From GPT to DeepSeek

Presenter : Jeongwan Shin

Kyungpook National University

February 19, 2025

Contents

- **Preliminary(History of GPT, RLHF)**

- **InstructGPT(+ChatGPT)**
  - Training language models to follow instructions with human feedback(NeurIPS 2022)
  - PPO : Proximal Policy Optimization Algorithms(arxiv, 2017)

- **After InstructGPT**
  - **DPO :** Direct Preference Optimization: Your Language Model is Secretly a Reward Model (NeurIPS 2023)
  - **DeepSeek-R1 -** GRPO : Group Relative Policy Optimization

GPT Series

Transformer

Paper :
Attention is All you need
(2017.06.12)

GPT-1

117 million parameters
Paper: improving Language Understanding by Generative Pre-Training
(2018.06.11)

GPT-2

1.5 billion parameters
Paper: Language Models are Unsupervised Multitask Learners
(2019.02.14)

GPT-3

175 billion parameters
Paper: Language Model are Few-shot Learners
(2020.05.28)

GPT-3* (InstructGPT)

GPT3+RLHF
Paper: Training language models to follow instructions with human feedback
(2022.05.04)

GPT-3.5 (ChatGPT)

175 billion parameters
Blog:https://openai.com/index/chatgpt/
(2022.11.30)

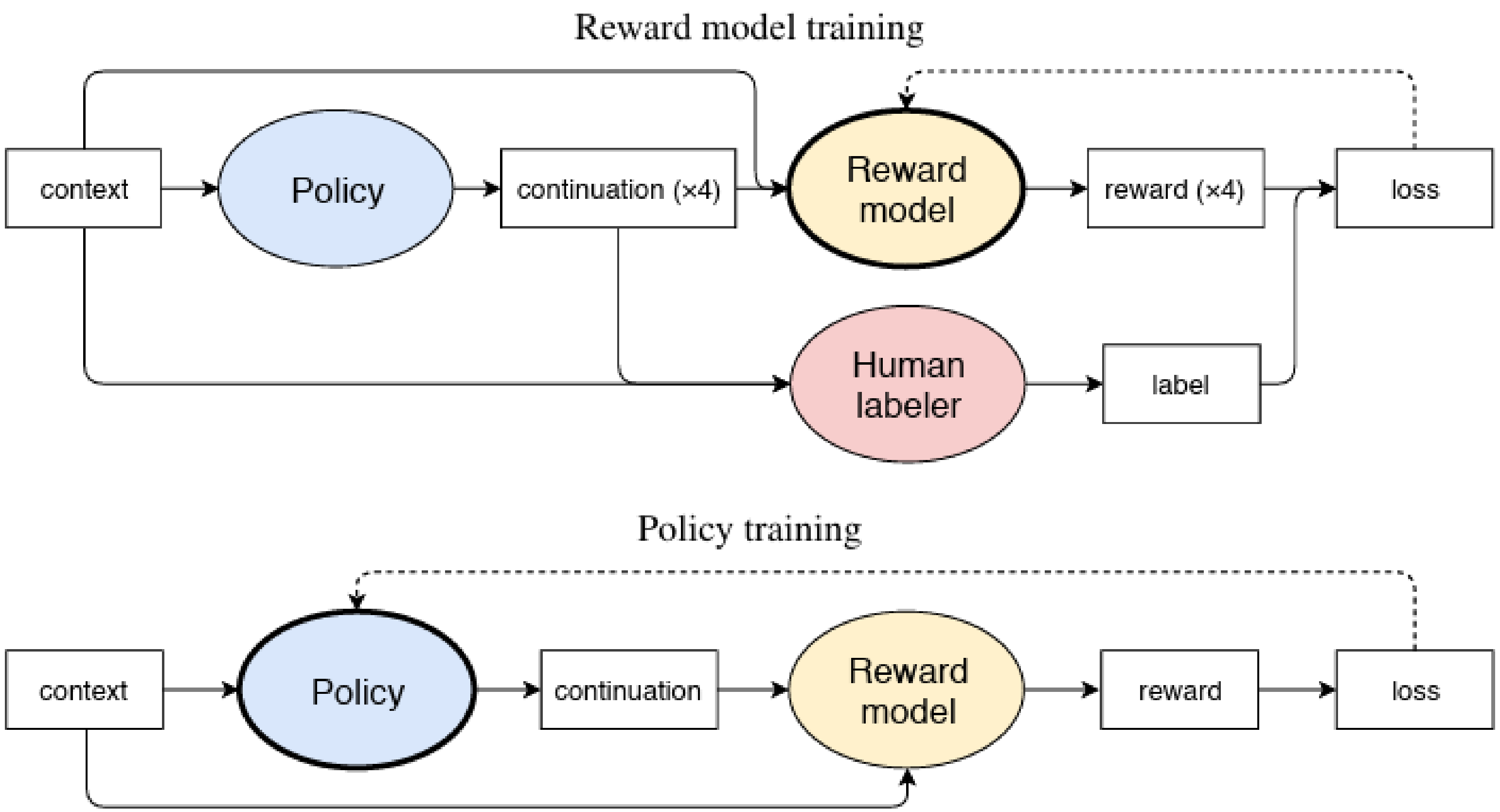
GPT4(ChatGPT)

170 trillion parameters
Paper: GPT-4 Technical Reports
(2023.03.15)

Fine-Tuning Language Models from Human Preferences

(OpenAI, 2019. 09. 18)

- Apply RL to tasks where reward is defined by human judgment



Training language models to follow instructions with human feedback

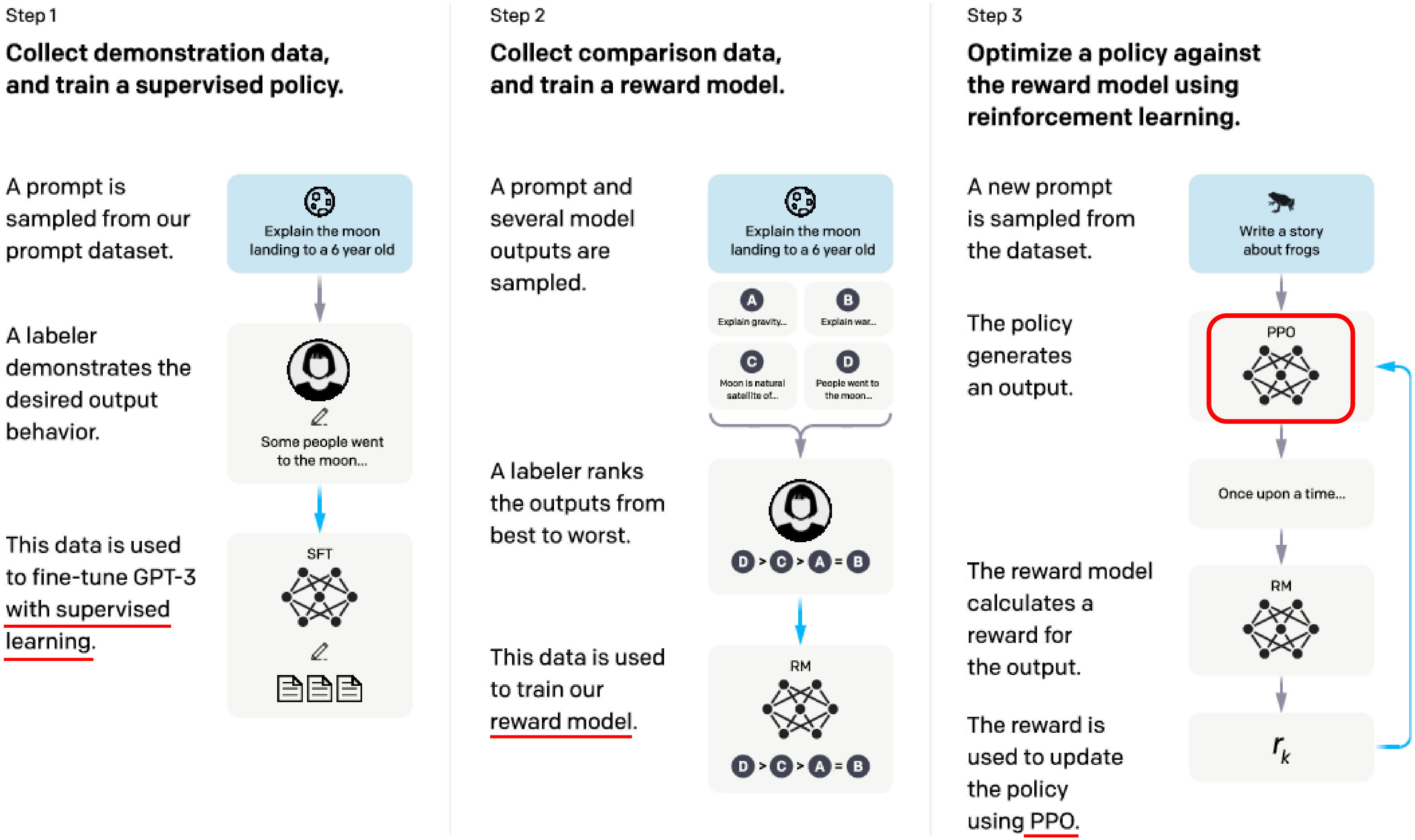
John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov

