

# 大语言模型中的强化学习

Fangfan Yu

Wuhan University

September 13, 2024



① Human Alignment

② RLHF

③ Exploration of PPO

④ 其他 RLHF 工作

⑤ RLHF and SFT

⑥ 非强化学习的对齐方法

- ▶ 大型语言模型可能会表现出无意的行为。

- ▶ 大型语言模型可能会表现出无意的行为。
- ▶ 需要使大型语言模型（LLMs）与人类价值观保持一致，例如：有帮助、诚实和无害（3H）。

- ▶ 大型语言模型可能会表现出无意的行为。
- ▶ 需要使大型语言模型（LLMs）与人类价值观保持一致，例如：有帮助、诚实和无害（3H）。
- ▶ OpenAI 和 Anthropic 验证了 RLHF 是将语言模型与用户意图对齐的有效途径，适用于广泛的任务。

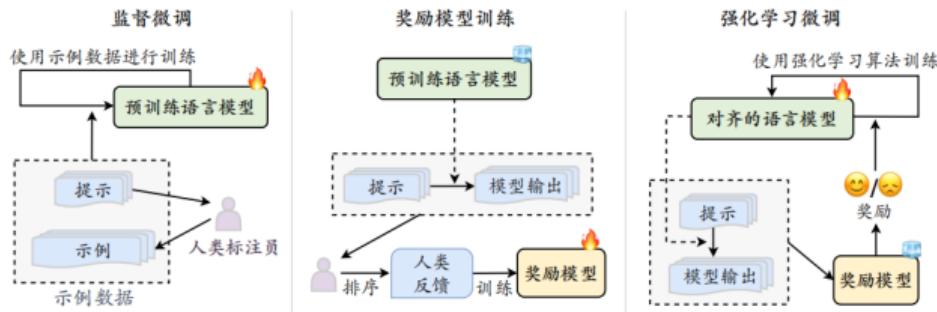
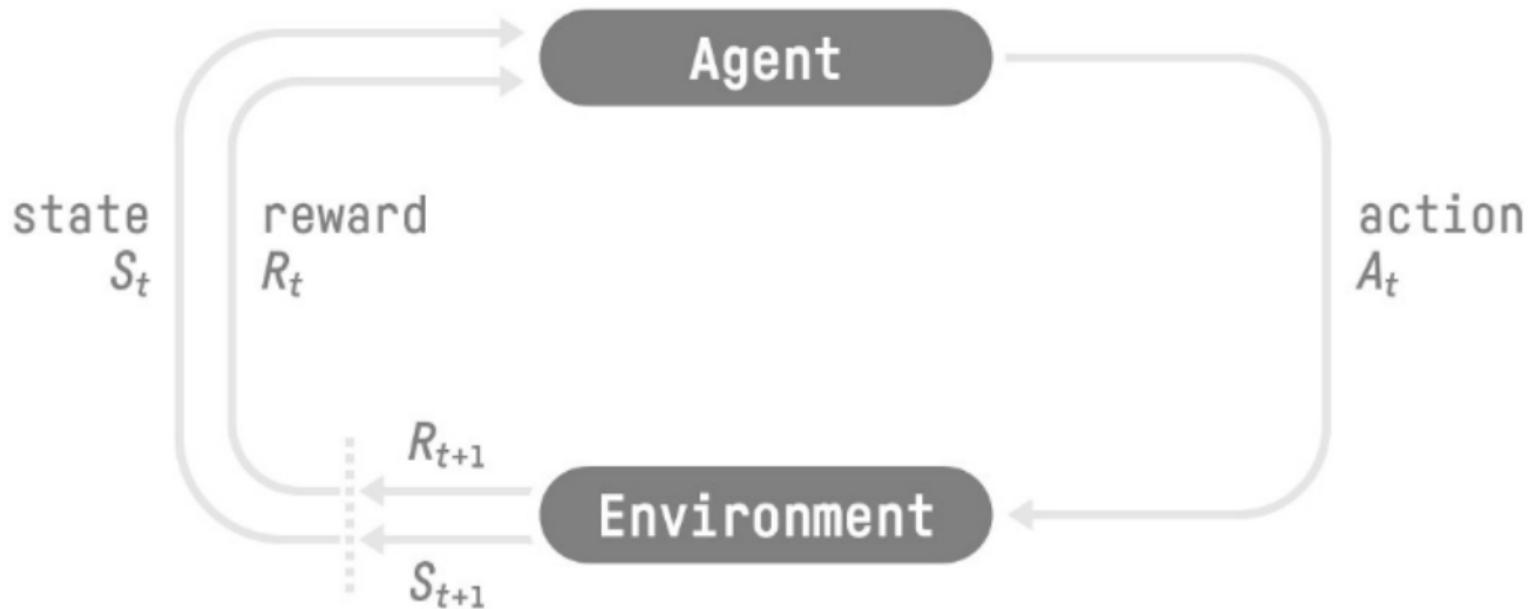


Figure: 基于人类反馈的强化学习的工作流程

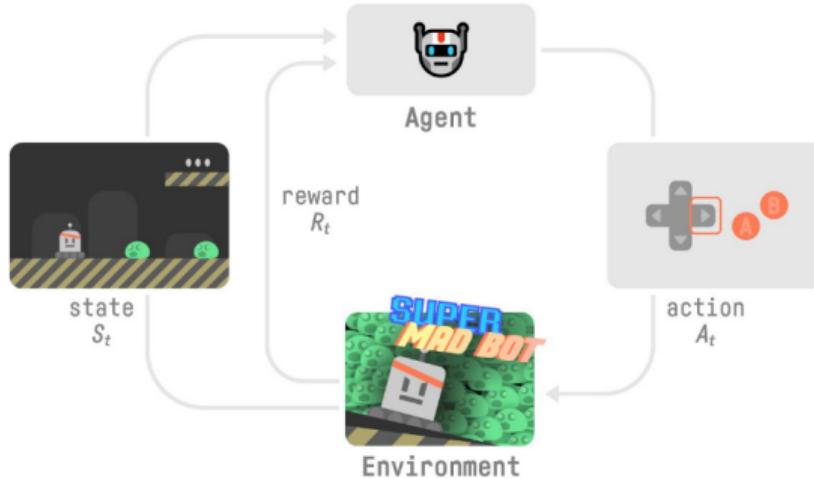




- ▶ 我们的代理从环境中接收到状态  $S_0$  – 我们接收到游戏的第一帧（环境）。



- ▶ 我们的代理从环境中接收到状态  $S_0$  — 我们接收到游戏的第一帧（环境）。
- ▶ 基于状态  $S_0$ ，智能体采取动作  $A_0$  —— 我们的智能体会向右移动。



- ▶ 我们的代理从环境中接收到状态  $S_0$  — 我们接收到游戏的第一帧（环境）。
- ▶ 基于状态  $S_0$ ，智能体采取动作  $A_0$  —— 我们的智能体会向右移动。
- ▶ 环境转移到新状态  $S_1$  —— 新的一帧。



- ▶ 我们的代理从环境中接收到状态  $S_0$  — 我们接收到游戏的第一帧（环境）。
- ▶ 基于状态  $S_0$ ，智能体采取动作  $A_0$  —— 我们的智能体会向右移动。
- ▶ 环境转移到新状态  $S_1$  —— 新的一帧。
- ▶ 环境给予智能体奖励  $R_1$  —— 我们没有死亡（正向奖励 +1）。

## 定义

马尔可夫决策过程 (MDPs)。一个 MDP 是一个五元组， $\langle S, A, R, P, \rho_0 \rangle$ ，其中

- ▶  $S$  是所有有效状态的集合，

## 定义

马尔可夫决策过程 (MDPs)。一个 MDP 是一个五元组， $\langle S, A, R, P, \rho_0 \rangle$ ，其中

- ▶  $S$  是所有有效状态的集合，
- ▶  $A$  是所有有效动作的集合，

马尔可夫决策过程 (MDPs)。一个 MDP 是一个五元组， $\langle S, A, R, P, \rho_0 \rangle$ ，其中

- ▶  $S$  是所有有效状态的集合，
- ▶  $A$  是所有有效动作的集合，
- ▶  $R : S \times A \times S \rightarrow \mathbb{R}$  是奖励函数，其中  $r_{t+1} = R(s_t, a_t, s_{t+1})$ ，

## 定义

马尔可夫决策过程 (MDPs)。一个 MDP 是一个五元组， $\langle S, A, R, P, \rho_0 \rangle$ ，其中

- ▶  $S$  是所有有效状态的集合，
- ▶  $A$  是所有有效动作的集合，
- ▶  $R : S \times A \times S \rightarrow \mathbb{R}$  是奖励函数，其中  $r_{t+1} = R(s_t, a_t, s_{t+1})$ ，
- ▶  $P : S \times A \rightarrow \mathcal{P}(S)$  是状态转移概率函数，其中  $P(s'|s, a)$  表示从状态  $s$  采取动作  $a$  后转移到状态  $s'$  的概率，

马尔可夫决策过程 (MDPs)。一个 MDP 是一个五元组， $\langle S, A, R, P, \rho_0 \rangle$ ，其中

- ▶  $S$  是所有有效状态的集合，
- ▶  $A$  是所有有效动作的集合，
- ▶  $R : S \times A \times S \rightarrow \mathbb{R}$  是奖励函数，其中  $r_{t+1} = R(s_t, a_t, s_{t+1})$ ，
- ▶  $P : S \times A \rightarrow \mathcal{P}(S)$  是状态转移概率函数，其中  $P(s'|s, a)$  表示从状态  $s$  采取动作  $a$  后转移到状态  $s'$  的概率，
- ▶  $\rho_0$  是初始状态分布。

- ▶ 一个策略可以是确定性的，在这种情况下，它用以下方式表示  $\mu$ :

$$a_t = \mu(s_t)$$

或者它可能是随机的，在这种情况下用以下符号表示:  $\pi$ :

$$a_t \sim \pi(\cdot | s_t)$$

- 一个策略可以是确定性的，在这种情况下，它用以下方式表示  $\mu$ :

$$a_t = \mu(s_t)$$

或者它可能是随机的，在这种情况下用以下符号表示:  $\pi$ :

$$a_t \sim \pi(\cdot | s_t)$$



$$G_\tau^\pi \triangleq \sum_{t=\tau}^T \gamma^{k-1} R_t$$

- 一个策略可以是确定性的，在这种情况下，它用以下方式表示  $\mu$ :

$$a_t = \mu(s_t)$$

或者它可能是随机的，在这种情况下用以下符号表示:  $\pi$ :

$$a_t \sim \pi(\cdot | s_t)$$



$$G_\tau^\pi \triangleq \sum_{t=\tau}^T \gamma^{t-1} R_t$$



$$V^\pi(x) \triangleq \mathbb{E} \left[ \sum_{t=1}^T \gamma^{t-1} R_t \mid X_1 = x \right]$$

- 一个策略可以是确定性的，在这种情况下，它用以下方式表示  $\mu$ :

$$a_t = \mu(s_t)$$

或者它可能是随机的，在这种情况下用以下符号表示:  $\pi$ :

$$a_t \sim \pi(\cdot | s_t)$$

- $$G_\tau^\pi \triangleq \sum_{t=\tau}^T \gamma^{t-1} R_t$$
- $$V^\pi(x) \triangleq \mathbb{E} \left[ \sum_{t=1}^T \gamma^{t-1} R_t \mid X_1 = x \right]$$
- $$Q^\pi(x, a) \triangleq \mathbb{E} \left[ \sum_{t=1}^{\infty} \gamma^{t-1} R_t \mid X_1 = x, A_1 = a \right]$$

- ▶ 状态 ( $s$ ): 文本生成过程中的某一步的上下文。

- ▶ 状态 ( $s$ ): 文本生成过程中的某一步的上下文。
- ▶ 动作 ( $a$ ): 模型在当前步骤生成的输出, 例如一个词或一个标记 (Token)。

- ▶ 状态 ( $s$ ): 文本生成过程中的某一步的上下文。
- ▶ 动作 ( $a$ ): 模型在当前步骤生成的输出, 例如一个词或一个标记 (Token)。
- ▶ 策略 ( $\pi(a|s)$ ): 给定当前状态 (上下文), 模型生成各种可能动作 (标记) 的概率分布。

