PPO (Proximal Policy Optimization)

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

- 1. 目标：训练一个Policy（LLM），在所有的状态 S （generated tokens）下，给出相应的Action（next token），得到的Return （累积Reward）的期望最大。
- 2. 期望：$q\sim P(Q)$ 表示对一个batch的数据求期望，$o\sim\pi_{old}(O|q)$表示对所有的Trajectory求期望。
- 3. Importance Sampling (online-2-offline)：
  - $E(f(x))_{x\sim p(x)} = \sum_x f(x)*q(x) = \sum_x f(x)*q(x)\frac{p(x)}{p(x)} = \sum_x f(x)*p(x)\frac{q(x)}{p(x)} = E_{x\sim q(x)}[f(x)\frac{q(x)}{p(x)}]$
  - 以更基本的Policy Gradient为例:

$$\begin{aligned}\mathcal{J}_{PG}(\theta) &= E_{(s_t,a_t)\sim\pi_\theta}\left[A^\theta(s_t,a_t)p_\theta(a_t^n \mid s_t^n)\right]\\ &= E_{(s_t,a_t)\sim\pi_{\theta'}}\left[\frac{p_\theta(a_t\mid s_t)}{p_{\theta'}(a_t\mid s_t)}A^{\theta'}(s_t,a_t)p_\theta(a_t^n\mid s_t^n)\right]\end{aligned} \tag{1}$$

$$\begin{aligned}\nabla\mathcal{J}_{PG}(\theta) &= E_{(s_t,a_t)\sim\pi_\theta}\left[A^\theta(s_t,a_t)\nabla\log p_\theta(a_t^n\mid s_t^n)\right]\\ &= E_{(s_t,a_t)\sim\pi_{\theta'}}\left[\frac{p_\theta(a_t\mid s_t)}{p_{\theta'}(a_t\mid s_t)}A^{\theta'}(s_t,a_t)\nabla\log p_\theta(a_t^n\mid s_t^n)\right]\end{aligned} \tag{2}$$

- 4. Advantage 计算：
  - 为什么要算Advantage？在好的局势下，可能所有action的reward都是正的，坏的局势下，可能都是负的
  - Advantage = Returns - Baseline：$A_\theta(s,a) = Q_\theta(s,a) - V_\theta(s)$ (在state s下，做出Action a，比其他动作能带来多少优势)
  - 动作价值函数 $Q_\theta(s,a) = r_t + \gamma * V_\theta(s_{t+1})$ (TD error) ,($V_\theta(s)$为状态价值函数)
  - $A_\theta(s_t,a) = r_t + \gamma * V_\theta(s_{t+1}) - V_\theta(s_t)$，此时，仅需要状态价值函数
  - $V_\theta(s_{t+1}) \approx r_{t+1} + \gamma * V_\theta(s_{t+2})$ (Bellman Equation)
  - $A^1_\theta(s_t,a) = r_t + \gamma * V_\theta(s_{t+1}) - V_\theta(s_t)$
  - $A^2_\theta(s_t,a) = r_t + \gamma * r_{t+1} + \gamma^2 * V_\theta(s_{t+2}) - V_\theta(s_t)$
  - $A^T_\theta(s_t,a) = r_t + \gamma * r_{t+1} + \gamma^2 * r_{t+2} + \ldots + \gamma^T * r_T - V_\theta(s_t)$
  - $\delta^V_t = r_t + \gamma * V_\theta(s_{t+1}) - V_\theta(s_t)$
  - $\delta^V_{t+1} = r_{t+1} + \gamma * V_\theta(s_{t+2}) - V_\theta(s_{t+1})$
  - $A^1_\theta(s_t,a) = \delta^V_t$，$A^2_\theta(s_t,a) = \delta^V_t + \gamma\delta^V_{t+1}$，$A^3_\theta(s_t,a) = \delta^V_t + \gamma\delta^V_{t+1} + \gamma^2\delta^V_{t+2}\ \ldots$
  - **Generalized Advantage Estimation (GAE)**
    - $A^{GAE}_\theta(s_t,a) = (1-\lambda)(A^1_\theta + \lambda * A^2_\theta + \lambda^2 A^3_\theta + \ldots)$ (For Example: $\lambda = 0.9$，$A^{GAE}_\theta(s_t,a) = 0.1A^1_\theta + 0.09A^2_\theta + 0.081A^3_\theta + \ldots$))
    - $A^{GAE}_\theta(s_t,a) = \sum_{b=0}^{\infty}(\gamma\lambda)^b\delta^V_{t+b}$
    - 采样步数越少方差越小、偏差越大；采样步数越大方差越大、偏差越小；因此，GAE函数是为了平衡方差与偏差，保证训练稳定性。
- 5. Clip算子：
  - 为保证训练的稳定性，需要限制policy在每一个training epoch的改变程度

