# PROXIMAL POLICY OPTIMISATION

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov
OpenAI

**{joschu, filip, prafulla, alec, oleg}@openai.com**

Андрей Ткачев
ВШЭ, 2019

# Введение: терминология

- $\pi_\theta(a, s)$ — политика, функция описывающая поведение нашего агента в зависимости от состояния среды и параметров агента
- $R(a, s)$ — вознаграждение агента за действие
- $V(s)$ — ценность состояния, потенциальный выигрыш
- $A(a, s)$ — полезность, описывает, насколько данное действие лучше остальных

# Подходы в RL: Q-learning

- Оцениваем полезность действия при текущем состоянии

$$Q^{\pi} = E[R_t]$$

- Выбираем оптимальное действие

$$\pi^*(s) = \underset{a}{\operatorname{argmax}} \; Q(s, a)$$

- Плохо работает для задач с непрерывным пространством действий

# Подходы в RL: actor-critic

- Не все действия одинаково хороши, даже если за них агент получает вознаграждение

$$A(a, s) = Q(s, a) - V(s)$$

- Будем смотреть на улучшение относительно максимально возможного в данном состоянии

# Подходы в RL: policy gradient

- Оптимизируем политику напрямую, считая, что градиент это

$$\hat{g} = \hat{E}_{\tau \sim \pi_{\theta_{old}}}[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t \mid s_t)\hat{A}_t]$$

- $\hat{A}_t$ это оценка полезности действия на шаге $t$

# Подходы в RL: policy gradient

- Сложно подобрать оптимальный learning-rate
- Если после последовательности действий мы получаем нулевой суммарный выигрыш, то веса не обновятся

# Подходы в RL: policy gradient

**Рис. 1:** проблемы с learning rate в оригинальном policy gradient

# Подходы в RL: trust regions

- Запрещаем политике обновляться слишком сильно

$$\underset{\theta}{\text{maximize}}\, \hat{E}_t \left[ \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)} \hat{A}_t \right]$$

$$\text{subject to } \hat{E}_t \left[ \text{KL}[\pi_\theta(\,.\mid s_t), \pi_{\theta_{old}}(\,.\mid s_t)] \right] < \delta$$

- В форме без ограничений

$$\underset{\theta}{\text{maximize}}\, \hat{E}_t \left[ \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)} \hat{A}_t - \beta \text{KL}[\pi_\theta(\,.\mid s_t), \pi_{\theta_{old}}(\,.\mid s_t)] \right]$$

# Подходы в RL: trust regions

- Для решения задачи с ограничениями применяются методы оптимизации второго порядка.
- К тому же не можем применять, напмример, dropout.
- В задаче без явных ограничений нужно подбирать оптимальный $\beta$

# PPO: Clipped Objective

- Для краткости $r_t(\theta) = \dfrac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)}$, тогда суррогатная часть функционала TRPO:

$$L^{CPI} = \hat{E}_t[r_t(\theta)\hat{A}_t]$$

- Но мы будем оптимизировать

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

# PPO: Clipped Objective

$$L^{CPI} = \hat{E}_t[r_t(\theta)\hat{A}_t]$$

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

- $clip(r_t(\theta),1-\epsilon,1+\epsilon)\hat{A}_t$ запрещает сильное изменение политики, а $min$ делает $L^{CLIP}$ нижней оценкой $L^{CPI}$

