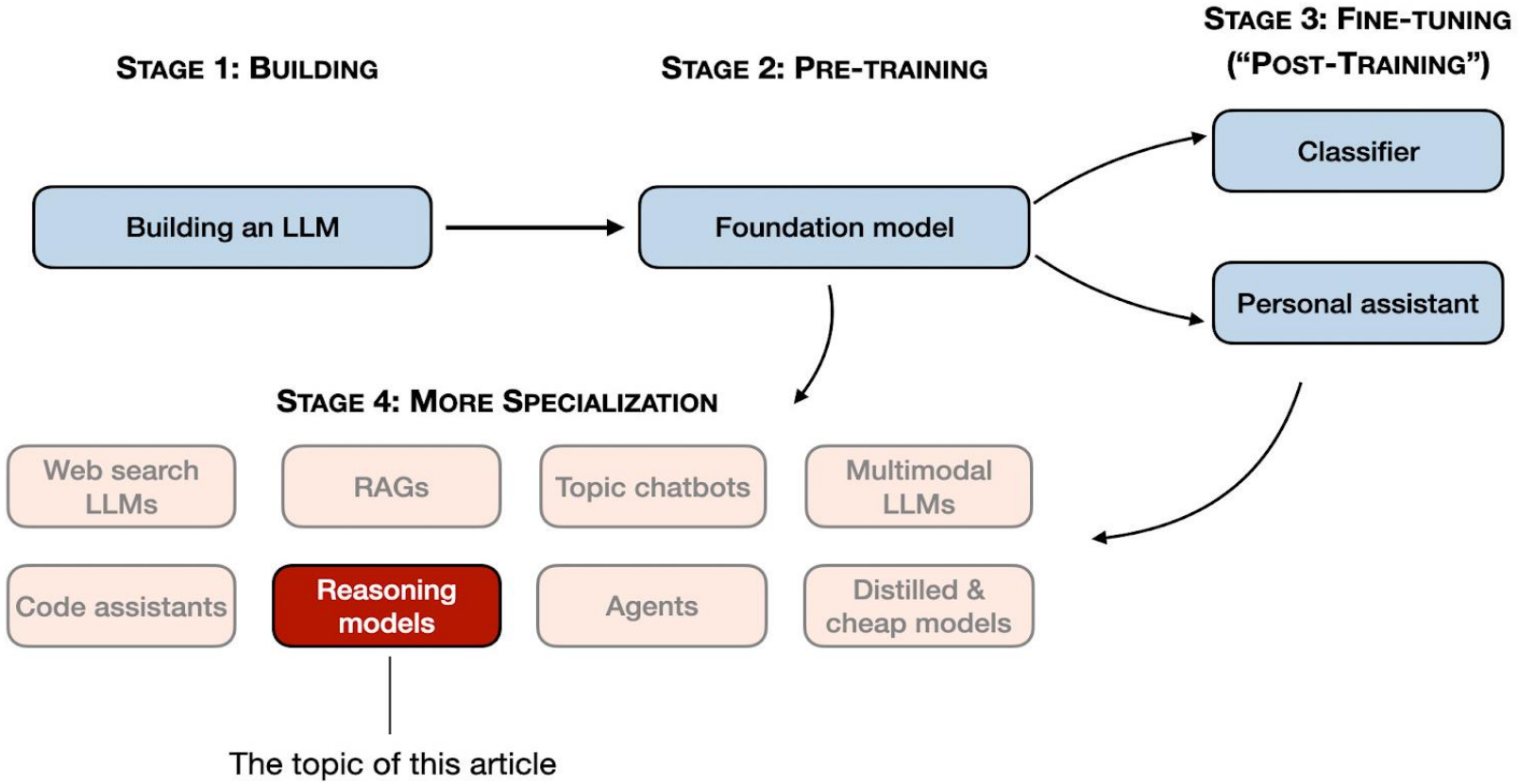
大模型训练方式发展趋势

Updated: 2025.02.26

- LLM诞生之初，常见的训练方法从Self-supervised Pre-training发展到Supervised Fine-tuning
- 近几年，面向特定问题（STEM等）的先进优化算法逐渐诞生，如RLHF、RAG、Test-time Computing、RL-based Reasoning Model等