- ▶ 状态 ( $s$ ): 文本生成过程中的某一步的上下文。
- ▶ 动作 ( $a$ ): 模型在当前步骤生成的输出, 例如一个词或一个标记 (Token)。
- ▶ 策略 ( $\pi(a|s)$ ): 给定当前状态 (上下文), 模型生成各种可能动作 (标记) 的概率分布。
- ▶ 奖励 ( $r$ ): 由奖励模型提供的标量值, 用于评估生成的动作或序列的质量。

- 三个阶段：监督微调（SFT）、奖励模型（RM）训练，以及基于奖励模型的近端策略优化（PPO）。

- ▶ 三个阶段：监督微调（SFT）、奖励模型（RM）训练，以及基于奖励模型的近端策略优化（PPO）。
- ▶ SFT 阶段：模型通过模仿人工标注的对话示例，学习进行类似人类的通用对话。

- ▶ 三个阶段：监督微调（SFT）、奖励模型（RM）训练，以及基于奖励模型的近端策略优化（PPO）。
- ▶ SFT 阶段：模型通过模仿人工标注的对话示例，学习进行类似人类的通用对话。
- ▶ 奖励模型：模型基于人类反馈，学习比较不同回复的偏好。

- ▶ 三个阶段：监督微调（SFT）、奖励模型（RM）训练，以及基于奖励模型的近端策略优化（PPO）。
- ▶ SFT 阶段：模型通过模仿人工标注的对话示例，学习进行类似人类的通用对话。
- ▶ 奖励模型：模型基于人类反馈，学习比较不同回复的偏好。
- ▶ PPO 阶段：模型根据奖励模型的反馈进行更新，通过探索和利用来寻找最优策略。

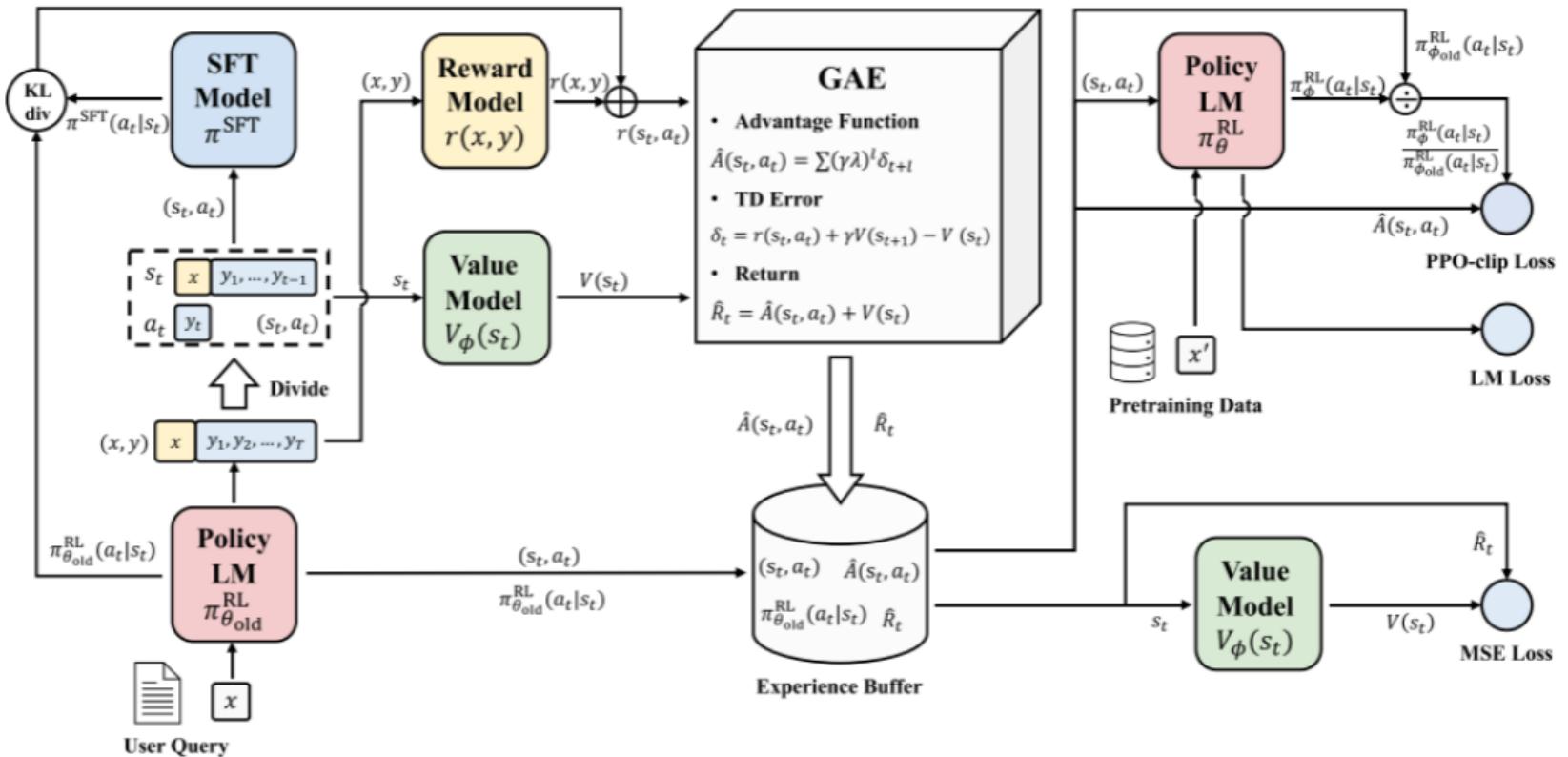


Figure: RLHF workflow

## ► 标注人类选择

- ▶ 标注人类选择
- ▶ 人类反馈形式：
  - 1、基于评分的人类反馈.

- ▶ 标注人类选择
- ▶ 人类反馈形式：
  - 1、基于评分的人类反馈.

- ▶ 标注人类选择
- ▶ 人类反馈形式：
  - 1、基于评分的人类反馈.
  - 2、基于排序的人类反馈.
- ▶ Reward Modeling(RM): 替代人类在 RLHF 训练过程中实时提供反馈.

- ▶ 标注人类选择
- ▶ 人类反馈形式：
  - 1、基于评分的人类反馈.
  - 2、基于排序的人类反馈.
- ▶ Reward Modeling(RM): 替代人类在 RLHF 训练过程中实时提供反馈.
- ▶ **训练方法：**
  - 1、打分式.

- ▶ 标注人类选择
- ▶ 人类反馈形式：
  - 1、基于评分的人类反馈.
  - 2、基于排序的人类反馈.
- ▶ Reward Modeling(RM): 替代人类在 RLHF 训练过程中实时提供反馈.
- ▶ 训练方法：
  - 1、打分式.

- ▶ 标注人类选择
- ▶ 人类反馈形式：
  - 1、基于评分的人类反馈.
  - 2、基于排序的人类反馈.
- ▶ Reward Modeling(RM): 替代人类在 RLHF 训练过程中实时提供反馈.
- ▶ 训练方法：
  - 1、打分式.
  - 2、对比式.

- ▶ 标注人类选择
- ▶ 人类反馈形式：
  - 1、基于评分的人类反馈.
  - 2、基于排序的人类反馈.
- ▶ Reward Modeling(RM): 替代人类在 RLHF 训练过程中实时提供反馈.
- ▶ 训练方法：
  - 1、打分式.
  - 2、对比式.
  - 3、排序式.

- 每对偏好样本和不偏好样本的建模损失是:

$$\mathcal{L}(\psi) = \log \sigma(r(x, y_w) - r(x, y_l)),$$

- ▶ 每对偏好样本和不偏好样本的建模损失是:

$$\mathcal{L}(\psi) = \log \sigma(r(x, y_w) - r(x, y_l)),$$

- ▶ 提升:

$$\begin{aligned}\mathcal{L}(\psi) = & -\lambda \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}_{rm}} [\log \sigma(r(x, y_w) - r(x, y_l))] \\ & + \beta_{rm} \mathbb{E}_{(x, y_w) \sim \mathcal{D}_{rm}} [\log(r'(x, y_w))],\end{aligned}$$

- ▶ 每对偏好样本和不偏好样本的建模损失是:

$$\mathcal{L}(\psi) = \log \sigma(r(x, y_w) - r(x, y_l)),$$

- ▶ 提升:

$$\begin{aligned}\mathcal{L}(\psi) = & -\lambda \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}_{\text{rm}}} [\log \sigma(r(x, y_w) - r(x, y_l))] \\ & + \beta_{\text{rm}} \mathbb{E}_{(x, y_w) \sim \mathcal{D}_{\text{rm}}} [\log(r'(x, y_w))],\end{aligned}$$

- ▶ KL 散度:

$$r_{\text{total}} = r(x, y) - \eta \text{KL} \left( \pi_{\phi}^{\text{RL}}(y|x), \pi^{\text{SFT}}(y|x) \right),$$

- ▶ 对于英文，从原始的 LLaMA-7B 开始，这是一种仅包含解码器的架构. 使用了 HH-RLHF 数据集中的 160k 对样本进行训练.

- ▶ 对于英文，从原始的 LLaMA-7B 开始，这是一种仅包含解码器的架构。使用了 HH-RLHF 数据集中的 160k 对样本进行训练。
- ▶ 对于中文，使用了 OpenChineseLLaMA。该模型通过在中文数据集上的增量预训练开发的，基于 LLaMA-7B 的基础，显著提高了其在中文上的理解和生成能力。

- ▶ 对于英文，从原始的 LLaMA-7B 开始，这是一种仅包含解码器的架构。使用了 HH-RLHF 数据集中的 160k 对样本进行训练。
- ▶ 对于中文，使用了 OpenChineseLLaMA。该模型通过在中文数据集上的增量预训练开发的，基于 LLaMA-7B 的基础，显著提高了其在中文上的理解和生成能力。
- ▶ 学习率设置为  $5e-6$ ，并在前 10% 的步骤中进行预热。使用动态批量方法，而不是固定值，以尽可能平衡每个批次中的标记数量，从而实现更高效和稳定的训练阶段。批量大小根据批次中的标记数量变化，最大值为 128，最小值为 4。将训练步骤固定为 1000，相当于整个训练集的约 1.06 个 epoch。设置  $\beta_{rm} = 1$ ，这表示在整个实验中使用 LM 损失权重来训练奖励模型。

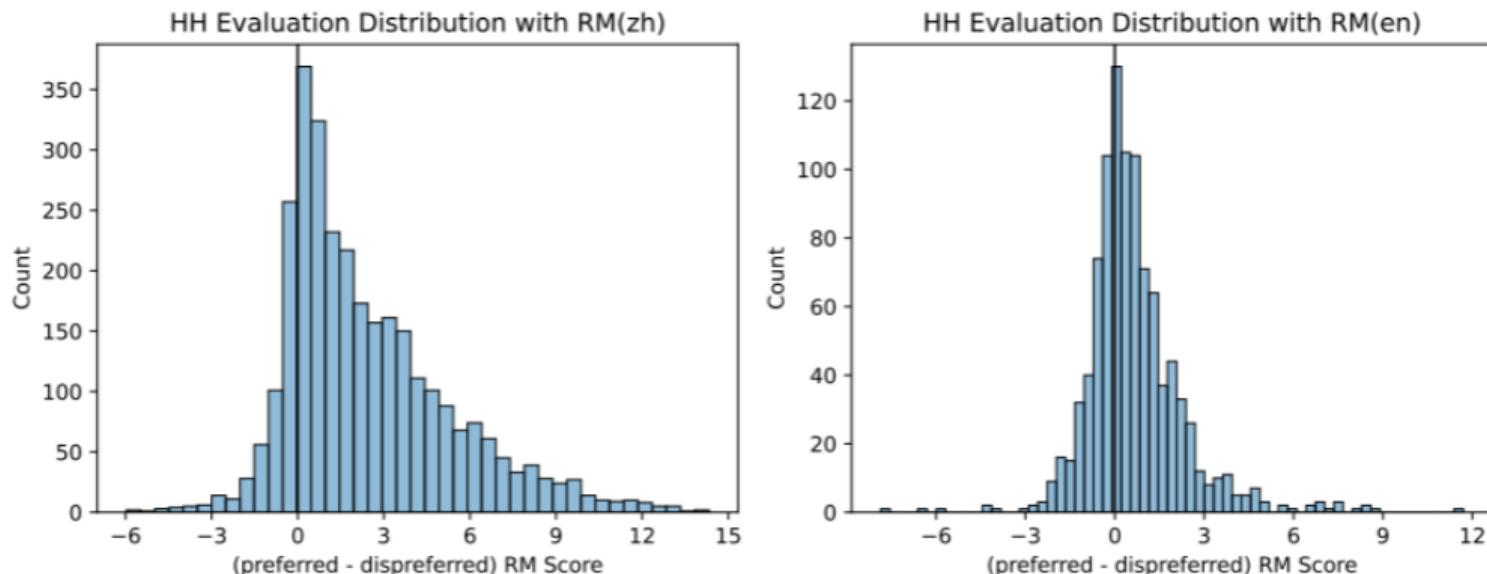


Figure 2: Histograms of the RM predictions for the HH evaluations. The left figure shows the score distribution for a PM trained on manually labeled Chinese data, while the right one shows that of HH<sub>WHLHF</sub> data. Both models roughly align with human preferences, especially the RM trained on HH<sub>WHLHF</sub> data.