OpenAI

NeurIPS 2022

InstructGPT: Training language models to follow instructions with human feedback

(2022.05.04, OpenAI)



PPO: Proximal Policy Optimization Algorithms

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

OpenAI

arxiv, 2017.07.20

PPO : Proximal Policy Optimization Algorithms

- **Policy Gradient Methods :** $L^{PG}(\theta) = \hat{\mathbb{E}}_t\left[\log \pi_\theta(a_t \mid s_t)\hat{A}_t\right]$
  - policy : $\pi_\theta(a_t|s_t)$
  - advantage : $\hat{A}_t$

- **TRPO:** Trust Region Policy Optimization(ICML 2015)

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

- **PPO:** Proximal Policy Optimization Algorithms(arxiv 2017)

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

$$\boxed{r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}}$$

- **RLHF(PPO) :** Training language models to follow instructions with human feedback(NeurIPS 2022)

$$\begin{aligned}\text{objective}\,(\phi) =& E_{(x,y)\sim D_{\pi_\phi^{\mathrm{RL}}}}\left[r_\theta(x,y) - \beta\log\left(\pi_\phi^{\mathrm{RL}}(y \mid x)/\pi^{\mathrm{SFT}}(y \mid x)\right)\right] + \\ & \gamma E_{x\sim D_{\text{pretrain}}}\left[\log(\pi_\phi^{\mathrm{RL}}(x))\right]\end{aligned}$$

$(\epsilon = 0.2, \beta = 0.02, \gamma = 27.8)$

PPO : Proximal Policy Optimization Algorithms

- **Policy Gradient Methods :** $L^{PG}(\theta) = \hat{\mathbb{E}}_t \left[\log \pi_\theta(a_t \mid s_t) \hat{A}_t \right]$
  - policy : $\pi_\theta(a_t|s_t)$
  - advantage : $\hat{A}_t$

- **TRPO:** Trust Region Policy Optimization(ICML 2015)

Important sampling

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

- **PPO:** Proximal Policy Optimization Algorithms(arxiv 2017)

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

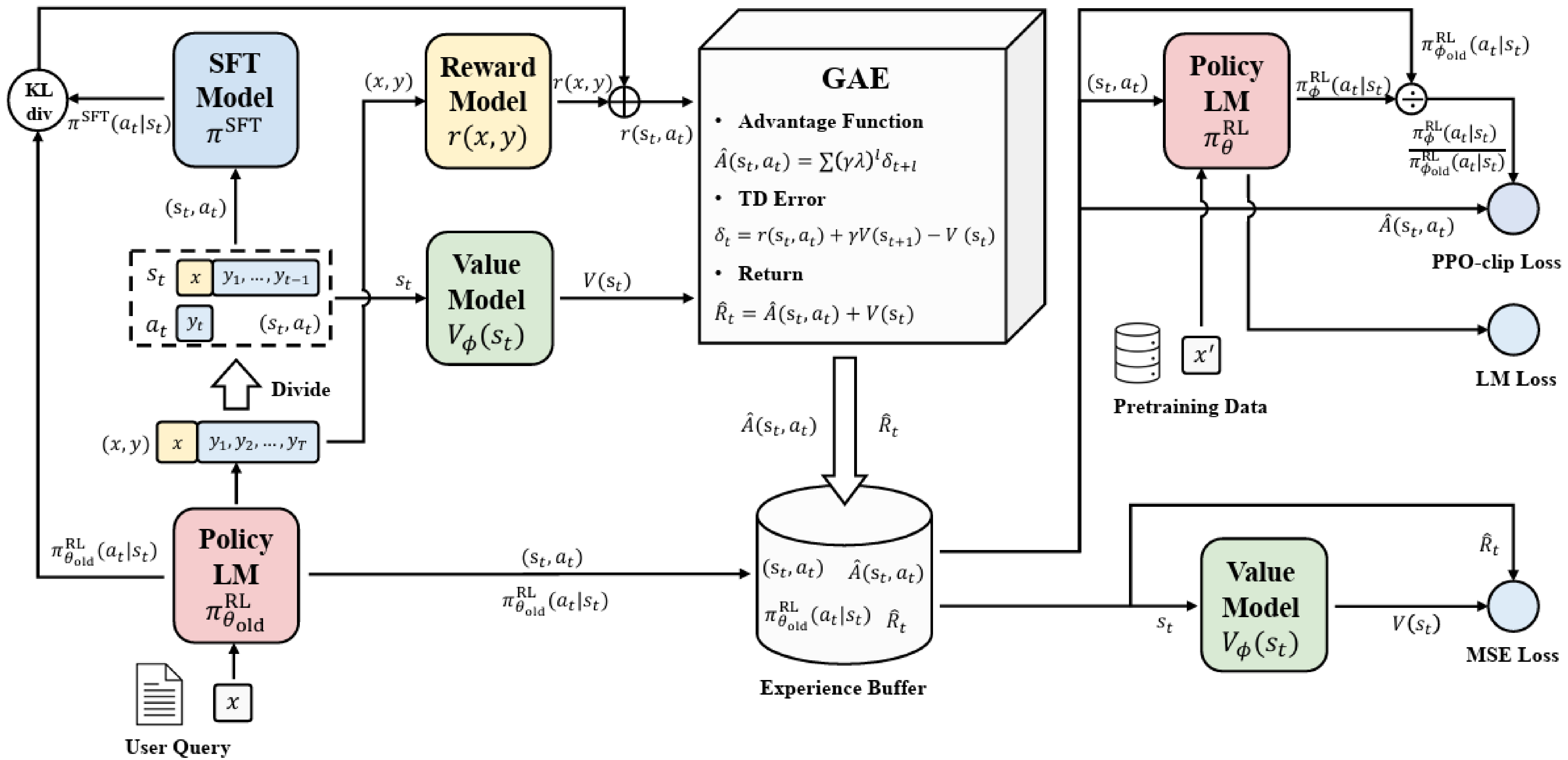
$$\boxed{r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}}$$

- **RLHF(PPO) :** Training language models to follow instructions with human feedback(NeurIPS 2022)

$$\text{objective}(\phi) = E_{(x,y)\sim D_{\pi_\phi^{\mathrm{RL}}}} \left[\boxed{r_\theta(x,y)} - \beta \log \left(\pi_\phi^{\mathrm{RL}}(y \mid x) / \pi^{\mathrm{SFT}}(y \mid x) \right) \right] + \gamma E_{x\sim D_{\text{pretrain}}} \left[\log(\pi_\phi^{\mathrm{RL}}(x)) \right]$$

$(\epsilon = 0.2, \beta = 0.02, \gamma = 27.8)$

Secrets of RLHF in Large Language Models (arXiv 2023)



PPO : Proximal Policy Optimization Algorithms

- **Policy Gradient Methods :** $L^{PG}(\theta) = \hat{\mathbb{E}}_t\left[\log \pi_\theta(a_t \mid s_t)\hat{A}_t\right]$
  - policy : $\pi_\theta(a_t|s_t)$
  - advantage : $\hat{A}_t$
- **TRPO:** Trust Region Policy Optimization(ICML 2015) Important sampling

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

- **PPO:** Proximal Policy Optimization Algorithms(arxiv 2017)

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min(\boxed{r_t(\theta)}\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right] \qquad \boxed{r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}}$$

- **RLHF(PPO) :** Training language models to follow instructions with human feedback(NeurIPS 2022)

$$\text{objective}(\phi) = E_{(x,y)\sim D_{\pi_\phi^{\text{RL}}}}\left[r_\theta(x,y) - \beta\log\left(\pi_\phi^{\text{RL}}(y \mid x)/\pi^{\text{SFT}}(y \mid x)\right)\right] + \gamma E_{x\sim D_{\text{pretrain}}}\left[\log(\pi_\phi^{\text{RL}}(x))\right]$$

$(\epsilon = 0.2, \beta = 0.02, \gamma = 27.8)$

- **Policy ?**
- **Advantage ?**
- **Important sampling ?**

PPO : Proximal Policy Optimization Algorithms

- **Policy Gradient Methods :** $L^{PG}(\theta) = \hat{\mathbb{E}}_t\left[\log \pi_\theta(a_t \mid s_t)\hat{A}_t\right]$
  - policy : $\pi_\theta(a_t|s_t)$
  - advantage : $\hat{A}_t$
- **TRPO:** Trust Region Policy Optimization(ICML 2015) Important sampling

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

- **PPO:** Proximal Policy Optimization Algorithms(arxiv 2017)

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min(\boxed{r_t(\theta)}\hat{A}_t, \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right] \qquad \boxed{r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}}$$

