# Robot Learning

GPS, TRPO, PPO

Prof. Songhwai Oh
ECE, SNU

Sergey Levine, Vladlen Koltun, "**Guided Policy Search**," in Proc. of the International Conference on Machine Learning (ICML), Jun. 2013.

# GUIDED POLICY SEARCH

# Expected Return

Expected return: $J(\theta) = \frac{1}{a_\Sigma} \mathbb{E}\left(\sum_{k=0}^{H} a_k r_k\right)$

- $H$, time horizon
- $r_k = r(x_k, u_k)$, reward at state $x_k$ and action $u_k$
- $u_k = \pi_\theta(u_k|x_k)$
- $a_k = \gamma^k$ and $a_\Sigma = (1-\gamma)^{-1}$ or $a_k = 1$ and $a_\Sigma = H$

Trajectory: $\tau = (x_{0:H}, u_{0:H})$

- $\tau \sim P_\theta(\tau) = P(\tau|\theta)$
- $r(\tau) = \sum_{k=0}^{H} a_k r_k$

$J(\theta) = \int_{\mathcal{T}} P_\theta(\tau) r(\tau) d\tau$

- $\mathcal{T}$ is a set of all trajectories

# Policy Gradient

$$J(\theta) = \int_{\mathcal{T}} P_\theta(\tau) r(\tau) d\tau$$

$$\begin{aligned}\nabla_\theta J(\theta) &= \int_{\mathcal{T}} \nabla_\theta P_\theta(\tau) r(\tau) d\tau \\ &= \int_{\mathcal{T}} P_\theta(\tau) \left(\nabla_\theta \log P_\theta(\tau) r(\tau)\right) d\tau \qquad \left(\because \frac{\partial \log P_\theta}{\partial \theta} = \frac{1}{P_\theta} \cdot \frac{\partial P_\theta}{\partial \theta}\right) \\ &= \mathbb{E}\left(\nabla_\theta \log P_\theta(\tau) r(\tau)\right)\end{aligned}$$

$$P_\theta(\tau) = P(x_0) \prod_{k=0}^{H} P(x_{k+1}|x_k, u_k) \pi_\theta(u_k|x_k)$$

$$\begin{aligned}\nabla \log P_\theta(\tau) &= \nabla_\theta \left( \log P(x_0) + \sum_{k=0}^{H} \log P(x_{k+1}|x_k, u_k) + \log \pi_\theta(u_k|x_k) \right) \\ &= \sum_{k=0}^{H} \nabla_\theta \log \pi_\theta(u_k|x_k)\end{aligned}$$

# Policy Gradient

$$\nabla \log P_\theta(\tau) = \sum_{k=0}^{H} \nabla_\theta \log \pi_\theta(u_k|x_k)$$

$$\begin{aligned}\nabla_\theta J(\theta) &= \mathbb{E}\left(\nabla_\theta \log P_\theta(\tau) r(\tau)\right) \\ &= \mathbb{E}\left(\left(\sum_{k=0}^{H} \nabla_\theta \log \pi_\theta(u_k|x_k)\right) r(\tau)\right)\end{aligned}$$

$$\nabla_\theta J(\theta) = \mathbb{E}\left(\nabla_\theta \log \pi_\theta(\tau) r(\tau)\right)$$

Useful fact

$$\begin{aligned}\int_{\mathcal{T}} P_\theta(\tau) \nabla_\theta \log P_\theta(\tau) d\tau &= \int_{\mathcal{T}} \nabla_\theta P_\theta(\tau) d\tau \\ &= \nabla_\theta \int_{\mathcal{T}} P_\theta(\tau) d\tau = 0\end{aligned}$$

Hence

$$\begin{aligned}\nabla_\theta J(\theta) &= \mathbb{E}\left(\nabla_\theta \log P_\theta(\tau)\left(r(\tau) - b\right)\right) \qquad \text{for any } b \in \mathbb{R} \\ &= \mathbb{E}\left(\nabla_\theta \log \pi_\theta(\tau)\left(r(\tau) - b\right)\right),\end{aligned}$$

where $b$ can be chosen to minimize variance.

