Proximal Policy Optimization

Deep Reinforcement Learning

Sai Sampath Kedari
sampath@umich.edu

Contents

1 Why PPO Is Needed **2**
1.1 Fixed On-Policy Data . . . 2
1.2 First-Order Actor–Critic Update . . . 2
1.3 What the First-Order Update Does Not Control . . . 3
1.4 From NPG to TRPO . . . 3
1.5 The Question PPO Answers . . . 4

2 PPO with an Adaptive KL Penalty **4**
2.1 From Constraint to Penalty . . . 4
2.2 Why the Penalty Coefficient Is Adapted . . . 5
2.3 Empirical Objective on a Fixed Rollout Batch . . . 6
2.4 PPO-Penalty Algorithm . . . 6

3 PPO with the Clipped Surrogate Objective **7**
3.1 Interpreting the Advantage and Likelihood Ratio . . . 8
3.2 The Unclipped Surrogate Objective . . . 9
3.3 Clipping the Likelihood Ratio . . . 9
3.4 Why the Minimum Is Required . . . 10
3.5 Case 1: Positive Advantage . . . 10
3.6 Case 2: Negative Advantage . . . 11
3.7 Shape of a Single Clipped Contribution . . . 11
3.8 Pessimistic Surrogate and First-Order Agreement . . . 12
3.9 Empirical Minibatch Objective . . . 13
3.10 Complete PPO Actor-Critic Objective . . . 13
3.11 PPO-Clip Algorithm . . . 14
3.12 What Clipping Does and Does Not Guarantee . . . 14

1 Why PPO Is Needed

Readers already familiar with A2C, NPG, and TRPO may skip directly to Section 2.

This section explains the problem that led to PPO. After estimating a policy-gradient direction, we must decide how far the policy should move. A2C uses a learning rate, NPG measures local movement in policy space, and TRPO checks the proposed policy before accepting it. PPO keeps the idea of limiting policy movement while returning to ordinary first-order optimization.

1.1 Fixed On-Policy Data

Freeze the current policy as

$$\pi_{\text{old}} = \pi_{\theta_{\text{old}}},$$

and run it in N parallel environments for T steps. This produces $B = TN$ raw transitions

$$Z_i^{\text{raw}} = \left(S_i, A_i, R_i, S_i^+, D_i^{\text{term}}, D_i^{\text{trunc}}\right).$$

Using the frozen policy π_{old} and the current value approximation $\widehat{V}(\cdot; w_{\text{old}})$, compute the advantage estimates, value targets, and old action probabilities. The resulting training batch is

$$\boxed{\mathcal{B}_{\text{old}} = \left\{ S_i,\ A_i,\ \widehat{A}_i^{\text{GAE},\pi_{\text{old}}},\ \widehat{V}_i^{\text{target}},\ \log \pi_{\text{old}}(A_i \mid S_i) \right\}_{i=1}^{B}.}$$

Every quantity in this batch is fixed during policy optimization. This is a *rollout buffer*, not a replay buffer: it contains data from one policy, is used for one optimization round, and is then discarded.

1.2 First-Order Actor–Critic Update

At the old parameters, the policy-gradient theorem gives

$$\begin{aligned} g_{\text{old}} &:= \nabla_\theta J(\theta)|_{\theta=\theta_{\text{old}}} \\ &= \frac{1}{1-\gamma} \mathbb{E}_{S \sim d^{\pi_{\text{old}}}, \rho_0} \left[\mathbb{E}_{A \sim \pi_{\text{old}}(\cdot|S)} \left[Q^{\pi_{\text{old}}}(S, A)\ \nabla_\theta \log \pi_\theta(A \mid S)|_{\theta=\theta_{\text{old}}} \right] \right]. \end{aligned}$$

Subtracting the state-value baseline gives the equivalent expression

$$\boxed{g_{\text{old}} = \frac{1}{1-\gamma} \mathbb{E}_{S \sim d^{\pi_{\text{old}}}, \rho_0} \left[\mathbb{E}_{A \sim \pi_{\text{old}}(\cdot|S)} \left[A^{\pi_{\text{old}}}(S, A)\ \nabla_\theta \log \pi_\theta(A \mid S)|_{\theta=\theta_{\text{old}}} \right] \right].}$$

The baseline does not change the expected gradient; it reduces the variance of its sample estimator. Using GAE gives

$$\boxed{\widehat{g}_{\text{old}} = \frac{1}{B} \sum_{i=1}^{B} \widehat{A}_i^{\text{GAE},\pi_{\text{old}}}\ \nabla_\theta \log \pi_\theta(A_i \mid S_i)|_{\theta=\theta_{\text{old}}}.}$$

A first-order actor–critic method then performs

$$\boxed{\theta_{\text{new}} = \theta_{\text{old}} + \alpha \widehat{g}_{\text{old}},}$$

or applies an optimizer such as Adam.

The important limitation is simple: GAE improves the estimate of the *direction*, but it does not determine how far the policy may safely move in that direction.