- **RLHF(PPO) :** Training language models to follow instructions with human feedback(NeurIPS 2022)

$$\text{objective}(\phi) = E_{(x,y)\sim D_{\pi_\phi^{\mathrm{RL}}}}\left[r_\theta(x,y) - \beta\log\left(\pi_\phi^{\mathrm{RL}}(y \mid x)/\pi^{\mathrm{SFT}}(y \mid x)\right)\right] + \gamma E_{x\sim D_{\text{pretrain}}}\left[\log(\pi_\phi^{\mathrm{RL}}(x))\right]$$

$(\epsilon = 0.2, \beta = 0.02, \gamma = 27.8)$

- **Policy ?**
  - Value based learning, policy based learning
- **Advantage ?**
  - GAE, TD Error
- **Important sampling ?**

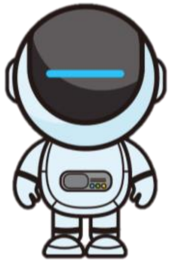
Reinforcement Learning

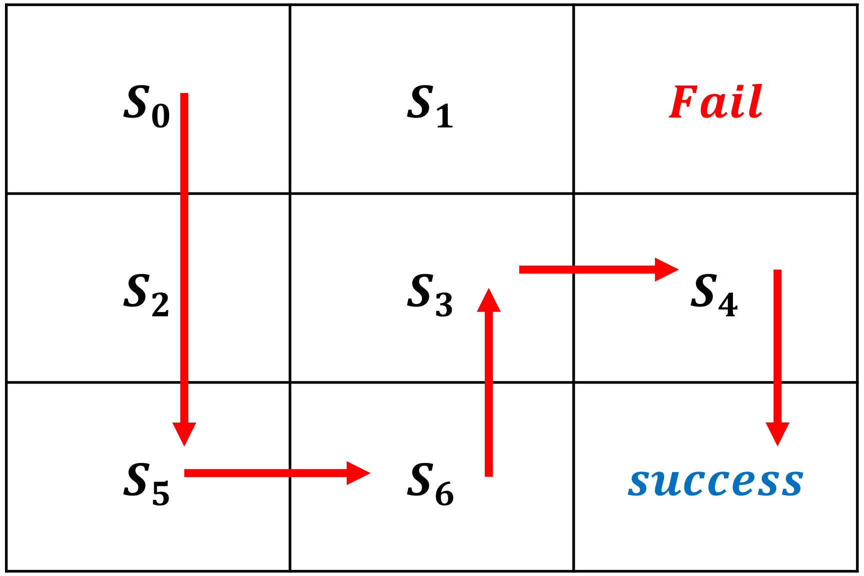
S_0	S_1	$Fail$
S_2	S_3	S_4
S_4	S_5	$success$

- 상태(State, s)
  - $\boldsymbol{s} = \{s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, success, fail\}$
- 행동(Action, a)
  - $\boldsymbol{a} = \{up, down, left, right\}$
- Reward
  - $success$: +10
  - $fail$: -10
  - 이동 : -1
- Episode
  - $s_0, a_D \rightarrow s_2, a_D \rightarrow s_4, a_R \rightarrow s_5, a_R \rightarrow Success$

Reinforcement Learning(Q-Value function)

Agent





- Q-value function(state-action value function)

$$Q_\theta(s,a) = \mathbb{E}\left[G_t \mid s_t = s, a_t = a\right]$$

G_t : t에서의 Return

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k} = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ \dots$$

discount factor

- r_t
  - *success*: +10
  - *fail*: -10
  - 이동 : -1

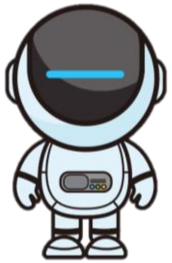
- Policy : $\pi(s) = \arg\max_a Q_\theta(s,a)$
- Ex) 초기 Policy 상태

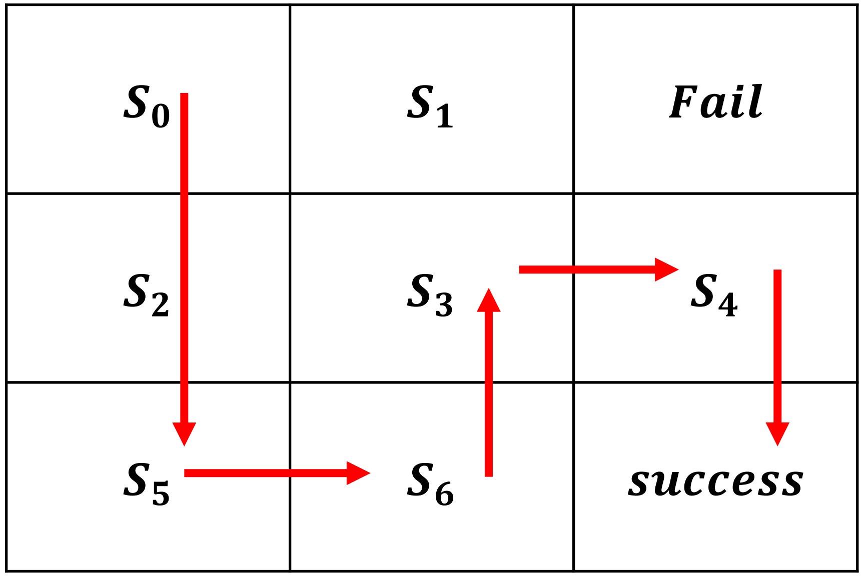
Q-value	s_0	s_1	s_2	s_3	s_4	s_5	s_6
Up	-	-	0.1	0.2	0.2	0.5	0.5
Down	0.4	0.5	0.3	0.3	0.4	-	-
Right	0.2	0.2	0.1	0.5	-	0.8	0.4
left	-	0.3	-	0.1	0.3	-	0.4

Reinforcement Learning(Q-Value function)

$$\boxed{G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k}} = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ \dots$$

Agent





- Q-value function(state-action value function)

$$Q_\theta(s,a) = \mathbb{E}\left[G_t \mid s_t = s, a_t = a\right]$$

- Policy: $\pi(s) = \arg\max_a Q_\theta(s,a)$

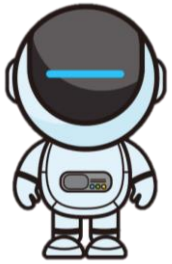
	s_0	s_1	s_2	s_3	s_4	s_5	s_6
Up	-	-	0.1	0.2	0.2	0.5	0.5
Down	0.4	0.4	0.3	0.3	0.4	-	-
Right	0.2	0.5	0.1	0.5	-	0.8	0.4
left	-	0.3	-	0.1	0.3	-	0.2

- **TD Error:** 현재 상태에서 예측된 보상과 미래의 실제 경험한 보상의 차이
  - *Ex)* $Q(s_6, up)$ 의 TD Error
  - $-1(r_t) + 0.9(\gamma) \times 0.5 - 0.5 = -1.05$
  - $Q(s_6, up) = 0.5 - \alpha \times 1.05 = -0.55 \ (\alpha = 1)$

$$\delta_t = r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)$$

Reinforcement Learning(Q-Value function)

Agent



s_0	s_1	*Fail*
s_2	s_3	s_4
s_5	s_6	*success*

- Q-value function(state-action value function)

$$Q_\theta(s,a) = \mathbb{E}\big[G_t \mid s_t = s, a_t = a\big]$$

- Policy: $\pi(s) = \arg\max_a Q_\theta(s,a)$

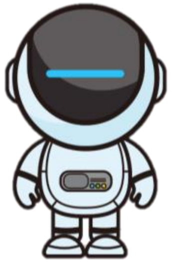
	s_0	s_1	s_2	s_3	s_4	s_5	s_6
Up	-	-	0.1	0.2	0.2	0.5	-0.55
Down	0.4	0.4	0.3	0.3	0.4	-	-
Right	0.2	0.5	0.1	0.5	-	0.8	10
left	-	0.3	-	0.1	0.3	-	0.2

- **TD Error:**
  - *Ex)* $Q(s_6, up)$ 의 TD Error
  - $-1(r_t) + 0.9(\gamma) \times 0.5 - 0.5 = -1.05$
  - $Q(s_6, up) = 0.5 + \alpha \times -1.05 = -0.55$ $(\alpha = 1)$

- **TD Error:**
  - *Ex)* $Q(s_6, right)$ 의 TD Error
  - $10(r_t) + 0.9(\gamma) \times 0 - 0.5 = 9.5$
  - $Q(s_6, right) = 0.5 + \alpha \times 9.5 = 10$ $(\alpha = 1)$

Reinforcement Learning(Q-Value function)

Agent



S_0	S_1	$Fail$
S_2	S_3	S_4
S_5	S_6	$success$

- **TD Error:**
  - $Ex) V(s_{t=6})$ 와 $V(s_{t+1=3})$의 TD Error
  - $-1 + 0.9 \times 0.25 - 0.4 = 0.275$
  - $0.4{+}0.275 = 0.675$

