RLHF 方法

应锦程

武汉大学数学与统计学院

2025 年 5 月 8 日

RL Enhanced LLMs	Organization	# Params	RL Methods
DeepSeek-R1 (DeepSeek-AI et al., 2025)	deepseek	671B-A31B	RL through CoT
Kimi-k1.5 (Team et al., 2025)	Moonshot AI	-	RL through CoT
o1 (OpenAI, 2024b)	OpenAI	-	RL through CoT
Hermes 3 (Teknium et al., 2024)	NOUS RESEARCH	8B, 70B, 405B	DPO
Athene-70B (Nexusflow, 2024)	Nexusflow	70B	RLHF
Starling-7B (Zhu et al., 2024)	Berkeley UNIVERSITY OF CALIFORNIA	7B	RLAIF, PPO
Gemma2 (Team et al., 2024b)	Google	2B, 9B, 27B	RLHF
Qwen2 (Yang et al., 2024a)	Alibaba	(0.5-72)B, 57B-A14B	DPO
Llama 3 (Dubey et al., 2024)	Meta	8B, 70B, 405B	DPO
Nemotron-4 340B (Adler et al., 2024)	NVIDIA	340B	DPO, RPO
ChatGLM (GLM et al., 2024)	ZHIPU·AI	6B, 9B	ChatGLM-RLHF
DeepSeek-V2 (Liu et al., 2024a)	deepseek	236B-A21B	GRPO
Phi-3 (Abdin et al., 2024)	Microsoft	3.8B, 7B, 14B	DPO
Zephyr (HuggingFaceH4, 2024)	Argilla	141B-A39B	ORPO
Reka (Team et al., 2024c)	Reka	7B, 21B	RLHF, PPO
Claude 3 (Anthropic, 2024)	ANTHROPIC	-	RLAIF
InternLM2 (Cai et al., 2024)	上海人工智能实验室 Shanghai Artificial Intelligence Laboratory	1.8B, 7B, 20B	RLHF, PPO
Gemini (Team et al., 2023)	Google	-	RLHF
GPT-4 (OpenAI, 2023)	OpenAI	-	RLHF, PPO, RBRM
Instruct-GPT (Ouyang et al., 2022)	OpenAI	1.3B, 6B, 175B	RLHF, PPO

Table 1: An overview of RL Enhanced LLMs. The format '141B-A39B' refers to MoE models with 141B total and 39B active parameters.

Reinforcement Learning with Human Feedback

- 基于人类反馈的强化学习 (RLHF) 是一种使 LLM 与人类的目标 (有帮助, 无害, 诚实) 对齐 (alignment) 的技术

可以分为三步:

- step1: 收集人类反馈数据, 有监督微调模型 (传统评价指标 BLEU,Ronge,BERTscore)
- step2: 人类偏好训练奖励模型 (引入人类偏好)
- step3: 强化学习训练策略, 微调大模型

RL 简介

- 强化学习: 智能体与环境交互，得到环境的反馈奖励, 学习最优行为策略以最大化累积奖励.

Markov 决策过程 (MDP):

$$M = (S, A, P, r, \gamma, \mu)$$

。

- S: 状态空间,A: 动作空间,P: 转移概率,r: 奖励
- γ: 折扣因子,μ 为初始状态分布
- $\pi(a|s)$ 一般表示智能体的策略, 状态 s 下选择动作 a 的概率
- 状态价值函数 $V(s)$,$Q(s,a)$ 状态动作价值函数

RL for LLM

强化学习方法用于大模型微调对齐人类偏好.

- 智能体: 待微调的大模型
- 环境: 用户输入的 Prompt 和 Reward model 的反馈
- 动作: 生成的 Token
- 奖励:Reward model 给出的反馈分数
- 优化目标: 最大化文本生成的累积奖励

策略梯度算法

将策略参数化为一个可学习的概率分布 $\pi_\theta(a|s)$, 智能体在该策略下有 n 步轨迹 $\tau = \{s_0, a_0, r_0, s_1, a_1, r_1, \ldots, s_n, a_n, r_n\}$, $R(\tau)$ 是轨迹的累积折扣期望奖励, 强化学习的目标函数可写为:

$$J(\theta) = \max_\theta \mathbb{E}_{\tau\sim\pi_\theta}\left[R(\tau)\right] = \max_\theta \sum_\tau P(\tau;\theta)R(\tau)$$

$$P(\tau;\theta) = \left[\prod_{t=0}^{T} P(s_{t+1} \mid s_t, a_t)\cdot \pi_\theta(a_t \mid s_t)\right]$$

