01

策略梯度的缺点

策略梯度算法回顾

- 蒙特卡洛策略梯度（REINFORCE）算法

  initialize θ arbitrarily

  for each episode $\{s_1, a_1, r(s_1, a_1), \dots, s_T, a_T, r(s_T, a_T)\} \sim \pi_\theta$ do

  for $t = 1$ to T do

  $$\theta \leftarrow \theta + \alpha \frac{\partial}{\partial \theta} \log \pi_\theta(a_t|s_t) G_t$$

  end for

  end for

  return θ

相关定义

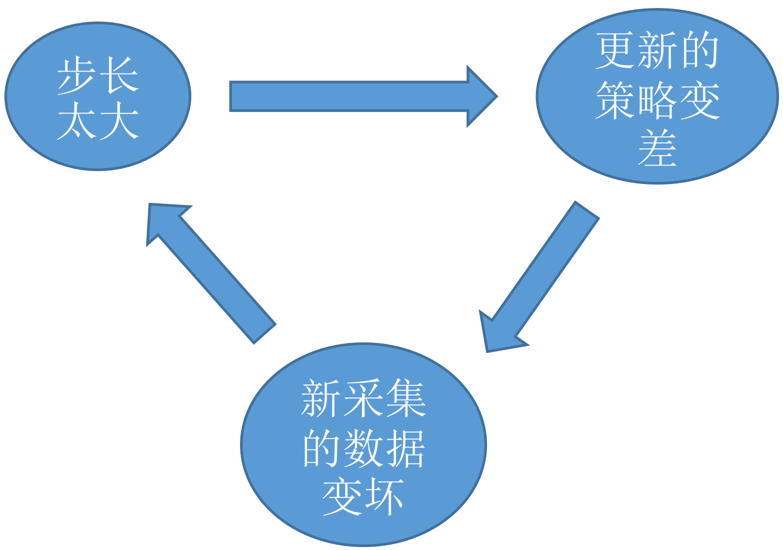
- s_t ， a_t ，$r(s_t, a_t)$ ： t 时刻的状态，动作和奖励
- π_θ ，θ ：使用的策略，表示策略所使用的参数
- G_t ：累计奖励
- α ：步长

“OPTIMIZING EXPECTATIONS: FROM DEEP REINFORCEMENT LEARNING TO STOCHASTIC COMPUTATION GRAPHS” ，John Schulman. (2016)

策略梯度的缺点

步长

- 步长难以确定
  - 采集到的数据的分布会随策略的更新而变化。
  - 较差的步长产生的影响大。



“OPTIMIZING EXPECTATIONS: FROM DEEP REINFORCEMENT LEARNING TO STOCHASTIC COMPUTATION GRAPHS” ，John Schulman. (2016)

02

TRPO

策略梯度的优化目标

- 优化目标的两种形式
  - 第一种：$J(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)}[\sum_t \gamma^t r(s_t, a_t)]$
  - 因为 $V^{\pi_\theta}(s) = \mathbb{E}_{a \sim \pi_\theta(s)}[Q^{\pi_\theta}(s,a)] = \mathbb{E}_{a \sim \pi_\theta(s)}\left[\mathbb{E}_{\tau \sim p_\theta(\tau)}[\sum_{s_k=s, a_k=a} \sum_{t=k}^{\infty} \gamma^{t-k} r(s_t, a_t)]\right]$。
  - 所以优化目标的第二种形式是： $J(\theta) = \mathbb{E}_{s_0 \sim p_\theta(s_0)}[V^{\pi_\theta}(s_0)]$

相关定义

- τ：轨迹
- s_0：初始状态
- s_t， a_t，$r(s_t, a_t)$： t 时刻的状态，动作和奖励
- π_θ：使用的策略
- θ：表示策略所使用的参数
- Q^{π_θ} 和 V^{π_θ}：策略 π_θ 下的 Q 值与状态值函数

"OPTIMIZING EXPECTATIONS: FROM DEEP REINFORCEMENT LEARNING TO STOCHASTIC COMPUTATION GRAPHS"，John Schulman. (2016)