# PPO: Clipped Objective

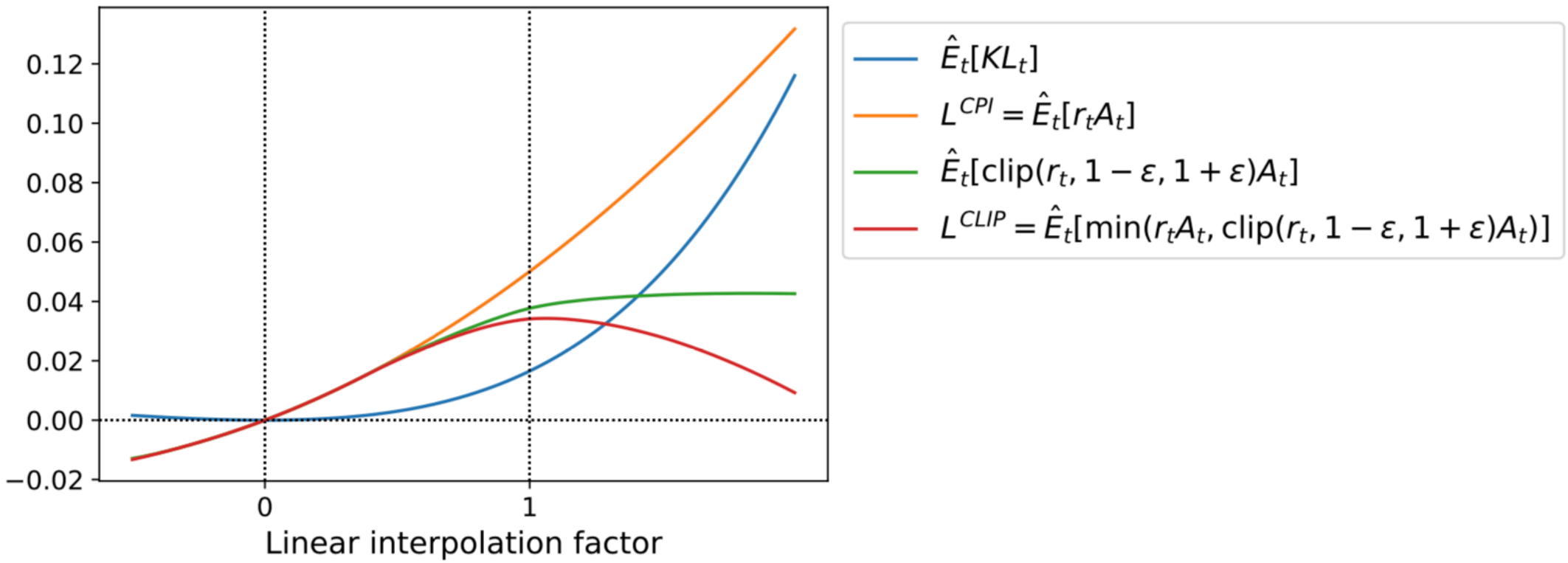


**Рис. 2:** поведение суррогатных функций при интерполяции параметров политики от $\theta_{old}$ до $\theta$ (одна итерация PPO)

# PPO: Clipped Objective

**Algorithm** PPO-Clip

1: Input: initial policy parameters $\theta_0$, initial value function parameters $\phi_0$
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Collect set of trajectories $\mathcal{D}_k = \{\tau_i\}$ by running policy $\pi_k = \pi(\theta_k)$ in the environment.
4: Compute rewards-to-go $\hat{R}_t$.
5: Compute advantage estimates, $\hat{A}_t$ (using any method of advantage estimation) based on the current value function $V_{\phi_k}$.
6: Update the policy by maximizing the PPO-Clip objective:

$$\theta_{k+1} = \arg\max_{\theta} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \min \left( \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} A^{\pi_{\theta_k}}(s_t, a_t), \quad g(\epsilon, A^{\pi_{\theta_k}}(s_t, a_t)) \right),$$

typically via stochastic gradient ascent with Adam.
7: Fit value function by regression on mean-squared error:

$$\phi_{k+1} = \arg\min_{\phi} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left( V_\phi(s_t) - \hat{R}_t \right)^2,$$

typically via some gradient descent algorithm.
8: **end for**

# PPO: Adaptive KL Penalty

- После очередной оптимизации политики сравним значение текущей KL-дивергенции с некоторой эталонной

- Увеличим или уменьшим $\beta$

$$L^{KLPEN}(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)} \hat{A}_t - \beta \mathrm{KL}[\pi_\theta(\,.\mid s_t), \pi_{\theta_{old}}(\,.\mid s_t)]$$

# PPO: Adaptive KL Penalty

**Algorithm** PPO with Adaptive KL Penalty

Input: initial policy parameters $\theta_0$, initial KL penalty $\beta_0$, target KL-divergence $\delta$
**for** $k = 0, 1, 2, \ldots$ **do**
  Collect set of partial trajectories $\mathcal{D}_k$ on policy $\pi_k = \pi(\theta_k)$
  Estimate advantages $\hat{A}_t^{\pi_k}$ using any advantage estimation algorithm
  Compute policy update

$$\theta_{k+1} = \arg\max_{\theta} \mathcal{L}_{\theta_k}(\theta) - \beta_k \bar{D}_{KL}(\theta||\theta_k)$$

  by taking $K$ steps of minibatch SGD (via Adam)
  **if** $\bar{D}_{KL}(\theta_{k+1}||\theta_k) \geq 1.5\delta$ **then**
    $\beta_{k+1} = 2\beta_k$
  **else if** $\bar{D}_{KL}(\theta_{k+1}||\theta_k) \leq \delta/1.5$ **then**
    $\beta_{k+1} = \beta_k/2$
  **end if**
**end for**