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

优势函数

优势函数一般是

$$A^{\pi}(s_t, a_t) = Q^{\pi}(s_t, a_t) - V^{\pi}(s_t)$$

PPO

- 策略梯度方法

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

- Trust Region Policy Optimization(TRPO)

$$\begin{aligned} \underset{\theta}{\text{maximize}} \quad & \hat{\mathbb{E}}_t \left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)} \hat{A}_t \right] \\ \text{subject to} \quad & \hat{\mathbb{E}}_t \left[\text{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)] \right] \leq \delta. \end{aligned}$$

其中 $\hat{A}_t$ 是对优势函数的估计, 常用广义优势估计 (GAE)

PPO= 策略梯度 + 重要性采样 +KL 散度约束

将 TRPO 的约束写成惩罚.

$$J_{\mathrm{PPO}}(\theta) = J^{\theta}(\theta) - \beta \mathrm{KL}[\pi_{\theta_{\mathrm{old}}}(\cdot \mid s_t), \pi_{\theta}(\cdot \mid s_t)] \tag{1}$$

$$J^{\theta}(\theta) \approx \sum_{(s_t, a_t)} \frac{\pi_{\theta}(a_t \mid s_t)}{\pi_{\theta^{old}}(a_t \mid s_t)} A^{\theta}(s_t, a_t) \tag{2}$$

PPO 惩罚

- 自适应 KL 惩罚: 希望策略变化不要太大

对更新的 KL 散度设定上下界, 当更新的散度过大, 即策略变化太大, 就增大 β, 加大对策略变化的惩罚力度, 限制更新幅度, 反之则减少 β 惩罚系数.

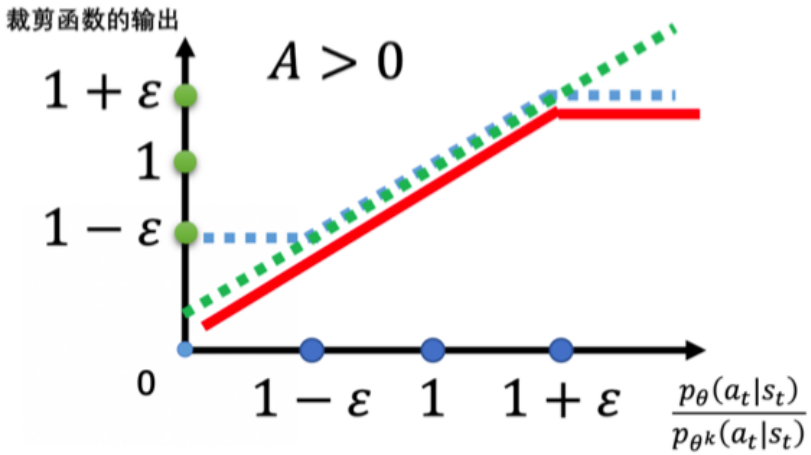
$$L^{KLPEN}(\theta) = \hat{\mathbb{E}}_t \left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \hat{A}_t - \beta \text{KL}[\pi_{\theta_{\text{old}}}(\cdot|s_t), \pi_\theta(\cdot|s_t)] \right]$$

PPO 剪裁

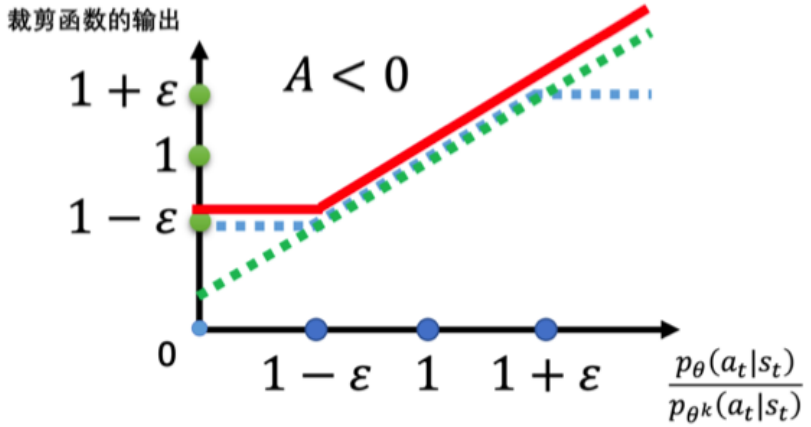
目标函数:

$$J^{\theta}_{\mathrm{PPO2}}(\theta) \approx \sum_{(s_t,a_t)} \min \left(\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)} A^{\theta}(s_t,a_t), \mathrm{clip}\left(\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)}, 1-\varepsilon, 1+\varepsilon \right) A^{\theta}(s_t,a_t) \right)$$