1.3 What the First-Order Update Does Not Control

The learning rate α measures movement in parameter space, not movement of the policy distribution. Its effect depends on the reward scale, advantage scale, current network parameters, and sensitivity of the action distribution.

For example, consider a Gaussian policy with fixed standard deviation:

$$D_{\mathrm{KL}}\left(\mathcal{N}(\mu,\sigma^2)\|\mathcal{N}(\mu+\Delta\mu,\sigma^2)\right)=\frac{(\Delta\mu)^2}{2\sigma^2}.$$

The same change $\Delta\mu$ produces a small policy change when σ is large and a much larger policy change when σ is small. Therefore, equal parameter steps need not represent equal policy changes.

A first-order update also does not check the policy it produces. The gradient is only a local direction estimated from finite data. If the step is too large, the new policy may perform worse, but ordinary actor–critic training contains no explicit test that detects this before accepting the update.

The problem becomes more important when the same rollout batch is optimized repeatedly. Its empirical surrogate objective is

$$\boxed{\widehat{\mathcal{L}}^{\mathrm{sur}}_{\pi_{\mathrm{old}}}(\theta)=\frac{1}{B}\sum_{i=1}^{B} r_i(\theta)\widehat{A}_i^{\mathrm{GAE},\pi_{\mathrm{old}}},\qquad r_i(\theta)=\frac{\pi_\theta(A_i\mid S_i)}{\pi_{\mathrm{old}}(A_i\mid S_i)}.}$$

At $\theta=\theta_{\mathrm{old}}$,

$$r_i(\theta_{\mathrm{old}})=1,\qquad \nabla_\theta\widehat{\mathcal{L}}^{\mathrm{sur}}_{\pi_{\mathrm{old}}}(\theta)\Big|_{\theta=\theta_{\mathrm{old}}}=\widehat{g}_{\mathrm{old}}.$$

The ratio corrects the probability of the sampled action, but the states and advantage estimates still come from π_{old}. Therefore, the surrogate is most reliable near θ_{old}. After many updates, the candidate policy may be far from the policy that generated the data, while optimization continues to use the same old states and old advantage estimates.

Thus, repeated optimization requires some mechanism that limits movement away from π_{old}.

1.4 From NPG to TRPO

Natural policy gradient replaces Euclidean parameter distance with the local KL geometry of the policy. Using the empirical Fisher matrix $\widehat{F}_{\mathrm{old}}$, NPG solves

$$\boxed{\begin{aligned}\Delta\theta_{\mathrm{NPG}}=\arg\max_{\Delta\theta}\quad & \widehat{g}_{\mathrm{old}}^{\top}\Delta\theta\\ \text{subject to}\quad & \frac{1}{2}\Delta\theta^{\top}\widehat{F}_{\mathrm{old}}\Delta\theta\le\delta.\end{aligned}}$$

Its solution is

$$\boxed{\Delta\theta_{\mathrm{NPG}}=\sqrt{\frac{2\delta}{\widehat{g}_{\mathrm{old}}^{\top}\widehat{F}_{\mathrm{old}}^{-1}\widehat{g}_{\mathrm{old}}}}\,\widehat{F}_{\mathrm{old}}^{-1}\widehat{g}_{\mathrm{old}}.}$$

The radius δ now has a policy-space interpretation: it approximately limits how much the action distribution may change. However, NPG constrains only the local quadratic approximation

$$\widehat{\bar{D}_{\mathrm{KL}}}\left(\pi_{\mathrm{old}} \| \pi_{\theta_{\mathrm{old}}+\Delta\theta}\right) \approx \frac{1}{2}\Delta\theta^{\top}\widehat{F}_{\mathrm{old}}\Delta\theta.$$

TRPO adds an explicit check by considering

$$\begin{array}{rl} \max\limits_{\theta} & \mathbb{E}_{S\sim d^{\pi_{\mathrm{old}},\rho_0}}\left[\mathbb{E}_{A\sim\pi_{\mathrm{old}}(\cdot|S)}\left[\dfrac{\pi_\theta(A\mid S)}{\pi_{\mathrm{old}}(A\mid S)}A^{\pi_{\mathrm{old}}}(S,A)\right]\right] \\ \text{subject to} & \mathbb{E}_{S\sim d^{\pi_{\mathrm{old}},\rho_0}}\left[D_{\mathrm{KL}}\left(\pi_{\mathrm{old}}(\cdot\mid S)\|\pi_\theta(\cdot\mid S)\right)\right] \le \delta. \end{array}$$

In practice, TRPO computes an approximate NPG direction using Fisher-vector products and conjugate gradient. It then tests

$$\theta_j = \theta_{\mathrm{old}} + \beta^j \Delta\theta_{\mathrm{NPG}}, \qquad 0<\beta<1,$$

and accepts the first candidate satisfying

$$\widehat{\mathcal{L}}^{\mathrm{sur}}_{\pi_{\mathrm{old}}}(\theta_j) > \widehat{\mathcal{L}}^{\mathrm{sur}}_{\pi_{\mathrm{old}}}(\theta_{\mathrm{old}}), \qquad \widehat{\bar{D}_{\mathrm{KL}}}\left(\pi_{\mathrm{old}}\|\pi_{\theta_j}\right) \le \delta.$$

Therefore,

A2C:	estimate a gradient and choose a learning rate,
NPG:	measure the step using local KL geometry,
TRPO:	check the empirical surrogate and empirical KL.