# Importance Sampling

Importance sampling with proposal distribution $q$:

$$\mathbb{E}_p(f(X)) = \mathbb{E}_q\left(\frac{p(X)}{q(X)}f(X)\right) =: A$$

$$\begin{aligned}\mathbb{E}_p(f(X)) &= \int f(x)p(x)dx = \int f(x)p(x)\frac{q(x)}{q(x)}dx \\ &= \int \left(\frac{p(x)}{q(x)}f(x)\right)q(x)dx = \mathbb{E}_q\left(\frac{p(x)}{q(x)}f(x)\right)\end{aligned}$$

Hence

$$A \approx \frac{1}{Z}\sum_{i=1}^{m}\frac{p(x_i)}{q(x_i)}f(x_i)$$

- $m$, the number of samples from $q(\cdot)$
- $\frac{p(x_i)}{q(x_i)}$, importance weight
- $Z = \sum_i \frac{p(x_i)}{q(x_i)}$, normalization constant

# Learning using off-policy samples

Using importance sampling, we can approximate the expected return using the proposal distribution $q$

$$\mathbb{E}(J(\theta)) \approx \frac{1}{Z(\theta)} \sum_{i=1}^{m} \frac{\pi_\theta(\tau_i)}{q(\tau_i)} r(\tau_i)$$

Using causality (i.e., past rewards do not depend on future actions)

$$\mathbb{E}(J(\theta)) \approx \sum_{t=1}^{H} \frac{1}{Z_t(\theta)} \sum_{i=1}^{m} \frac{\pi_\theta(\tau_{i,1:t})}{q(\tau_{i,1:t})} r(x_t^i, u_t^i)$$

# Guided Policy Search

$$\Phi(\theta) = \sum_{t=1}^{H} \left( \frac{1}{Z_t(\theta)} \sum_{i=1}^{m} \frac{\pi(\tau_{i,1:t})}{q(\tau_{i,1:t})} r(x_t^i, u_t^i) + w_r \log Z_t(\theta) \right)$$

- $w_r \log Z_t(\theta)$, regularizer to reduce the variance
- With high $w_r$, it makes the policy follow samples more closely
- $\nabla_\theta \Phi(\theta)$ is derived in the paper

Iterative LQR (iLQR, a differential dynamic programming (DDP) method for trajectory optimization): repeat (1) and (2)

(1) Trajectory (nominal): $(\bar{x}_1, \bar{u}_1), (\bar{x}_2, \bar{u}_2), \ldots, (\bar{x}_H, \bar{u}_H)$

(2) LQR (optimal policy for a linearized model): $g(x_t) = \bar{u}_t + k_t + K_t(x_t - \bar{x}_t)$

Guided trajectory samples: $\pi_G(u_t|x_t) = \mathcal{N}\left(u_t; g(x_t), -Q_{uut}^{-1}\right)$

# Guided Policy Search

**Algorithm 1** Guided Policy Search

1: Generate DDP solutions $\pi_{\mathcal{G}_1}, \ldots, \pi_{\mathcal{G}_n}$
2: Sample $\zeta_1, \ldots, \zeta_m$ from $q(\zeta) = \frac{1}{n} \sum_i \pi_{\mathcal{G}_i}(\zeta)$
3: Initialize $\theta^\star \leftarrow \arg\max_\theta \sum_i \log \pi_{\theta^\star}(\zeta_i)$
4: Build initial sample set $\mathcal{S}$ from $\pi_{\mathcal{G}_1}, \ldots, \pi_{\mathcal{G}_n}, \pi_{\theta^\star}$
5: **for** iteration $k = 1$ to $K$ **do**
6: Choose current sample set $\mathcal{S}_k \subset \mathcal{S}$
7: Optimize $\theta_k \leftarrow \arg\max_\theta \Phi_{\mathcal{S}_k}(\theta)$
8: Append samples from $\pi_{\theta_k}$ to $\mathcal{S}_k$ and $\mathcal{S}$
9: Optionally generate adaptive guiding samples
10: Estimate the values of $\pi_{\theta_k}$ and $\pi_{\theta^\star}$ using $\mathcal{S}_k$
11: **if** $\pi_{\theta_k}$ is better than $\pi_{\theta^\star}$ **then**
12: Set $\theta^\star \leftarrow \theta_k$
13: Decrease $w_r$
14: **else**
15: Increase $w_r$
16: Optionally, resample from $\pi_{\theta^\star}$
17: **end if**
18: **end for**
19: Return the best policy $\pi_{\theta^\star}$