剪裁的目的是限制更新后的策略与原策略不要太远

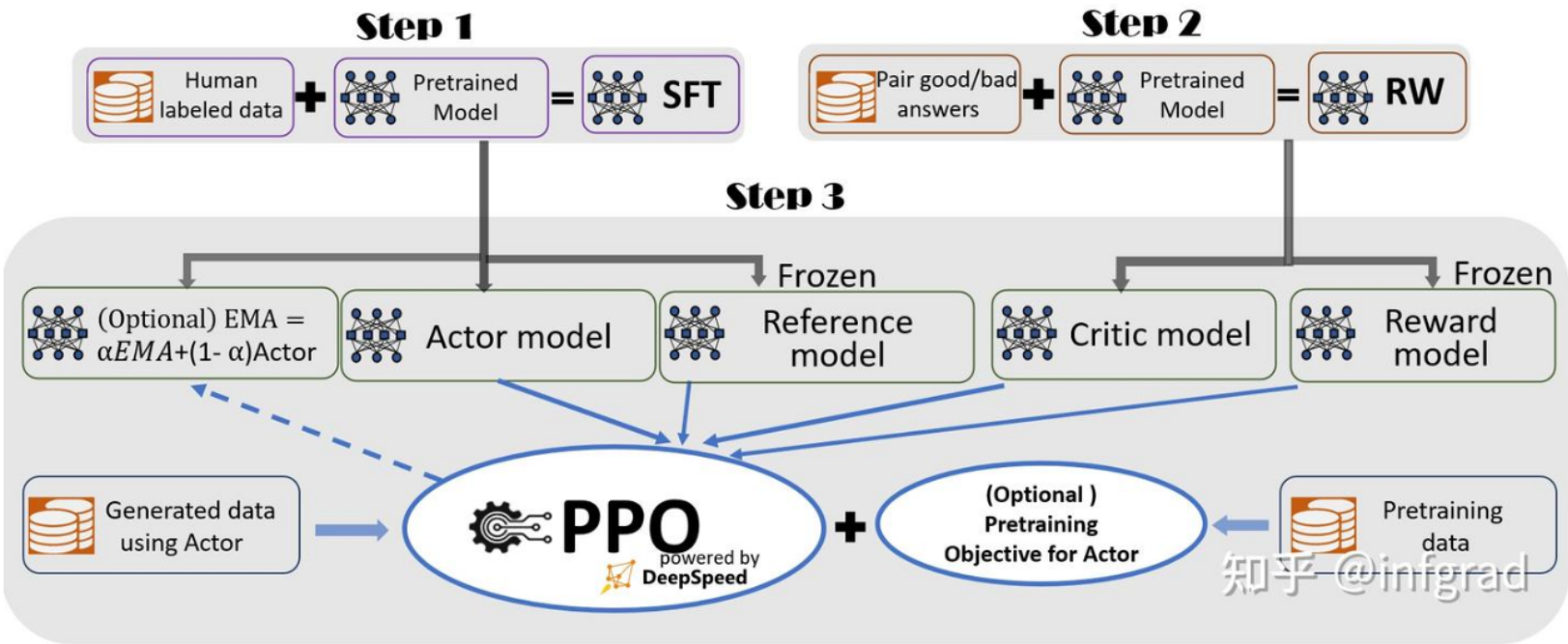

裁剪函数的输出
$A > 0$
$1+\varepsilon$
1
$1-\varepsilon$
0
$1-\varepsilon$
1
$1+\varepsilon$
$\frac{p_\theta(a_t|s_t)}{p_{\theta^k}(a_t|s_t)}$

(a) $A > 0$


裁剪函数的输出
$A < 0$
$1+\varepsilon$
1
$1-\varepsilon$
0
$1-\varepsilon$
1
$1+\varepsilon$
$\frac{p_\theta(a_t|s_t)}{p_{\theta^k}(a_t|s_t)}$

(b) $A < 0$

PPO for LLM



Critic model 用来预估总收益 V_t, reward model 对 prompt-response 问答对输出相应的得分 (计算 R_t), 两个模型用于估计优势函数, 冻结的 reference model 保留策略 π_{ref}.