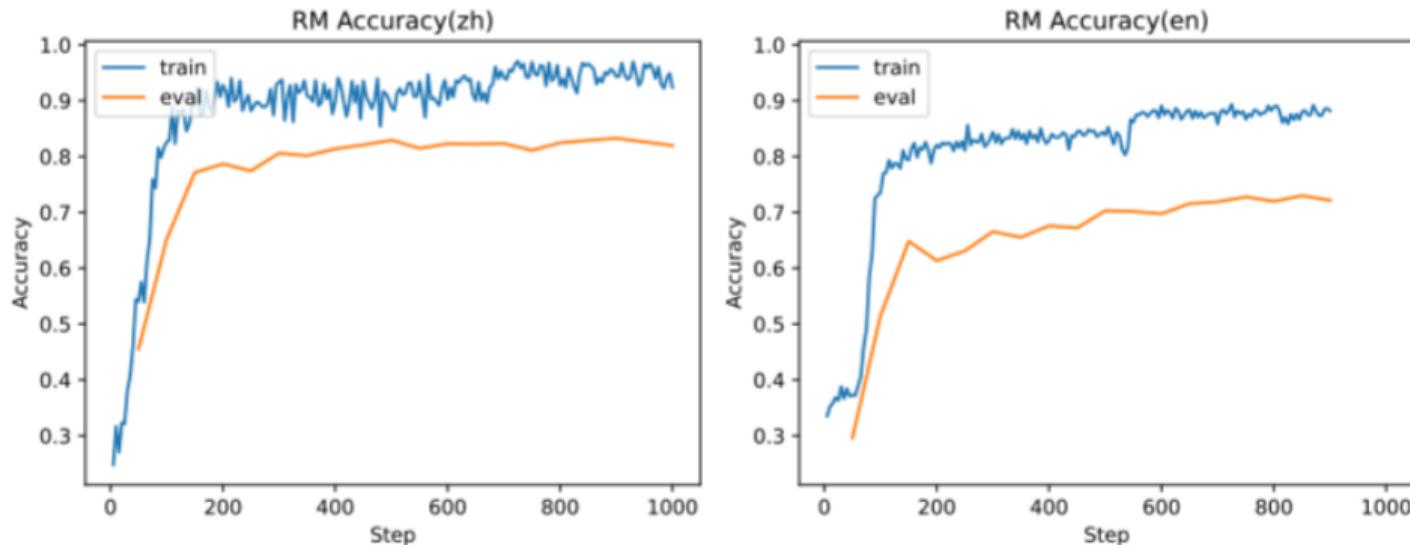


Figure 3: We show the variation of RM accuracy during training. The performance of both models steadily improves on the validation set. The RM trained on Chinese data shows a higher accuracy for the greater dissimilarity between the two responses within a pair in the Chinese data, and becomes relatively easier for the RM to model the distinctive features between them when trained

策略  $\pi$  由参数  $\theta$  参数化，我们将其表示为  $\pi(a|s, \theta)$ 。更新规则为：

$$\theta \leftarrow \theta + \alpha \nabla J(\theta),$$

其中  $\alpha$  是学习率， $J(\theta)$  表示遵循策略  $\pi_\theta$  时的期望回报， $\nabla_\theta J(\theta)$  是策略梯度。

策略  $\pi$  由参数  $\theta$  参数化，我们将其表示为  $\pi(a|s, \theta)$ 。更新规则为：

$$\theta \leftarrow \theta + \alpha \nabla J(\theta),$$

其中  $\alpha$  是学习率， $J(\theta)$  表示遵循策略  $\pi_\theta$  时的期望回报， $\nabla_\theta J(\theta)$  是策略梯度。策略梯度的一般形式可以表示为：

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[ \sum_{t=0}^T \nabla_\theta \log \pi_\theta(a_t | s_t) \Phi_t \right],$$

其中  $\Phi_t$  可以是以下任意一种形式： $\Phi_t = R(\tau)$  或  $\Phi_t = \sum_{t'=t}^T R(s_{t'}, a_{t'})$  或  $\Phi_t = \sum_{t'=t}^T R(s_{t'}, a_{t'}) - b(s_t)$  (其中  $b$  是基线函数)。尽管这些选择会导致不同的方差，但它们都会得到相同的策略梯度期望值。

优势函数  $A(s_t, a_t)$  表示在状态  $s_t$  下采取特定动作  $a_t$  相比在该状态下同一策略下动作的平均质量有多好。因此，

$$\Phi_t = A(s_t, a_t) = Q(s_t, a_t) - V(s_t).$$



优势函数  $A(s_t, a_t)$  表示在状态  $s_t$  下采取特定动作  $a_t$  相比在该状态下同一策略下动作的平均质量有多好。因此，

$$\Phi_t = A(s_t, a_t) = Q(s_t, a_t) - V(s_t).$$

优势函数的估计方法在不同算法中有显著差异，我们引入广义优势估计（GAE）。

优势函数  $A(s_t, a_t)$  表示在状态  $s_t$  下采取特定动作  $a_t$  相比在该状态下同一策略下动作的平均质量有多好。因此，

$$\Phi_t = A(s_t, a_t) = Q(s_t, a_t) - V(s_t).$$

优势函数的估计方法在不同算法中有显著差异，我们引入广义优势估计（GAE）。

GAE 算法使用时序差分（TD）回报和完整的蒙特卡洛回报来平衡偏差和方差。

TD- $k$  回报  $\hat{R}_t^k$  是实际奖励和估计回报的组合：

TD- $k$  回报  $\hat{R}_t^k$  是实际奖励和估计回报的组合：

$$\hat{R}_t^k = r_t + \gamma r_{t+1} + \cdots + \gamma^{(k-1)} r_{t+k-1} + \gamma^k V(s_{t+k}),$$

TD- $k$  回报  $\hat{R}_t^k$  是实际奖励和估计回报的组合：

$$\hat{R}_t^k = r_t + \gamma r_{t+1} + \cdots + \gamma^{(k-1)} r_{t+k-1} + \gamma^k V(s_{t+k}),$$

其中  $\gamma$  是折扣因子。使用 TD- $k$  回报的优势估计称为  $k$ -步优势，定义为：

TD- $k$  回报  $\hat{R}_t^k$  是实际奖励和估计回报的组合：

$$\hat{R}_t^k = r_t + \gamma r_{t+1} + \cdots + \gamma^{(k-1)} r_{t+k-1} + \gamma^k V(s_{t+k}),$$

其中  $\gamma$  是折扣因子。使用 TD- $k$  回报的优势估计称为  $k$ -步优势，定义为：

$$\begin{aligned}\hat{A}_t^k &= \hat{R}_t^k - V(s_t) = \sum_{l=1}^k \gamma^{l-1} \delta_{t+l} \\ &= -V(s_t) + r_t + \gamma r_{t+1} + \cdots + \gamma^{k-1} r_{t+k-1} + \gamma^k V(s_{t+k}),\end{aligned}$$

TD- $k$  回报  $\hat{R}_t^k$  是实际奖励和估计回报的组合：

$$\hat{R}_t^k = r_t + \gamma r_{t+1} + \cdots + \gamma^{(k-1)} r_{t+k-1} + \gamma^k V(s_{t+k}),$$

其中  $\gamma$  是折扣因子。使用 TD- $k$  回报的优势估计称为  $k$ -步优势，定义为：

$$\begin{aligned}\hat{A}_t^k &= \hat{R}_t^k - V(s_t) = \sum_{l=1}^k \gamma^{l-1} \delta_{t+l} \\ &= -V(s_t) + r_t + \gamma r_{t+1} + \cdots + \gamma^{k-1} r_{t+k-1} + \gamma^k V(s_{t+k}),\end{aligned}$$

其中  $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$  是 TD 误差。

$$\begin{aligned}
\hat{A}_t^{\text{GAE}(\gamma, \lambda)} &= (1 - \lambda) \left( \hat{A}_t^{(1)} + \lambda \hat{A}_t^{(2)} + \lambda^2 \hat{A}_t^{(3)} + \dots \right) \\
&= (1 - \lambda) \left( \delta_t + \lambda(\delta_t + \gamma \delta_{t+1}) + \lambda^2(\delta_t + \gamma \delta_{t+1} + \gamma^2 \delta_{t+2}) + \dots \right) \\
&= (1 - \lambda) (\delta_t(1 + \lambda + \lambda^2 + \dots) + \gamma \delta_{t+1}(\lambda + \lambda^2 + \lambda^3 + \dots) \\
&\quad + \gamma^2 \delta_{t+2}(\lambda^2 + \lambda^3 + \lambda^4 + \dots) + \dots) \\
&= (1 - \lambda) \left( \delta_t \left( \frac{1}{1 - \lambda} \right) + \gamma \delta_{t+1} \left( \frac{\lambda}{1 - \lambda} \right) + \gamma^2 \delta_{t+2} \left( \frac{\lambda^2}{1 - \lambda} \right) + \dots \right) \\
&= \sum_{l=0}^{\infty} (\gamma \lambda)^l \delta_{t+l}.
\end{aligned}$$

$$\begin{aligned}
\hat{A}_t^{\text{GAE}(\gamma, \lambda)} &= (1 - \lambda) \left( \hat{A}_t^{(1)} + \lambda \hat{A}_t^{(2)} + \lambda^2 \hat{A}_t^{(3)} + \dots \right) \\
&= (1 - \lambda) \left( \delta_t + \lambda(\delta_t + \gamma \delta_{t+1}) + \lambda^2(\delta_t + \gamma \delta_{t+1} + \gamma^2 \delta_{t+2}) + \dots \right) \\
&= (1 - \lambda) (\delta_t(1 + \lambda + \lambda^2 + \dots) + \gamma \delta_{t+1}(\lambda + \lambda^2 + \lambda^3 + \dots) \\
&\quad + \gamma^2 \delta_{t+2}(\lambda^2 + \lambda^3 + \lambda^4 + \dots) + \dots) \\
&= (1 - \lambda) \left( \delta_t \left( \frac{1}{1 - \lambda} \right) + \gamma \delta_{t+1} \left( \frac{\lambda}{1 - \lambda} \right) + \gamma^2 \delta_{t+2} \left( \frac{\lambda^2}{1 - \lambda} \right) + \dots \right) \\
&= \sum_{l=0}^{\infty} (\gamma \lambda)^l \delta_{t+l}.
\end{aligned}$$

**GAE**( $\gamma, 0$ ) :  $\hat{A}_t = \delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$ .

$$\begin{aligned}
\hat{A}_t^{\text{GAE}(\gamma, \lambda)} &= (1 - \lambda) \left( \hat{A}_t^{(1)} + \lambda \hat{A}_t^{(2)} + \lambda^2 \hat{A}_t^{(3)} + \dots \right) \\
&= (1 - \lambda) \left( \delta_t + \lambda(\delta_t + \gamma \delta_{t+1}) + \lambda^2(\delta_t + \gamma \delta_{t+1} + \gamma^2 \delta_{t+2}) + \dots \right) \\
&= (1 - \lambda) (\delta_t (1 + \lambda + \lambda^2 + \dots) + \gamma \delta_{t+1} (\lambda + \lambda^2 + \lambda^3 + \dots) \\
&\quad + \gamma^2 \delta_{t+2} (\lambda^2 + \lambda^3 + \lambda^4 + \dots) + \dots) \\
&= (1 - \lambda) \left( \delta_t \left( \frac{1}{1 - \lambda} \right) + \gamma \delta_{t+1} \left( \frac{\lambda}{1 - \lambda} \right) + \gamma^2 \delta_{t+2} \left( \frac{\lambda^2}{1 - \lambda} \right) + \dots \right) \\
&= \sum_{l=0}^{\infty} (\gamma \lambda)^l \delta_{t+l}.
\end{aligned}$$

$$\text{GAE}(\gamma, 0) : \hat{A}_t = \delta_t = r_t + \gamma V(s_{t+1}) - V(s_t).$$

$$\text{GAE}(\gamma, 1) : \hat{A}_t = \sum_{l=0}^{\infty} \gamma^l \delta_{t+l} = \sum_{l=0}^{\infty} \gamma^l r_{t+l+1} - V(s_t).$$