优化目标的优化量

$$J(\theta) = \mathbb{E}_{\tau\sim p_\theta(\tau)}[\sum_t \gamma^t r(s_t, a_t)]$$
$$J(\theta) = \mathbb{E}_{s_0\sim p_\theta(s_0)}[V^{\pi_\theta}(s_0)]$$

$$\begin{aligned}
J(\theta') - J(\theta) &= J(\theta') - \mathbb{E}_{s_0\sim p(s_0)}[\mathrm{V}^{\pi_\theta}(s_0)] \\
&= J(\theta') - \mathbb{E}_{\tau\sim p_{\theta'}(\tau)}[\mathrm{V}^{\pi_\theta}(s_0)] \\
&= J(\theta') - \mathbb{E}_{\tau\sim p_{\theta'}(\tau)}\left[\sum_{t=0}^{\infty}\gamma^t \mathrm{V}^{\pi_\theta}(s_t) - \sum_{t=1}^{\infty}\gamma^t \mathrm{V}^{\pi_\theta}(s_t)\right] \\
&= J(\theta') + \mathbb{E}_{\tau\sim p_{\theta'}(\tau)}\left[\sum_{t=0}^{\infty}\gamma^t (\gamma \mathrm{V}^{\pi_\theta}(s_{t+1}) - \mathrm{V}^{\pi_\theta}(s_t))\right] \\
&= \mathbb{E}_{\tau\sim p_{\theta'}(\tau)}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t)\right] + \mathbb{E}_{\tau\sim p_{\theta'}(\tau)}\left[\sum_{t=0}^{\infty}\gamma^t (\gamma \mathrm{V}^{\pi_\theta}(s_{t+1}) - \mathrm{V}^{\pi_\theta}(s_t))\right] \\
&= \mathbb{E}_{\tau\sim p_{\theta'}(\tau)}\left[\sum_{t=0}^{\infty}\gamma^t (r(s_t, a_t) + \gamma \mathrm{V}^{\pi_\theta}(s_{t+1}) - \mathrm{V}^{\pi_\theta}(s_t))\right] \\
&= \mathbb{E}_{\tau\sim p_{\theta'}(\tau)}[\sum_{t=0}^{\infty}\gamma^t A^{\pi_\theta}(s_t, a_t)]
\end{aligned}$$

初始状态的分布与θ无关

$J(\theta')$的定义

$Q^{\pi_\theta}(s_t, a_t)$

$A^{\pi_\theta}(s_t, a_t)$
$= Q^{\pi_\theta}(s_t, a_t) - \mathrm{V}^{\pi_\theta}(s_t)$

不方便采样

“OPTIMIZING EXPECTATIONS: FROM DEEP REINFORCEMENT LEARNING TO STOCHASTIC COMPUTATION GRAPHS” , John Schulman. (2016)

使用重要性采样

- 使用重要性采样（Importance Sampling）

$$
\begin{aligned}
&J(\theta') - J(\theta) \\
&= \mathbb{E}_{\tau\sim p_{\theta'}(\tau)}\left[\sum_{t=0}^{\infty}\gamma^t A^{\pi_\theta}(s_t,a_t)\right] \\
&= \sum_t \mathbb{E}_{s_t\sim p_{\theta'}(s_t)}[\mathbb{E}_{a_t\sim\pi_{\theta'}(a_t|s_t)}[\gamma^t A^{\pi_\theta}(s_t,a_t)]] \\
&= \sum_t \mathbb{E}_{s_t\sim p_{\theta'}(s_t)}[\mathbb{E}_{a_t\sim\pi_\theta(a_t|s_t)}[\frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)}\gamma^t A^{\pi_\theta}(s_t,a_t)]]
\end{aligned}
$$

$$
\begin{aligned}
&A^{\pi_\theta}(s_t,a_t) \\
&= Q^{\pi_\theta}(s_t,a_t) - V^{\pi_\theta}(s_t)
\end{aligned}
$$