# Results

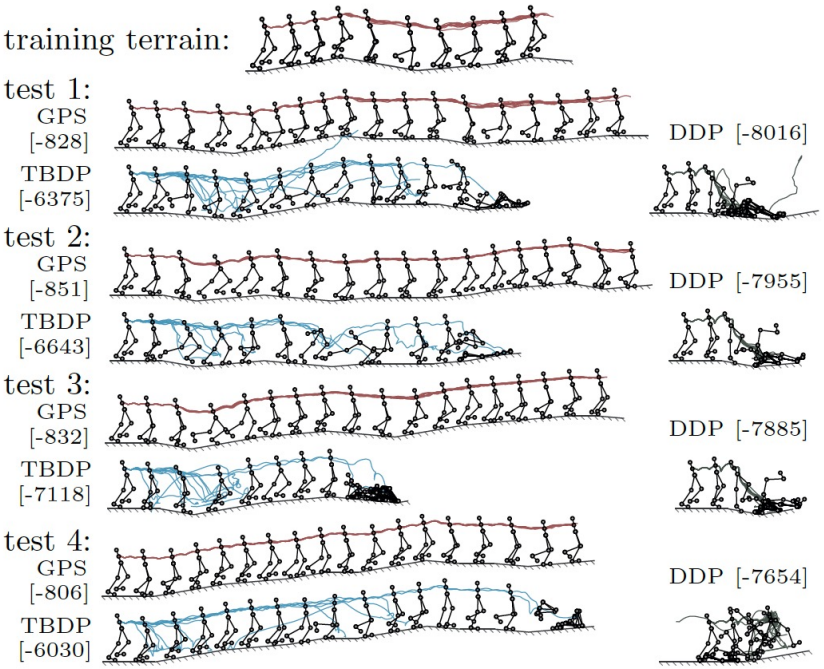


*Figure 4.* Rollouts of GPS, TBDP, and DDP on test terrains, shown as colored trajectories, with mean rewards in brackets. All DDP rollouts and most TBDP rollouts fall within a few steps, and all TBDP rollouts fall before reaching the end, while GPS generalizes successfully.

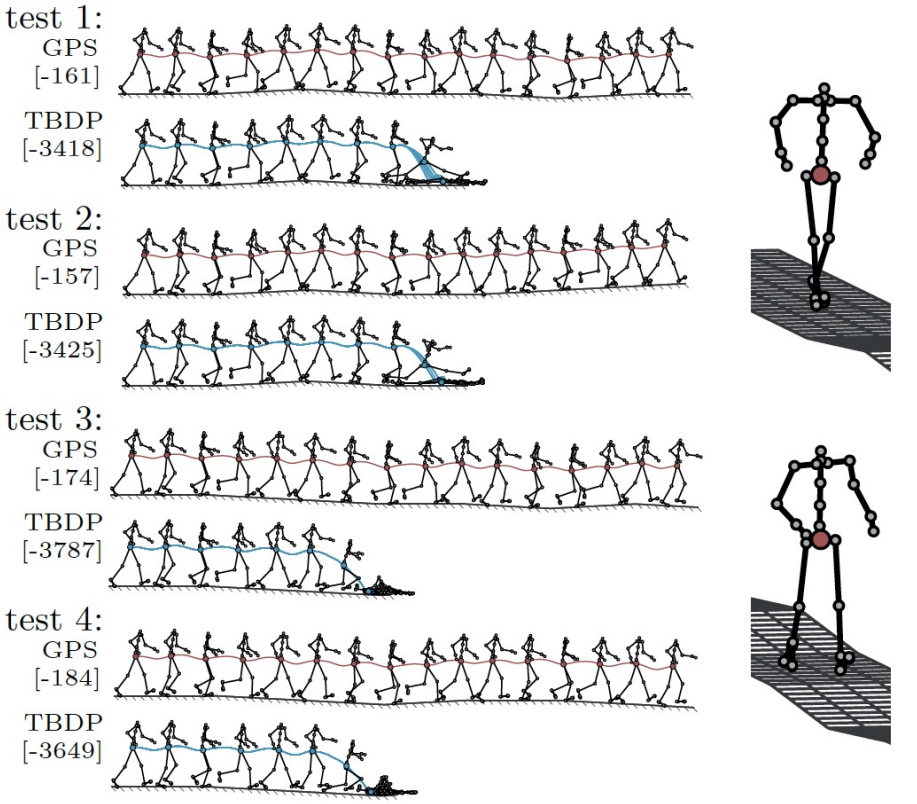


*Figure 5.* Humanoid running on test terrains, with mean rewards in brackets (left), along with illustrations of the 3D humanoid model (right). Our approach again successfully generalized to the test terrains, while the TBDP policy is unable to maintain balance.