通过 GAE，我们可以准确估计优势函数  $A(s_t, a_t)$  的  $\hat{A}_t$ 。这一估计在构建策略梯度估计器时将起到关键作用：

通过 GAE，我们可以准确估计优势函数  $A(s_t, a_t)$  的  $\hat{A}_t$ 。这一估计在构建策略梯度估计器时将起到关键作用：

$$\nabla_{\theta} \hat{J}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{\tau \in \mathcal{D}} \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \hat{A}_t,$$

通过 GAE，我们可以准确估计优势函数  $A(s_t, a_t)$  的  $\hat{A}_t$ 。这一估计在构建策略梯度估计器时将起到关键作用：

$$\nabla_{\theta} \hat{J}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{\tau \in \mathcal{D}} \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \hat{A}_t,$$

其中  $\mathcal{D}$  是一个有限的样本批次，我们用  $\hat{\mathbb{E}}_t$  表示上述的  $\frac{1}{|\mathcal{D}|} \sum_{\tau \in \mathcal{D}} \sum_{t=1}^T$ 。

通过 GAE，我们可以准确估计优势函数  $A(s_t, a_t)$  的  $\hat{A}_t$ 。这一估计在构建策略梯度估计器时将起到关键作用：

$$\nabla_{\theta} \hat{J}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{\tau \in \mathcal{D}} \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \hat{A}_t,$$

其中  $\mathcal{D}$  是一个有限的样本批次，我们用  $\hat{\mathbb{E}}_t$  表示上述的  $\frac{1}{|\mathcal{D}|} \sum_{\tau \in \mathcal{D}} \sum_{t=1}^T$ 。

价值函数估计：

$$\mathcal{L}_{\text{critic}}(\phi) = \hat{\mathbb{E}}_t \left[ \|V_{\phi}(s_t) - \hat{R}_t\|^2 \right].$$

这里， $V_{\phi}(s_t)$  表示参数为  $\phi$  的评论家模型对状态  $s_t$  的预测值，而  $\hat{R}_t$  表示状态  $s_t$  的实际回报值，其估计为：

$$\hat{R}_t = \sum_{l=0}^{\infty} \gamma^l r_{t+l},$$

其中  $\gamma$  是折扣因子。

## ► TRPO:KL divergence

$$\max_{\theta} \hat{\mathbb{E}}_t \left[ \frac{\pi_{\theta}(a_t | s_t)}{\pi_{\theta_{\text{old}}}(a_t | s_t)} \hat{A}_t \right],$$

subject to  $\hat{\mathbb{E}}_t [\text{KL}(\pi_{\theta_{\text{old}}}(\cdot | s_t), \pi_{\theta}(\cdot | s_t))] \leq \delta,$

## ► TRPO:KL divergence

$$\max_{\theta} \hat{\mathbb{E}}_t \left[ \frac{\pi_{\theta}(a_t | s_t)}{\pi_{\theta_{\text{old}}}(a_t | s_t)} \hat{A}_t \right],$$

subject to  $\hat{\mathbb{E}}_t [\text{KL}(\pi_{\theta_{\text{old}}}(\cdot | s_t), \pi_{\theta}(\cdot | s_t))] \leq \delta$ ,

## ► PPO-Penalty:

$$\mathcal{L}_{\text{ppo-penalty}}(\theta) = \hat{\mathbb{E}}_t \left[ \frac{\pi_{\theta}(a_t | s_t)}{\pi_{\theta_{\text{old}}}(a_t | s_t)} \hat{A}_t \right] - \beta \text{KL}(\pi_{\theta_{\text{old}}}(\cdot | s_t), \pi_{\theta}(\cdot | s_t)),$$

- TRPO:KL divergence

$$\max_{\theta} \hat{\mathbb{E}}_t \left[ \frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \hat{A}_t \right],$$

subject to  $\hat{\mathbb{E}}_t [\text{KL}(\pi_{\theta_{\text{old}}}(\cdot|s_t), \pi_{\theta}(\cdot|s_t))] \leq \delta,$

- PPO-Penalty:

$$\mathcal{L}_{\text{ppo-penalty}}(\theta) = \hat{\mathbb{E}}_t \left[ \frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \hat{A}_t \right] - \beta \text{KL}(\pi_{\theta_{\text{old}}}(\cdot|s_t), \pi_{\theta}(\cdot|s_t)),$$

- PPO-Clip:

$$\mathcal{L}_{\text{ppo-clip}}(\theta) = \hat{\mathbb{E}}_t \left[ \min \left( \frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \hat{A}_t, \text{clip} \left( \frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}, 1 - \epsilon, 1 + \epsilon \right) \hat{A}_t \right) \right],$$

## 算法 1 PPO

**输入:** 初始策略参数  $\theta_0$ , 初始价值函数参数  $\phi_0$ 。

$n = 0, 1, 2, \dots$  通过在环境中执行策略  $\pi(\theta_n)$  收集一组轨迹  $\mathcal{D}_n = \{\tau_i\}$ 。计算未来奖励  $\hat{R}_t$ 。基于当前价值函数  $V_{\phi_n}$  计算优势估计  $\hat{A}_t$  (使用任意优势估计方法)。通过最大化 PPO-clip 目标函数更新策略：

$$\theta_{n+1} = \arg \max_{\theta} \mathcal{L}_{\text{ppo-clip}}(\theta_n).$$

通过均方误差回归更新价值函数：

$$\phi_{n+1} = \arg \min_{\phi} \mathcal{L}_{\text{critic}}(\phi_n).$$

**输出:** 优化后的策略参数  $\theta$

- ▶ Reference Model and Policy Model: 用 7B SFT(监督微调) 模型初始化，SFT 模型基于 OpenChineseLLaMA 对 100 万条筛选后的指令数据进行了 2 个 epoch 的监督微调，这些数据包括 40 万单轮指令样本和 60 万多轮指令样本。学习率设置为  $9.5 \times 10^{-6}$ ，并采用余弦学习率调度，最终学习率会衰减至峰值学习率的 10%。全局批量大小设置为 1024。

- ▶ Reference Model and Policy Model: 用 7B SFT(监督微调) 模型初始化，SFT 模型基于 OpenChineseLLaMA 对 100 万条筛选后的指令数据进行了 2 个 epoch 的监督微调，这些数据包括 40 万单轮指令样本和 60 万多轮指令样本。学习率设置为  $9.5 \times 10^{-6}$ ，并采用余弦学习率调度，最终学习率会衰减至峰值学习率的 10%。全局批量大小设置为 1024。
- ▶ Critic Model and Reward Model: 用奖励模型初始化.

- ▶ Reference Model and Policy Model: 用 7B SFT(监督微调) 模型初始化，SFT 模型基于 OpenChineseLLaMA 对 100 万条筛选后的指令数据进行了 2 个 epoch 的监督微调，这些数据包括 40 万单轮指令样本和 60 万多轮指令样本。学习率设置为  $9.5 \times 10^{-6}$ ，并采用余弦学习率调度，最终学习率会衰减至峰值学习率的 10%。全局批量大小设置为 1024。
- ▶ Critic Model and Reward Model: 用奖励模型初始化。
- ▶ 一个手动构建的 HH 数据集上训练模型，该数据集包含 8k 无害查询和 20k 有帮助的查询，并且固定训练步数。在实验中，将从环境中采样的批量大小设置为 128，用于训练策略模型和评价模型的批量大小为 32。策略模型和评价模型的学习率分别设置为  $5 \times 10^{-7}$  和  $1.65 \times 10^{-6}$ ，并在前 10% 的训练步数中进行预热。

- ▶ Reference Model and Policy Model: 用 7B SFT(监督微调) 模型初始化，SFT 模型基于 OpenChineseLLaMA 对 100 万条筛选后的指令数据进行了 2 个 epoch 的监督微调，这些数据包括 40 万单轮指令样本和 60 万多轮指令样本。学习率设置为  $9.5 \times 10^{-6}$ ，并采用余弦学习率调度，最终学习率会衰减至峰值学习率的 10%。全局批量大小设置为 1024。
- ▶ Critic Model and Reward Model: 用奖励模型初始化。
- ▶ 一个手动构建的 HH 数据集上训练模型，该数据集包含 8k 无害查询和 20k 有帮助的查询，并且固定训练步数。在实验中，将从环境中采样的批量大小设置为 128，用于训练策略模型和评价模型的批量大小为 32。策略模型和评价模型的学习率分别设置为  $5 \times 10^{-7}$  和  $1.65 \times 10^{-6}$ ，并在前 10% 的训练步数中进行预热。
- ▶ 所有实验均在相同配置的机器上进行。每台机器包含 8 个 80G A100 GPU、1TB 内存和 128 个 CPU。在训练阶段，使用 ZERO2 和梯度检查点（gradient checkpoint）以减少 GPU 内存开销。

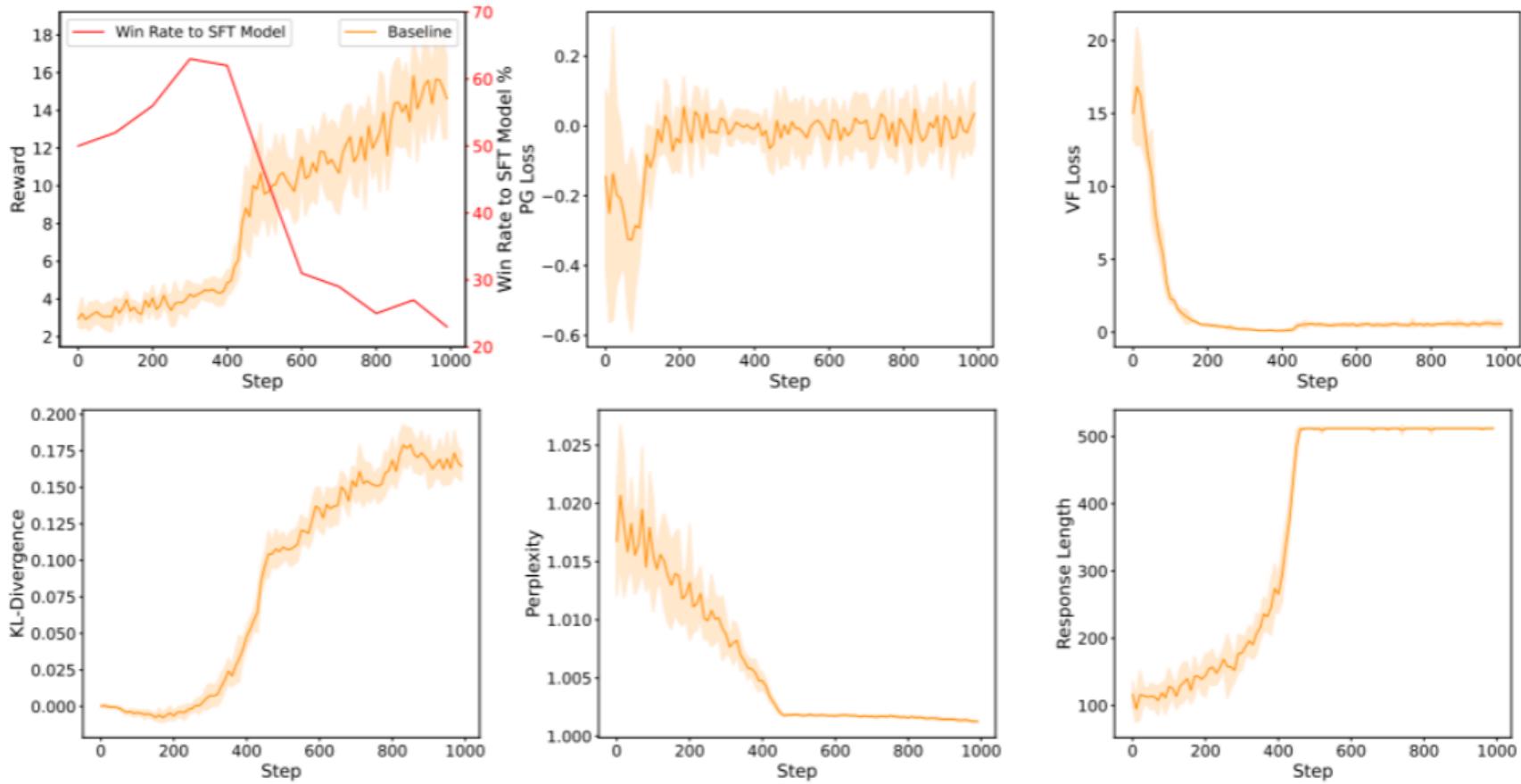


Figure 4: (Top) We show the response reward and training loss under vanilla PPO implementation. (Bottom) We show the divergence, perplexity, and response length under RLHF in Large Language Models.