仍然是 $p_{\theta'}$
（近似操作）

重要性采样

忽略状态分布的差异

- 当策略更新前后的变化较小时，可以令 $p_\theta(s_t) \approx p_{\theta'}(s_t)$。
  - 假设使用确定性策略，当 $\pi_{\theta'}(s_t) \neq \pi_\theta\,(s_t)$ 的概率小于ϵ时
  - 或者假设使用随机策略，当$a'\sim\pi_{\theta'}(\cdot\,|s_t) \neq a\sim\,\pi_\theta(\cdot\,|s_t)$的概率小于$\epsilon$时
  - $p_{\theta'}(s_t) = (1-\epsilon)^t p_\theta(s_t) + (1-(1-\epsilon)^t)p_{mistake}(s_t)$
  - $|p_{\theta'}(s_t) - p_\theta(s_t)| = (1-(1-\epsilon)^t)|p_{mistake}(s_t) - p_\theta(s_t)| \leq 2(1-(1-\epsilon)^t) \leq 2\epsilon t$

$$(1-\epsilon)^t \geq 1-\epsilon t \; for \; \epsilon \in [0,1]$$

$$J(\theta') - J(\theta) \approx \sum_t \mathbb{E}_{s_t\sim p_\theta(s_t)}[\mathbb{E}_{a_t\sim\pi_\theta(a_t|s_t)}[\frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)}\gamma^t A^{\pi_\theta}(s_t,a_t)]]$$

"OPTIMIZING EXPECTATIONS: FROM DEEP REINFORCEMENT LEARNING TO STOCHASTIC COMPUTATION GRAPHS"，John Schulman. (2016)

约束策略的变化

- 使用KL散度约束策略更新的幅度

$$\theta' \leftarrow \arg\max_{\theta'} \sum_t \mathbb{E}_{s_t \sim p_\theta(s_t)}[\mathbb{E}_{a_t \sim \pi_\theta(a_t|s_t)}[\frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)}\gamma^t A^{\pi_\theta}(s_t,a_t)]]$$
$$such\ that\ \mathbb{E}_{s_t \sim p(s_t)}[D_{KL}\left(\pi_{\theta'}(a_t|s_t) \parallel \pi_\theta(a_t|s_t)\right)] \le \epsilon$$

- 实际多使用constraint violate as penalty

$$\theta' \leftarrow \arg\max_{\theta'} \sum_t \mathbb{E}_{s_t \sim p_\theta(s_t)}[\mathbb{E}_{a_t \sim \pi_\theta(a_t|s_t)}[\frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)}\gamma^t A^{\pi_\theta}(s_t,a_t)]]$$
$$-\lambda(D_{KL}(\pi_{\theta'}(a_t|s_t) \parallel \pi_\theta(a_t|s_t)) - \epsilon)$$

1. 优化上式，更新θ'
2. 更新 $\lambda \leftarrow \lambda + \alpha(D_{KL}(\pi_{\theta'}(a_t|s_t) \parallel \pi_\theta(a_t|s_t)) - \epsilon)$

"OPTIMIZING EXPECTATIONS: FROM DEEP REINFORCEMENT LEARNING TO STOCHASTIC COMPUTATION GRAPHS" , John Schulman. (2016)
http://rail.eecs.berkeley.edu/deeprlcourse/static/slides/lec-9.pdf

TRPO的原理

Line search
(like gradient ascent)

