PPO and GRPO for LLM Training

LLM Lab

December 2025

Contents

Policy Training in Online RL: PPO and GRPO

This tutorial explains **Proximal Policy Optimization (PPO)** and **Group Relative Policy Optimization (GRPO)**, two algorithms used for reinforcement learning in LLM training.

---

1. Background: Policy Gradient Methods

In RL for LLMs, we want to update a policy π_θ (the language model) to generate better responses. The basic policy gradient objective is:

$$\mathcal{J}(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_t R_t \right]$$

where τ is a trajectory (sequence of tokens) and R_t is the reward at step t.

---

2. The Advantage Function

2.1 Definition

The **advantage** A_t measures how much better an action is compared to the expected value:

$$A_t = Q(s_t, a_t) - V(s_t)$$

Symbol	Meaning
$Q(s_t, a_t)$	Expected return from taking action a_t in state s_t
$V(s_t)$	Expected return from state s_t (baseline)
$A_t > 0$	Action is better than average $\rightarrow$ reinforce
$A_t < 0$	Action is worse than average $\rightarrow$ discourage

2.2 Why Use Advantage?

Using advantage instead of raw rewards:

- **Reduces variance** in gradient estimates
- **Centers the signal** around zero (relative improvement)
- **Faster convergence** during training

---

3. Generalized Advantage Estimation (GAE)

3.1 The Bias-Variance Trade-off

There are two extreme ways to estimate advantage:

Method 1: One-step TD (Temporal Difference)

$$\hat{A}_t^{(1)} = \delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$$

- **Low variance** (uses single reward)
- **High bias** (relies heavily on value estimate V)

Method 2: Monte Carlo

$$\hat{A}_t^{(\infty)} = \sum_{k=0}^{\infty} \gamma^k r_{t+k} - V(s_t)$$

- **Low bias** (uses actual returns)
- **High variance** (sums many random rewards)

3.2 GAE Formula

GAE interpolates between these extremes using parameter $\lambda \in [0, 1]$:

$$A_t^{\mathrm{GAE}(\gamma,\lambda)} = \sum_{k=0}^{\infty} (\gamma\lambda)^k \delta_{t+k}$$

where the **TD residual** is:

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$$

3.3 GAE Parameter Effects

λ Value	Behavior	Bias	Variance
$\lambda = 0$	Pure 1-step TD	High	Low
$\lambda = 1$	Monte Carlo-like	Low	High
$\lambda \approx 0.95$	Balanced (common choice)	Medium	Medium

3.4 Notation Summary for GAE

Symbol	Meaning
γ	Discount factor (typically 0.99)
λ	GAE parameter (typically 0.95)
r_t	Reward at time step t
$V(s_t)$	Value function estimate at state s_t
δ_t	TD residual at time t

---

4. Proximal Policy Optimization (PPO)

4.1 The PPO Objective

The PPO loss function is:

$$\mathcal{J}_{\mathrm{PPO}}(\theta) = \mathbb{E}\left[\min\left(r_t(\theta)A_t,\ \mathrm{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)A_t\right)\right]$$

where the **probability ratio** is:

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\mathrm{old}}}(a_t|s_t)}$$

4.2 PPO Notation Table

Symbol	Meaning
θ	Current policy parameters (being optimized)
θ_{old}	Old policy parameters (fixed during update)
$\pi_\theta(a_t \mid s_t)$	Probability of action a_t given state s_t under policy θ
$r_t(\theta)$	Probability ratio between new and old policy
A_t	Advantage estimate (from GAE)

Symbol	Meaning
ε	Clipping parameter (typically 0.1 to 0.2)
$\text{clip}(x, a, b)$	Clamp x to range $[a, b]$

4.3 For LLM Training

In the context of LLMs, the PPO objective becomes:

$$\mathcal{J}_{\text{PPO}}(\theta) = \mathbb{E}_{q\sim P(Q),\, o\sim\pi_{\theta_{\text{old}}}} \left[\frac{1}{|o|} \sum_{t=1}^{|o|} \min\left(r_t(\theta)A_t,\ \text{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)A_t\right) \right]$$