* TBDP (Trajectory-Based Dynamic Programming)

J. Schulman, S. Levine, P. Abbeel, M. I. Jordan, and P. Moritz, "**Trust Region Policy Optimization**," in Proc. of the International Conference on Machine Learning (ICML), Jul. 2015.

Sham M. Kakade, and John Langford, "**Approximately optimal approximate reinforcement learning**," in Proc. of the International Conference on Machine Learning (ICML), 2002.

# TRUST REGION POLICY OPTIMIZATION (TRPO)

# Advantage Function

- Stochastic policy function, $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$
- $\eta(\pi) = \mathbb{E}_{s_0, a_0, \ldots} \left( \sum_{t=0}^{\infty} \gamma^t c(s_t) \right)$
  - $c(s_t)$, cost (or negative reward)
  - $s_0 \sim \rho_0(s_0)$, initial state distribution
  - $a_t \sim \pi(a_t | s_t)$
  - $s_{t+1} \sim P(s_{t+1} | s_t, a_t)$
- Advantage function:

$$\begin{aligned} A_\pi(s, a) &= Q_\pi(s, a) - V_\pi(s) \\ &= \mathbb{E}_{s' \sim P(s'|s,a)} \left( c(s) + \gamma V_\pi(s') - V_\pi(s) \right) \end{aligned}$$

# Policy Improvement Bound

- Two policies: $\pi_{\text{old}}$, $\pi_{\text{new}}$
- Trajectory $\tau = s_0, a_0, s_1, a_1, \ldots,$ where $\tau \sim \pi_{\text{new}}, P$

$$\mathbb{E}_\tau \left( \sum_{t=0}^{\infty} \gamma^t A_{\pi_{\text{old}}}(s_t, a_t) \right) = \mathbb{E}_\tau \left( \sum_{t=0}^{\infty} \gamma^t \left( c(s_t) + \gamma V_{\pi_{\text{old}}}(s_{t+1}) - V_{\pi_{\text{old}}}(s_t) \right) \right)$$

$$\begin{aligned}
&\text{(telescopic sum)} \qquad = \mathbb{E}_\tau \left( -V_{\pi_{\text{old}}}(s_0) + \sum_{t=0}^{\infty} \gamma^t c(s_t) \right) \\
&= -\mathbb{E}_{s_0} \left( V_{\pi_{\text{old}}}(s_0) \right) + \mathbb{E}_\tau \left( \sum_{t=0}^{\infty} \gamma^t c(s_t) \right) \\
&= -\eta(\pi_{\text{old}}) + \eta(\pi_{\text{new}})
\end{aligned}$$

Hence

$$\eta(\pi_{\text{new}}) = \eta(\pi_{\text{old}}) + \mathbb{E}_{\tau \sim \pi_{\text{new}}, P} \left( \sum_{t=0}^{\infty} \gamma^t A_{\pi_{\text{old}}}(s_t, a_t) \right)$$

With the discounted occupancy

$$\rho_\pi(s) = \sum_{t=0}^{\infty} \gamma^t P(s_t = s)$$

$$\eta(\pi_{\text{new}}) = \eta(\pi_{\text{old}}) + \underbrace{\sum_s \rho_{\pi_{\text{new}}}(s) \sum_a \pi_{\text{new}}(s,a) A_{\pi_{\text{old}}}(s,a)}_{\text{difficult to compute since } \rho_{\pi_{\text{new}}} \text{ depends on } \pi_{\text{new}}}$$

Local approximation:

$$L_{\pi_{\text{old}}}(\pi_{\text{new}}) := \eta(\pi_{\text{old}}) + \sum_s \rho_{\pi_{\text{old}}}(s) \sum_a \pi_{\text{new}}(s,a) A_{\pi_{\text{old}}}(s,a)$$

$L_\pi$ matches $\eta$ upto the first order

- $L_{\pi_{\theta_0}}(\pi_{\theta_0}) = \eta(\pi_{\theta_0})$
- $\nabla_\theta L_{\pi_{\theta_0}}(\pi_\theta)|_{\theta=\theta_0} = \nabla_\theta \eta(\pi_\theta)|_{\theta=\theta_0}$

# (Kakade & Langford, 2002)

Given

$$\pi' = \arg\min_{\pi} L_{\pi_{\text{old}}}(\pi)$$

$$\pi_{\text{new}}(a|s) = (1-\alpha)\pi_{\text{old}}(a|s) + \alpha\pi'(a|s),$$

we have

$$\eta(\pi_{\text{new}}) \leq L_{\pi_{\text{old}}}(\pi_{\text{new}}) + \frac{2\epsilon\gamma}{(1-\gamma)^2}\alpha^2,$$

where

$$\epsilon = \max_{s} \left| \mathbb{E}_{a\sim\pi'(a|s)} \left( A_{\pi_{\text{old}}}(s,a) \right) \right|$$

is the maximum advantage of $\pi'$ relative to $\pi_{\text{old}}$.