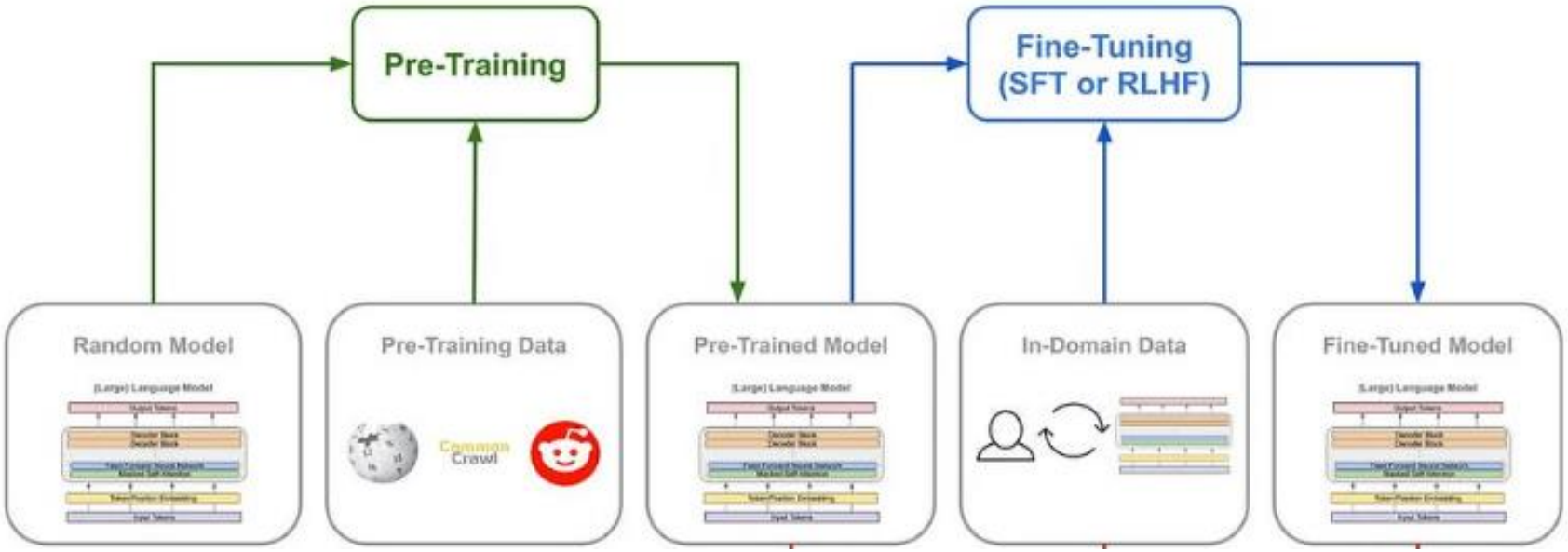


大模型训练方式介绍

Pre-training → **Fine-tuning**

- 通常是**无监督学习**，采用掩码机制或自回归机制训练
- 对算力与数据规模**需求大**，在大量的无标签数据中学习**通用特征**和先验知识
- 基础模型通常采用Multi-head-Attention-based **Encoder/Decoder**架构

- 通常是**有监督学习**，针对特定领域对预训练模型微调
- 对算力与数据规模**需求小**，在小规模有标签数据中学习**领域特征**
- 在预训练模型基础上可增加针对下游任务的**其它模型**
- 通常会引入**ICL、LoRA**等方法来提升训练速度与性能



什么是“推理模型”？

- 普通的大语言模型（**LLM**）通常只会提供一个**简短的答案**
- **推理模型**通常会包含中间步骤，揭示**思考**过程，擅长解决更复杂的推理任务，例如解谜、数学证明

什么是推理?

1. 中国的首都是哪里? （事实问题）

2. 如果一列火车以每小时60英里的速度行驶3小时，它会行驶多远? （推理问题）

If a train is moving at 60 mph and travels for 3 hours, how far does it go?

LLM基础模型回答:

The train travels 180 miles.

Plain response

推理模型回答:

To determine the distance traveled, use the formula:

Distance = Speed × Time

Given that the speed is 60 mph and the time is 3 hours:

Distance = 60 mph × 3 hours = 180 miles

So, the train travels 180 miles.

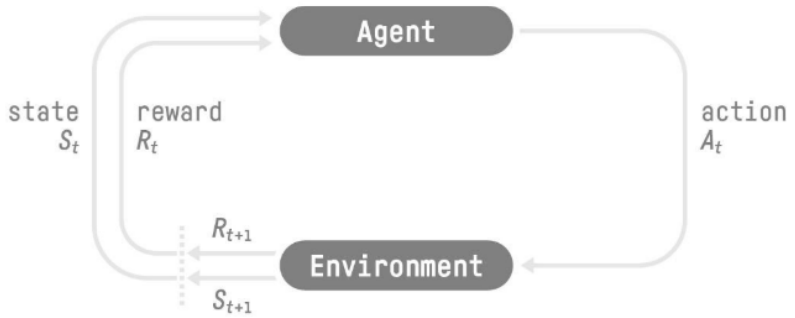
Response with intermediate reasoming steps

实现推理模型的核心方法：强化学习

- **强化学习基础知识：**
  - **RL**：一个机器人（Agent）看到一些信息（Observation）后，做出一个决策（Action），随即根据采取决策后得到的反馈（Reward）来进行**自我学习**（Learning）的过程
  - **RL与LLM对应**：

概念	RL	LLM
Agent/Policy	智能体	LLM Model (GPT、Bert、...)
Observation/State	对环境在某一时刻的描述	当前给定的输入/输出
Action	智能体在给定状态下进行的行为	预测过程的search/sampling过程（随机性）
Environment	智能体所处并阈值互动的外部世界	人工有监督反馈，或格式判定自监督反馈等
Reward	环境对智能体执行特定动作的及时反馈	当前输出的奖励（如句子是否更通顺）