PPO for LLM

PPO 的目标函数

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

直接优化策略模型可能会导致策略崩溃（Policy Collapse），使用 Refernce Model 计算完 KL 散度后直接与 Reward 结合 Reward 的计算: 在 reward model 打分上，加一个 per-token 的 KL 散度惩罚

$$r_t = r_\varphi(q, o_{\le t}) - \beta \log \frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{ref}(o_t|q,o_{<t})}$$

Reward model

奖励模型通过最大化选中回复与拒绝回复得分的差异，学习人类的偏好，即成对排序损失（Pairwise Ranking Loss）：

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

DPO

- bradley-terry model: 成对比较, 预测一个对象比另一个对象表现更好的概率

设 p_i 和 p_j 分别是能力值, 则 i 击败 j 的概率为

$$P(i \text{ beat } j) = \frac{p_i}{p_i + p_j}$$

由点态奖励 $r^*(x, y)$ 生成偏好概率:

$$p^*(y_1 \succ y_2 \mid x) = \frac{\exp(r^*(x, y_1))}{\exp(r^*(x, y_1)) + \exp(r^*(x, y_2))}.$$

PPO 到 DPO

对目标函数改写

$$\max_{\pi_\theta} \left\{ \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)}[r_\phi(x,y)] - \beta \mathbb{D}_{\mathsf{KL}}[\pi_\theta(y|x)||\pi_{\mathsf{ref}}(y|x)] \right\}$$

$$= \max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)} \left[r_\phi(x,y) - \beta \log \frac{\pi_\theta(y|x)}{\pi_{\mathsf{ref}}(y|x)} \right]$$

$$= \min_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)} \left[\log \frac{\pi_\theta(y|x)}{\pi_{\mathsf{ref}}(y|x)} - \frac{1}{\beta} r_\phi(x,y) \right]$$

$$= \min_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)} \left[\log \frac{\pi_\theta(y|x)}{\pi_{\mathsf{ref}}(y|x) e^{r_\phi(x,y)/\beta}} \right]$$

把 log 内写成分布，归一化分母处理：

$$Z(x) = \sum_y \pi_{\mathsf{ref}}(y|x) e^{r_\phi(x,y)/\beta}$$

PPO 到 DPO

$$\pi^*(y|x) = \frac{\pi_{\text{ref}}(y|x)e^{r_\phi(x,y)/\beta}}{Z(x)}$$

优化目标可以进一步写为:

$$\min_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(y|x)} \left[\log \frac{\pi_\theta(y|x)}{\pi^*(y|x)} - \log Z(x)\right]$$

$$= \min_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(y|x)} \left[\log \frac{\pi$$

$$= \min_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}} \mathbb{D}_{\text{KL}} \left(\pi_\theta(y|x)||\pi$$

策略分布相等时目标函数最小, 因此得到了最优策略与最优奖励的关系:$\pi^*(y|x) = \frac{\pi_{\text{ref}}(y|x)e^{r_\phi(x,y)/\beta}}{Z(x)}$, 绕过建立 reward model 的步骤, 直接对策略优化

DPO 优化目标

得到 $r_\phi(x,y) = \beta \log \frac{\pi^*(y|x)}{\pi_{\text{ref}}(y|x)} + \beta \log Z(x)$ 代入奖励模型的目标函数:

$$L_R(r_\phi, D) = -\mathbb{E}_{(x,y_w,y_l)\sim D}\left[\log P\left(y_w \succ y_l \mid x\right)\right]$$

得到

$$\max_{\pi^*}\left\{\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi^*(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta\log\frac{\pi^*(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right)\right]\right\}$$

- σ 指 sigmoid 函数
- 选择/拒绝的奖励的绝对大小实际上并不重要, 相对 reference 策略的变化相对差距变大是目标
- 综合了奖励模型与策略训练的目标, 得到偏好模型的目标, 即为 DPO 的目标
- 多个候选 response 的偏好模型:Plackett-Luce 偏好模型