- Value function(state-action value function)

$$V_\theta(s) = \mathbb{E}[G_t \mid s_t = s]$$

$$\pi(s) = argmax_a[r(s,a) + \gamma V(s')]$$

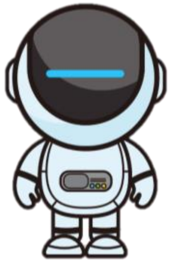
Value	s_0	s_1	s_2	s_3	s_4	s_5	s_6
Up	-	-	0.1	0.1	0.2	0.4	0.6
Down	0.4	0.4	0.3	0.3	0.4	-	-
Right	0.2	0.5	0.2	0.5	-	0.8	0.4
left	-	0.3	-	0.1	0.3	-	0.2
AVG	0.3	0.4	0.2	0.25	0.3	0.6	0.4

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

- **TD Error:**
  - $Ex) V(s_{t=6})$와 $V(s_{t+1=s})$ 의 TD Error
  - $-1 + 0.9 \times 10 - 0.4 = 7.6$
  - $0.4{+}7.6 = 8$

Reinforcement Learning(Q-Value function)

Agent



S_0	S_1	$Fail$
S_2	S_3	S_4
S_5	S_6	$success$

- **TD Error:**
  - *Ex)* $V(s_{t=6})$ 와 $V(s_{t+1=3})$의 TD Error
  - $-1 + 0.9 \times 0.25 - 0.4 = 0.275$
  - $0.4 + 0.275 = 0.675$

- Value function(state-action value function)

$$V_\theta(s) = \mathbb{E}[G_t \mid s_t = s]$$

$$\pi(s) = argmax_a[r(s,a) + \gamma V(s')]$$

Value	s_0	s_1	s_2	s_3	s_4	s_5	s_6
Up	-	-	0.1	0.1	0.2	0.4	0.675
Down	0.4	0.4	0.3	0.3	0.4	-	-
Right	0.2	0.5	0.2	0.5	-	0.8	10
left	-	0.3	-	0.1	0.3	-	0.2
AVG	0.3	0.4	0.2	0.25	0.3	0.6	3.685

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

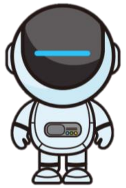
- **TD Error:**
  - *Ex)* $V(s_{t=6})$와 $V(s_{t+1=S})$ 의 TD Error
  - $10 + 0.9 \times 0 - 0.4 = 9.6$
  - $0.4 + 9.6 = 10$

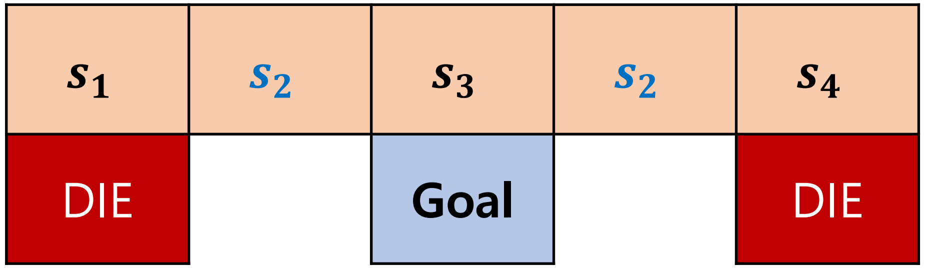
Reinforcement Learning

But!

- Value-based 학습처럼 특정 상태에서 즉각적인 보상을 학습하면 **연속적인 행동 공간(Continuous action)**이나 **확률적 정책(Stochastic policy)**을 다루기 어려움

EX) 조금 극단적인 상황을 가정 (왼쪽 s_2와 오른쪽 s_2 같음) **:**





$$s_t = \{L, R, U, D\}$$
$$s_1 = \{0, 1, 0, 1\}$$
$$s_2 = \{1, 1, 0, 0\}$$
$$s_3 = \{1, 1, 0, 1\}$$
$$s_4 = \{1, 0, 0, 1\}$$

Value	s_1	s_2	s_3	s_4
Down	0.0	-	1.0	0.0
Right	1.0	1.0	0.0	-
left	-	0.0	0.0	1.0

문제점 :
s_2 일 때,
$Q(s_2, R) > Q(s_2, L)$ 이면, 반복이 발생.

다시 학습하면
$Q(s_2, R) < Q(s_2, L)$, 또 반복...

Reinforcement Learning

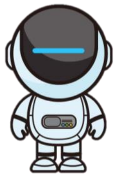
But!

- Value-based 학습처럼 특정 상태에서 즉각적인 보상을 학습하면 **연속적인 행동 공간(Continuous action)**이나 **확률적 정책(Stochastic policy)**을 다루기 어려움

- ## Policy-based Learning
  - **목표** : 장기적인 보상(Expected Return) 최적화
  - 강화학습의 궁극적인 목표는 단기 보상이 아니라 **장기적인 성과를 높이는 것**입니다.

  - 즉, 단순히 **현재 상태에서 가장 높은 보상을 받는 행동을 선택하는 것**이 아니라, **미래**까지 고려한 최적의 행동을 학습해야 합니다.

Reinforcement Learning(Policy-based Learning)



S_0	S_1	$Fail$
S_2	S_3	S_4
S_5	S_6	$success$

- **에피소드 예제**$(\gamma = 1, \hat{A}_t \approx G_t)$**:**

$(s_0, a_R \rightarrow s_1, a_D \rightarrow s_3, a_D \rightarrow s_6, a_R \rightarrow s_S)$, G_t=$(-1)^4$+10

$(s_0, a_R \rightarrow s_1, a_R \rightarrow s_F)$, G_t=$(-1)^2$-10

$(s_0, a_R \rightarrow s_1)$, G_t=-1 + ?

$(s_0, a_D \rightarrow s_2)$, G_t=-1 + ?

$(s_0, a_D \rightarrow s_2, a_u \rightarrow s_0, a_R \rightarrow s_1 \ldots)$, G_t=?

$(s_0, a_D \rightarrow s_2, a_u \rightarrow s_0, a_D \rightarrow s_2, a_D \rightarrow s_5 \ldots)$, G_t=?

…

- **Policy based learning**
  - $\boldsymbol{\pi_\theta(a|s)}$를 직접 학습, 즉 확률적 선택 (sampling)이 가능.

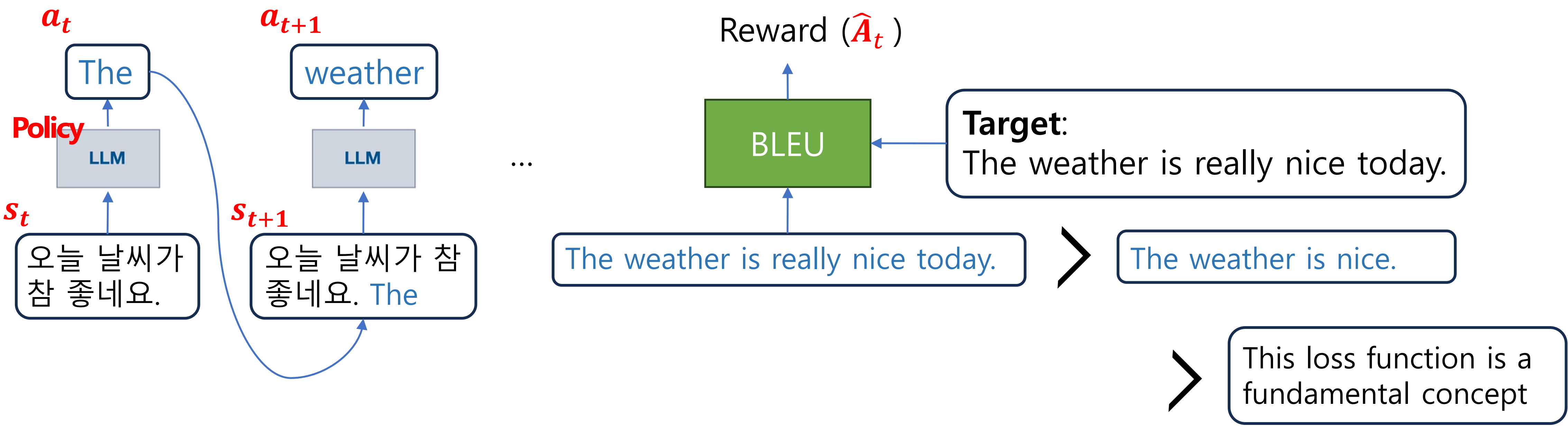
- **Policy Gradient Objective :**
  - 정책이 **이득이 큰 행동**에는 더 높은 확률을 부여하도록 학습

$$L^{PG}(\theta) = \mathbb{E}_t\left[\log \pi_\theta(a_t \mid s_t)\hat{A}_t\right]$$