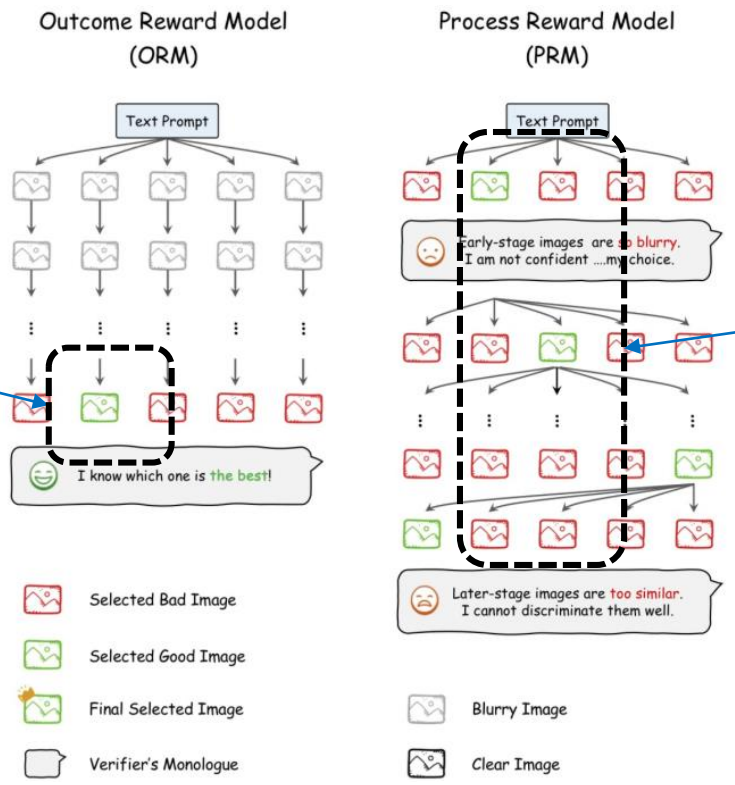
  - **RL迭代：**
    - **执行**：循环输出一系列状态、动作、奖励和下一个状态
    - **优化目标**：最大化整个RL迭代循环过程的**累积奖励** $\sum_{t=1}^{|o|} R_t$，从而提升智能体更准确执行任务的能力



推理模型技术路线1：有监督的RLHF

- **核心：**基于有监督数据，通过RLHF等方式提升模型推理以及Instruction-following能力
- **主流形式：**
  - **ORM** (Outcome Reward Model)：对推理**结果整体**打分评估
  - **PRM** (Process Reward Model)：对**每一步推理过程**进行更细粒度的打分，人工标注成本高，容易产生reward hacking

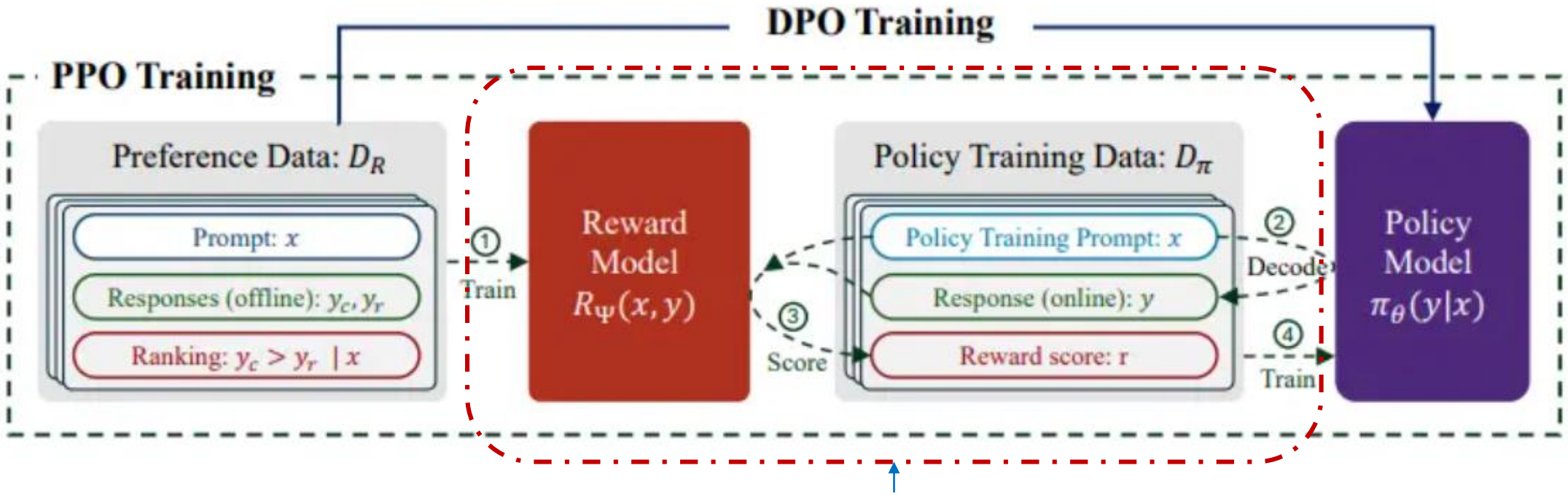
进行多步推理后，只评估最后的整体输出



对每一步的推理过程都进行评估

推理模型技术路线1：有监督的RLHF

- **常用优化策略：**
  - **PPO** (Proximal Policy Optimization)：基于策略梯度的优化算法，通过**最大化累积奖励**来优化策略
  - **DPO** (Direct Preference Optimization)：通过直接优化用户或系统的偏好来调整策略，不依赖于传统的奖励信号，而是通过**对比学习或直接反馈优化策略**