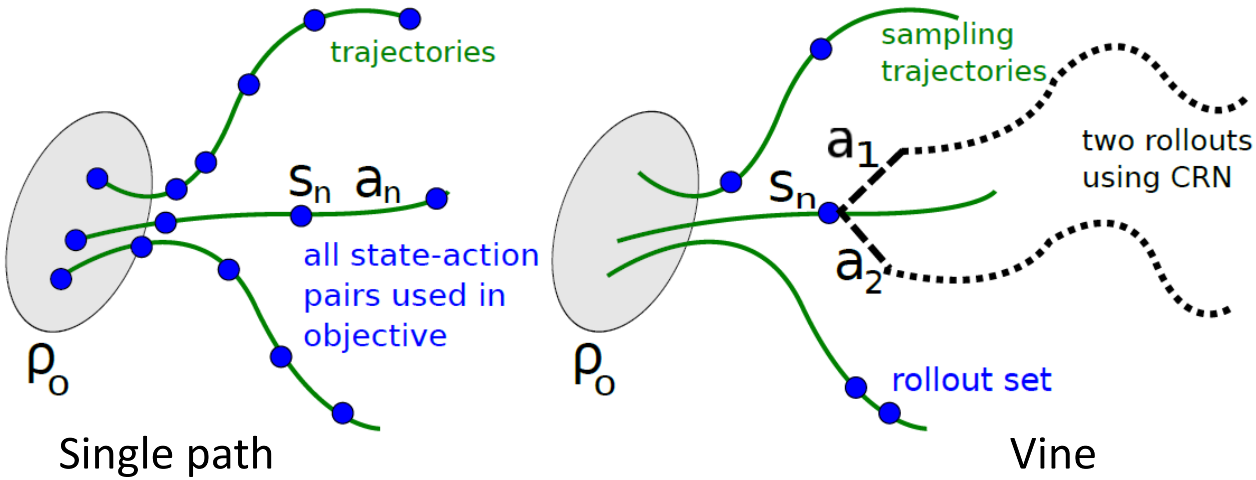
Optimization in Trust Region

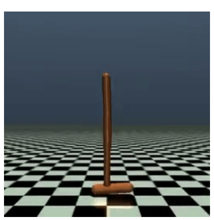
https://medium.com/@jonathan_hui/rl-trust-region-policy-optimization-trpo-explained-a6ee04eeeee9

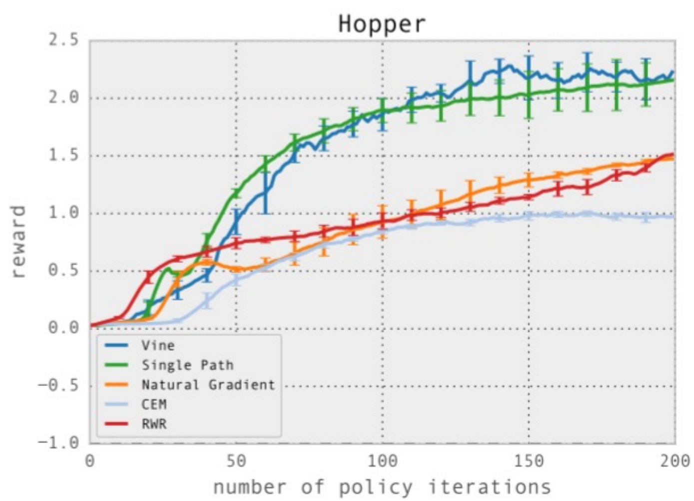
04

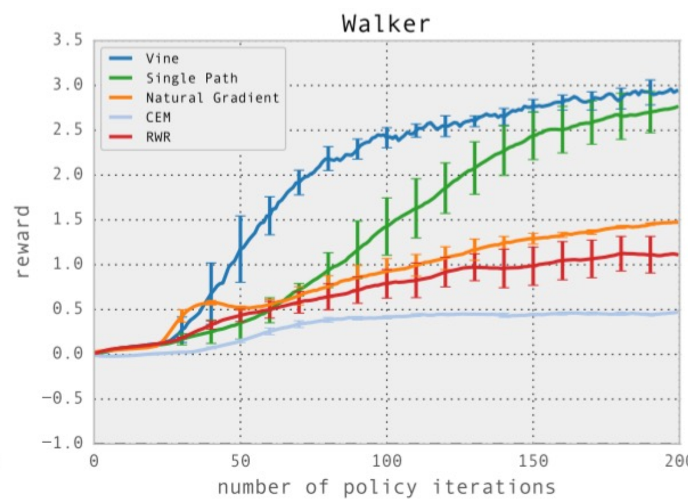
实验结果

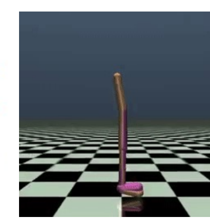
训练曲线











“Trust Region Policy Optimization”，John Schulman, et al. (2017)