# Main Result

- Extension from a mixture (Kakade & Langford, 2002) to a general policy
- Total variation: $D_{\mathrm{TV}}(p\|q) = \frac{1}{2}\sum_i |p_i - q_i|$
  - In general, $D_{\mathrm{TV}}(\mu\|\nu) = \sup_{A\in\mathcal{B}} |\mu(A) - \nu(A)|$
- $D_{\mathrm{TV}}^{\max}(\pi, \tilde{\pi}) = \max_s D_{\mathrm{TV}}(\pi(\cdot, s)\|\tilde{\pi}(\cdot|s))$
- Let $\alpha = D_{\mathrm{TV}}^{\max}(\pi_{\mathrm{old}}, \pi_{\mathrm{new}})$
- Then

$$\eta(\pi_{\mathrm{new}}) \leq L_{\pi_{\mathrm{old}}}(\pi_{\mathrm{new}}) + \frac{2\epsilon\gamma}{(1-\gamma)^2}\alpha^2$$

- Since $D_{\mathrm{TV}}(p\|q)^2 \leq D_{\mathrm{KL}}(p\|q)$, (where $D_{\mathrm{KL}}(p\|q) = -\sum p_i \log \frac{q_i}{p_i}$) if $D_{\mathrm{KL}}^{\max}(\pi, \tilde{\pi}) = \max_s D_{\mathrm{KL}}(\pi(\cdot|s)\|\tilde{\pi}(\cdot|s))$, then

$$\eta(\pi_{\mathrm{new}}) \leq L_{\pi_{\mathrm{old}}}(\pi_{\mathrm{new}}) + \frac{2\epsilon\gamma}{(1-\gamma)^2} D_{\mathrm{KL}}^{\max}(\pi_{\mathrm{new}}, \pi_{\mathrm{old}})$$

- For a sequence of policies $(\pi_0, \pi_1, \pi_2, \ldots)$ generated by Algorithm 1, we have $\eta(\pi_0) \geq \eta(\pi_1) \geq \eta_(\pi_2) \geq \ldots$

**Algorithm 1** Approximate policy iteration algorithm guaranteeing non-increasing expected cost $\eta$

Initialize $\pi_0$.

**for** $i = 0, 1, 2, \ldots$ until convergence **do**

Compute all advantage values $A_{\pi_i}(s, a)$.

Solve the constrained optimization problem

$$\pi_{i+1} = \arg\min_{\pi} \left[ L_{\pi_i}(\pi) + \left( \frac{2\epsilon\gamma}{(1-\gamma)^2} \right) D_{\mathrm{KL}}^{\max}(\pi_i, \pi) \right]$$

where $\epsilon = \max_s \max_a |A_\pi(s, a)|$

and $L_{\pi_i}(\pi) = \eta(\pi_i) + \sum_s \rho_{\pi_i}(s) \sum_a \pi(a|s) A_{\pi_i}(s, a)$

**end for**

# Practical Algorithm

- Optimization problem:

$$\min_{\theta} L_{\theta_{\text{old}}}(\theta) + C D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta)$$

- Trust region constraint

$$\min_{\theta} L_{\theta_{\text{old}}}(\theta) \qquad \text{subject to } D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta) \leq \delta$$

- $D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta)$ requires to check all states. Hence, we use an approximation, the average KL divergence, $\bar{D}_{\text{KL}}$

$$\min_{\theta} L_{\theta_{\text{old}}}(\theta) \qquad \text{subject to } \bar{D}_{\text{KL}}^{\rho_{\theta_{\text{old}}}}(\theta_{\text{old}}, \theta) \leq \delta,$$

where $\bar{D}_{\text{KL}}^{\rho}(\theta_1, \theta_2) = \mathbb{E}_{s \sim \rho}\left(D_{\text{KL}}(\pi_{\theta_1}(\cdot|s) \| \pi_{\theta_2}(\cdot|s))\right)$.