DPO省略了Reward Model计算，基于人工标注的数据直接更新策略

推理模型技术路线1：有监督的Reward Model

- **代表性工作：ChatGPT**，OpenAI于**2022.11**发布，带动基于有监督数据集的推理模型发展
- **其它工作**： Sparrow、InstructGPT、ChatGPT、EAGER、ELLM、Eureka、GPT4、...

AlpacaEval Leaderboard

An Automatic Evaluator for Instruction-following Language Models

Length-controlled (LC) win rates alleviate length biases of GPT-4, but it may favor models finetuned on its outputs.

Version: AlpacaEval | AlpacaEval 2.0 Filter: Community | Verified

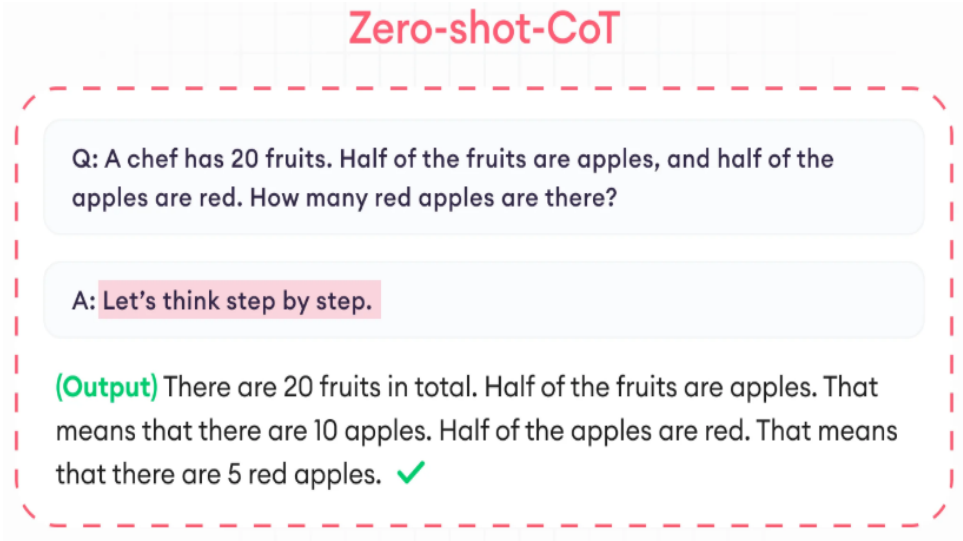
Baseline: GPT-4 Preview (11/06) | Auto-annotator: GPT-4 Preview (11/06)

Rank	Model Name	LC Win Rate	Win Rate
1	GPT-4 Omni (05/13)	57.5%	51.3%
2	GPT-4 Turbo (04/09)	55.0%	46.1%
3	Yi-Large Preview	51.9%	57.5%
4	GPT-4o Mini (07/18)	50.7%	44.7%
5	GPT-4 Preview (11/06)	50.0%	50.0%
6	Claude 3 Opus (02/29)	40.5%	29.1%
7	Llama 3.1 405B Instruct	39.3%	39.1%
8	GPT-4	38.1%	23.6%
9	Qwen2 72B Instruct	38.1%	29.9%
10	Llama 3.1 70B Instruct	38.1%	39.1%