- $r_n(x, y)$  表示每批奖励的结果；

- ▶  $r_n(x, y)$  表示每批奖励的结果；
- ▶  $\{r(x, y)\} \triangleq \{r_n(x, y)\}_{n=1}^B$  表示训练中的奖励序列；

- ▶  $r_n(x, y)$  表示每批奖励的结果；
- ▶  $\{r(x, y)\} \triangleq \{r_n(x, y)\}_{n=1}^B$  表示训练中的奖励序列；
- ▶  $\sigma(A)$  和  $\bar{A}$  分别表示变量  $A$  的标准差和均值。

- ▶  $r_n(x, y)$  表示每批奖励的结果；
- ▶  $\{r(x, y)\} \triangleq \{r_n(x, y)\}_{n=1}^B$  表示训练中的奖励序列；
- ▶  $\sigma(A)$  和  $\bar{A}$  分别表示变量  $A$  的标准差和均值。
- ▶ 奖励缩放：

$$\frac{r_n(x, y)}{\sigma(r(x, y))}.$$

- ▶  $r_n(x, y)$  表示每批奖励的结果；
- ▶  $\{r(x, y)\} \triangleq \{r_n(x, y)\}_{n=1}^B$  表示训练中的奖励序列；
- ▶  $\sigma(A)$  和  $\bar{A}$  分别表示变量  $A$  的标准差和均值。
- ▶ 奖励缩放：

$$\frac{r_n(x, y)}{\sigma(r(x, y))}.$$

- ▶ 奖励归一化与裁剪：

$$\tilde{r}(x, y) = \text{clip}\left(\frac{r_n(x, y) - \bar{r}(x, y)}{\sigma(r(x, y))}, -\delta, \delta\right).$$

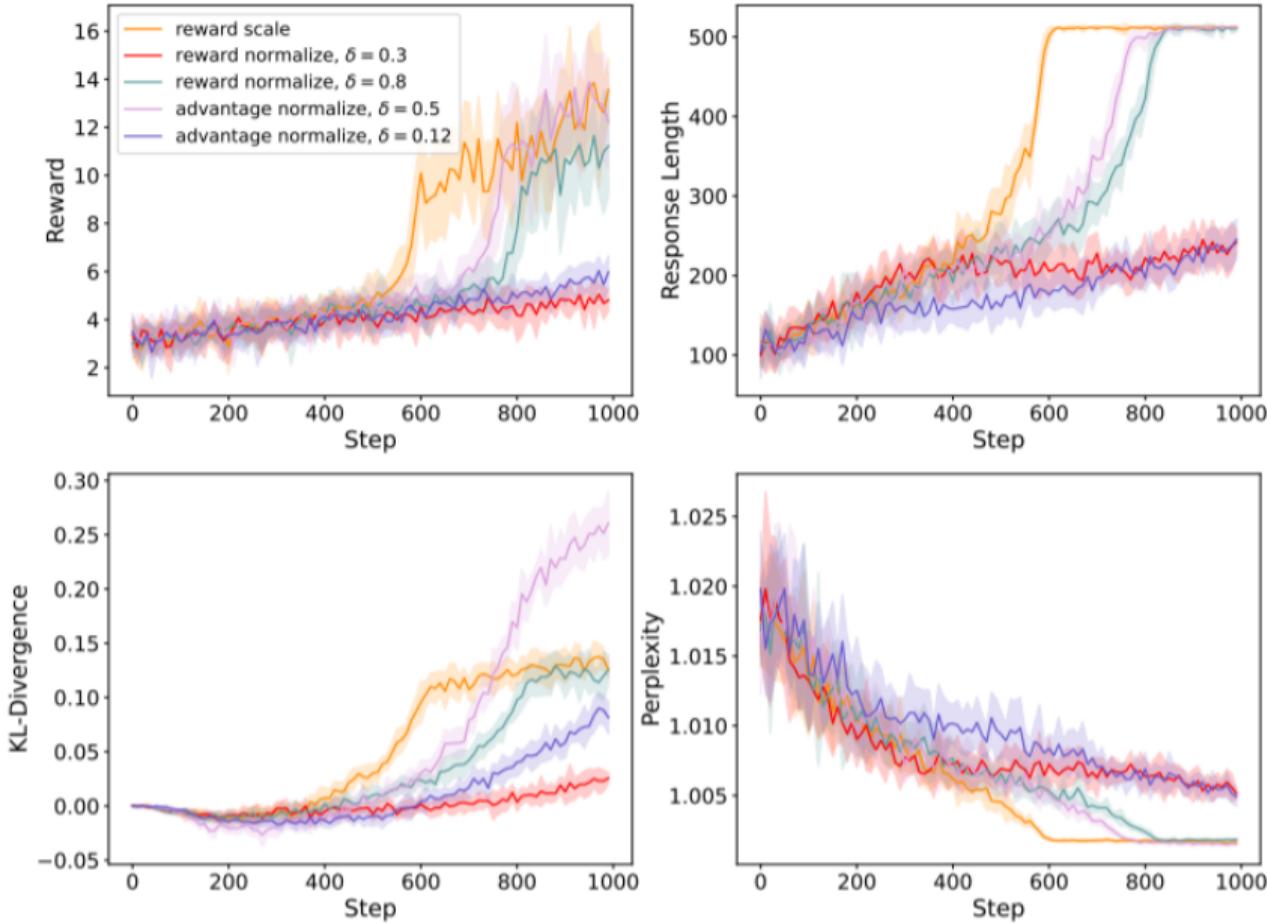
- ▶  $r_n(x, y)$  表示每批奖励的结果；
- ▶  $\{r(x, y)\} \triangleq \{r_n(x, y)\}_{n=1}^B$  表示训练中的奖励序列；
- ▶  $\sigma(A)$  和  $\bar{A}$  分别表示变量  $A$  的标准差和均值。
- ▶ 奖励缩放：

$$\frac{r_n(x, y)}{\sigma(r(x, y))}.$$

- ▶ 奖励归一化与裁剪：

$$\tilde{r}(x, y) = \text{clip}\left(\frac{r_n(x, y) - \bar{r}(x, y)}{\sigma(r(x, y))}, -\delta, \delta\right).$$

- ▶ 优势归一化与裁剪：减去其均值并除以其标准差。



► 词元级别的 KL 惩罚:

$$r_{\text{total}}(x, y_i) = r(x, y_i) - \eta \text{KL}(\pi_{\theta}^{\text{RL}}(y_i|x), \pi^{\text{SFT}}(y_i|x)).$$

- ▶ 词元级别的 KL 惩罚:

$$r_{\text{total}}(x, y_i) = r(x, y_i) - \eta \text{KL}(\pi_{\theta}^{\text{RL}}(y_i|x), \pi^{\text{SFT}}(y_i|x)).$$

- ▶ 重要性采样: 在利用经验缓冲区中的响应优化策略模型时, 修正历史生成模型与当前模型之间的策略差异。

- ▶ 词元级别的 KL 惩罚:

$$r_{\text{total}}(x, y_i) = r(x, y_i) - \eta \text{KL}(\pi_{\theta}^{\text{RL}}(y_i|x), \pi^{\text{SFT}}(y_i|x)).$$

- ▶ 重要性采样: 在利用经验缓冲区中的响应优化策略模型时, 修正历史生成模型与当前模型之间的策略差异。

▶

$$\begin{aligned}\mathbb{E}_{x \sim q}[f(x)] &= \int q(x) \cdot f(x) dx = \int \frac{p(x)}{p(x)} \cdot q(x) \cdot f(x) dx \\ &= \int p(x) \cdot \left[ \frac{q(x)}{p(x)} \cdot f(x) \right] dx \\ &= \mathbb{E}_{x \sim p} \left[ \frac{q(x)}{p(x)} \cdot f(x) \right],\end{aligned}$$

- ▶ 词元级别的 KL 惩罚:

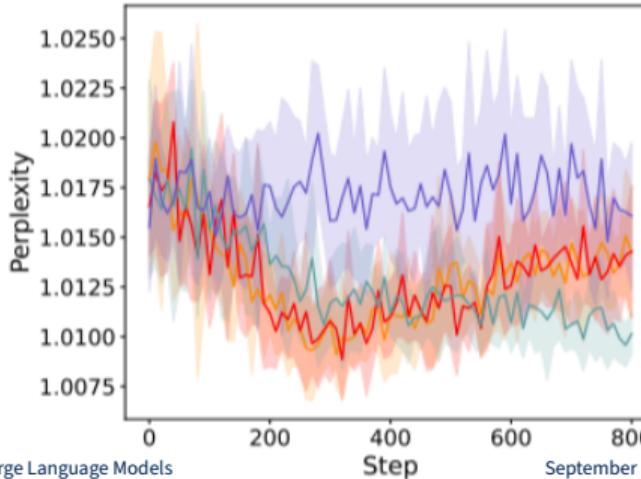
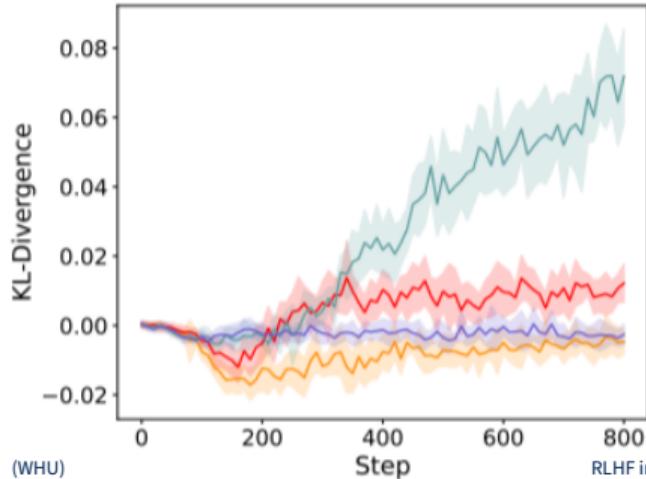
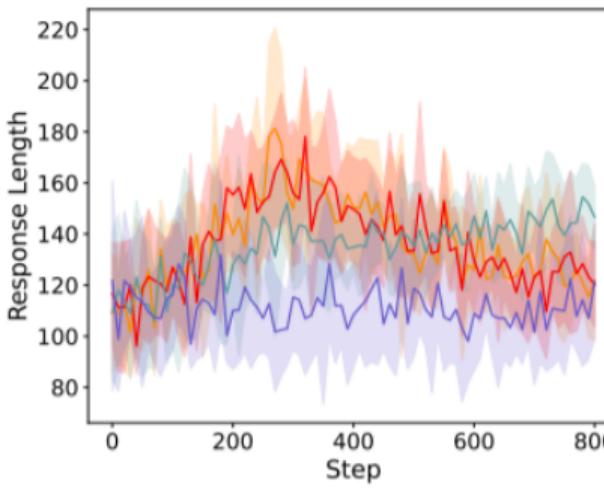
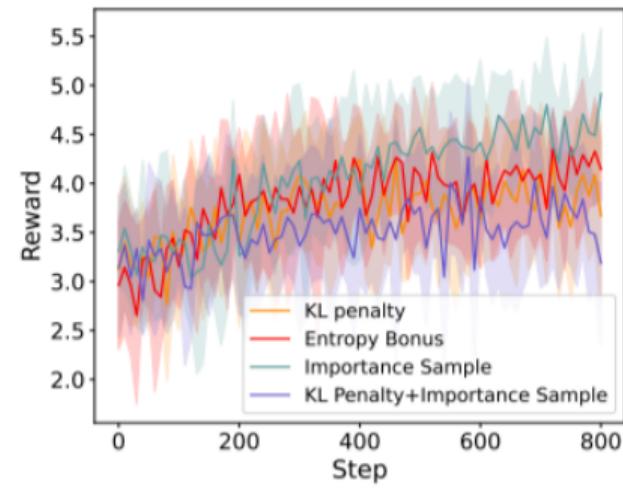
$$r_{\text{total}}(x, y_i) = r(x, y_i) - \eta \text{KL}(\pi_{\theta}^{\text{RL}}(y_i|x), \pi^{\text{SFT}}(y_i|x)).$$