1.5 The Question PPO Answers

TRPO controls policy movement, but one update requires Fisher-vector products, conjugate gradient, trust-region scaling, and backtracking line search. These computations use the whole rollout batch to produce one accepted policy step.

PPO asks whether the same basic control can be obtained with a simpler first-order objective, allowing several minibatch updates and several epochs over the rollout batch.

PPO provides two answers:

- **PPO-Penalty** subtracts an adaptive average-KL penalty from the surrogate objective.
- **PPO-Clip** modifies each likelihood-ratio contribution so that excessively large favorable changes are no longer rewarded.

Both variants remove TRPO's second-order optimization machinery and permit ordinary optimizers such as Adam. In exchange, they do not impose TRPO's explicit empirical KL constraint.

2 PPO with an Adaptive KL Penalty

2.1 From Constraint to Penalty

TRPO maximizes the old-policy surrogate subject to a hard constraint on the average KL divergence. PPO-Penalty removes this feasibility constraint and places the same average KL divergence

directly in the objective:

$$\boxed{\begin{aligned}\mathcal{L}^{\text{KL-PEN}}_{\pi_{\text{old}}}(\theta;\beta_{\text{KL}}) = \mathbb{E}_{S\sim d^{\pi_{\text{old}}},\rho_0}\left[\mathbb{E}_{A\sim\pi_{\text{old}}(\cdot|S)}\left[\frac{\pi_\theta(A\mid S)}{\pi_{\text{old}}(A\mid S)}A^{\pi_{\text{old}}}(S,A)\right]\right] \\ -\beta_{\text{KL}}\mathbb{E}_{S\sim d^{\pi_{\text{old}}},\rho_0}\left[D_{\text{KL}}\left(\pi_{\text{old}}(\cdot\mid S)\|\pi_\theta(\cdot\mid S)\right)\right].\end{aligned}}$$

The update is obtained by maximizing

$$\boxed{\theta^\star = \arg\max_\theta \mathcal{L}^{\text{KL-PEN}}_{\pi_{\text{old}}}(\theta;\beta_{\text{KL}}).}$$

This is a different optimization problem from TRPO.

In TRPO, any candidate violating the KL constraint is infeasible. In PPO-Penalty, every candidate is feasible, but candidates farther from the old policy receive a larger penalty.

The coefficient β_{KL} controls the relative cost of policy movement. A larger β_{KL} discourages movement more strongly, whereas a smaller β_{KL} allows the surrogate objective to dominate.

However, β_{KL} does not specify a maximum permitted KL divergence. Consequently, no fixed value of β_{KL} guarantees that the resulting policy remains inside a prescribed KL radius.

2.2 Why the Penalty Coefficient Is Adapted

A single fixed penalty coefficient is unreliable across different problems and different stages of training. A value that produces an appropriate update early in training may later produce an update that is either too small or too large.

PPO-Penalty therefore introduces a fixed target average KL divergence

$$d_{\text{target}} > 0.$$

After completing the policy update, let

$$d = \mathbb{E}_{S\sim d^{\pi_{\text{old}}},\rho_0}\left[D_{\text{KL}}\left(\pi_{\text{old}}(\cdot\mid S)\|\pi_{\theta_{\text{new}}}(\cdot\mid S)\right)\right]$$

denote the achieved average KL divergence.

The penalty coefficient for the next policy update is

$$\boxed{\beta^{\text{next}}_{\text{KL}} = \begin{cases}\dfrac{\beta_{\text{KL}}}{2}, & d < \dfrac{d_{\text{target}}}{1.5}, \\ 2\beta_{\text{KL}}, & d > 1.5 d_{\text{target}}, \\ \beta_{\text{KL}}, & \text{otherwise.}\end{cases}}$$

If the policy moved substantially less than the target, the penalty is weakened. If it moved substantially more than the target, the penalty is strengthened. The interval

$$\left[\frac{d_{\text{target}}}{1.5},\ 1.5 d_{\text{target}}\right]$$

forms a dead zone in which the coefficient remains unchanged.

The completed policy update is always accepted. There is no backtracking and no rejection. The measured divergence changes β_{KL} only for the next policy update.

Thus, d_{target} is a target used by the adaptation rule, not a hard constraint on the current update.