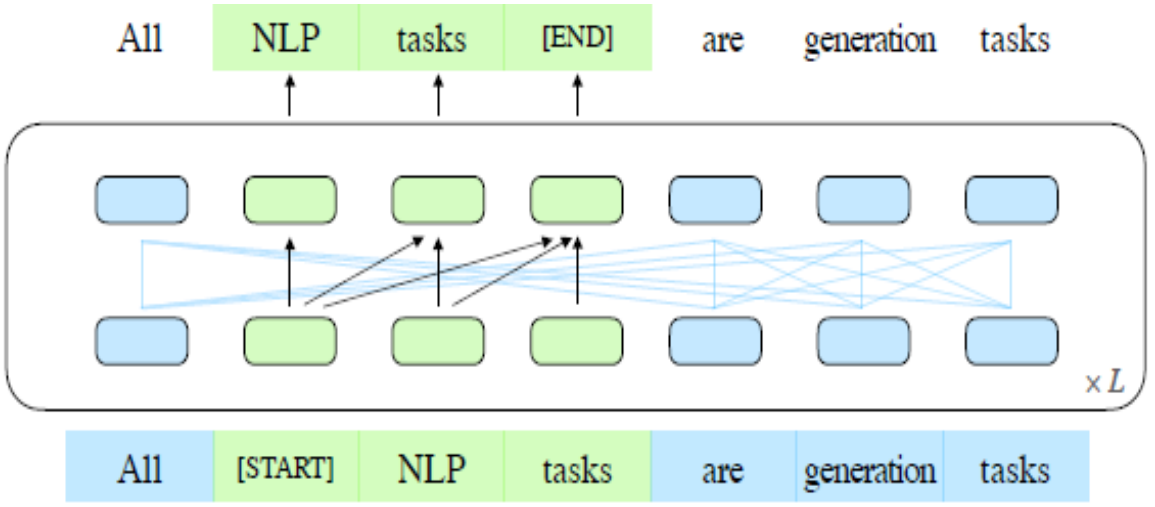
有监督训练的推理模型相比非推理模型，在推理任务上表现更好表现

推理模型技术路线2：无监督的RL

- **核心**：采用RL方法，**无需依赖有标注数据**，大幅提升模型的推理能力
- **关键算法**：
  - **极简的奖励策略**：如格式奖励等，无需人工打分或标注
  - **冷启动**：基于少量高质量CoT数据的模型微调
  - **多阶段训练**：结合规则奖励（如语言一致性）与偏好奖励，确保模型在开放任务的安全性