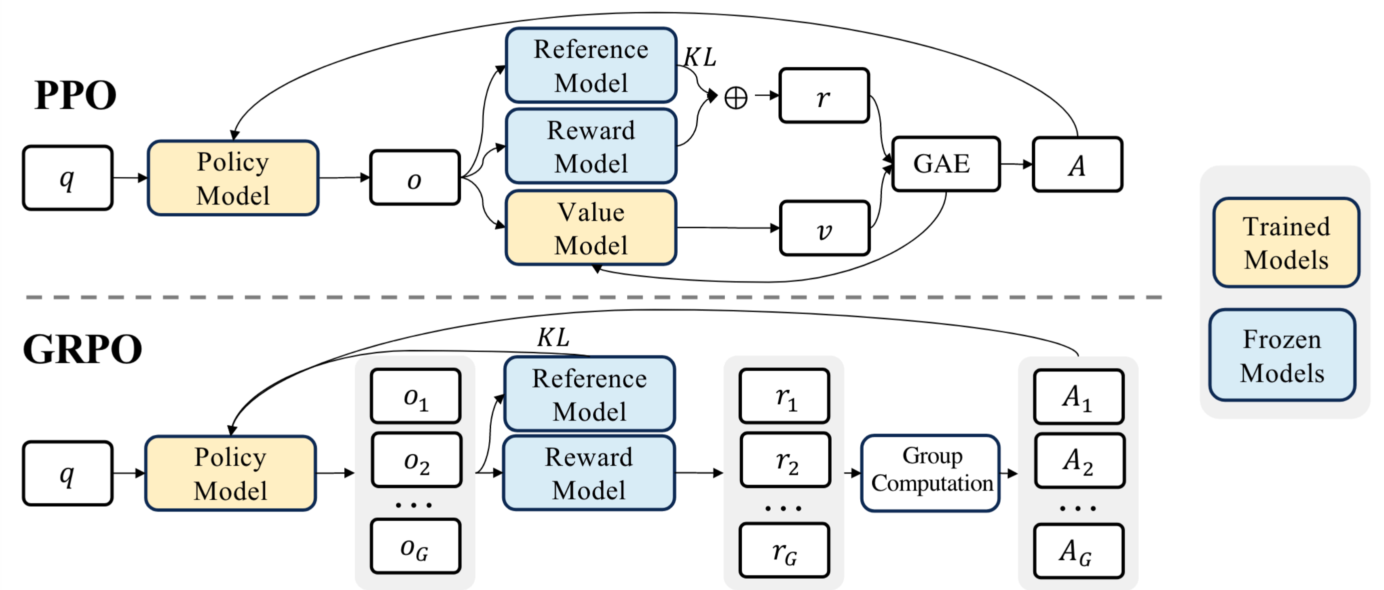
- 
- 为什么要求min？1）clip掉之后该条数据是没有梯度的；2）Advantage有正负。

