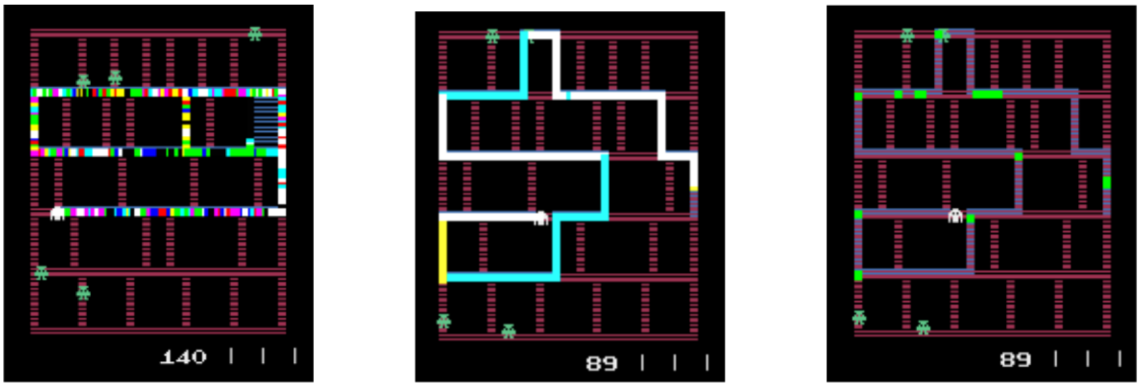


Algorithm 1: Asynchronous Advantage Option-Critic

```
Initialize global counter $T \leftarrow 1$
Initialize thread counter $t \leftarrow 1$
$c \leftarrow 0$
repeat
    $t_{start} = t$
    $s_t \leftarrow s_0$
    Reset gradients: $dw \leftarrow 0$, $d\theta_\beta \leftarrow 0$ and $d\theta_\pi \leftarrow 0$
    Choose $o_t$ with an $\epsilon$-soft policy over options $\mu(s_t)$
    repeat
        Choose $a_t$ according to $\pi_\theta(\cdot|s_t)$
        Take action $a_t$ in $s_t$, observe $r_t, s_{t+1}$
        $\widetilde{r}_t \leftarrow r_t + c_t$
        if the current option $o_t$ terminates in $s_{t+1}$ then
            choose new $o_{t+1}$ with $\epsilon$-soft($\mu(s_{t+1})$)
            $c \leftarrow \eta$
        else
            $c \leftarrow 0$
        end
        $t \leftarrow t + 1$
        $T \leftarrow T + 1$
    until episode ends or $t - t_{start} == t_{max}$ or
      $(t - t_{start} > t_{min}$ and $o_t$ terminated)
    $G = V_\theta(s_t)$
    for $k \in t - 1, ..., t_{start}$ do
        $G \leftarrow \widetilde{r}_k + \gamma G$
        Accumulate thread specific gradients:
        $dw \leftarrow dw - \alpha_w \frac{\partial(G - Q_\theta(s_k, o_k))^2}{\partial w}$
        $d\theta_\pi \leftarrow d\theta_\pi + \alpha_{\theta_\pi} \frac{\partial \log \pi_\theta(a_k|s_k)}{\partial \theta_\pi} (G - Q_\theta(s_k, o_k))$
        $d\theta_\beta \leftarrow$
          $d\theta_\beta - \alpha_{\theta_\beta} \frac{\partial \beta_\theta(s_k)}{\partial \theta_\beta} (Q_\theta(s_k, o_k) - V_\theta(s_k) + \eta)$
    end
    Update global parameters with thread gradients
until $T > T_{\max}$
```

5. Experiment

- 加了deliberation cost之后每个option持续时间更长了



(a) Without a deliberation cost, options terminate instantly and are used in any scenario without specialization.

(b) Options are used for extended periods and in specific scenarios through a trajectory, when using a deliberation cost.

(c) Termination is sparse when using the deliberation cost. The agent terminates options at intersections requiring high level decisions.

Figure 2: We show the effects of using deliberation costs on both the option termination and policies. In figures (a) and (b), every color in the agent trajectory represents a different option being executed. This environment is the game Amidar, of the Atari 2600 suite.

- 加了deliberation cost就可以有很明显效果, 不需要加很多

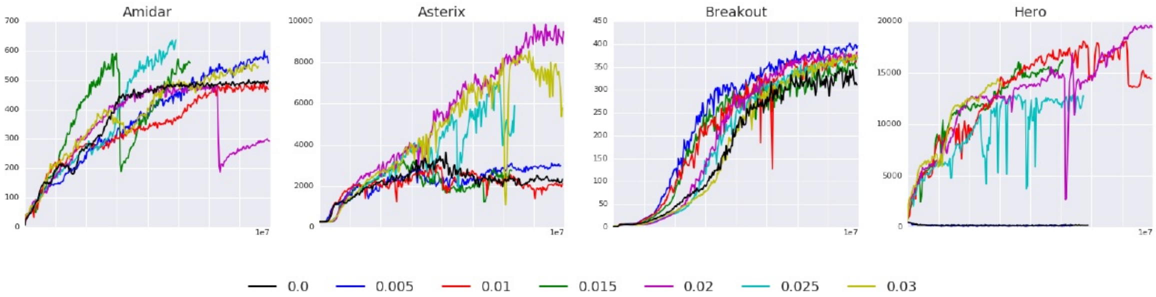


Figure 3: Training curves with different deliberation costs on 4 Atari 2600 games. Trained for up to 80M frames.