CoT生成：基于推理模型，通过zero-shot/prompt learning等方式生成



SFT: 通过自回归监督训练方式，基于CoT推理数据微调模型

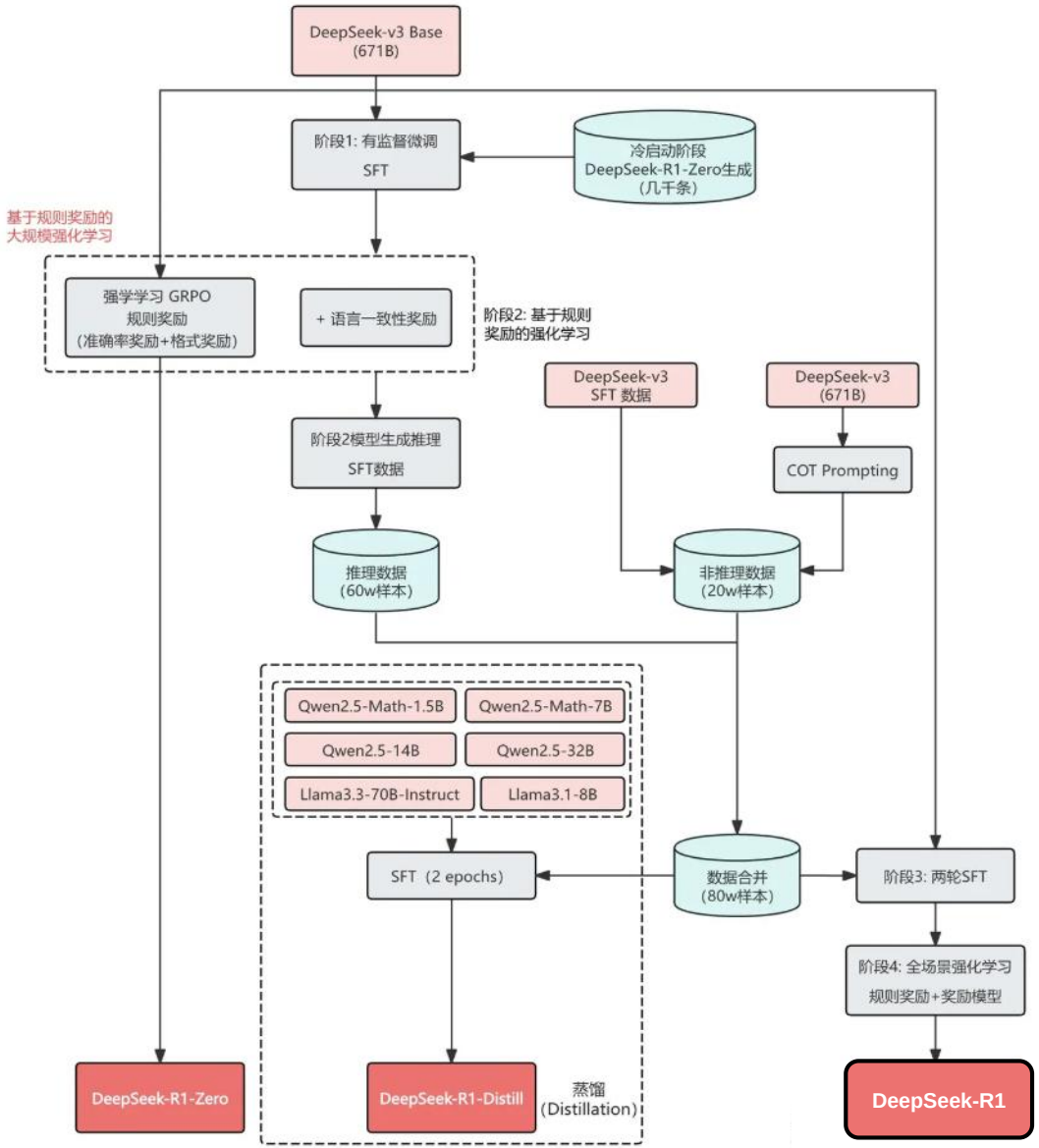
推理模型技术路线2：无监督的RL

- **代表性工作：DeepSeek-R1：采用RL方法，大幅提升模型推理能力**
- **涌现的其它工作**：
  - **2025.01 rStar-Math**：基于MCTS实现深度思考，专注于数学领域的推理优化
  - **2025.01 DeepSeek-R1**：基于纯RL的推理模型，在Math等领域的SOTA模型，颠覆推理模型实现范式
  - **2025.02 LIMO**：验证基于少量(800+)精心设计的样本+RL，可实现数学推理能力的质的飞跃
  - **2025.02 OpenAI-o1-IOI**：基于RL+TTC打造的在编程竞赛中表现超群的AI模型
  - **2025.02 ReasonFlux**：基于少量(500+)精巧设计的CoT数据+分层强化学习，验证创新的方法可弥补算力的差距，并大幅提升数学推理能力
  - **More and more……**

DeepSeek系列模型构建过程总览

- **模型列表：**
  - **Stage 1**: DeepSeek-V3
  - **Stage 2**: DeepSeek-R1-Zero
  - **Stage 3**: DeepSeek-R1、DeepSeek-R1 Distill
- **拒绝采样方法生成数据：**
  - 推理数据：**60w条**
  - 非推理数据：**20w条**
  - 冷启动数据：**几千条**

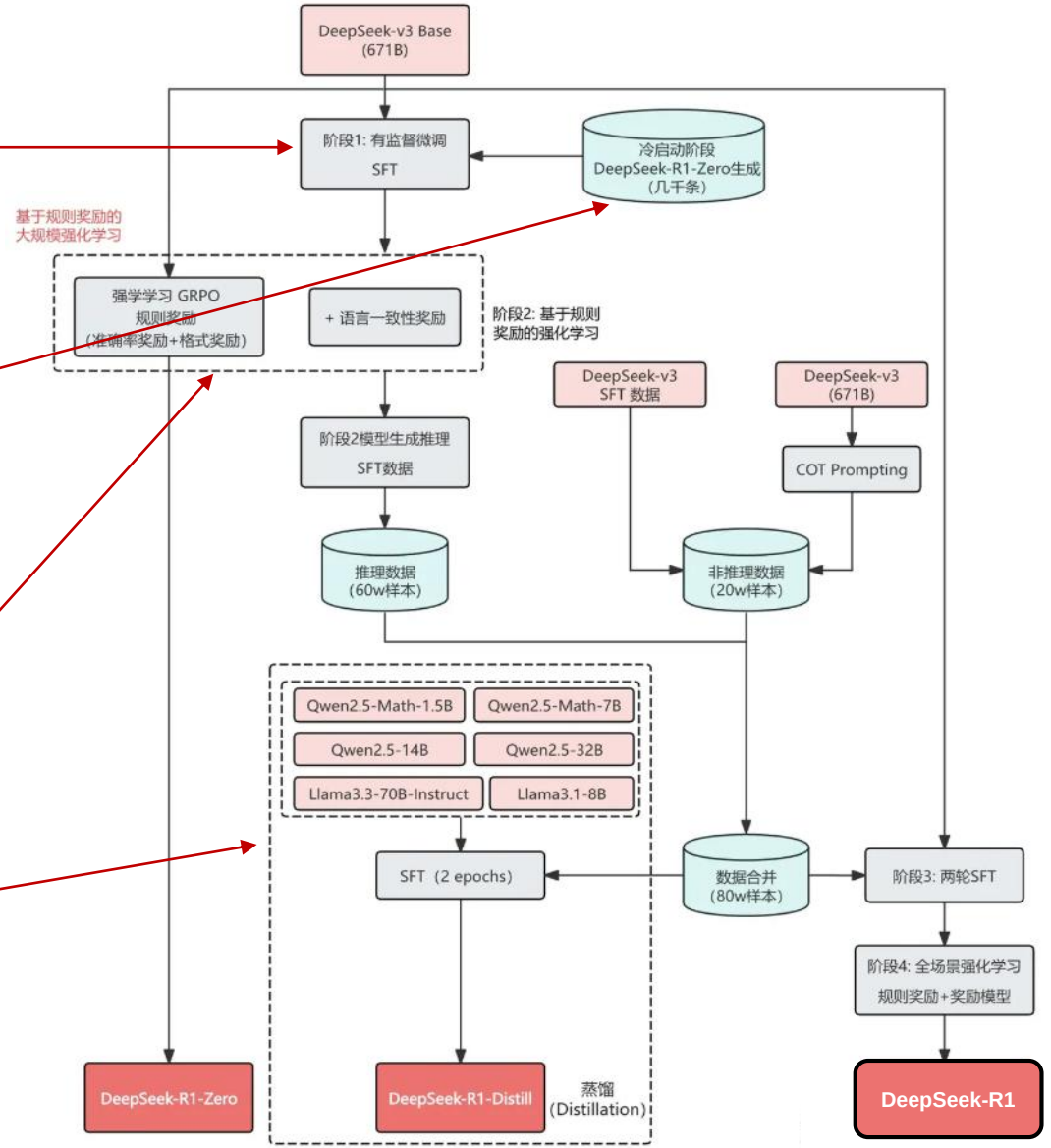
模型	训练方法
DeepSeek-V3 Base	Pre-train + Fine-tune + RLHF
DeepSeek-R1-Zero	纯RL（GRPO）
DeepSeek-R1 Distill	SFT
DeepSeek-R1	SFT + 纯RL + RLHF