  - $\hat{A}_t$: Advantage Function
    - $\hat{A}_t$ > 0 : $log\pi_\theta(a_t|s_t)$ 값을 최대화
    - $\hat{A}_t$ < 0 : $log\pi_\theta(a_t|s_t)$ 값을 최소화

- **학습의 어려움**
  1. **에피소드**에 대한 경우의 수가 많아 학습이 어려움.
  2. G_t 에 대한 variance가 높음

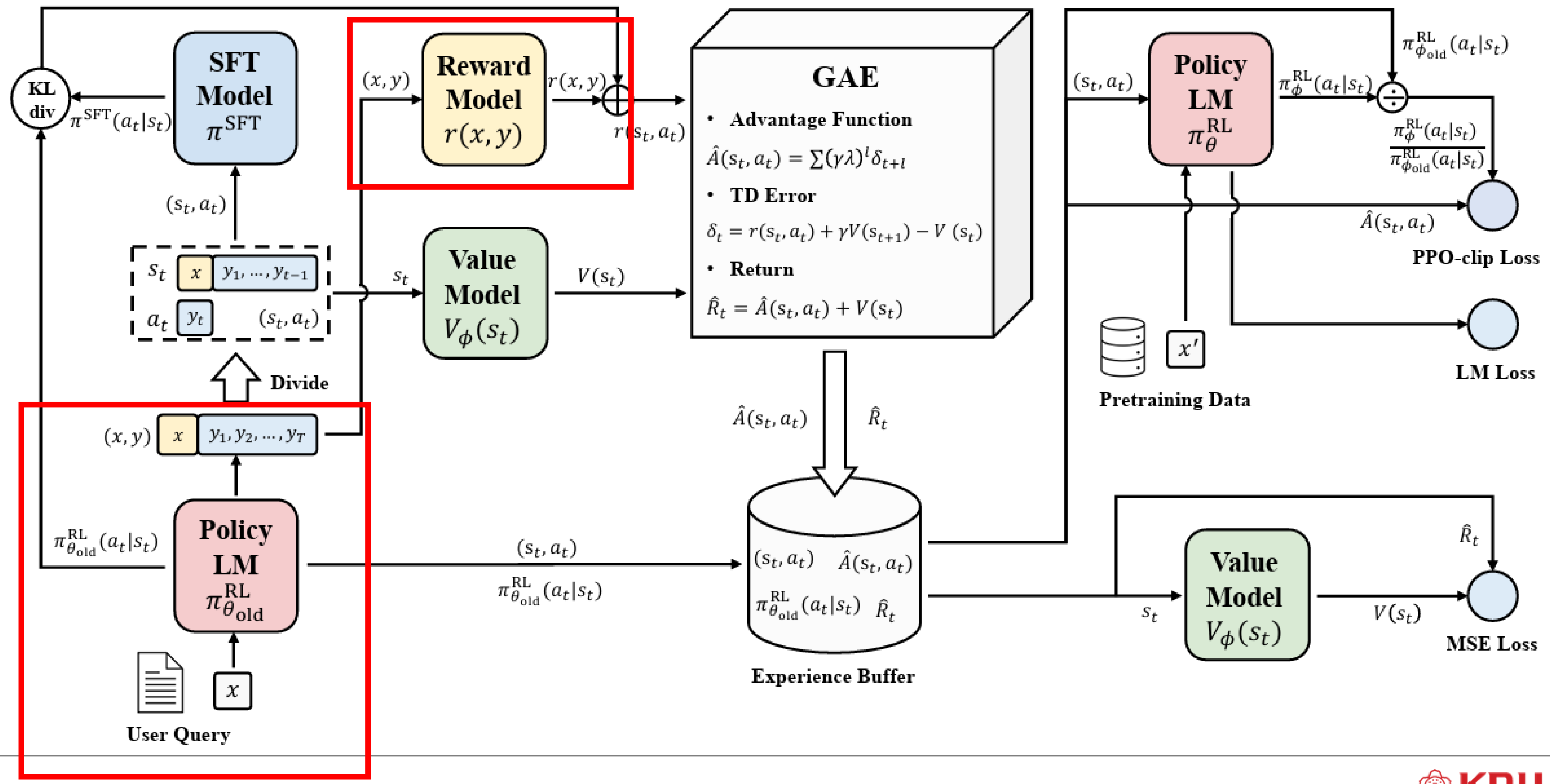
Reinforcement Learning (Policy based text generation)



Policy based RL

- *state* : 토큰 시퀀스(Token Sequence) 또는 이전 문장이 상태
- *action* : 다음 단어(Token)
- BLEU : BLEU 점수

Secrets of RLHF in Large Language Models (arXiv 2023)



PPO : Proximal Policy Optimization Algorithms

- **Policy Gradient Methods :** $L^{PG}(\theta) = \hat{\mathbb{E}}_t \left[\log \pi_\theta(a_t \mid s_t) \hat{A}_t \right]$
  - policy : $\pi_\theta(a_t|s_t)$
  - advantage : $\hat{A}_t$
- **TRPO:** Trust Region Policy Optimization(ICML 2015) Important sampling

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

- **PPO:** Proximal Policy Optimization Algorithms(arxiv 2017)

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min(\boxed{r_t(\theta)} \hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon) \hat{A}_t) \right] \qquad \boxed{r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}}$$

- **RLHF(PPO) :** Training language models to follow instructions with human feedback(NeurIPS 2022)

$$\text{objective}(\phi) = E_{(x,y)\sim D_{\pi_\phi^{\text{RL}}}} \left[r_\theta(x,y) - \beta \log\left(\pi_\phi^{\text{RL}}(y \mid x) / \pi^{\text{SFT}}(y \mid x) \right) \right] + \gamma E_{x \sim D_{\text{pretrain}}} \left[\log(\pi_\phi^{\text{RL}}(x)) \right]$$

$(\epsilon = 0.2, \beta = 0.02, \gamma = 27.8)$

- Policy?
- **Advantage?**
  - **GAE, TD Error**
- **Important sampling?**

Reinforcement Learning(Policy-based Learning)

GAE

- **Advantage Function**

$\hat{A}(s_t, a_t) = \sum(\gamma\lambda)^l \delta_{t+l}$

- **TD Error**

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

- **Return**

$\hat{R}_t = \hat{A}(s_t, a_t) + V(s_t)$

- **GAE**(Generalized Advantage Estimation)
  - 강화학습에서 **Advantage Function**($\hat{A}_t$)의 Variance를 줄이기 위해 사용됩니다.

    기존의 Advantage Function(Ex:G_t)은 Variance가 높아서 학습이 불안정할 수 있는데, GAE는 **TD Error**를 시간에 따라 **감쇠**시켜 더 안정적이고 효율적으로 학습합니다.
  - **TD Error :** 현재 상태에서 예측된 보상과 미래의 실제 경험한 보상의 차이
    - $\delta_t = r(s_t, a_t) + \gamma V(s_{t+1}) - V(s_t)$

      미래의 보상 현재 상태
  - $\hat{A}_t(s_t, a_t) = \sum(\gamma\lambda)^l \delta_{t+l}$
    - γ : 미래의 보상에 대한 **가치**를 얼마나 중요하게 생각할지를 결정($0 \le \gamma \le 1$)
    - λ : TD Error의 **누적**을 제어 ($0 \le \lambda \le 1$)