- ▶ 重要性采样: 在利用经验缓冲区中的响应优化策略模型时, 修正历史生成模型与当前模型之间的策略差异。
- ▶

$$\begin{aligned}\mathbb{E}_{x \sim q}[f(x)] &= \int q(x) \cdot f(x) dx = \int \frac{p(x)}{p(x)} \cdot q(x) \cdot f(x) dx \\ &= \int p(x) \cdot \left[ \frac{q(x)}{p(x)} \cdot f(x) \right] dx \\ &= \mathbb{E}_{x \sim p} \left[ \frac{q(x)}{p(x)} \cdot f(x) \right],\end{aligned}$$

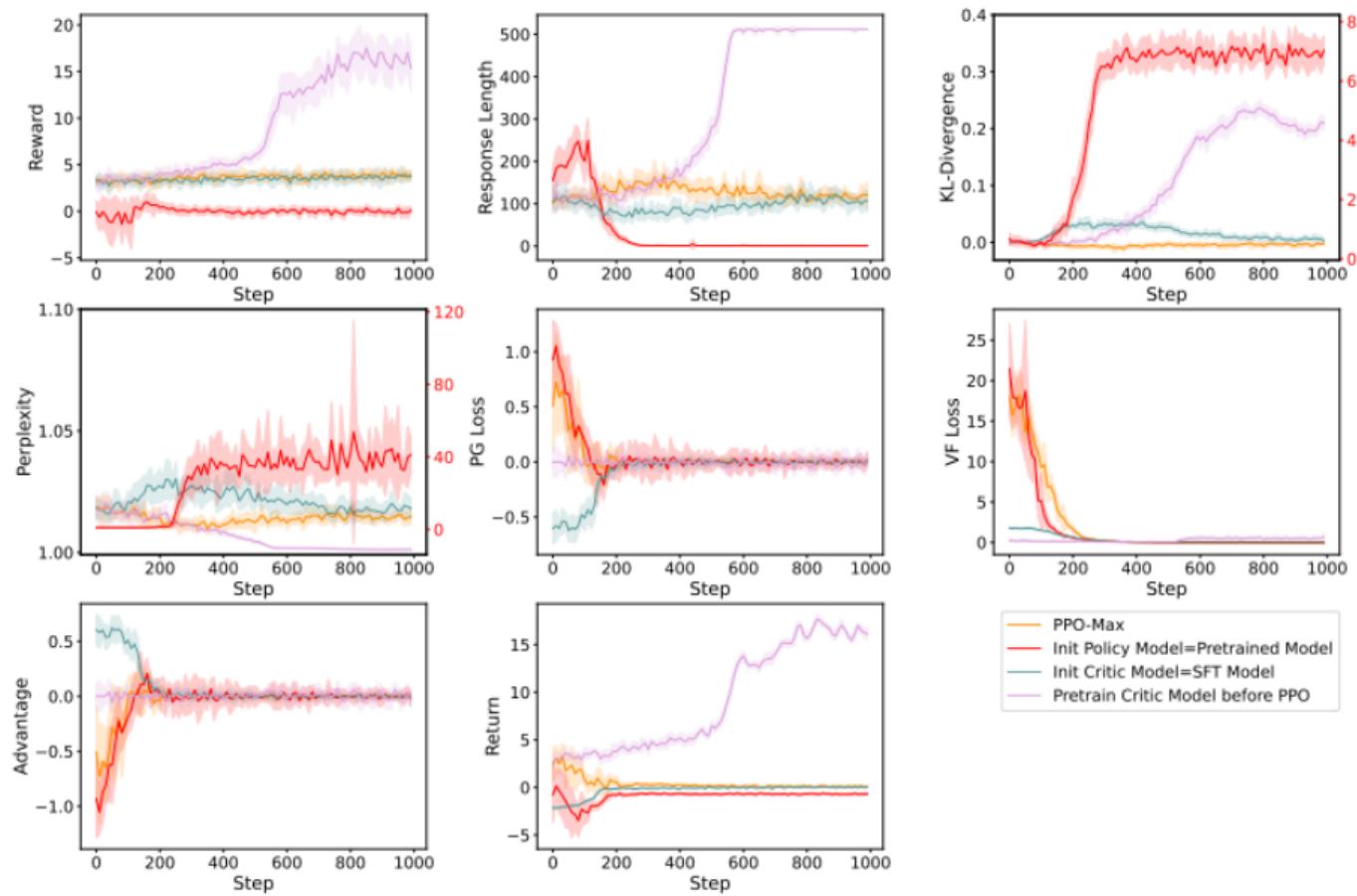
- ▶ 熵奖励:

$$H(\pi) = - \sum_a \pi(a|s) \log \pi(a|s).$$

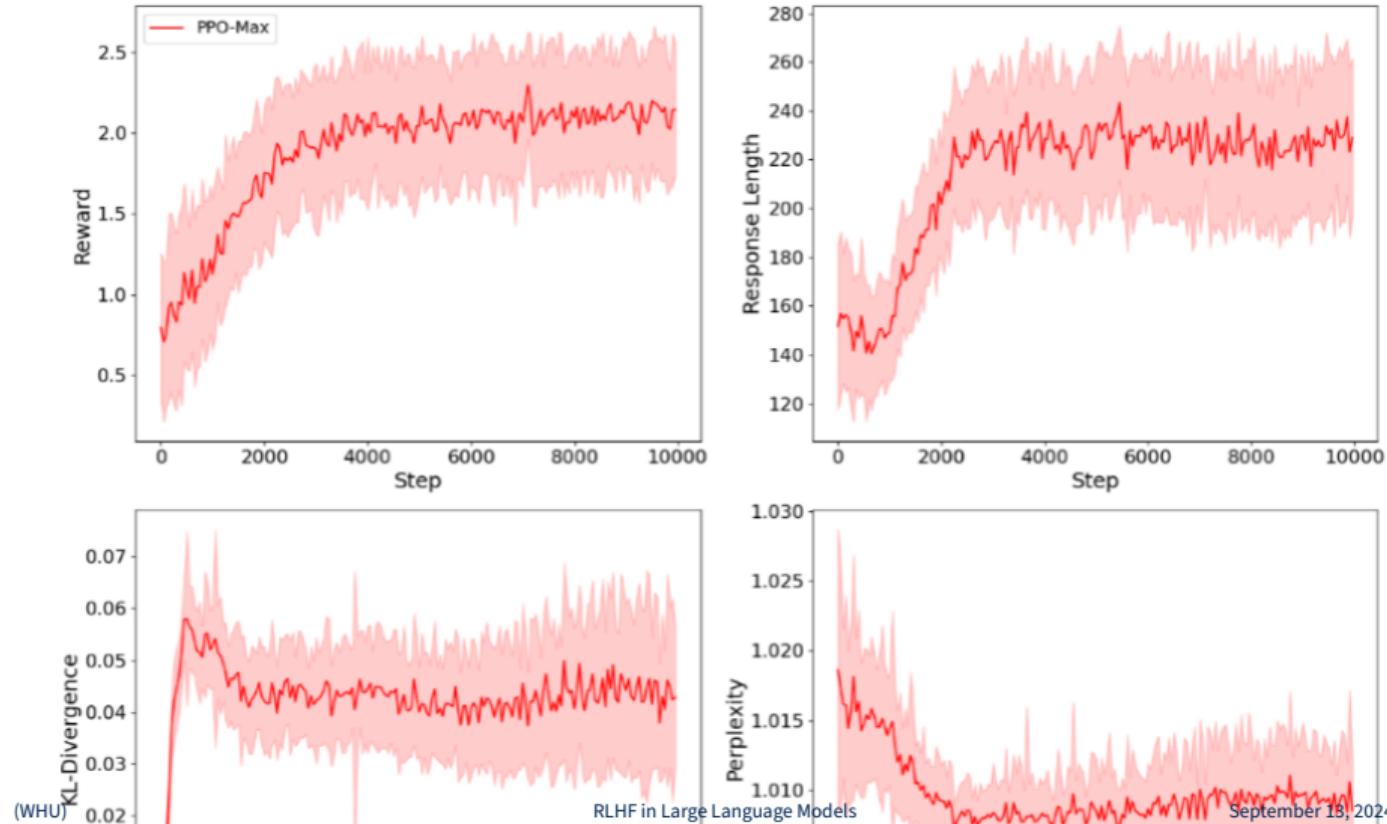


## ► Critic Model Initialization: SFT

- ▶ Critic Model Initialization: SFT
- ▶ Policy Model Initialization: pre-train



(WHY) Figure 8: We show the necessity regarding RLHF in Large Language Models



- ▶ 监督信号：结果监督信号和过程监督信号。

- ▶ 监督信号：结果监督信号和过程监督信号。
- ▶ 数据集：带有细粒度标注信息的数据集合 PRM800K。

- ▶ 监督信号：结果监督信号和过程监督信号。
- ▶ 数据集：带有细粒度标注信息的数据集合 PRM800K。
- ▶ **RLHF 训练方法：**专家迭代方法：通过向专家策略学习进而改进基础策略的强化学习方法。包含两个主要阶段：策略改进和蒸馏。在策略改进阶段，专家策略进行广泛的搜索并生成样本，过程监督奖励模型引导专家策略在搜索过程中生成高质量的样本。具体来说，在专家策略搜索的过程中，过程监督奖励模型基于当前的状态和决策轨迹，对专家策略的下一步决策进行打分，辅助专家策略选取更好的决策（即分数更高的决策）。随后，在蒸馏阶段，进一步使用第一阶段由专家策略生成的样本对基础策略（即待对齐的语言模型）进行监督微调。

- ▶ 监督信号：结果监督信号和过程监督信号。
- ▶ 数据集：带有细粒度标注信息的数据集合 PRM800K。
- ▶ RLHF 训练方法：专家迭代方法：通过向专家策略学习进而改进基础策略的强化学习方法。包含两个主要阶段：策略改进和蒸馏。在策略改进阶段，专家策略进行广泛的搜索并生成样本，过程监督奖励模型引导专家策略在搜索过程中生成高质量的样本。具体来说，在专家策略搜索的过程中，过程监督奖励模型基于当前的状态和决策轨迹，对专家策略的下一步决策进行打分，辅助专家策略选取更好的决策（即分数更高的决策）。随后，在蒸馏阶段，进一步使用第一阶段由专家策略生成的样本对基础策略（即待对齐的语言模型）进行监督微调。
- ▶ 过程监督奖励模型的扩展功能：辅助大语言模型完成下游任务。