结果比较

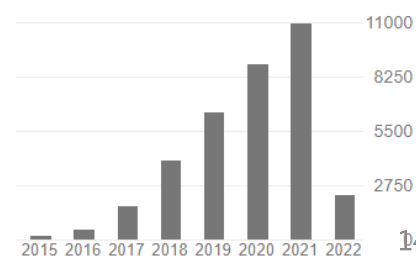
	B. Rider	*Breakout*	*Enduro*	*Pong*	*Q*bert*	*Seaquest*	*S. Invaders*
Random	354	1.2	0	−20.4	157	110	179
Human (Mnih et al., 2013)	7456	31.0	368	−3.0	18900	28010	3690
Deep Q Learning (Mnih et al., 2013)	4092	168.0	470	20.0	1952	1705	581
UCC-I (Guo et al., 2014)	5702	380	741	21	20025	2995	692
TRPO - single path	1425.2	10.8	534.6	20.9	1973.5	1908.6	568.4
TRPO - vine	859.5	34.2	430.8	20.9	7732.5	788.4	450.2

"Trust Region Policy Optimization", John Schulman, et al. (2017)

推荐阅读

PPO

- TRPO的不足
  - 近似带来误差
  - 求解约束优化问题的困难
- PPO算法
  - 理论更简洁，操作更简单，实验效果更好
  - 推荐阅读 Proximal Policy Optimization Algorithms, John Schulman, et al. (2017)



John Schulman

Research Scientist, OpenAI
Verified email at openai.com - Homepage
Artificial Intelligence Robotics Neuroscience

FOLLOW

TITLE	CITED BY	YEAR
Proximal policy optimization algorithms J Schulman, F Wolski, P Dhariwal, A Radford, O Klimov arXiv preprint arXiv:1707.06347	7028	2017
Trust region policy optimization J Schulman, S Levine, P Abbeel, M Jordan, P Moritz International conference on machine learning, 1889-1897	4720	2015

Google Scholar

Cited by

	All	Since 2017
Citations	35222	34267
h-index	42	41
i10-index	58	57

近端策略优化

Proximal Policy Optimization

回顾TRPO

- TRPO使用KL散度约束策略更新的幅度

$$\theta' \leftarrow \arg\max_{\theta'} \sum_t \mathbb{E}_{s_t \sim p_\theta(s_t)}[\mathbb{E}_{a_t \sim \pi_\theta(a_t|s_t)}[\frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)} \gamma^t A^{\pi_\theta}(s_t, a_t)]]$$

$$such\ that\ D_{KL}(\pi_{\theta'}(a_t|s_t) \parallel \pi_\theta(a_t|s_t)) \leq \epsilon$$

- 使用constraint violate as penalty

$$\theta' \leftarrow \arg\max_{\theta'} \sum_t \mathbb{E}_{s_t \sim p_\theta(s_t)}[\mathbb{E}_{a_t \sim \pi_\theta(a_t|s_t)}[\frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)} \gamma^t A^{\pi_\theta}(s_t, a_t)]] - \lambda(D_{KL}(\pi_{\theta'}(a_t|s_t) \parallel \pi_\theta(a_t|s_t)) - \epsilon)$$

1. 优化上式，更新θ'
2. 更新 $\lambda \leftarrow \lambda + \alpha(D_{KL}(\pi_{\theta'}(a_t|s_t) \parallel \pi_\theta(a_t|s_t)) - \epsilon)$