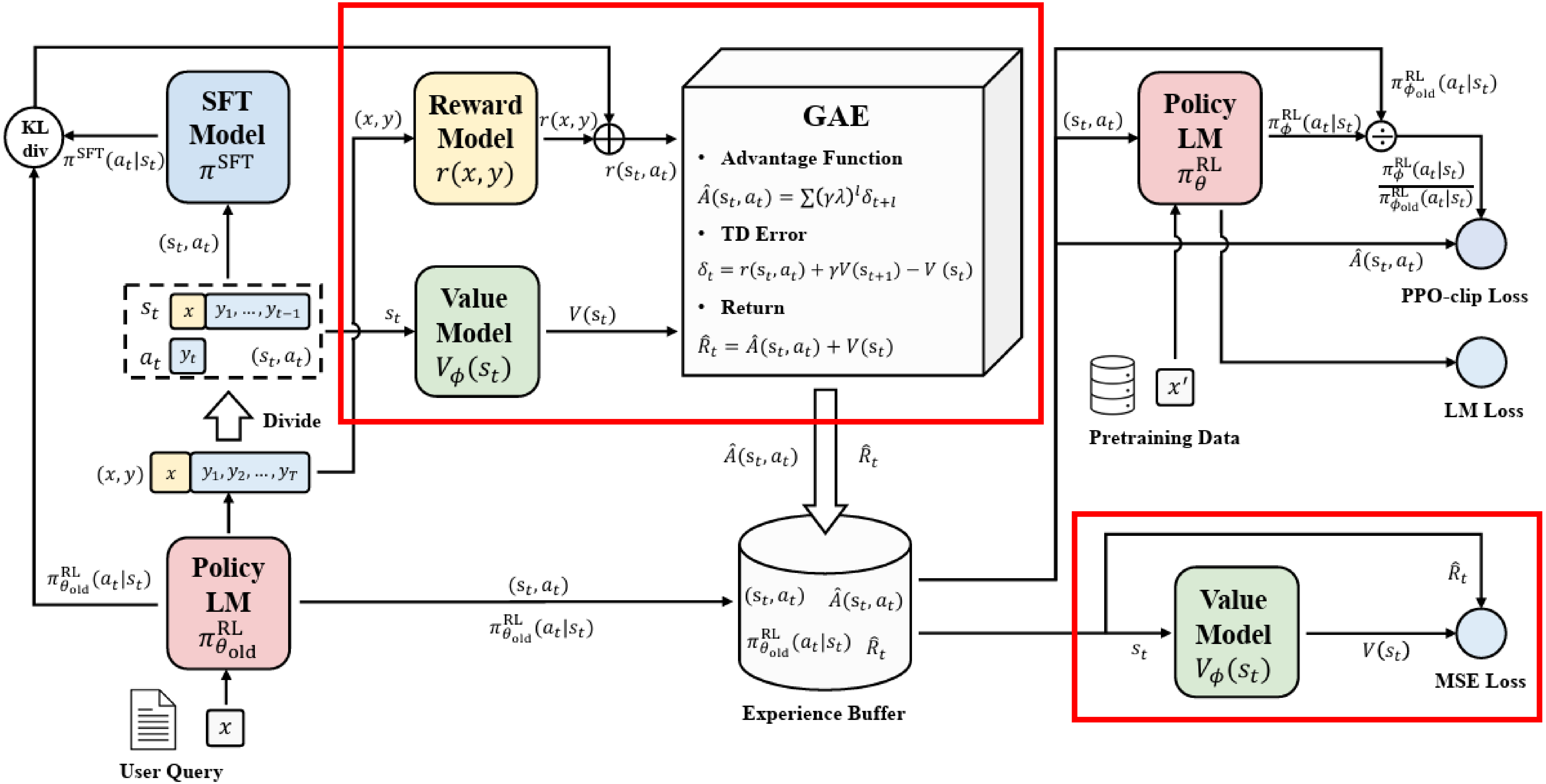
- $\mathrm{R}(\mathrm{s_t}, \mathrm{a_t}) = -1, \mathrm{V}(\mathrm{s_{t+1}}) = 6, \mathrm{V}(\mathrm{s_t}) = 5$
  - $\delta_t = 0$ *예측이 정확함.*
- $\mathrm{R}(\mathrm{s_t}, \mathrm{a_t}) = -1, \mathrm{V}(\mathrm{s_{t+1}}) = 8, \mathrm{V}(\mathrm{s_t}) = 5$
  - $\delta_t > 0$: *예측*보다 좋음 → *Value function 증가*
- $\mathrm{R}(\mathrm{s_t}, \mathrm{a_t}) = -1, \mathrm{V}(\mathrm{s_{t+1}}) = 4, \mathrm{V}(\mathrm{s_t}) = 5,$
  - $\delta_t < 0$: *예측*보다 나쁨 → *Value function 감소*

- $\hat{A}_t(s_t, a_t) = \delta_t + (\gamma\lambda)^1 \delta_{t+1} + (\gamma\lambda)^2 \delta_{t+2} + \cdots$

  $= [r(s_t, q_t) + \gamma V(s_{t+1}) - V(s_t)] + [r(s_{t+1}, q_{t+1}) + \gamma V(s_{t+2}) - V(s_{t+1})] + \ldots$

★ 즉... 현재 상태 예측된 보상 에서 여러 누적 **TD Error**를 합쳐서**(GAE)** Return값을 만든다

Secrets of RLHF in Large Language Models (arXiv 2023)



PPO : Proximal Policy Optimization Algorithms

- **Policy Gradient Methods :** $L^{PG}(\theta) = \hat{\mathbb{E}}_t\left[\log \pi_\theta(a_t \mid s_t)\hat{A}_t\right]$
  - policy : $\pi_\theta(a_t|s_t)$
  - advantage : $\hat{A}_t$
- **TRPO:** Trust Region Policy Optimization(ICML 2015) Important sampling

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

- **PPO:** Proximal Policy Optimization Algorithms(arxiv 2017)

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

$$\boxed{r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}}$$

- **RLHF(PPO) :** Training language models to follow instructions with human feedback(NeurIPS 2022)

$$\text{objective}(\phi) = E_{(x,y)\sim D_{\pi_\phi^{\mathrm{RL}}}}\left[r_\theta(x,y) - \beta \log\left(\pi_\phi^{\mathrm{RL}}(y \mid x)/\pi^{\mathrm{SFT}}(y \mid x)\right)\right] + \gamma E_{x\sim D_{\text{pretrain}}}\left[\log(\pi_\phi^{\mathrm{RL}}(x))\right]$$

$(\epsilon = 0.2, \beta = 0.02, \gamma = 27.8)$

- **Policy?**
- **Advantage?**
- **Important sampling?**

PPO : Proximal Policy Optimization Algorithms

Problems in Policy-based RL for Text Learning

- **Sample**
  - 학습을 위해 sampling 해야 할 Text(s_t, a_t)가 너무 많음
  - 업데이트 후 새로운 Policy(π_θ)에서 sampling을 해야함

- **Reward**(Advantage)
  - Reward 의 입력인 sampling text의 특성으로 인해 variance가 높음

Solving with PPO

- Sample variance → Importance sampling
- Cliping
- Reward variance → GAE

PPO : Proximal Policy Optimization Algorithms

- **Importance sampling**
  - 확률분포 $P(x)$에서 sampling하기 어렵지만, 유사한 분포 $Q(x)$에서는 샘플링이 쉽고, 이를 통해 $P(x)$ 에 대한 기댓값을 추정할 때 사용됩니다.

  - $Q(x)$ 에서 sampling한 후 비율 $\frac{P(x)}{Q(x)}$를 곱해 보정합니다.

$$\mathbb{E}_P[f(x)] = \sum_x f(x)\frac{P(x)}{Q(x)}Q(x)$$

$$= \int f(x)\frac{P(x)}{Q(x)}Q(x)\,dx$$

PPO : Proximal Policy Optimization Algorithms

- **Importance sampling : 주사위 예제**
  - 6면 주사위를 던질 때, 나오는 숫자가 4 이상일 확률을 계산한다고 가정합니다.
    - 실제 분포 $P(x)$: 각 숫자(1~6)는 $\frac{1}{6}$.
    - 중요도 분포 $Q(x)$: 예를 들어, $Q(x)$를 다음과 같이 설정:
      - $x = 1, 2, 3$: 각각 $\frac{1}{12}$
      - $x = 4, 5, 6$: 각각 $\frac{1}{4}$
        (이렇게 하면 $\frac{1}{12} \times 3 + \frac{1}{4} \times 3 = 1$)

PPO : Proximal Policy Optimization Algorithms

- **Importance sampling : 주사위 예제**

이제 Importance Sampling을 통해 $\mathbb{E}_P[f(x)]$를 구해보겠습니다. 여기서:

- $f(x) = 1$ if $x \geq 4$, otherwise $f(x) = 0$. 목표는 $P(x \geq 4)$이므로, $f(x) = 1$인 경우만 고려합니다. 따라서:

$$\mathbb{E}_P[f(x)] = \sum_{x=4}^{6} \frac{P(x)}{Q(x)} Q(x)$$

각 경우를 계산해보면:

- $x = 4$: $\frac{1/6}{1/4} = \frac{2}{3}$
- $x = 5$: $\frac{1/6}{1/4} = \frac{2}{3}$
- $x = 6$: $\frac{1/6}{1/4} = \frac{2}{3}$

각각의 $Q(x)$는 $\frac{1}{4}$이므로:

$$\mathbb{E}_P[f(x)] = \frac{2}{3} \times \frac{1}{4} + \frac{2}{3} \times \frac{1}{4} + \frac{2}{3} \times \frac{1}{4}$$

$$= 3 \times \frac{2}{3} \times \frac{1}{4} = \frac{1}{2}$$

PPO : Proximal Policy Optimization Algorithms

- **Policy Gradient Methods :** $L^{PG}(\theta) = \hat{\mathbb{E}}_t\left[\log \pi_\theta(a_t \mid s_t)\hat{A}_t\right]$
  - policy : $\pi(a_t|s_t)$
  - advantage : $\hat{A}_t$

- **TRPO:** Trust Region Policy Optimization(ICML 2015)

Important sampling

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

- **PPO:** Proximal Policy Optimization Algorithms(arxiv 2017)