DPO for LLM

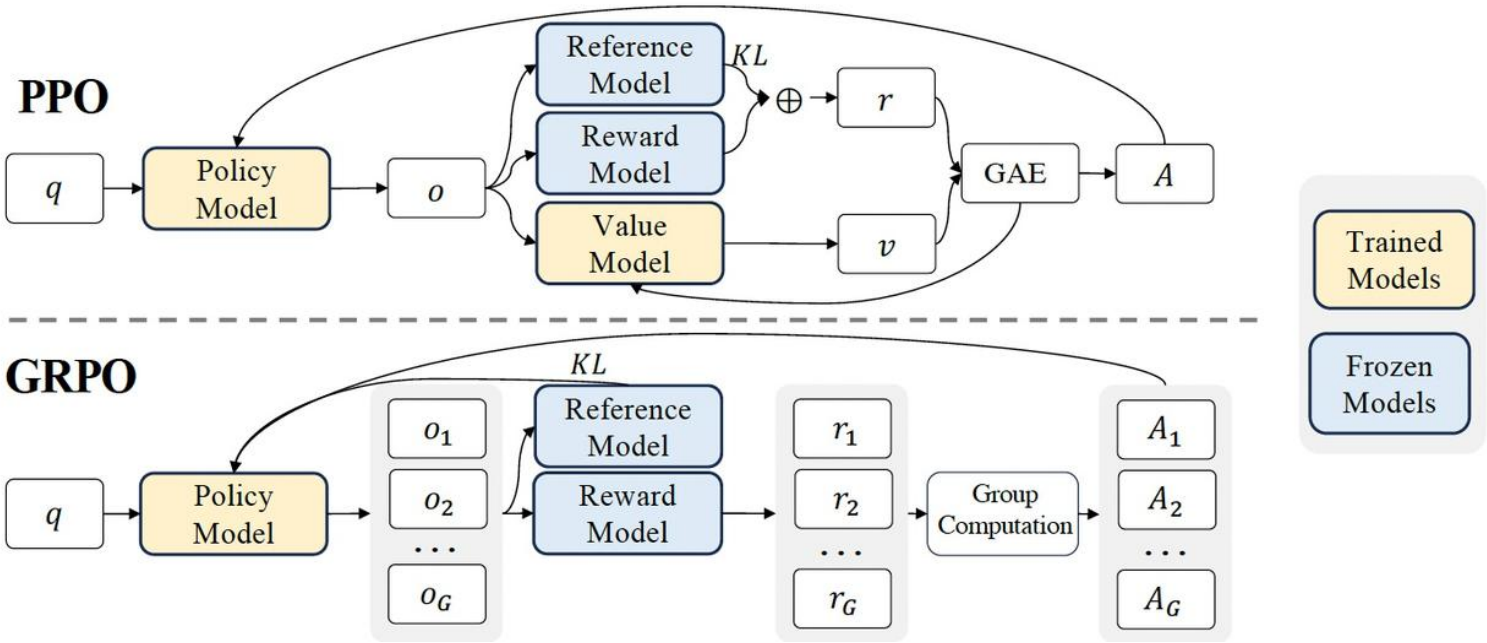
二元交叉熵损失:

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

梯度:

$$\nabla_\theta\mathcal{L}_{\mathrm{DPO}}(\theta) = -\beta\mathbb{E}_{(x,y_w,y_l)\sim D}[\underbrace{\sigma\left(\hat{r}_\theta(x,y_l) - \hat{r}_\theta(x,y_w)\right)}_{\text{higher weight when reward estimate is wrong}}[\underbrace{\nabla_\theta\log\pi(y_w \mid x)}_{\text{increase likelihood of } y_w} - \underbrace{\nabla_\theta\log\pi(y_l \mid x)}_{\text{decrease likelihood of } y_l}]]$$

GRPO for LLM



通过组内相对奖励来估 V_t，从而避免使用额外的价值函数模型 (critic model)．传统的 PPO 算法需要训练一个价值函数来估计优势函数 (advantage function)，而 GRPO 通过从同一问题的多个输出中计算平均奖励来替代这一过程，显著减少了内存和计算资源的消耗