[1]. https://www.zhihu.com/tardis/bd/art/19868935152?source_id=1001

DeepSeek系列模型训练：关键技术

- 基于**大量推理+非推理**CoT数据，微调**基础**模型
- 基于**少量推理**CoT数据，微调**推理**模型，让模型更快进入**稳定训练**阶段
- RL+RLHF，基于**大量推理+非推理**CoT数据+**有标签**数据，通过优化Reward Model，微调**推理**模型
- 基于**大量推理+非推理**CoT数据，通过SFT微调**现有**模型（Llama、Qwen）



策略优化算法：PPO

- **PPO算法：RLHF训练常采用的优化算法，关键步骤包含**
  - ① **收集数据**：基于当前模型（policy）生成一组**样本**，包括query、当前状态/output、reward、value
  - ② **计算优势估计**：计算优势函数（advantage function），根据reward和value值评价当前状态相对于平均水平的好坏，常用的优势函数包括时间差分（TD）估计、**广义优势估计**（GAE）等
  - ③ **下一个状态估计**：根据优势估计结果，获得可能的**下一个状态**/输出
  - ④ **优化Critic/Value Model**：用第二步所得优势估计值取代真实reward，并用更新前的value model限制更新后的奖励模型的范围，从而保证value-model**向正确的方向完成小步更新**
  - ⑤ **更新Actor/Policy Model**：提出PPO Penalty、**PPO Clip**等优化，解决传统PO算法存在的过冲（错过奖励峰值点）、二阶优化慢、无法处理大参数矩阵等瓶颈
  - ⑥ **迭代优化**：使用新的策略参数**重复**以上步骤，直到满足某些停止准则，如策略性能不再提升、达到设定的迭代次数等

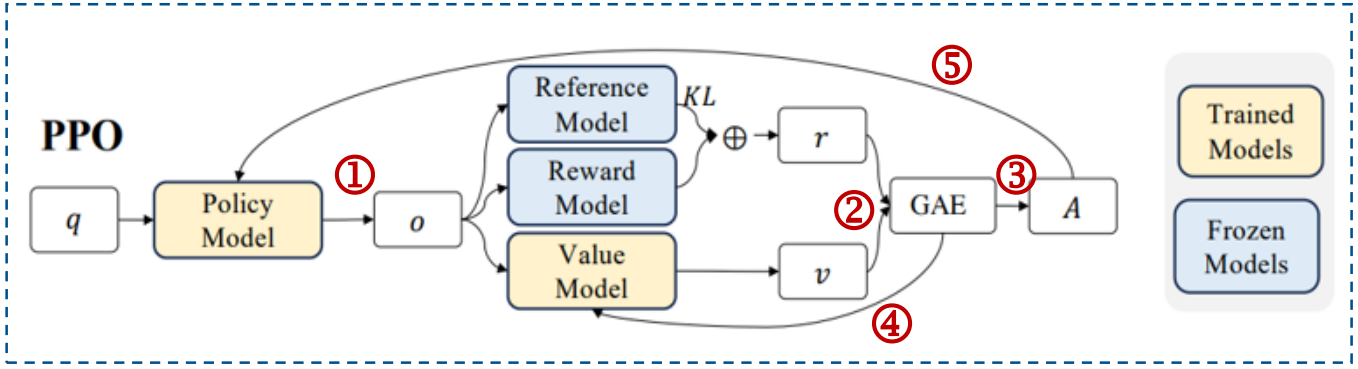
Critic优化目标:

$$V_\pi^{CLIP}(s_t) = clip(V_\pi(s_t), V_\pi^{old}(s_t) - \epsilon, V_\pi^{old}(s_t) + \epsilon)$$

$$\arg\min_{V_\pi} L(V_\pi) = E_t[max((A_t^{GAE} + V_\pi^{old}(s_t)) - V_\pi(s_t))^2, (A_t^{GAE} + V_\pi^{old}(s_t)) - V_\pi^{CLIP}(s_t))^2)]$$

Actor优化目标:

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



[1]. Proximal policy optimization algorithms. https://arxiv.org/pdf/1707.06347, 2017. OpenAI.
[2]. https://zhuanlan.zhihu.com/p/13467768873

策略优化算法：PPO

- **PPO算法中Policy/Actor优化：**

① **定义policy**：通常是一个神经网络π_θ，输入为某一个时刻的状态/输入s_t，输出为当前可能执行的动作$a_t \sim \pi_\theta(\cdot|s_t)$