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

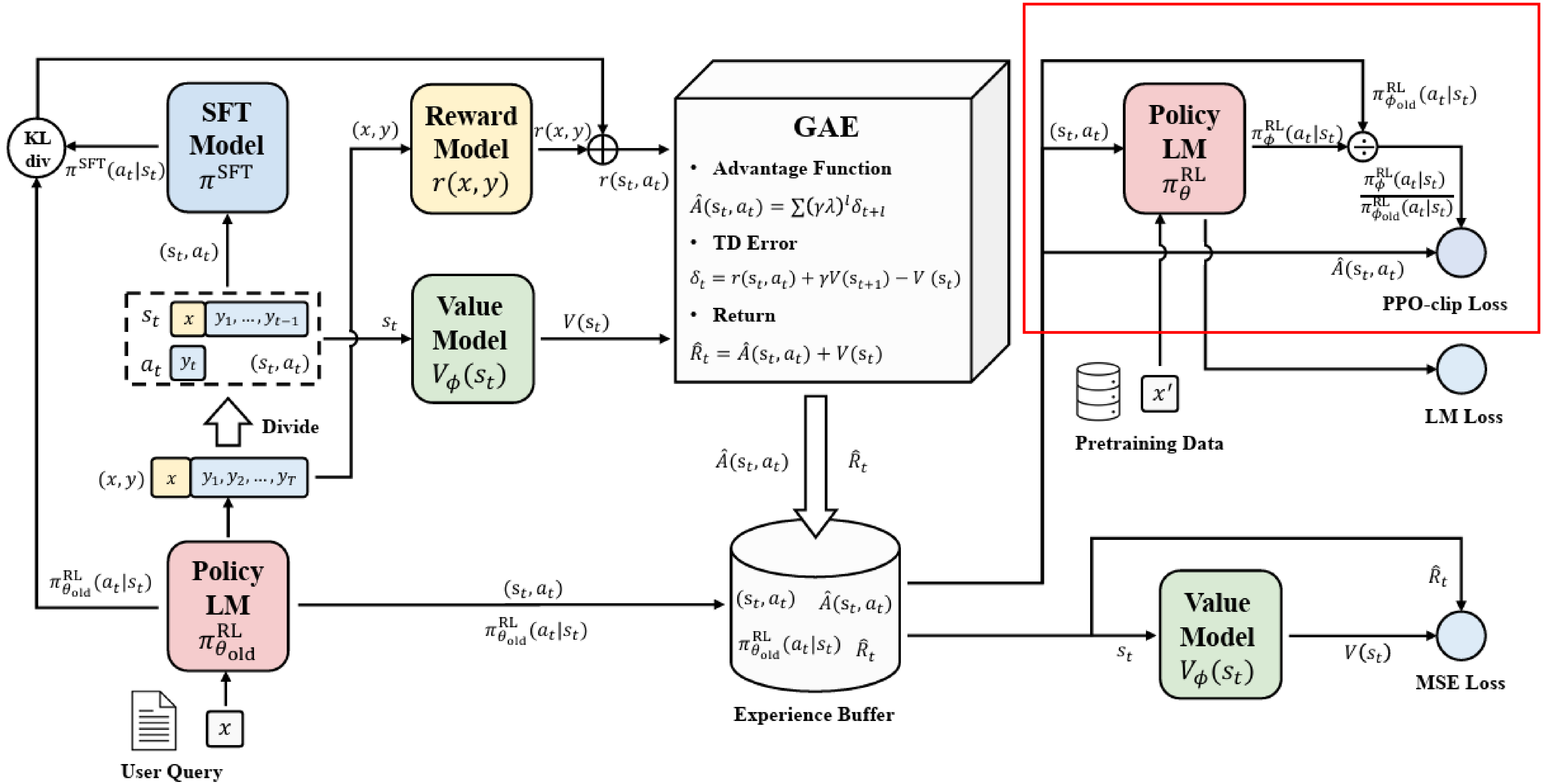
$$\boxed{r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}}$$

- **RLHF(PPO) :** Training language models to follow instructions with human feedback

$$\text{objective}(\phi) = E_{(x,y)\sim D_{\pi_\phi^{\text{RL}}}}\left[r_\theta(x,y) - \beta\log\left(\pi_\phi^{\text{RL}}(y \mid x)/\pi^{\text{SFT}}(y \mid x)\right)\right] + \gamma E_{x\sim D_{\text{pretrain}}}\left[\log(\pi_\phi^{\text{RL}}(x))\right]$$

$(\epsilon = 0.2, \beta = 0.02, \gamma = 27.8)$

Secrets of RLHF in Large Language Models (arXiv 2023)



PPO : Proximal Policy Optimization Algorithms

- **Policy Gradient Methods :** $L^{PG}(\theta) = \hat{\mathbb{E}}_t \left[\log \pi_\theta(a_t \mid s_t) \hat{A}_t \right]$
  - policy : $\pi(a_t|s_t)$
  - advantage : $\hat{A}_t$

- **TRPO:** Trust Region Policy Optimization(ICML 2015)

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

- **PPO:** Proximal Policy Optimization Algorithms(arxiv 2017)

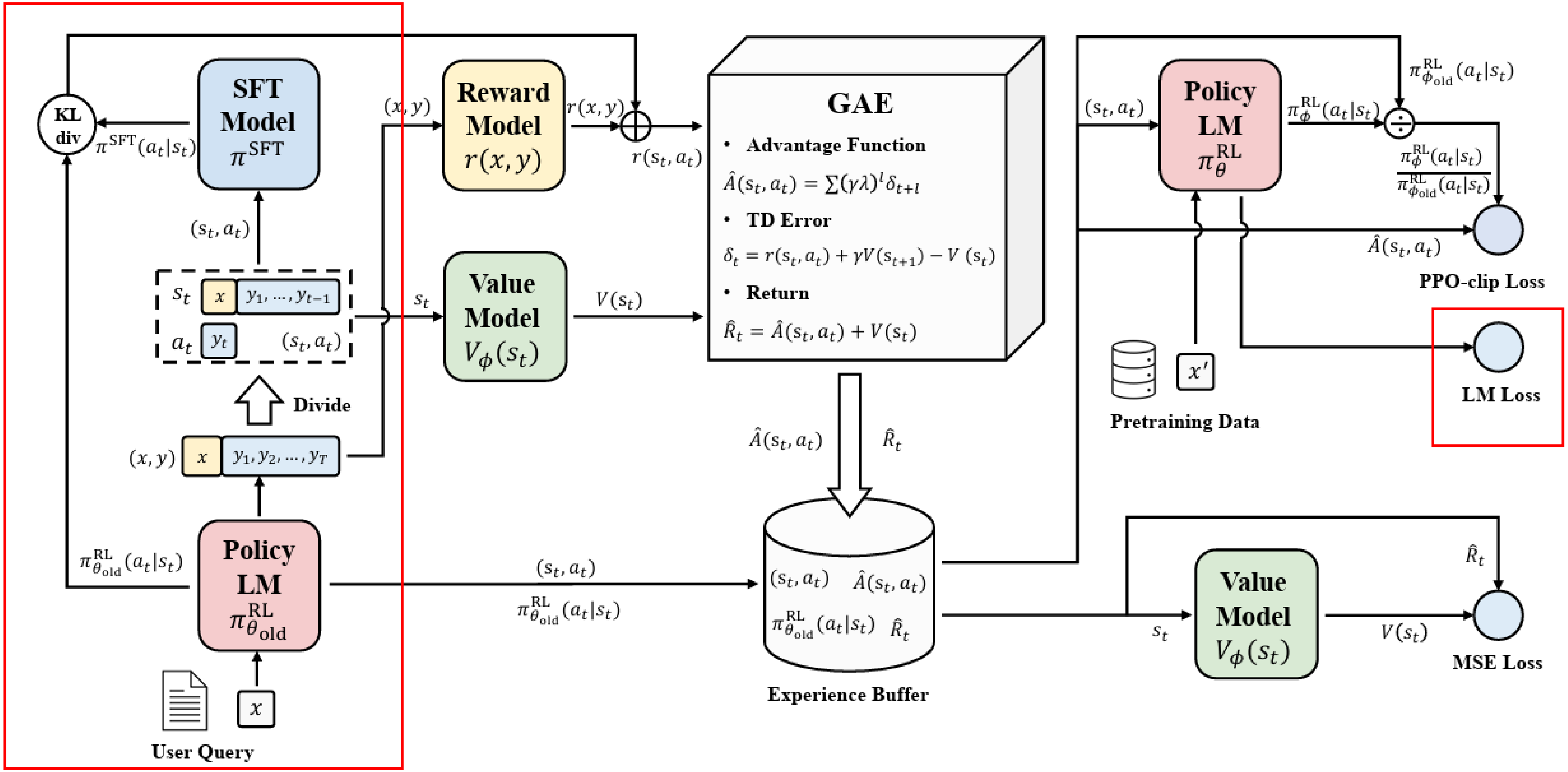
$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min(r_t(\theta)\hat{A}_t, \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t) \right]$$