GRPO 的目标函数

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

将 KL 散度抑制，移到了优势函数计算的外面。KL 散度的计算也进行了改进，

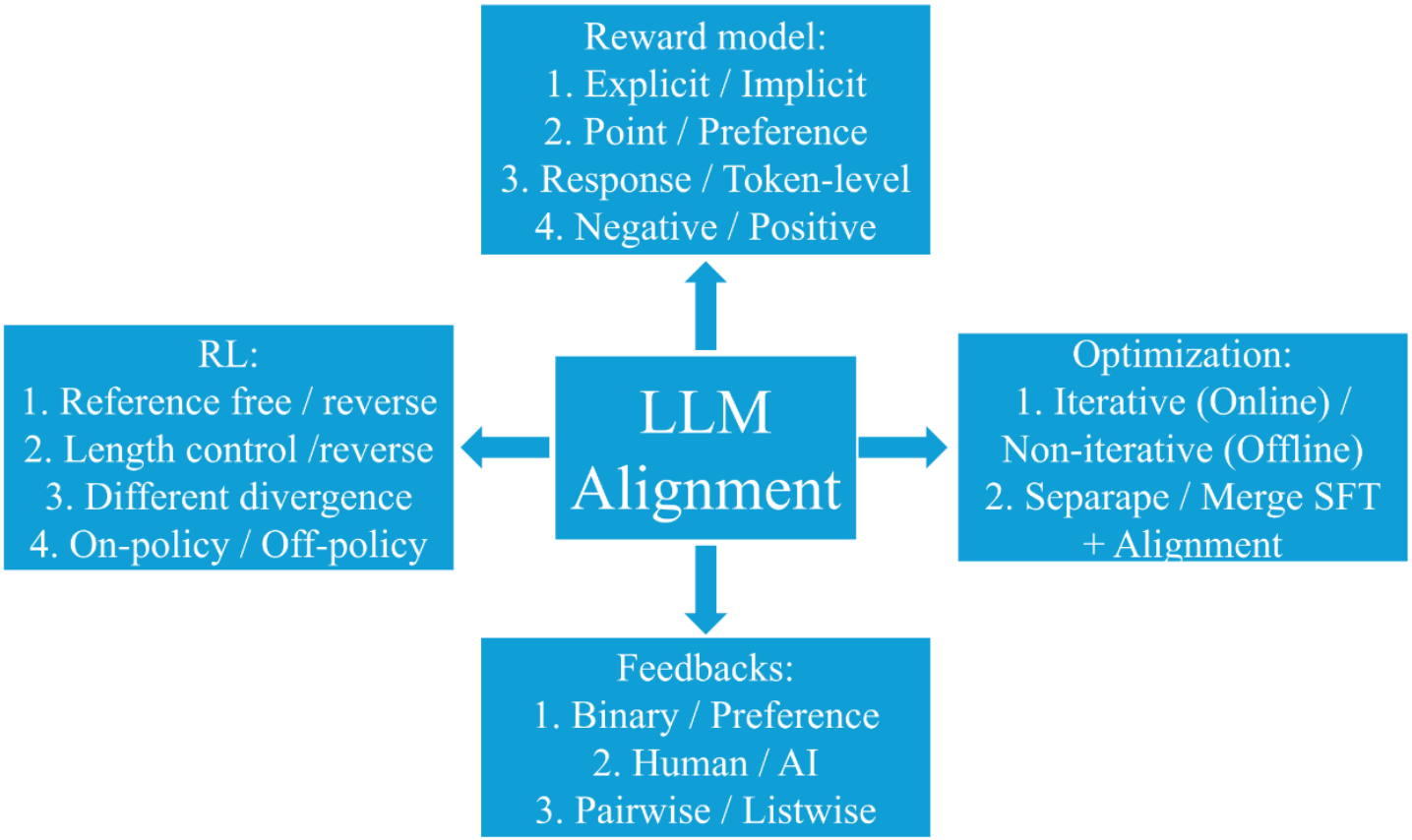
$$\mathbb{D}_{KL}\left[\pi_\theta||\pi_{ref}\right] = \frac{\pi_{ref}(o_{i,t}|q,o_{i,<t})}{\pi_\theta(o_{i,t}|q,o_{i,<t})} - \log\frac{\pi_{ref}(o_{i,t}|q,o_{i,<t})}{\pi_\theta(o_{i,t}|q,o_{i,<t})} - 1,$$

其中,q 是问题 query,o_i 是输出的 G 个 response,G 是采样组的样本数，归一化处理 reward 得到优势 $\hat{A}_{i,t} = \tilde{r}_i = \frac{r_i - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r})}$

观察公式,GRPO 是 token level 的

GRPO 与 PPO 的主要区别有：

- GRPO 省略了 value function model.(rlhf 中即为 critic model)，可以理解为摒弃了对训练过程的监督，直接以结果作为考量.
- GRPO reward 计算，改成了一个 query 生成多个 response(输出 o)，然后 reward 打分。
- PPO 优势函数计算时，KL 是包含在 GAE 内部的。GRPO 直接挪到了外面，同时修改了计算方法。
- 每一轮训练后，根据新的采样输出更新 reward model.