- Important sampling

$$\min_{\theta} \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}},a\sim q}\left(\frac{\pi_\theta(a|s)}{q(a|s)}Q_{\theta_{\text{old}}}(s,a)\right)$$

$$\text{subject to } \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}}\left(D_{\text{KL}}(\pi_{\theta_{\text{old}}}(\cdot|s)\|\pi_\theta(\cdot|s))\right) \leq \delta$$

- $q(a|s)$ is a proposal distribution, e.g., $\pi_{\theta_{\text{old}}}$

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "**Proximal policy optimization algorithms**," arXiv preprint arXiv:1707.06347, 2017

# PROXIMAL POLICY OPTIMIZATION (PPO)

# PPO

- TRPO

$$\max_{\theta} \hat{\mathbb{E}}_t \left( \frac{\pi_\theta(a_t|s_t)}{q(a_t|s_t)} \hat{A}_t \right)$$

$$\text{subject to } \hat{\mathbb{E}}_t \left( D_{\mathrm{KL}}(\pi_{\theta_{\text{old}}}(\cdot|s_t) \| \pi_\theta(\cdot|s_t)) \right) \leq \delta$$

  - $\hat{\mathbb{E}}_t$: empirical average over a batch of samples

- Let $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$. Then TRPO maximizes

$$L^{\mathrm{CPI}}(\theta) = \hat{\mathbb{E}}_t \left( r_t(\theta) \hat{A}_t \right)$$

  - CPI: conservative policy iteration
  - Without constraint, the maximization of $L^{\mathrm{CPI}}$ can lead to a large policy update

- PPO

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

  - clip makes $r_t$ inside of $[1-\epsilon, 1+\epsilon]$
  - Ignores changes in probability ratio when it improves the objective
  - Include changes when it makes the objective worse

# $L^{CLIP}$

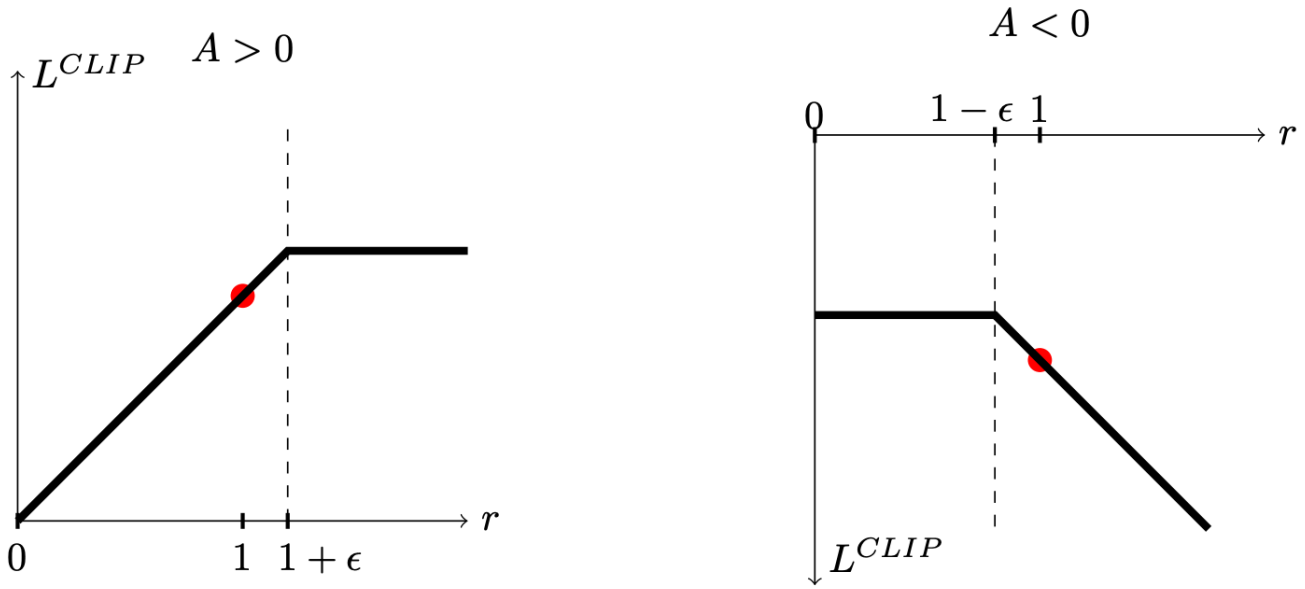


Figure 1: Plots showing one term (i.e., a single timestep) of the surrogate function $L^{CLIP}$ as a function of the probability ratio $r$, for positive advantages (left) and negative advantages (right). The red circle on each plot shows the starting point for the optimization, i.e., $r = 1$. Note that $L^{CLIP}$ sums many of these terms.