2.3 Empirical Objective on a Fixed Rollout Batch

Consider a fixed rollout batch generated by π_{old}:

$$\mathcal{B}_{\mathrm{old}} = \left\{ S_i,\, A_i,\, \widehat{A}_i^{\pi_{\mathrm{old}}},\, \pi_{\mathrm{old}}(A_i \mid S_i),\, \pi_{\mathrm{old}}(\cdot \mid S_i) \right\}_{i=1}^{B}, \qquad B = TN.$$

For each sample, define the likelihood ratio

$$\boxed{r_i(\theta) = \frac{\pi_\theta(A_i \mid S_i)}{\pi_{\mathrm{old}}(A_i \mid S_i)}.}$$

Let M denote the minibatch size, and let

$$\mathcal{M} \subseteq \{1, \ldots, B\}, \qquad |\mathcal{M}| = M,$$

denote the index set of one minibatch. The empirical PPO-Penalty objective is

$$\boxed{\widehat{\mathcal{L}}_{\mathcal{M}}^{\text{KL-PEN}}(\theta; \beta_{\mathrm{KL}}) = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \left[r_i(\theta) \widehat{A}_i^{\pi_{\mathrm{old}}} - \beta_{\mathrm{KL}} D_{\mathrm{KL}} \left(\pi_{\mathrm{old}}(\cdot \mid S_i) \| \pi_\theta(\cdot \mid S_i) \right) \right].}$$

During all minibatch updates,

$$\boxed{\mathcal{B}_{\mathrm{old}}, \qquad \pi_{\mathrm{old}}, \qquad \widehat{A}_i^{\pi_{\mathrm{old}}}, \qquad \beta_{\mathrm{KL}} \quad \text{remain fixed.}}$$

Only the candidate parameters θ change. Therefore, even after several minibatch updates, every likelihood ratio is still computed relative to the policy that generated the rollout batch.

After the final epoch, the achieved empirical average KL divergence is

$$\boxed{d = \frac{1}{B} \sum_{i=1}^{B} D_{\mathrm{KL}} \left(\pi_{\mathrm{old}}(\cdot \mid S_i) \| \pi_{\theta_{\mathrm{new}}}(\cdot \mid S_i) \right).}$$

2.4 PPO-Penalty Algorithm

Given initial policy parameters θ_0, an initial penalty coefficient β_{KL}, target divergence d_{target}, number of epochs E, and minibatch size M, repeat the following for $k = 0, 1, 2, \ldots$:

1. Freeze the old policy:

$$\pi_{\mathrm{old}} \leftarrow \pi_{\theta_k}.$$

2. Collect a rollout batch $\mathcal{B}_{\text{old}}$ of $B = TN$ transitions under π_{old}. Compute the fixed advantage estimates $\widehat{A}_i^{\pi_{\text{old}}}$ and store the old action probabilities and old policy distributions.

3. Initialize the candidate parameters:
$$\theta \leftarrow \theta_k.$$

4. For each epoch $e = 1, \ldots, E$, shuffle $\mathcal{B}_{\text{old}}$ and divide it into minibatches of size M.

5. For each minibatch $\mathcal{M}$, compute
$$\widehat{\mathcal{L}}_{\mathcal{M}}^{\text{KL-PEN}}(\theta; \beta_{\text{KL}})$$
and apply one first-order ascent step to θ using an optimizer such as Adam.

6. Accept the parameters obtained after the final epoch:
$$\theta_{k+1} \leftarrow \theta.$$

7. Compute the full-batch average KL divergence:
$$d \leftarrow \frac{1}{B} \sum_{i=1}^{B} D_{\text{KL}} \left(\pi_{\text{old}}(\cdot \mid S_i) \| \pi_{\theta_{k+1}}(\cdot \mid S_i) \right).$$

8. Adapt the penalty coefficient for the next policy update:
$$\beta_{\text{KL}} \leftarrow \begin{cases} \beta_{\text{KL}}/2, & d < d_{\text{target}}/1.5, \\ 2\beta_{\text{KL}}, & d > 1.5 d_{\text{target}}, \\ \beta_{\text{KL}}, & \text{otherwise.} \end{cases}$$

The value-function update is independent of the adaptive-KL mechanism and is omitted here. It will be added when constructing the complete PPO actor-critic objective.

Compared with TRPO, PPO-Penalty removes the Fisher-vector products, conjugate-gradient solve, trust-region scaling, and backtracking line search. The policy can therefore be trained using ordinary first-order minibatch updates for several epochs over the same rollout batch.

The price is that policy movement is controlled only through feedback on β_{KL}. The target d_{target} encourages a particular average KL divergence, but does not enforce it.

The next section develops PPO-Clip, which modifies the likelihood-ratio objective directly rather than penalizing the KL divergence.

3 PPO with the Clipped Surrogate Objective

PPO-Clip uses the same fixed rollout batch generated by $\pi_{\text{old}} = \pi_{\theta_{\text{old}}}$, but it does not add a KL-divergence penalty. Instead, it modifies each sample's likelihood-ratio contribution to the old-policy surrogate. This is a separate construction from PPO-Penalty; it is not obtained by algebraically expanding the TRPO KL constraint.

3.1 Interpreting the Advantage and Likelihood Ratio

For a state-action pair (S, A), the old-policy advantage is

$$A^{\pi_{\text{old}}}(S, A) = Q^{\pi_{\text{old}}}(S, A) - V^{\pi_{\text{old}}}(S).$$

A positive advantage means that the action is better than the old policy's average behavior at that state, so its probability should increase. A negative advantage means that it is worse than the old policy's average behavior, so its probability should decrease.