- **RLHF(PPO) :** Training language models to follow instructions with human feedback

$$\text{objective}(\phi) = E_{(x,y)\sim D_{\pi_\phi^{\mathrm{RL}}}} \left[r_\theta(x,y) - \boxed{\beta \log\left(\pi_\phi^{\mathrm{RL}}(y \mid x) / \pi^{\mathrm{SFT}}(y \mid x)\right)} \right] + \boxed{\gamma E_{x\sim D_{\text{pretrain}}} \left[\log(\pi_\phi^{\mathrm{RL}}(x)) \right]}$$

$(\epsilon = 0.2, \beta = 0.02, \gamma = 27.8)$

Secrets of RLHF in Large Language Models (arXiv 2023)



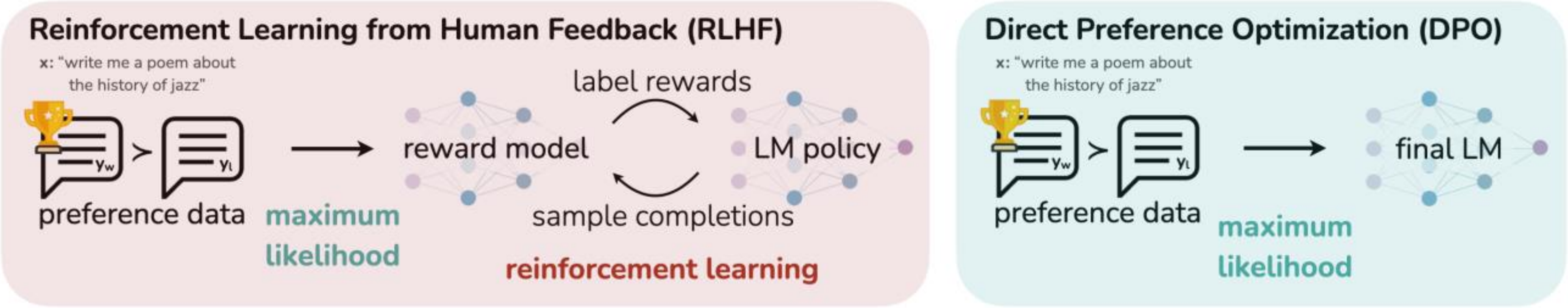
After InstructGPT

- **DPO :** Direct Preference Optimization: Your Language Model is Secretly a Reward Model (NeurIPS 2023)

- **DeepSeek-R1 -** GRPO : Group Relative Policy Optimization

DPO : Direct Preference Optimization: Your Language Model is Secretly a Reward Model

(NeurIPS 2023)



$$\mathcal{L}_{\mathrm{DPO}}(\pi_\theta; \pi_{\mathrm{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\mathrm{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\mathrm{ref}}(y_l \mid x)} \right) \right]$$

- y_w : "winning" response (the better or more preferred response)
- y_l : "losing" response (the less preferred response)

DPO : Direct Preference Optimization: Your Language Model is Secretly a Reward Model

(NeurIPS 2023)

$$\mathcal{L}_{\mathrm{DPO}}(\pi_\theta; \pi_{\mathrm{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\mathrm{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\mathrm{ref}}(y_l \mid x)} \right) \right]$$

$$\pi^*(y|x) = \frac{1}{z(x)} \pi_{ref}(y|x) \exp\left(\frac{1}{\beta} RM_\phi(x, y)\right)$$

$$\mathbb{E}_{y \sim \pi_\theta}\left[RM_\phi(x, y) - \beta \log\left(\frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)}\right)\right]$$

DPO : Direct Preference Optimization: Your Language Model is Secretly a Reward Model (NeurIPS 2023)

$$\boxed{\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right]}$$

$$\pi^*(y|x) = \frac{1}{z(x)}\pi_{ref}(y|x)\exp(\frac{1}{\beta}RM_\phi(x,y)) \rightarrow$$

$$\log \pi^*(y \mid x) = \log \pi_{ref}(y \mid x) + \frac{1}{\beta}RM_\phi(x,y)$$

$$RM_\phi(x,y) = \beta\left[\log \pi^*(y \mid x) - \log \pi_{ref}(y \mid x)\right]$$

$$\mathbb{E}_{y\sim\pi_\theta}[RM_\phi(x,y) - \beta\log(\frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)})]$$

$$\mathbb{E}_{y\sim\pi_\theta}\left[\beta\left[\log \pi^*(y \mid x) - \log \pi_{ref}(y \mid x)\right] - \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{ref}(y \mid x)}\right]$$

$$\downarrow$$

$$\mathbb{E}_{y\sim\pi_\theta}\left[\beta\left(\log \pi^*(y \mid x) - \log \pi_\theta(y \mid x)\right)\right]$$

GRPO: Group Relative Policy optimization

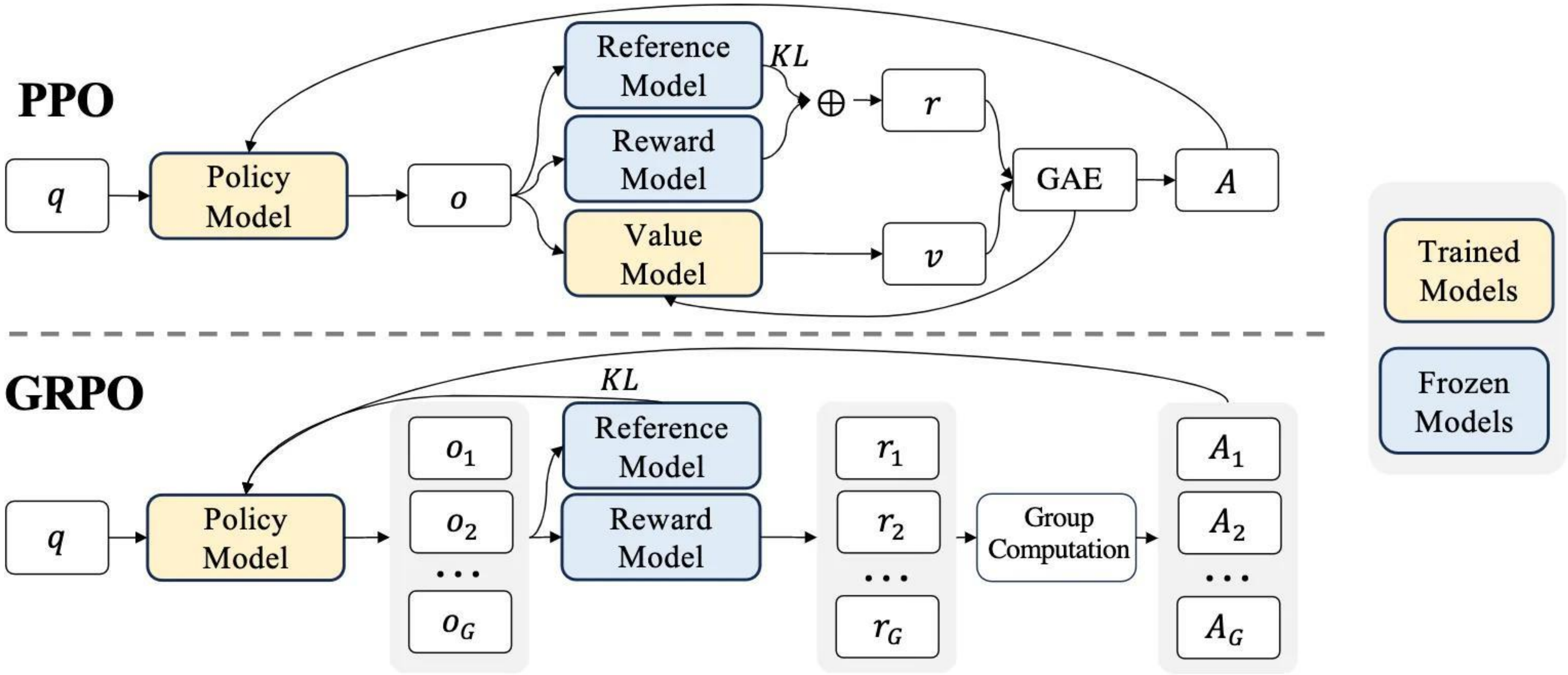


Figure 4 | Demonstration of PPO and our GRPO. GRPO foregoes the value model, instead estimating the baseline from group scores, significantly reducing training resources.

GRPO: Group Relative Policy optimization

- **Objective Function:**

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)]$$

$$\frac{1}{G}\sum_{i=1}^{G}\left(\min\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}A_i, \text{clip}\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}, 1-\varepsilon, 1+\varepsilon\right)A_i\right) - \beta\mathbb{D}_{KL}\left(\pi_\theta||\pi_{ref}\right)\right),$$

$$\mathbb{D}_{KL}\left(\pi_\theta||\pi_{ref}\right) = \frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - \log\frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - 1,$$

- **Advantage :**

$$A_i = \frac{r_i - \text{mean}(\{r_1, r_2, \cdots, r_G\})}{\text{std}(\{r_1, r_2, \cdots, r_G\})}.$$