# Эксперименты

| 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 |

**Табл. 1:** Сравнение разных подходов в PPO. Средний счет на различных заданиях в окружении OpenAI Gym

# Эксперименты

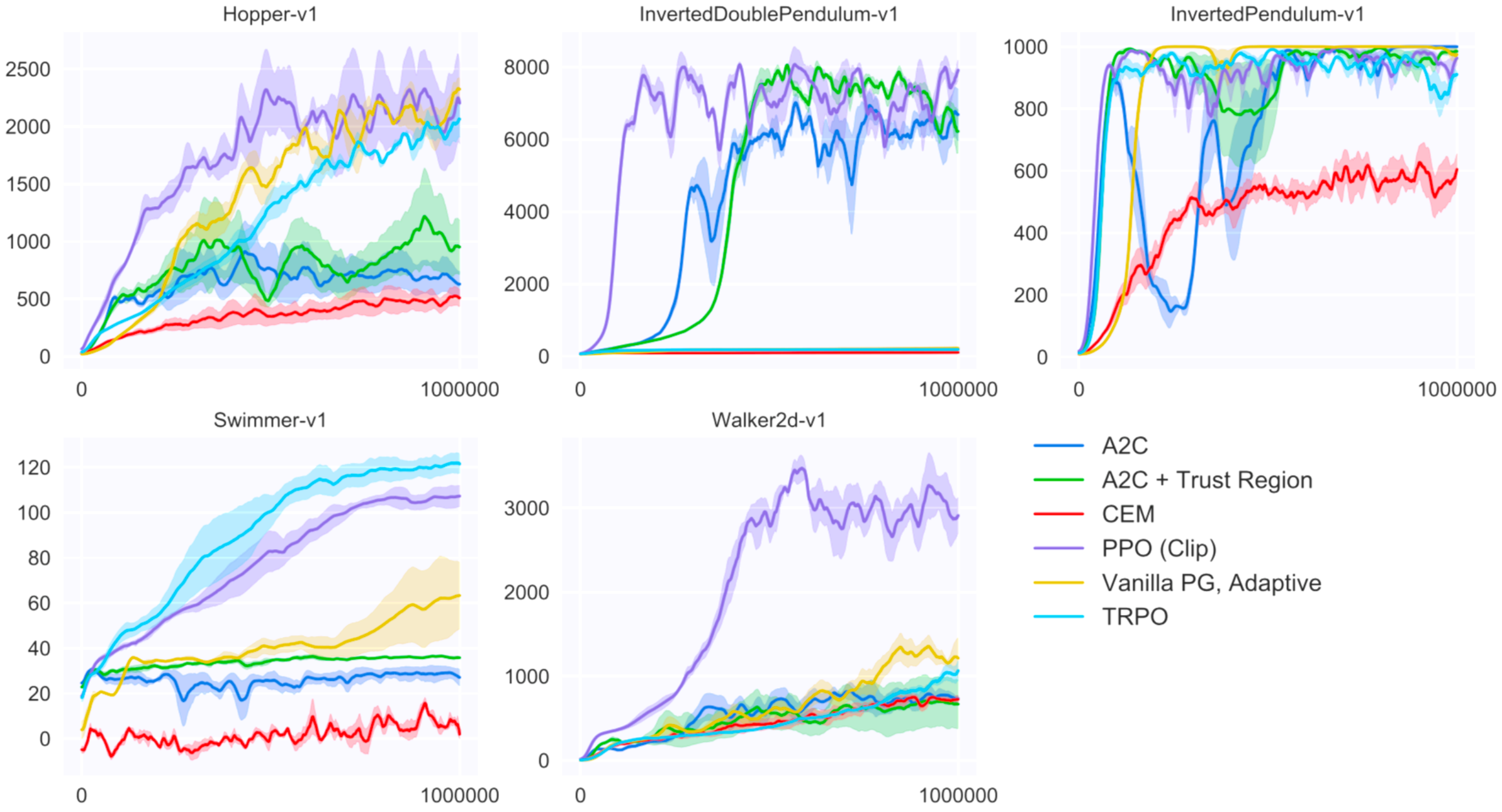


**Рис. 3:** Сравнение PPO с Clipped Objective с другими подходами на нескольких окружениях MuJoCo

# Эксперименты

| | A2C | ACER | PPO | Tie |
|---|---|---|---|---|
| (1) avg. episode reward over all of training | 1 | 18 | **30** | 0 |
| (2) avg. episode reward over last 100 episodes | 1 | **28** | 19 | 1 |

**Табл. 2:** Arcade Learning Environment. Количество пройденных игр (из 49)

# Выводы

- Методы PPO просты в реализации, выигрывают по производительности TRPO, а по стабильности и результатам сравнимы с TRPO.

# Вопросы

- В чем проблема Trust Regions Methods? Как эта проблема решается в PPO?
- Какие подходы проксимальной оптимизации предлагаются авторами? Кратко опишите один из них.
- Опишите общий алгоритм проксимальной оптимизации политики с clipped objective