This comparison is relative, not absolute. An action can receive positive reward and still have negative advantage if the old policy usually obtains an even larger return from that state.

For the sampled pair (S_i, A_i), define the likelihood ratio

$$\boxed{r_i(\theta) = \frac{\pi_\theta(A_i \mid S_i)}{\pi_{\text{old}}(A_i \mid S_i)}.}$$

This ratio compares the candidate policy with the policy that generated the sample. It is never negative:

$$\begin{aligned} r_i(\theta) = 1 \quad &\Longrightarrow \quad \text{the sampled action has the same probability,} \\ r_i(\theta) > 1 \quad &\Longrightarrow \quad \text{the candidate makes the sampled action more likely,} \\ 0 < r_i(\theta) < 1 \quad &\Longrightarrow \quad \text{the candidate makes the sampled action less likely.} \end{aligned}$$

For example, ratios 1.2 and 0.8 mean 20% more and 20% less probability, respectively. For a continuous-action policy, the ratio compares probability densities evaluated at the sampled action.

The contribution of sample i to the unclipped surrogate is

$$\ell_i^{\text{CPI}}(\theta) = r_i(\theta)\widehat{A}_i^{\pi_{\text{old}}}.$$

Viewed as a function of the ratio,

$$\frac{\partial \ell_i^{\text{CPI}}}{\partial r_i} = \widehat{A}_i^{\pi_{\text{old}}}.$$

Therefore, maximizing this contribution produces four possible cases:

Advantage	Change in ratio	Meaning
$\widehat{A}_i^{\pi_{\text{old}}} > 0$	r_i increases	reinforce a better action
$\widehat{A}_i^{\pi_{\text{old}}} > 0$	r_i decreases	suppress a better action
$\widehat{A}_i^{\pi_{\text{old}}} < 0$	r_i decreases	suppress a worse action
$\widehat{A}_i^{\pi_{\text{old}}} < 0$	r_i increases	reinforce a worse action

The first and third rows improve the old-policy surrogate. The second and fourth rows make it worse.

3.2 The Unclipped Surrogate Objective

The population form of the unclipped surrogate is

$$\boxed{\mathcal{L}^{\mathrm{CPI}}_{\pi_{\mathrm{old}}}(\theta) = \mathbb{E}_{S\sim d^{\pi_{\mathrm{old}}},\rho_0}\left[\mathbb{E}_{A\sim\pi_{\mathrm{old}}(\cdot|S)}\left[r_\theta(S,A)A^{\pi_{\mathrm{old}}}(S,A)\right]\right],}$$

where

$$r_\theta(S,A) = \frac{\pi_\theta(A \mid S)}{\pi_{\mathrm{old}}(A \mid S)}.$$

The empirical version on a rollout batch of B samples is

$$\widehat{\mathcal{L}}^{\mathrm{CPI}}_{\pi_{\mathrm{old}}}(\theta) = \frac{1}{B}\sum_{i=1}^{B} r_i(\theta)\widehat{A}_i^{\pi_{\mathrm{old}}}.$$

At the beginning of optimization,

$$\theta = \theta_{\mathrm{old}}, \qquad r_i(\theta_{\mathrm{old}}) = 1$$

for every sample.

During repeated minibatch updates, however, the candidate parameters change while the rollout states, actions, and advantage estimates remain fixed. The unclipped objective continues rewarding movement in the favorable direction, regardless of how far the candidate policy has already moved from the rollout policy.

For example, if $\widehat{A}_i^{\pi_{\mathrm{old}}} > 0$, increasing $r_i(\theta)$ always increases the sample objective

$$r_i(\theta)\widehat{A}_i^{\pi_{\mathrm{old}}}.$$

Likewise, if $\widehat{A}_i^{\pi_{\mathrm{old}}} < 0$, decreasing $r_i(\theta)$ always makes the sample objective less negative. The unclipped objective contains no point at which it says that the probability has changed sufficiently.

3.3 Clipping the Likelihood Ratio

Choose a clipping parameter

$$\epsilon \in (0,1),$$

and define

$$\boxed{\mathrm{clip}\left(r_i(\theta), 1-\epsilon, 1+\epsilon\right) = \begin{cases} 1-\epsilon, & r_i(\theta) < 1-\epsilon, \\ r_i(\theta), & 1-\epsilon \le r_i(\theta) \le 1+\epsilon, \\ 1+\epsilon, & r_i(\theta) > 1+\epsilon. \end{cases}}$$

The interval

$$[1-\epsilon, 1+\epsilon]$$

is centered at the old-policy ratio $r_i = 1$. For example, if $\epsilon = 0.2$, the clipping interval is

$$[0.8, 1.2].$$

A first attempt might be to replace the likelihood ratio directly:

$$r_i(\theta)\widehat{A}_i^{\pi_{\text{old}}} \quad \longrightarrow \quad \text{clip}\left(r_i(\theta), 1-\epsilon, 1+\epsilon\right)\widehat{A}_i^{\pi_{\text{old}}}.$$

This alone is incorrect because it clips both favorable and unfavorable changes.