Reward model:
1. Explicit / Implicit
2. Point / Preference
3. Response / Token-level
4. Negative / Positive
RL:
1. Reference free / reverse
2. Length control /reverse
3. Different divergence
4. On-policy / Off-policy
LLM Alignment
Optimization:
1. Iterative (Online) / Non-iterative (Offline)
2. Separape / Merge SFT + Alignment
Feedbacks:
1. Binary / Preference
2. Human / AI
3. Pairwise / Listwise

Reward model

奖励模型是一个微调的 LLM，根据 prompt 和生成的 response 分配了分数

- 点态奖励，奖励形式:$r(x, y)$
- 其中 x 为输入大模型的 prompt 指令,y 为 response

有如下几种分类

- 显式奖励 or 隐式奖励
- 点态奖励 or 偏好模型
- token level or response level
- 正向奖励 or 负向奖励

显式奖励 or 隐式奖励

三元组构成的偏好数据集:(x, y_w, y_l)

- prompt :x, 人类期望的响应 y_w 和一个人类不期望的响应 y_l
- 显式奖励模型: 表示为 $r_\phi(x, y)$ (例如 PPO,GRPO)
- 隐式奖励模型: 表示为 $r_\theta(x, y)$, 绕过了训练显式奖励模型的过程 (例如 DPO 绕过了直接建立 reward model)

点态奖励 or 偏好模型

对于同一个 prompt, 有一个期望的 response 和一个不期望的 response 对应的两个点态奖励分数 $r(x, y_w)$ 和 $r(x, y_l)$. Bradley-Terry (BT) 模型得到期望 response 优先于不期望 response 的概率

$$P(y_w > y_l|x) = \sigma(r(x, y_w) - r(x, y_l))$$

BT 模型的方法, 从点态奖励到偏好模型无法直接获得成对偏好, 也无法适应人类标注中的不一致性. 为了解决这个问题, 提出了 Nash 学习方法.(Nash Learning from Human Feedback,SPPO,DNO)

token level or response level

在原始数据集中以三元组形式收集，即 x, y_w, y_l，奖励是按 response 给出的。因此，在 RLHF 和 DPO 中，奖励是在 response 级别构建的；在 RL 的 MDP 中，奖励是在每个动作 (token) 之后给出的，为了在每个动作之后实现对齐，引入了 token 级奖励模型.(TDPO)
DeepSeekMath 中所有 token 共享 response 的奖励信号; 或者把每个 token 的 KL 散度作为当前生成步的 reward.

正向奖励 or 负向奖励

在 RLHF 数据集中，人类标注了期望和不期望的响应。最近，随着 LLM 能力的进步，一些研究人员提出，LLM 可以生成比人类标注者更高质量的期望响应。因此，他们选择仅使用收集数据集中的提示和不期望响应，利用 LLM 生成期望响应

feedback

- 偏好反馈与二元反馈: 形式如 $y_w > y_l$ 的偏好反馈, 收集困难, 采用二元 (binary) 反馈 (只给出积极" 赞" 或消极" 踩" 反馈)(KTO,DRO)
- 成对反馈与列表反馈; 成对反馈针对给定 prompt, 向评估者展示两个 response, 选择更喜欢的一个; 列表反馈对 K 个候选 response 做 C_k^2 次比较确定顺序 (LIPO)
- 人类反馈与 AI 反馈:AI 反馈提供偏好降低标注成本

优化

- 迭代/在线偏好优化与非迭代/离线偏好优化: 仅利用收集的数据集进行对齐的过程被称为非迭代/离线偏好优化. 迭代/在线偏好优化: 人类标注新数据或 LLMs 扮演双重角色——既生成响应又评估它们.
- 分离 SFT 与对齐与合并 SFT 与对齐; 传统上先 SFT 后对齐 (可能既繁琐又容易导致灾难性遗忘), 但也可以将 SFT 与对齐整合到一个过程中以简化微调.