Symbol	Meaning in LLM Context
q	Query/prompt
o	Output/response sequence
o_t	Token at position t
$o_{<t}$	All tokens before position t
$\mid o \mid$	Length of output sequence
$P(Q)$	Distribution over prompts
$\pi_\theta(o_t \mid q, o_{<t})$	Probability of token o_t given prompt and context

The probability ratio for LLMs:

$$r_t(\theta) = \frac{\pi_\theta(o_t \mid q, o_{<t})}{\pi_{\theta_{\text{old}}}(o_t \mid q, o_{<t})}$$

4.4 How Clipping Works

The clipping mechanism prevents large policy updates:

Case 1: Good action ($A_t > 0$)

- We want to increase r_t (make action more likely)
- But clip prevents r_t from exceeding $1 + \varepsilon$
- Benefit is capped, preventing over-optimization

Case 2: Bad action ($A_t < 0$)

- We want to decrease r_t (make action less likely)
- But clip prevents r_t from going below $1 - \varepsilon$
- Penalty is capped, preventing destructive updates

Scenario	A_t	Unclipped Objective	Clipping Effect
Good action	> 0	Push $r_t \to \infty$	Cap at $r_t = 1 + \varepsilon$
Bad action	< 0	Push $r_t \to 0$	Cap at $r_t = 1 - \varepsilon$

4.5 PPO Architecture (for LLMs)

PPO requires **four models**:

1. **Policy Model** π_θ — The LLM being trained
2. **Reference Model** π_{ref} — Frozen copy for KL penalty

3. **Reward Model** r_ϕ — Scores response quality
4. **Value Model** V_ψ — Estimates expected return (for GAE)

---

5. Group Relative Policy Optimization (GRPO)

5.1 Motivation

GRPO was introduced by DeepSeek to improve efficiency:

> "A variant of PPO that enhances mathematical reasoning abilities while concurrently optimizing the memory usage of PPO."

Key insight: Instead of learning a value function $V(s)$, compute advantages by comparing multiple sampled outputs.

5.2 GRPO Advantage Computation

For a query q, sample G outputs: $\{o_1, o_2, \ldots, o_G\}$

Get rewards: $\{r_1, r_2, \ldots, r_G\}$

Compute **group-relative advantage**:

$$A_i = \frac{r_i - \text{mean}(\{r_1, \ldots, r_G\})}{\text{std}(\{r_1, \ldots, r_G\})}$$

5.3 GRPO vs PPO Comparison

Aspect	PPO	GRPO
Value Model	Required	Not needed
Advantage	GAE (from value estimates)	Group-relative (compare samples)
Models needed	4 (Policy + Ref + Reward + Value)	3 (Policy + Ref + Reward)
Memory	Higher	Lower
Samples per query	1	Multiple (G)

5.4 GRPO Architecture

GRPO requires **three models**:

1. **Policy Model** π_θ — The LLM being trained
2. **Reference Model** π_{ref} — Frozen copy for KL penalty
3. **Reward Model** r_ϕ — Scores response quality

The value model is eliminated by using group comparisons.

---

6. Summary

Key Equations

Component	Formula
Advantage	$A_t = Q(s_t, a_t) - V(s_t)$
TD Residual	$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$
GAE	$A_t^{\text{GAE}} = \sum_{k=0}^{\infty} (\gamma\lambda)^k \delta_{t+k}$
Probability Ratio	$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}$
PPO Objective	$\min(r_t A_t, \text{clip}(r_t, 1-\varepsilon, 1+\varepsilon) A_t)$
GRPO Advantage	$A_i = (r_i - \bar{r})/\sigma_r$

When to Use What

Method	Best For
PPO	When you have compute for value model, need fine-grained credit assignment
GRPO	Memory-constrained settings, reasoning tasks with verifiable rewards

---

7. References

1. Schulman et al. (2017). "Proximal Policy Optimization Algorithms" arXiv:1707.06347
2. Schulman et al. (2015). "High-Dimensional Continuous Control Using Generalized Advantage Estimation" arXiv:1506.02438
3. DeepSeek-AI (2024). "DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models" arXiv:2402.03300