If $\widehat{A}_i^{\pi_{\text{old}}} > 0$, making the sampled action much less likely is unfavorable. However, directly clipping the ratio below $1-\epsilon$ would make the objective constant there and hide this harmful change.

Similarly, if $\widehat{A}_i^{\pi_{\text{old}}} < 0$, making the sampled action much more likely is unfavorable. Directly clipping above $1+\epsilon$ would hide this harmful change.

PPO must stop rewarding excessively large favorable changes without hiding unfavorable changes.

3.4 Why the Minimum Is Required

PPO compares the clipped and unclipped contributions and keeps the smaller one:

$$\boxed{\ell_i^{\text{CLIP}}(\theta) = \min\left(r_i(\theta)\widehat{A}_i^{\pi_{\text{old}}}, \text{clip}\left(r_i(\theta), 1-\epsilon, 1+\epsilon\right)\widehat{A}_i^{\pi_{\text{old}}}\right).}$$

Because this objective is maximized, the minimum chooses the less optimistic value. It removes the additional reward for an excessively large favorable change, while an unfavorable change remains visible to the optimizer.

The complete population objective is

$$\boxed{\mathcal{L}_{\pi_{\text{old}}}^{\text{CLIP}}(\theta) = \mathbb{E}_{S\sim d^{\pi_{\text{old}}},\rho_0}\left[\mathbb{E}_{A\sim\pi_{\text{old}}(\cdot|S)}\left[\min\Big(r_\theta(S,A)A^{\pi_{\text{old}}}(S,A), \text{clip}\left(r_\theta(S,A), 1-\epsilon, 1+\epsilon\right)A^{\pi_{\text{old}}}(S,A)\Big)\right]\right].}$$

The role of the minimum becomes explicit by separating the positive- and negative-advantage cases.

3.5 Case 1: Positive Advantage

Suppose

$$\widehat{A}_i^{\pi_{\text{old}}} > 0.$$

The sampled action was estimated to be better than the old policy's usual behavior, so increasing its probability is favorable.

Because multiplication by a positive number preserves inequalities,

$$\boxed{\ell_i^{\text{CLIP}}(\theta) = \begin{cases} r_i(\theta)\widehat{A}_i^{\pi_{\text{old}}}, & r_i(\theta) \le 1+\epsilon, \\ (1+\epsilon)\widehat{A}_i^{\pi_{\text{old}}}, & r_i(\theta) > 1+\epsilon. \end{cases}}$$

Equivalently,

$$\boxed{\ell_i^{\text{CLIP}}(\theta) = \widehat{A}_i^{\pi_{\text{old}}}\min\left(r_i(\theta), 1+\epsilon\right), \qquad \widehat{A}_i^{\pi_{\text{old}}} > 0.}$$

Therefore:

- Increasing the ratio from 1 toward $1+\epsilon$ improves the objective.
- Increasing it beyond $1+\epsilon$ gives no additional objective improvement from this sample.
- Decreasing the ratio below 1 makes the objective worse.
- A decrease below $1-\epsilon$ is not hidden by clipping.

Thus, for positive advantage, PPO clips only an excessively large increase in the probability of the sampled action.

3.6 Case 2: Negative Advantage

Now suppose

$$\widehat{A}_i^{\pi_{\text{old}}} < 0.$$

The sampled action was estimated to be worse than the old policy's usual behavior, so decreasing its probability is favorable.

Because multiplication by a negative number reverses inequalities,

$$\boxed{\ell_i^{\text{CLIP}}(\theta) = \begin{cases} (1-\epsilon)\widehat{A}_i^{\pi_{\text{old}}}, & r_i(\theta) < 1-\epsilon, \\ r_i(\theta)\widehat{A}_i^{\pi_{\text{old}}}, & r_i(\theta) \geq 1-\epsilon. \end{cases}}$$

Equivalently,

$$\boxed{\ell_i^{\text{CLIP}}(\theta) = \widehat{A}_i^{\pi_{\text{old}}} \max\left(r_i(\theta), 1-\epsilon\right), \qquad \widehat{A}_i^{\pi_{\text{old}}} < 0.}$$

Therefore:

- Decreasing the ratio from 1 toward $1-\epsilon$ improves the objective.
- Decreasing it below $1-\epsilon$ gives no additional objective improvement from this sample.
- Increasing the ratio above 1 makes the objective worse.
- An increase above $1+\epsilon$ is not hidden by clipping.

Thus, for negative advantage, PPO clips only an excessively large decrease in the probability of the sampled action.

3.7 Shape of a Single Clipped Contribution

Figure 1 shows the contribution of one sample. For positive advantage, the plateau appears on the right because increasing the action probability is favorable. For negative advantage, the plateau appears on the left because decreasing the action probability is favorable.

The unclipped side never becomes flat when movement is unfavorable.