	$p_t(\theta) > 0$	A_t	Return Value of *min*	Objective is Clipped	Sign of Objective	Gradient
1	$p_t(\theta) \in [1-\epsilon, 1+\epsilon]$	$+$	$p_t(\theta)A_t$	no	$+$	✓
2	$p_t(\theta) \in [1-\epsilon, 1+\epsilon]$	$-$	$p_t(\theta)A_t$	no	$-$	✓
3	$p_t(\theta) < 1-\epsilon$	$+$	$p_t(\theta)A_t$	no	$+$	✓
4	$p_t(\theta) < 1-\epsilon$	$-$	$(1-\epsilon)A_t$	yes	$-$	**0**
5	$p_t(\theta) > 1+\epsilon$	$+$	$(1+\epsilon)A_t$	yes	$+$	**0**
6	$p_t(\theta) > 1+\epsilon$	$-$	$p_t(\theta)A_t$	no	$-$	✓

- 6. PPO in LLMs
  - Policy—>LLM, state—>generated tokens, action—>predict next token
  - 如何计算 advantage —> 如何计算 reward & 如何计算状态价值$V_\theta(s)$
  - Reward: Reward Modeling，如ORM、PRM、LLM-as-a-Judge、Rule-Based Reward等
  - 状态价值$V_\theta(s)$: 一般做法：使用LLM作为 Critic/Value Model 来估计，将最后的 LM head 替换为 value head，得到截止到所有tokens的state value (注意，Critic Model需要更新参数)

GRPO (Group Relative Policy Optimization)

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q\sim P(Q),\, \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(O|q)} \left\{ \frac{1}{G}\sum_{i=1}^{G} \frac{1}{|o_i|}\sum_{t=1}^{|o_i|} \min\left[\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})}\hat{A}_{i,t},\ \text{clip}\left(\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})}, 1-\epsilon, 1+\epsilon \right)\hat{A}_{i,t} \right] - \beta\mathbb{D}_{\text{KL}}[\pi_\theta \,\|\, \pi_{\text{ref}}] \right\}$$



1. Motivation：舍弃掉 PPO 中的 Critic Model，1）直接来看是舍弃掉一个Model节省资源；2）实际上，在真实场景中，Policy 和 Critic Model通常是同一个模型，只是最后一层的 head 不同，因此存在训练目标的堆叠，影响训练效率和稳定性。

2. GRPO的Advantage如何计算?

$$\hat{A}_{i,t} = \frac{r_i - mean(\mathbf{r})}{std(\mathbf{r})}$$

3. 为什么要再加上KL散度?

$$\mathbb{D}_{KL}[\pi_\theta||\pi_{ref}] = \frac{\pi_{ref}(o_{i,t}|q,o_{i,<t})}{\pi_\theta(o_{i,t}|q,o_{i,<t})} - log\frac{\pi_{ref}(o_{i,t}|q,o_{i,<t})}{\pi_\theta(o_{i,t}|q,q_{i,<t})}$$

4. length bias?
5. question-level difficult bias?

Algorithm 1 Iterative Group Relative Policy Optimization

Input initial policy model $\pi_{\theta_{\text{init}}}$; reward models r_φ; task prompts $\mathcal{D}$; hyperparameters ε, β, μ

policy model $\pi_\theta \leftarrow \pi_{\theta_{\text{init}}}$
for iteration = 1, . . . , I **do**
    reference model $\pi_{ref} \leftarrow \pi_\theta$
    for step = 1, . . . , M **do**
        Sample a batch $\mathcal{D}_b$ from $\mathcal{D}$
        Update the old policy model $\pi_{\theta_{old}} \leftarrow \pi_\theta$
        Sample G outputs $\{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(\cdot \mid q)$ for each question $q \in \mathcal{D}_b$
        Compute rewards $\{r_i\}_{i=1}^{G}$ for each sampled output o_i by running r_φ
        Compute $\hat{A}_{i,t}$ for the t-th token of o_i through group relative advantage estimation.
        for GRPO iteration = 1, . . . , μ **do**
            Update the policy model π_θ by maximizing the GRPO objective (Equation 21)
    Update r_φ through continuous training using a replay mechanism.

Output π_θ

6.