TRPO的不足

- 重要性比例带来的大方差
- 求解约束优化问题的困难

PPO: Proximal Policy Optimization

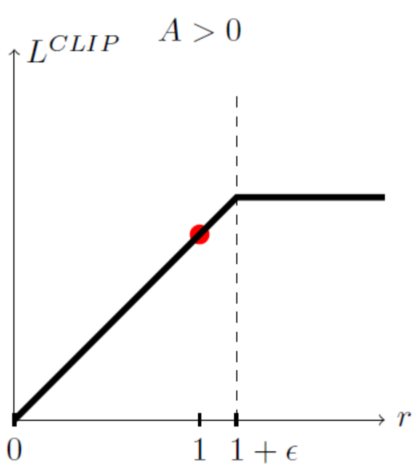
PPO在TRPO基础上的改进

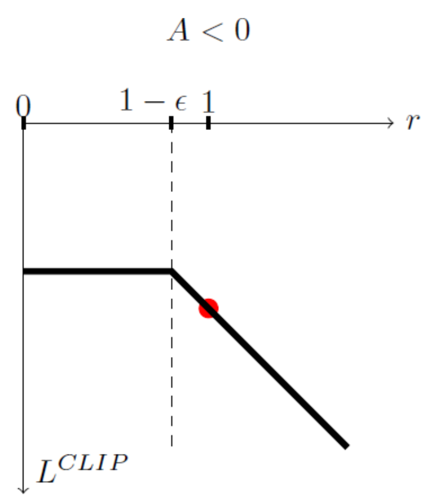
1. 截断式优化目标

conservative policy iteration

$$L^{CPI}(\theta) = \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}\hat{A}_t\right] = \hat{\mathbb{E}}_t\left[r_t(\theta)\hat{A}_t\right]$$

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min\left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right]$$





构建下界

$$L^{CLIP}(\theta) \leq L^{CPI}(\theta)$$

在$r = 1$附近相等

$$L^{CLIP}(\theta) = L^{CPI}(\theta)$$

Proximal Policy Optimization Algorithms, John Schulman, et al. (2017)

PPO: Proximal Policy Optimization

PPO在TRPO基础上的改进

1. 截断式优化目标

conservative policy iteration

$$L^{CPI}(\theta) = \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}\hat{A}_t\right] = \hat{\mathbb{E}}_t\left[r_t(\theta)\hat{A}_t\right]$$

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min\left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right]$$

2. 优势函数$\hat{A}_t$选用多步时序差分

$$\hat{A}_t = -V(s_t) + r_t + \gamma r_{t+1} + \cdots + \gamma^{T-t+1} r_{T-1} + \gamma^{T-t} V(s_T)$$

- 在每次迭代中，并行N个actor收集T步经验数据
- 计算每步的$\hat{A}_t$ 和$L^{CLIP}(\theta)$，构成mini-batch
- 更新参数θ，并更新$\theta_{\text{old}} \leftarrow \theta$

Proximal Policy Optimization Algorithms, John Schulman, et al. (2017)

PPO: Proximal Policy Optimization

PPO在TRPO基础上的改进

3. 自适应的KL惩罚项参数

$$L^{KLPEN}(\theta) = \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}\hat{A}_t - \beta \mathrm{KL}[\pi_{\theta_{\text{old}}}(\cdot\,|s_t)|\pi_\theta(\cdot\,|s_t)]\right]$$

动态调整β方法

- 计算KL值 $d = \hat{\mathbb{E}}_t\left[\mathrm{KL}[\pi_{\theta_{\text{old}}}(\cdot\,|s_t)|\pi_\theta(\cdot\,|s_t)]\right]$
  a) 如果$d < d_{\text{targ}}/1.5$，更新$\beta \leftarrow \beta/2$
  b) 如果$d > d_{\text{targ}}\times 1.5$，更新$\beta \leftarrow \beta\times 2$

注：这里1.5和2是经验参数，算法效能和它们并不是很敏感

Proximal Policy Optimization Algorithms, John Schulman, et al. (2017)