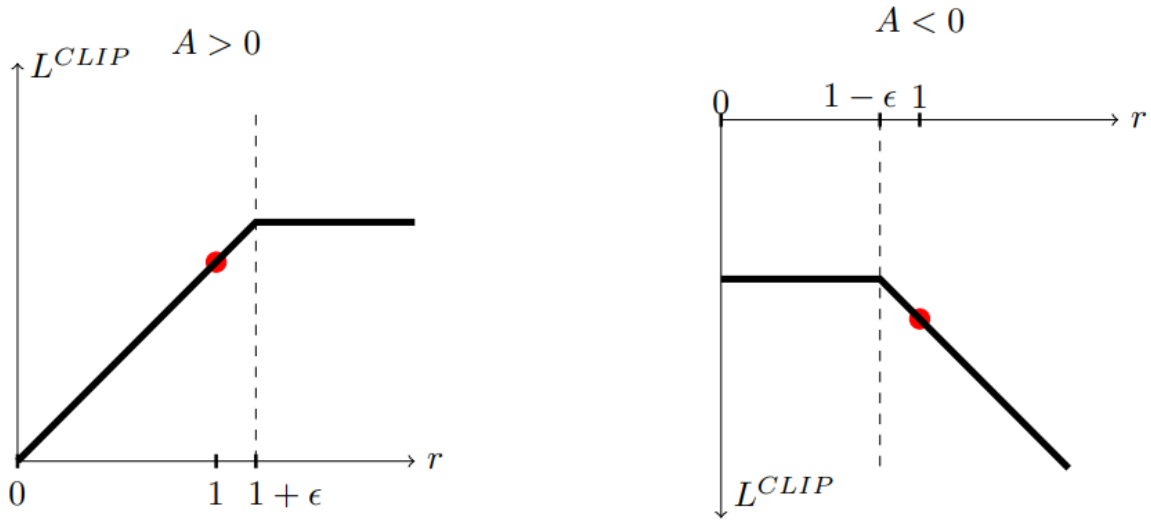


Figure 1: One timestep of the clipped surrogate as a function of the probability ratio. The left plot has positive advantage and becomes flat above $1+\epsilon$. The right plot has negative advantage and becomes flat below $1-\epsilon$. The red circles mark the starting point $r = 1$. The complete PPO objective averages many such terms. Source: Figure 1 of Schulman et al., *Proximal Policy Optimization Algorithms*.

3.8 Pessimistic Surrogate and First-Order Agreement

For every sample and every candidate policy,

$$\boxed{\ell_i^{\mathrm{CLIP}}(\theta) \le \ell_i^{\mathrm{CPI}}(\theta) = r_i(\theta)\widehat{A}_i^{\pi_{\mathrm{old}}}.}$$

This follows immediately because $\ell_i^{\mathrm{CLIP}}(\theta)$ is the minimum of the clipped and unclipped contributions.

The clipped objective is therefore a samplewise lower bound on the unclipped surrogate. It is called pessimistic because it retains the less favorable of the two predictions.

This does not mean that the clipped objective is a direct lower bound on the true return $J(\theta)$. It also does not establish monotonic policy improvement.

At the old parameters,

$$r_i(\theta_{\mathrm{old}}) = 1,$$

which lies strictly inside the clipping interval. Therefore, the clipped and unclipped objectives agree locally:

$$\boxed{\mathcal{L}_{\pi_{\mathrm{old}}}^{\mathrm{CLIP}}(\theta_{\mathrm{old}}) = \mathcal{L}_{\pi_{\mathrm{old}}}^{\mathrm{CPI}}(\theta_{\mathrm{old}}),}$$

and

$$\boxed{\nabla_\theta \mathcal{L}_{\pi_{\mathrm{old}}}^{\mathrm{CLIP}}(\theta)\big|_{\theta=\theta_{\mathrm{old}}} = \nabla_\theta \mathcal{L}_{\pi_{\mathrm{old}}}^{\mathrm{CPI}}(\theta)\big|_{\theta=\theta_{\mathrm{old}}}.}$$

PPO-Clip therefore begins with the ordinary policy-gradient direction. The objectives differ only after the candidate policy moves sufficiently far from the rollout policy.

3.9 Empirical Minibatch Objective

Let M denote the minibatch size, and let

$$\mathcal{M} \subseteq \{1, \dots, B\}, \qquad |\mathcal{M}| = M,$$

be the index set of one minibatch.

The empirical PPO-Clip policy objective is

$$\boxed{\widehat{\mathcal{L}}^{\mathrm{CLIP}}_{\mathcal{M}}(\theta) = \frac{1}{M} \sum_{i \in \mathcal{M}} \min\left(r_i(\theta) \widehat{A}_i^{\pi_{\mathrm{old}}}, \mathrm{clip}\left(r_i(\theta), 1-\epsilon, 1+\epsilon\right) \widehat{A}_i^{\pi_{\mathrm{old}}} \right).}$$

During every minibatch update and every epoch,

$$\boxed{S_i, \quad A_i, \quad \widehat{A}_i^{\pi_{\mathrm{old}}}, \quad \log \pi_{\mathrm{old}}(A_i \mid S_i) \quad \text{remain fixed.}}$$

Only the candidate parameters θ change. Therefore, even after several updates, every ratio is computed relative to the single policy that generated the rollout batch:

$$r_i\left(\theta^{(k)}\right) = \frac{\pi_{\theta^{(k)}}(A_i \mid S_i)}{\pi_{\mathrm{old}}(A_i \mid S_i)}.$$

In implementation, the ratio is normally computed using the stored old log probability:

$$\boxed{r_i(\theta) = \exp\left(\log \pi_\theta(A_i \mid S_i) - \log \pi_{\mathrm{old}}(A_i \mid S_i)\right).}$$

The old policy is not replaced after each minibatch update. It remains fixed until all epochs over the rollout batch are complete.