- Increase deliberation cost $\eta \rightarrow$ decrease average termination probabilities

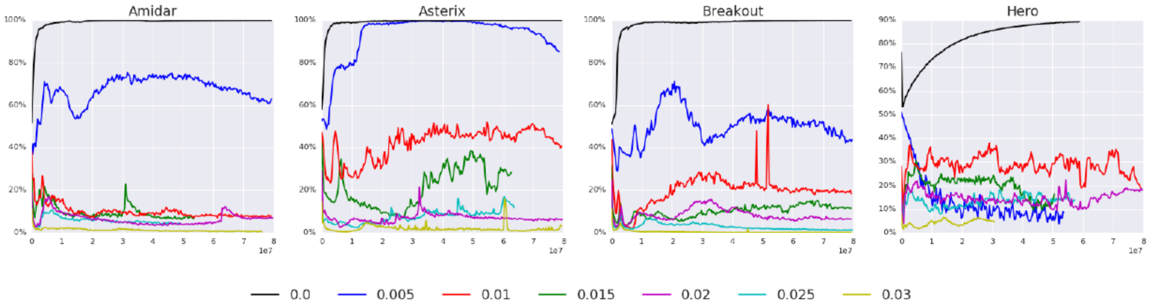
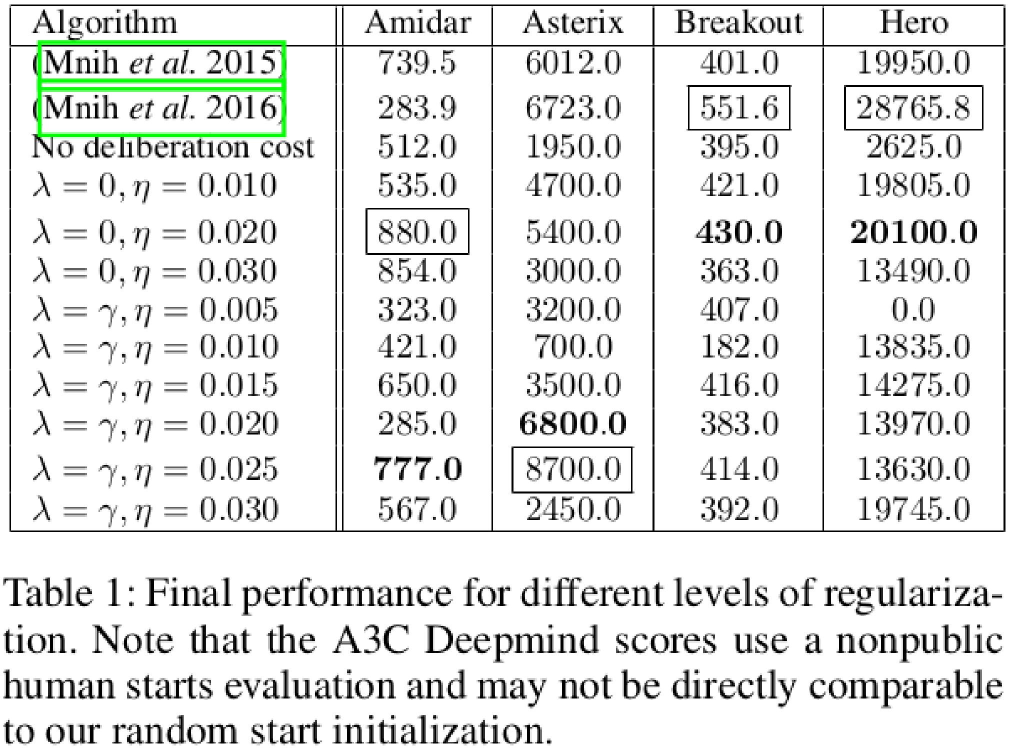


Figure 4: Average termination probabilities through training, with varying amounts of deliberation costs. With no deliberation, the termination rate quickly goes to 100% (black curve).

- Result of different discount factor (regularization)



Algorithm	Amidar	Asterix	Breakout	Hero
(Mnih *et al.* 2015)	739.5	6012.0	401.0	19950.0
(Mnih *et al.* 2016)	283.9	6723.0	$\boxed{551.6}$	$\boxed{28765.8}$
No deliberation cost	512.0	1950.0	395.0	2625.0
$\lambda = 0, \eta = 0.010$	535.0	4700.0	421.0	19805.0
$\lambda = 0, \eta = 0.020$	$\boxed{880.0}$	5400.0	**430.0**	**20100.0**
$\lambda = 0, \eta = 0.030$	854.0	3000.0	363.0	13490.0
$\lambda = \gamma, \eta = 0.005$	323.0	3200.0	407.0	0.0
$\lambda = \gamma, \eta = 0.010$	421.0	700.0	182.0	13835.0
$\lambda = \gamma, \eta = 0.015$	650.0	3500.0	416.0	14275.0
$\lambda = \gamma, \eta = 0.020$	285.0	**6800.0**	383.0	13970.0
$\lambda = \gamma, \eta = 0.025$	**777.0**	$\boxed{8700.0}$	414.0	13630.0
$\lambda = \gamma, \eta = 0.030$	567.0	2450.0	392.0	19745.0

Table 1: Final performance for different levels of regularization. Note that the A3C Deepmind scores use a nonpublic human starts evaluation and may not be directly comparable to our random start initialization.

6. Conclusion

- Extend OC to A2OC: 采用和A3C类似的构架, 提升性能
- Add deliberation cost: 防止 option 终止过于频繁
- 这篇论文重在实现好嘞~