RL

- 基于参考的 RL 与无参考的 RL: 引入参考策略是为了避免微调后策略相较原策略太远, 带来了显著的内存负担. 一些方法避免使用了参考策略.
- 长度控制 RL:LLM 倾向于偏好冗长的 response, 但没有更多的信息量,R-DPO 和 SimPO ,RLOO 引入了对长度控制的考虑
- RL 中的不同散度:KL 散度已被发现会降低 response 的多样性. 为了解决这一问题, 研究了探索不同散度度量的效果
- 在线 RL 与离线 RL: 在线 RL 从策略的最新版本中采样 response, 使得策略与 response 一致. 离线 RL 依赖早期的 response, 在策略更新后可能不一致.

Step 1

Collect demonstration data and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3.5 with supervised learning.

Step 2

Collect comparison data and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.

Step 3

Optimize a policy against the reward model using the PPO reinforcement learning algorithm.

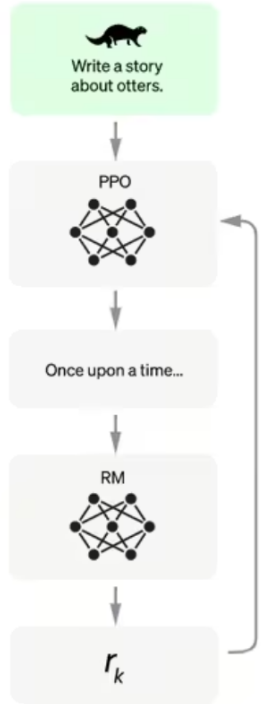
A new prompt is sampled from the dataset.

The PPO model is initialized from the supervised policy.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



reward learning

$$L_{\mathrm{RM}}(r_\phi) = -\frac{1}{C_K^2}\mathbb{E}_{(x,y_w,y_l)\sim D}\left[\log\left(\sigma\left(r_\phi(x,y_w) - r_\phi(x,y_l)\right)\right)\right]$$

考虑到同一个 prompt 的 response 都具有一定的相关性，如果完全打乱进行随机训练会导致过拟合，在同一个 prompt 下回复了 K 个 response，共有 C_k^2w loss response 对进行输入进行训练，改善了过拟合

缺点是忽略了相对之间的关系，没有说明 response 之间的相对得分。也就是说，对分数相似的对响应或分数差异很大的响应被对待相同。后续有人用 listwise 加以改进

RL 训练

$$\pi_\theta^*(y|x) = \max_{\pi_\theta} \mathbb{E}_{x\sim D}\left[\mathbb{E}_{y\sim\pi_\theta(y|x)} r_\phi(x,y) - \beta D_{\mathrm{KL}}(\pi_\theta(y|x)||\pi_{\mathrm{ref}}(y|x))\right] + \gamma \mathbb{E}_{x\sim D_{\mathrm{pretrain}}}\left[\log(\pi_\theta(x))\right]$$

- 最大化奖励
- 最小化与 referennce 的差距, 避免策略偏离太多
- 避免 alignment tax , 加入 $\gamma \neq 0$ 后, 避免了公共 NLP 数据集 (即预训练数据集) 上的表现降级 (ppo-ptx)

所谓「对齐税」(Alignment Tax), 指的是在使人工智能系统符合人类偏好的过程中, 所不可避免付出的性能损失或代价。在对齐后, 在下游任务中的表现降级

对齐指标:Helpful, Honest, Harms

- Helpful: 遵循指示, 且能推断 prompt 的意图
- honest : 评估该模型在封闭领域任务上编造信息的倾向; 真实问答（TruthfulQA）基准测试上的表现
- harms: 某项输出是否不恰当

Anthropic:

- 较小模型中体现了 alignment tax, 但有利于较大模型 (13B,52B), 对于规模较大的模型, 仅仅使用 ppo 就可以在 NLP 的下游任务中得到很高的对齐奖励. 并且认为 RL 训练过程中 $\beta = 0.001$ 是最佳参数
- 奖励模型的准确度和参数 size 成对数线性关系, 参数规模越大的奖励模型越稳健
- 奖励与 $D_{KL}(\pi_\theta || \pi_{ref})$ 之间存在线性趋势
- 分布外（OOD）技术来检测并拒绝不合理的请求。
- 在线训练模式，通过与众包工作者互动获取新的人类偏好数据，每周对奖励模型和强化学习策略进行更新

Thanks!