PPO实验对比

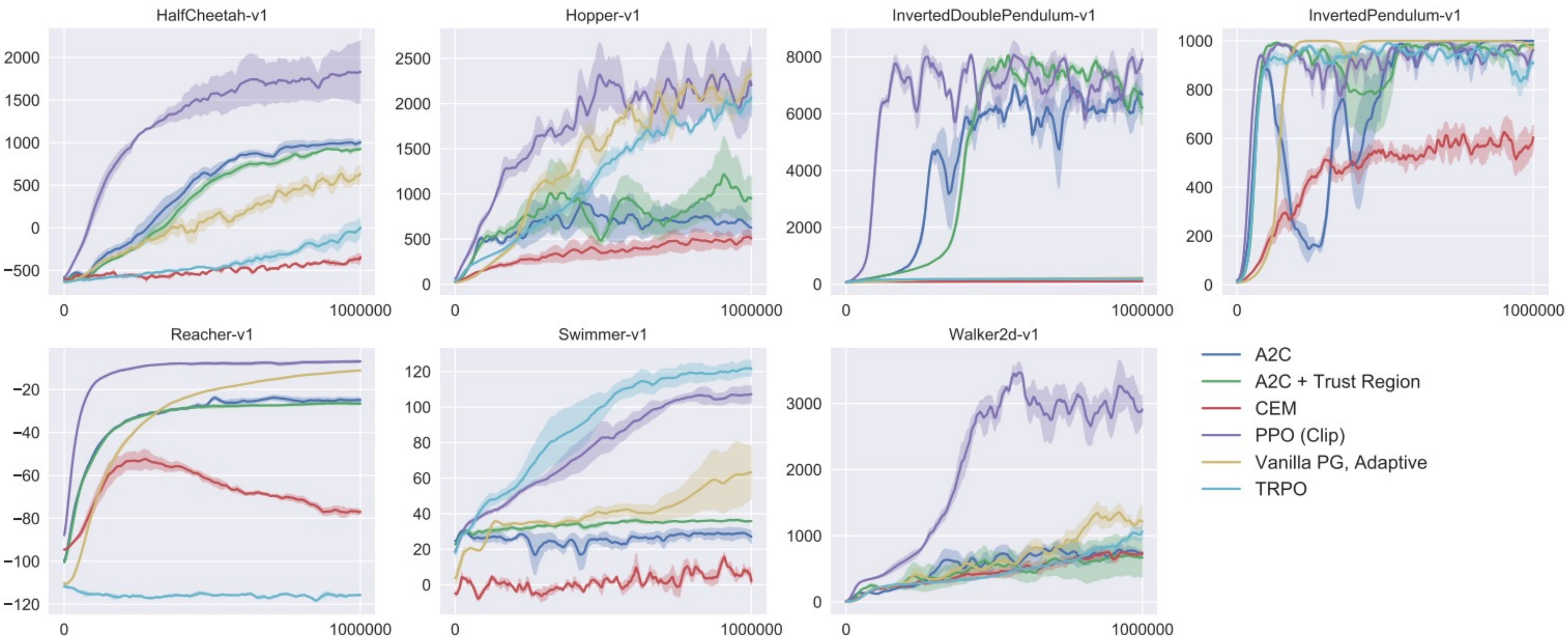
No clipping or penalty:	$L_t(\theta) = r_t(\theta)\hat{A}_t$
Clipping:	$L_t(\theta) = \min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta)), 1-\epsilon, 1+\epsilon)\hat{A}_t$
KL penalty (fixed or adaptive)	$L_t(\theta) = r_t(\theta)\hat{A}_t - \beta\,\text{KL}[\pi_{\theta_{\text{old}}}, \pi_\theta]$

- 7个连续控制的环境
- 3个random seed
- 每个算法跑100个episode，跑21遍，做平均值计算
- 最佳score归一化为1

algorithm	avg. normalized score
No clipping or penalty	-0.39
Clipping, $\epsilon = 0.1$	0.76
Clipping, $\epsilon = 0.2$	**0.82**
Clipping, $\epsilon = 0.3$	0.70
Adaptive KL $d_{\text{targ}} = 0.003$	0.68
Adaptive KL $d_{\text{targ}} = 0.01$	0.74
Adaptive KL $d_{\text{targ}} = 0.03$	0.71
Fixed KL, $\beta = 0.3$	0.62
Fixed KL, $\beta = 1.$	0.71
Fixed KL, $\beta = 3.$	0.72
Fixed KL, $\beta = 10.$	0.69

Proximal Policy Optimization Algorithms, John Schulman, et al. (2017)

PPO实验对比



Proximal Policy Optimization Algorithms, John Schulman, et al. (2017)