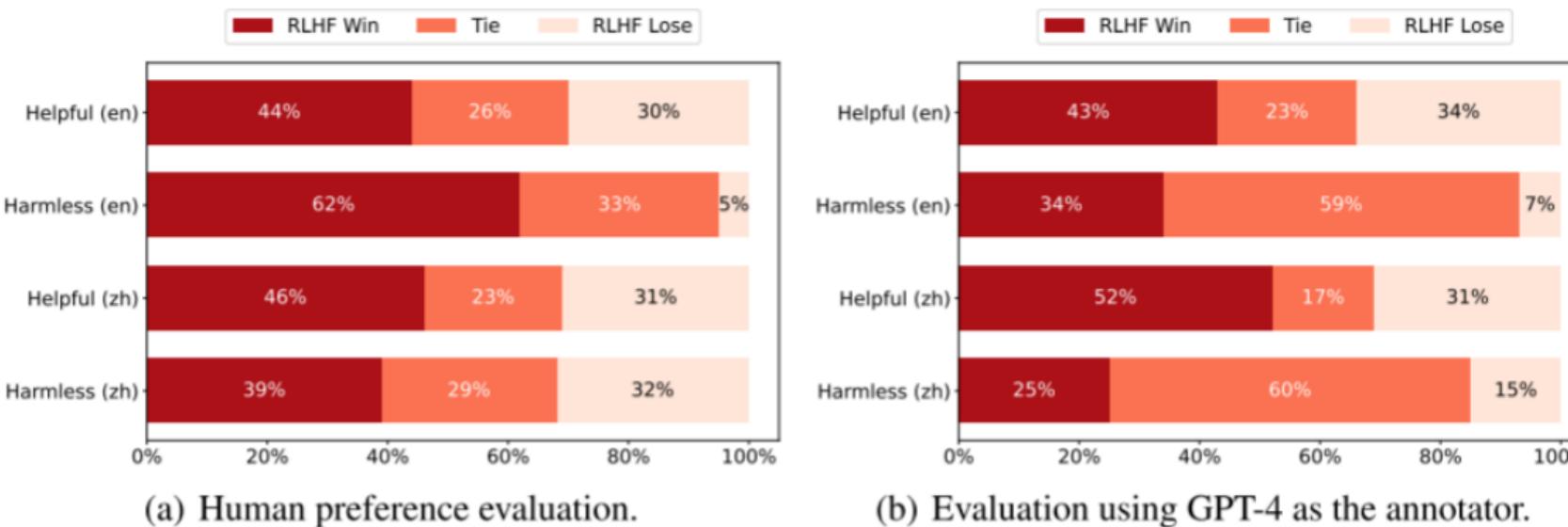
- ▶ 已对齐大语言模型的反馈：Constitutional AI 的算法为例：该算法分为监督微调与强化学习两个步骤。首先利用经过 RLHF 训练的大语言模型，针对输入的问题生成初步回复。为确保生成的回复与人类价值观和偏好相符，算法进一步采用评价和修正的方法对初步回复进行调整和修改。经过微调的模型通过奖励模型的反馈进行强化学习，得到与人类偏好对齐的大语言模型。

- ▶ 已对齐大语言模型的反馈：Constitutional AI 的算法为例：该算法分为监督微调与强化学习两个步骤。首先利用经过 RLHF 训练的大语言模型，针对输入的问题生成初步回复。为确保生成的回复与人类价值观和偏好相符，算法进一步采用评价和修正的方法对初步回复进行调整和修改。经过微调的模型通过奖励模型的反馈进行强化学习，得到与人类偏好对齐的大语言模型。
- ▶ 待对齐大语言模型的自我反馈：RLAIF 算法：使用策略模型对自己的输出进行反馈，通过自我反馈进行对齐训练。

表 8.1 SFT 和 RLHF 的优缺点对比

	<b>优点</b>	1、提高大语言模型在各种基准测试中的性能 2、增强大语言模型在不同任务上的泛化能力 3、提升大语言模型在专业领域的能力
<b>SFT</b>	<b>缺点</b>	1、当数据超出大语言模型的知识范围时，模型易产生幻觉 2、通过对教师模型的蒸馏，会增加学生模型出现幻觉的可能性 3、不同标注者对实例数据标注的差异，会影响 SFT 的学习性能 4、指令数据的质量会影响大语言模型的训练效果
	<b>优点</b>	1、进一步增强模型的能力，提高模型有用性 2、有效减轻大语言模型出现有害响应的可能性 3、有效减轻大语言模型出现幻觉的可能性 4、偏好标注可以减轻实例生成过程中的不一致情况
<b>RLHF</b>		

# 对比



(a) Human preference evaluation.

(b) Evaluation using GPT-4 as the annotator.

Figure 10: Preference evaluations, compared RLHF models with SFT models in human evaluation (left) and GPT-4 evaluation (right).

- ▶ 基于监督微调的对齐方法，通过更简洁、更直接的方式来实现大语言模型与人类价值观的对齐，进而避免复杂的强化学习算法所带来的种种问题。

- ▶ 基于监督微调的对齐方法，通过更简洁、更直接的方式来实现大语言模型与人类价值观的对齐，进而避免复杂的强化学习算法所带来的种种问题。
- ▶ 利用高质量的对齐数据集，通过特定的监督学习算法对于大语言模型进行微调。

- ▶ 基于监督微调的对齐方法，通过更简洁、更直接的方式来实现大语言模型与人类价值观的对齐，进而避免复杂的强化学习算法所带来的种种问题。
- ▶ 利用高质量的对齐数据集，通过特定的监督学习算法对于大语言模型进行微调。
- ▶ 在优化过程中使模型能够区分对齐的数据和未对齐的数据 (或者对齐质量的高低)，进而直接从这些数据中学习到与人类期望对齐的行为模式。

- ▶ 基于监督微调的对齐方法，通过更简洁、更直接的方式来实现大语言模型与人类价值观的对齐，进而避免复杂的强化学习算法所带来的种种问题。
- ▶ 利用高质量的对齐数据集，通过特定的监督学习算法对于大语言模型进行微调。
- ▶ 在优化过程中使模型能够区分对齐的数据和未对齐的数据(或者对齐质量的高低)，进而直接从这些数据中学习到与人类期望对齐的行为模式。
- ▶ **两个关键要素：构建高质量对齐数据集和设计监督微调对齐算法。**

- ▶ 基于奖励模型的方法：在 RLHF 方法中，由于奖励模型已经在包含人类偏好的反馈数据集上进行了训练，因此可以将训练好的奖励模型用于评估大语言模型输出的对齐程度。

- ▶ 基于奖励模型的方法：在 RLHF 方法中，由于奖励模型已经在包含人类偏好的反馈数据集上进行了训练，因此可以将训练好的奖励模型用于评估大语言模型输出的对齐程度。
- ▶ 基于大语言模型的方法：编写符合人类对齐标准的自然语言指令与相关示例，进而让大语言模型对其输出进行自我评价与检查，并针对有害内容进行迭代式修正，最终生成与人类价值观对齐的数据集。

- ▶ 主要思想：在强化学习的目标函数中建立决策函数与奖励函数之间的关系，以规避奖励建模的过程。

- ▶ 主要思想：在强化学习的目标函数中建立决策函数与奖励函数之间的关系，以规避奖励建模的过程。
- ▶ 形式化地，DPO 算法首先需要找到奖励函数  $r(x, y)$  与决策函数  $\pi_\theta(y|x)$  之间的关系，即使用  $\pi_\theta(y|x)$  表示  $r(x, y)$ 。然后，通过奖励建模的方法来直接建立训练目标和决策函数  $\pi_\theta(y|x)$  之间的关系。这样，大语言模型就能够通过与强化学习等价的形式学习到人类的价值观和偏好，并且去除了复杂的奖励建模过程。

- ▶ 主要思想：在强化学习的目标函数中建立决策函数与奖励函数之间的关系，以规避奖励建模的过程。
- ▶ 形式化地，DPO 算法首先需要找到奖励函数  $r(x, y)$  与决策函数  $\pi_\theta(y|x)$  之间的关系，即使用  $\pi_\theta(y|x)$  表示  $r(x, y)$ 。然后，通过奖励建模的方法来直接建立训练目标和决策函数  $\pi_\theta(y|x)$  之间的关系。这样，大语言模型就能够通过与强化学习等价的形式学习到人类的价值观和偏好，并且去除了复杂的奖励建模过程。
- ▶ 目标函数：

$$L(\theta) = \max_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta} [r(x, y)] - \beta \text{KL} [\pi_\theta(y|x), \pi_{\theta_{\text{old}}}(y|x)].$$

► 求解：

$$\begin{aligned}
 L(\theta) &= \max_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(\cdot|x)} \left[ r(x, y) - \beta \log \frac{\pi_\theta(y|x)}{\pi_{\theta_{\text{old}}}(y|x)} \right] \\
 &= \min_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(\cdot|x)} \left[ \log \frac{\pi_\theta(y|x)}{\pi_{\theta_{\text{old}}}(y|x)} - \frac{1}{\beta} r(x, y) \right] \\
 &= \min_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(\cdot|x)} \left[ \log \frac{\pi_\theta(y|x)}{\frac{1}{Z(x)} \pi_{\theta_{\text{old}}}(y|x) \exp\left(\frac{1}{\beta} r(x, y)\right)} - \log Z(x) \right],
 \end{aligned}$$

► 求解：

$$\begin{aligned}
 L(\theta) &= \max_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(\cdot|x)} \left[ r(x, y) - \beta \log \frac{\pi_\theta(y|x)}{\pi_{\theta_{\text{old}}}(y|x)} \right] \\
 &= \min_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(\cdot|x)} \left[ \log \frac{\pi_\theta(y|x)}{\pi_{\theta_{\text{old}}}(y|x)} - \frac{1}{\beta} r(x, y) \right] \\
 &= \min_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(\cdot|x)} \left[ \log \frac{\pi_\theta(y|x)}{\frac{1}{Z(x)} \pi_{\theta_{\text{old}}}(y|x) \exp\left(\frac{1}{\beta} r(x, y)\right)} - \log Z(x) \right],
 \end{aligned}$$

►  $Z(x)$  是一个配分函数：

$$Z(x) = \sum_y \pi_{\theta_{\text{old}}}(y|x) \exp\left(\frac{1}{\beta} r(x, y)\right).$$

► 求解：

$$\begin{aligned}
 L(\theta) &= \max_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(\cdot | x)} \left[ r(x, y) - \beta \log \frac{\pi_\theta(y | x)}{\pi_{\theta_{\text{old}}}(y | x)} \right] \\
 &= \min_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(\cdot | x)} \left[ \log \frac{\pi_\theta(y | x)}{\pi_{\theta_{\text{old}}}(y | x)} - \frac{1}{\beta} r(x, y) \right] \\
 &= \min_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(\cdot | x)} \left[ \log \frac{\pi_\theta(y | x)}{\frac{1}{Z(x)} \pi_{\theta_{\text{old}}}(y | x) \exp\left(\frac{1}{\beta} r(x, y)\right)} - \log Z(x) \right],
 \end{aligned}$$

►  $Z(x)$  是一个配分函数：

$$Z(x) = \sum_y \pi_{\theta_{\text{old}}}(y | x) \exp\left(\frac{1}{\beta} r(x, y)\right).$$

► 定义：

$$\pi^*(y | x) = \frac{1}{Z(x)} \pi_{\theta_{\text{old}}}(y | x) \exp\left(\frac{1}{\beta} r(x, y)\right).$$



$$\begin{aligned}
 & \min_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}} \mathbb{E}_{y \sim \pi_\theta(\cdot | x)} \left[ \log \frac{\pi_\theta(y | x)}{\frac{1}{Z(x)} \pi_{\theta_{\text{old}}}(y | x) \exp\left(\frac{1}{\beta} r(x, y)\right)} - \log Z(x) \right] \\
 &= \min_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}} \mathbb{E}_{y \sim \pi_\theta(\cdot | x)} \left[ \log \frac{\pi_\theta(y | x)}{\pi^*(y | x)} - \log Z(x) \right] \\
 &= \min_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}} \left[ \mathbb{E}_{y \sim \pi_\theta(\cdot | x)} \left[ \log \frac{\pi_\theta(y | x)}{\pi^*(y | x)} \right] - \log Z(x) \right] \\
 &= \min_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}} \left[ \text{KL}[\pi_\theta(y | x), \pi^*(y | x)] - \log Z(x) \right].
 \end{aligned}$$