3.10 Complete PPO Actor-Critic Objective

Let w denote the value-function parameters, and let $\widehat{V}_i^{\mathrm{target}}$ be the fixed value target. Define the minibatch value loss

$$\boxed{\widehat{\mathcal{L}}^{\mathrm{VF}}_{\mathcal{M}}(w) = \frac{1}{M} \sum_{i \in \mathcal{M}} \left(V_w(S_i) - \widehat{V}_i^{\mathrm{target}} \right)^2}$$

and the empirical entropy bonus

$$\boxed{\widehat{\mathcal{H}}_{\mathcal{M}}(\theta) = \frac{1}{M} \sum_{i \in \mathcal{M}} \mathcal{H}\left(\pi_\theta(\cdot \mid S_i)\right).}$$

The complete PPO actor-critic objective to maximize is

$$\boxed{\widehat{\mathcal{J}}^{\mathrm{PPO}}_{\mathcal{M}}(\theta, w) = \widehat{\mathcal{L}}^{\mathrm{CLIP}}_{\mathcal{M}}(\theta) - c_V \widehat{\mathcal{L}}^{\mathrm{VF}}_{\mathcal{M}}(w) + c_H \widehat{\mathcal{H}}_{\mathcal{M}}(\theta),}$$

where $c_V \geq 0$ and $c_H \geq 0$ are the value-loss and entropy coefficients.

An implementation written as minimization instead uses

$$\boxed{\mathrm{Loss}^{\mathrm{PPO}}_{\mathcal{M}}(\theta, w) = -\widehat{\mathcal{L}}^{\mathrm{CLIP}}_{\mathcal{M}}(\theta) + c_V \widehat{\mathcal{L}}^{\mathrm{VF}}_{\mathcal{M}}(w) - c_H \widehat{\mathcal{H}}_{\mathcal{M}}(\theta).}$$

3.11 PPO-Clip Algorithm

Given initial policy parameters θ_0, value parameters w_0, clipping parameter ϵ, number of epochs E, and minibatch size M, repeat the following for $k = 0, 1, 2, \ldots$:

1. Freeze the rollout policy:
$$\pi_{\text{old}} \leftarrow \pi_{\theta_k}.$$

2. Run N parallel environments for T steps and collect
$$B = TN$$
transitions under π_{old}.

3. Using the frozen critic V_{w_k}, compute the GAE advantages $\widehat{A}_i^{\pi_{\text{old}}}$ and value targets $\widehat{V}_i^{\text{target}}$. Store $\log \pi_{\text{old}}(A_i \mid S_i)$.

4. Initialize the candidate parameters:
$$\theta \leftarrow \theta_k, \qquad w \leftarrow w_k.$$

5. For each epoch $e = 1, \ldots, E$, shuffle the fixed rollout batch and divide it into minibatches $\mathcal{M}$ of size M.

6. For each minibatch, compute
$$r_i(\theta), \qquad \widehat{\mathcal{L}}_{\mathcal{M}}^{\text{CLIP}}, \qquad \widehat{\mathcal{L}}_{\mathcal{M}}^{\text{VF}}, \qquad \widehat{\mathcal{H}}_{\mathcal{M}}.$$
Apply one Adam update to maximize
$$\widehat{\mathcal{J}}_{\mathcal{M}}^{\text{PPO}}(\theta, w).$$

7. After the final epoch, set
$$\theta_{k+1} \leftarrow \theta, \qquad w_{k+1} \leftarrow w.$$
Discard the rollout batch and collect new data under the updated policy.

3.12 What Clipping Does and Does Not Guarantee

The interval
$$[1 - \epsilon, 1 + \epsilon]$$
is not a hard constraint on every likelihood ratio. Clipping only makes an individual sample's objective flat after a sufficiently large favorable change.

Because every sample shares the same policy parameters, gradients from other samples can continue changing a sample's ratio even when its own contribution is already clipped. Therefore, PPO-Clip does not guarantee
$$r_i(\theta) \in [1 - \epsilon, 1 + \epsilon]$$
for every sample.

It also does not impose a hard bound on the average KL divergence and does not provide an unconditional monotonic-improvement guarantee. The clipping parameter ϵ determines where the objective stops rewarding favorable ratio changes; it is not a maximum policy distance.

In practice, implementations commonly monitor the empirical KL divergence and the fraction of samples whose ratios are clipped. Early stopping when the KL divergence becomes too large is an additional safeguard, not part of the clipped objective itself.