# 代表性监督对齐算法 DPO



$$\begin{aligned}
 & \min_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}} \mathbb{E}_{y \sim \pi_\theta(\cdot | x)} \left[ \log \frac{\pi_\theta(y | x)}{\frac{1}{Z(x)} \pi_{\theta_{\text{old}}}(y | x) \exp\left(\frac{1}{\beta} r(x, y)\right)} - \log Z(x) \right] \\
 &= \min_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}} \mathbb{E}_{y \sim \pi_\theta(\cdot | x)} \left[ \log \frac{\pi_\theta(y | x)}{\pi^*(y | x)} - \log Z(x) \right] \\
 &= \min_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}} \left[ \mathbb{E}_{y \sim \pi_\theta(\cdot | x)} \left[ \log \frac{\pi_\theta(y | x)}{\pi^*(y | x)} \right] - \log Z(x) \right] \\
 &= \min_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}} \left[ \text{KL}[\pi_\theta(y | x), \pi^*(y | x)] - \log Z(x) \right].
 \end{aligned}$$



$$\pi_r(y | x) = \pi^*(y | x) = \frac{1}{Z(x)} \pi_{\theta_{\text{old}}}(y | x) \exp\left(\frac{1}{\beta} r(x, y)\right).$$

$$\begin{aligned}
 \pi_r(y \mid x) &= \frac{1}{Z(x)} \pi_{\theta_{\text{old}}}(y \mid x) \exp\left(\frac{1}{\beta} r(x, y)\right) \\
 \implies \log(\pi_r(y \mid x)) &= \log\left(\frac{1}{Z(x)} \pi_{\theta_{\text{old}}}(y \mid x) \exp\left(\frac{1}{\beta} r(x, y)\right)\right) \\
 \implies \log(\pi_r(y \mid x)) - \log\left(\frac{1}{Z(x)} \pi_{\theta_{\text{old}}}(y \mid x)\right) &= \log\left(\exp\left(\frac{1}{\beta} r(x, y)\right)\right) \\
 \implies r(x, y) &= \beta \log\left(\frac{\pi_r(y \mid x)}{\pi_{\theta_{\text{old}}}(y \mid x)}\right) + \beta \log(Z(x)).
 \end{aligned}$$

$$\begin{aligned}
 \pi_r(y \mid x) &= \frac{1}{Z(x)} \pi_{\theta_{\text{old}}}(y \mid x) \exp\left(\frac{1}{\beta} r(x, y)\right) \\
 \implies \log(\pi_r(y \mid x)) &= \log\left(\frac{1}{Z(x)} \pi_{\theta_{\text{old}}}(y \mid x) \exp\left(\frac{1}{\beta} r(x, y)\right)\right) \\
 \implies \log(\pi_r(y \mid x)) - \log\left(\frac{1}{Z(x)} \pi_{\theta_{\text{old}}}(y \mid x)\right) &= \log\left(\exp\left(\frac{1}{\beta} r(x, y)\right)\right) \\
 \implies r(x, y) &= \beta \log\left(\frac{\pi_r(y \mid x)}{\pi_{\theta_{\text{old}}}(y \mid x)}\right) + \beta \log(Z(x)).
 \end{aligned}$$

考慮奖励建模时使用的公式：

$$\begin{aligned}
 P(y^+ > y^- \mid x) &= \frac{\exp(r(x, y^+))}{\exp(r(x, y^+)) + \exp(r(x, y^-))} \\
 &= \frac{1}{1 + \frac{\exp(r(x, y^-))}{\exp(r(x, y^+))}}.
 \end{aligned}$$

代入得到：

$$\begin{aligned}
 P(y^+ > y^- \mid x) &= \frac{1}{\exp\left(\beta \log \frac{\pi_\theta(y^- \mid x)}{\pi_{\theta_{\text{old}}}(y^- \mid x)} + \beta \log Z(x)\right)} \\
 &\quad + \exp\left(\beta \log \frac{\pi_\theta(y^+ \mid x)}{\pi_{\theta_{\text{old}}}(y^+ \mid x)} + \beta \log Z(x)\right) \\
 &= \frac{1}{1 + \exp\left(\beta \log \frac{\pi_\theta(y^- \mid x)}{\pi_{\theta_{\text{old}}}(y^- \mid x)} - \beta \log \frac{\pi_\theta(y^+ \mid x)}{\pi_{\theta_{\text{old}}}(y^+ \mid x)}\right)} \\
 &= \sigma\left(\beta \log \frac{\pi_\theta(y^+ \mid x)}{\pi_{\theta_{\text{old}}}(y^+ \mid x)} - \beta \log \frac{\pi_\theta(y^- \mid x)}{\pi_{\theta_{\text{old}}}(y^- \mid x)}\right).
 \end{aligned}$$

代入得到：

$$\begin{aligned}
 P(y^+ > y^- \mid x) &= \frac{1}{\exp\left(\beta \log \frac{\pi_\theta(y^- \mid x)}{\pi_{\theta_{\text{old}}}(y^- \mid x)} + \beta \log Z(x)\right)} \\
 &\quad + \exp\left(\beta \log \frac{\pi_\theta(y^+ \mid x)}{\pi_{\theta_{\text{old}}}(y^+ \mid x)} + \beta \log Z(x)\right) \\
 &= \frac{1}{1 + \exp\left(\beta \log \frac{\pi_\theta(y^- \mid x)}{\pi_{\theta_{\text{old}}}(y^- \mid x)} - \beta \log \frac{\pi_\theta(y^+ \mid x)}{\pi_{\theta_{\text{old}}}(y^+ \mid x)}\right)} \\
 &= \sigma\left(\beta \log \frac{\pi_\theta(y^+ \mid x)}{\pi_{\theta_{\text{old}}}(y^+ \mid x)} - \beta \log \frac{\pi_\theta(y^- \mid x)}{\pi_{\theta_{\text{old}}}(y^- \mid x)}\right).
 \end{aligned}$$

最终优化函数：

$$L(\theta) = -\mathbb{E}_{(x, y^+, y^-) \sim \mathcal{D}} \left[ \log \sigma \left( \beta \log \left( \frac{\pi_\theta(y^+ \mid x)}{\pi_{\theta_{\text{old}}}(y^+ \mid x)} \right) - \beta \log \left( \frac{\pi_\theta(y^- \mid x)}{\pi_{\theta_{\text{old}}}(y^- \mid x)} \right) \right) \right].$$

令：

$$u = \beta \log \left( \frac{\pi_{\theta}(y^+ | x)}{\pi_{\theta_{\text{old}}}(y^+ | x)} \right) - \beta \log \left( \frac{\pi_{\theta}(y^- | x)}{\pi_{\theta_{\text{old}}}(y^- | x)} \right), \hat{r}_{\theta}(x, y) = \beta \log \left( \frac{\pi_{\theta}(y | x)}{\pi_{\theta_{\text{old}}}(y | x)} \right)$$

令：

$$u = \beta \log \left( \frac{\pi_{\theta}(y^+ | x)}{\pi_{\theta_{\text{old}}}(y^+ | x)} \right) - \beta \log \left( \frac{\pi_{\theta}(y^- | x)}{\pi_{\theta_{\text{old}}}(y^- | x)} \right), \hat{r}_{\theta}(x, y) = \beta \log \left( \frac{\pi_{\theta}(y | x)}{\pi_{\theta_{\text{old}}}(y | x)} \right)$$

有：

$$\nabla L(\theta) = -\mathbb{E}_{(x, y^+, y^-) \sim \mathcal{D}} \left[ \frac{\nabla \sigma(u)}{\sigma(u)} \right] = -\mathbb{E}_{(x, y^+, y^-) \sim \mathcal{D}} \left[ \frac{\sigma(u)(1 - \sigma(u))\nabla u}{\sigma(u)} \right].$$

令：

$$u = \beta \log \left( \frac{\pi_\theta(y^+ | x)}{\pi_{\theta_{\text{old}}}(y^+ | x)} \right) - \beta \log \left( \frac{\pi_\theta(y^- | x)}{\pi_{\theta_{\text{old}}}(y^- | x)} \right), \hat{r}_\theta(x, y) = \beta \log \left( \frac{\pi_\theta(y | x)}{\pi_{\theta_{\text{old}}}(y | x)} \right)$$

有：

$$\nabla L(\theta) = -\mathbb{E}_{(x, y^+, y^-) \sim \mathcal{D}} \left[ \frac{\nabla \sigma(u)}{\sigma(u)} \right] = -\mathbb{E}_{(x, y^+, y^-) \sim \mathcal{D}} \left[ \frac{\sigma(u)(1 - \sigma(u))\nabla u}{\sigma(u)} \right].$$

进一步推导：

$$-\mathbb{E}_{(x, y^+, y^-) \sim \mathcal{D}} \left[ \frac{\nabla \sigma(u)}{\sigma(u)} \nabla u \right] = -\mathbb{E}_{(x, y^+, y^-) \sim \mathcal{D}} [\sigma(-u)\nabla u].$$

$$\begin{aligned} &= -\beta \mathbb{E}_{(x, y^+, y^-) \sim \mathcal{D}} \left[ \sigma(\hat{r}_\theta(x, y^-) - \hat{r}_\theta(x, y^+)) \right. \\ &\quad \left. \left( \nabla \log \pi_\theta(y^+ | x) - \nabla \log \pi_\theta(y^- | x) \right) \right]. \end{aligned}$$

在实现中，DPO 采用梯度下降的方式来优化策略模型的参数  $\theta$ 。通过对上述目标函数的导数进行分析，可以发现优化过程中会增大  $\log \pi_\theta(y^+ | x)$  与  $\log \pi_\theta(y^- | x)$  之间的差异。这表明优化过程中训练模型向符合人类偏好的内容靠近 ( $y^+$ )，同时尽量避免生成不符合人类偏好的内容 ( $y^-$ )。此外，公式的前半部分  $\sigma(\hat{r}_\theta(x, y^+) - \hat{r}_\theta(x, y^-))$  可以看作是梯度的系数，动态地控制梯度下降的步长。

在实现中，DPO 采用梯度下降的方式来优化策略模型的参数  $\theta$ 。通过对上述目标函数的导数进行分析，可以发现优化过程中会增大  $\log \pi_\theta(y^+ | x)$  与  $\log \pi_\theta(y^- | x)$  之间的差异。这表明优化过程中训练模型向符合人类偏好的内容靠近 ( $y^+$ )，同时尽量避免生成不符合人类偏好的内容 ( $y^-$ )。此外，公式的前半部分  $\sigma(\hat{r}_\theta(x, y^+) - \hat{r}_\theta(x, y^-))$  可以看作是梯度的系数，动态地控制梯度下降的步长。

与 RLHF 算法相比，DPO 算法没有采用强化学习算法来训练奖励模型，而是通过监督微调的方式对于语言模型进行训练。与传统有监督微调方法不同，DPO 算法中不仅训练模型生成符合人类偏好的内容，同时降低模型生成不符合人类偏好内容的概率。相比于强化学习算法 PPO，DPO 在训练过程中只需要加载策略模型和参考模型，并不用加载奖励模型和评价模型。因此，DPO 算法占用的资源更少、运行效率更高，并且具有较好的对齐性能，在实践中得到了广泛应用。

## 其他有监督对齐算法

## ► 优化目标:

$$\mathcal{L}_{\text{total}} = -\mathbb{E}_{(x, y^+) \sim \mathcal{D}} \underbrace{\sum_{t=1}^T \log(y_t^+ \mid x, y_{<t}^+)}_{\text{主要训练目标}} + \underbrace{\mathcal{L}_{\text{aux}}(y^+, y^-, x)}_{\text{辅助训练目标}},$$

## 其他有监督对齐算法

- ▶ 优化目标：

$$\mathcal{L}_{\text{total}} = -\mathbb{E}_{(x, y^+) \sim \mathcal{D}} \underbrace{\sum_{t=1}^T \log(y_t^+ \mid x, y_{<t}^+)}_{\text{主要训练目标}} + \underbrace{\mathcal{L}_{\text{aux}}(y^+, y^-, x)}_{\text{辅助训练目标}},$$

- ▶ 除了主要的训练目标，现有监督对齐算法还设计了不同的辅助训练目标，以帮助大语言模型在监督微调过程中能够更好地区分正例和负例。

- ▶ 优化目标：

$$\mathcal{L}_{\text{total}} = -\mathbb{E}_{(x, y^+) \sim \mathcal{D}} \underbrace{\sum_{t=1}^T \log(y_t^+ \mid x, y_{<t}^+)}_{\text{主要训练目标}} + \underbrace{\mathcal{L}_{\text{aux}}(y^+, y^-, x)}_{\text{辅助训练目标}},$$

- ▶ 除了主要的训练目标，现有监督对齐算法还设计了不同的辅助训练目标，以帮助大语言模型在监督微调过程中能够更好地区分正例和负例。
- ▶ **基于质量提示的训练目标：使用提示技术来帮助模型区分正负例。**

- ▶ 优化目标：

$$\mathcal{L}_{\text{total}} = -\mathbb{E}_{(x, y^+) \sim \mathcal{D}} \underbrace{\sum_{t=1}^T \log(y_t^+ \mid x, y_{<t}^+)}_{\text{主要训练目标}} + \underbrace{\mathcal{L}_{\text{aux}}(y^+, y^-, x)}_{\text{辅助训练目标}},$$

- ▶ 除了主要的训练目标，现有监督对齐算法还设计了不同的辅助训练目标，以帮助大语言模型在监督微调过程中能够更好地区分正例和负例。
- ▶ 基于质量提示的训练目标：使用提示技术来帮助模型区分正负例。
- ▶ 基于质量对比的训练目标：让模型有更高的概率生成高质量的回答，更低的概率生成低质量的回答，更好地利用质量得分的偏序关系。

# Thanks!