② **定义状态转移**：服从随机状态转移，即$s_{t+1} \sim P(\cdot|s_t, a_t)$

③ **获得运动/生成轨迹**：$\tau = (s_1, a_1, s_2, a_2, s_3, a_3, \ldots, s_{|o|}, a_{|o|})$, $|o|$为总的输出token数量

④ **定义累积奖励**：轨迹中所有动作取得的(折扣)奖励之和 $R_\tau = \sum_{t=1}^{t=|o|} \gamma^t r_t$，其中$\gamma \in (0,1)$为折扣系数，$r_t$为每次输出的奖励

⑤ **评估policy好坏**：定义为动作累积奖励的期望，即$J_{\pi_\theta} = E_{\tau \sim \pi_\theta}[R_\tau]$,定义策略和状态转移都是随机的，优化目标是最大化累积折扣奖励的期望，对应着训练过程中我们**需要采样多次轨迹（search策略，多个轨迹τ）**，在足够多的轨迹上估计期望（擅用log导数特性推导梯度）

⑥ **优势估计修正**：使用动作的价值$A_\tau = \sum_{t=1}^{t=|o|} adv_t$修正奖励，降低采样随机性的影响，则$J_{\pi_\theta} = E_{\tau \sim \pi_\theta}[A_\tau(s_t, a_t)]$；以时间差分方法TD为例，$adv_t = r_t - (V_\pi(s_t) - \gamma V_\pi(s_{t+1}))$，其中$V_\pi$为不同动作的价值，通过Value Model计算得到

⑦ **重要性采样修正**：采用off-policy方法，将一批数据分别输入至不同的策略，降低采样成本，提升训练效率

a) 使用MC采样得到一批数据（输入），输入至旧策略π_{old}，假设输出（轨迹、奖励），假设服从分布$q(x)$

b) 将该批采样数据输入直至当前策略π_θ，获得对应输出（更新后的轨迹、奖励），假设服从分布$p(x)$

c) 由于无法从$p(x)$中采样，因此将奖励的期望修正为$J_{\pi_\theta} = E_{\tau \sim \pi_{\theta_{old}}}\left[\frac{p(x)}{q(x)} A_{\tau \sim \pi_\theta}(s_t, a_t)\right] = E_{\tau \sim \pi_{\theta_{old}}}\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} A_\pi(s_t, a_t)\right]$

⑧ **限制新旧策略间的差距**：限制策略变化幅度防止梯度过冲，即$J_{\pi_\theta} = E_{\tau \sim \pi_{\theta_{old}}}\left[clip(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} A_\pi(s_t, a_t), 1-\varepsilon', 1+\varepsilon')\right]$

⑨ **优化policy**：通过最大化奖励的期望更新策略，即$\nabla\theta = argmax_\theta J_{\pi_\theta} = E_{\tau \sim \pi_{\theta_{old}}}\left[\frac{\nabla \pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} A_\pi(s_t, a_t)\right]$

注: 上述步骤已省略KL penalty: $KL[\pi_\theta(a_t|s_t), \pi_{\theta_{old}}(a_t|s_t)]$，PPO文章实验验证其效果不如Clip.

补充细节1

- **价值函数（Value function）**：价值函数评估了状态或状态-动作对的价值，即在某状态或动作下，智能体预期能够获得的累积奖励，常见的价值函数有
  - 状态价值函数（State-value function）：刻画的是从某个状态s开始，依据策略π采取动作，智能体可以得到的累积折扣奖励的**期望值**，$V_\pi(s) = E_\pi[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | S_t = s] = E_\pi[R_\tau | S_t = s]$
  - 状态-动作价值函数 （Action-value function）：刻画的是对于每个**状态和动作pair**，如果智能体在该状态s下启动，执行该动作a，智能体预期得到的累积奖励，$Q_\pi(s,a) = E_\pi[R_\tau | S_t = s, A_t = a]$
- **Actor-Critic** 方法结合了基于策略和基于价值的优点。在这种方法中：
  - Actor：代表策略部分 (policy-based method)，负责生成动作。Actor 直接建模策略 $\pi(a|s)$ ，并根据 Critic 的反馈调整策略参数，以优化长期奖励。
  - Critic：代表价值部分 (value-based method)，负责评估采取的动作。Critic 学习价值函数$Q_\pi(s,a)$或 $V_\pi(s)$，并计算如 TD（时序差分）误差，这是当前估计的价值和实际获得的奖励之间的差异。
- 通过这种**结合**，Actor-Critic 方法可以
  - 利用 Critic 的稳定性和快速收敛的特点
  - 同时通过 Actor 学习更复杂的策略，包括在连续动作空间中的应用。
  - 此外，Critic 的反馈可以帮助减少 Actor 更新时的方差，从而提高学习的稳定性。
- **Actor对应的梯度**最终推导为：$\nabla J_{\pi_\theta} \approx \frac{1}{N}\sum_{n=1}^{N}\sum_{t=0}^{T_n-1} R_{\tau_n} \nabla log\pi_\theta(a_t|s_t) = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=0}^{T_n-1} R_{\tau_n} \frac{\nabla\pi_\theta(a_t|s_t)}{\pi_\theta(a_t|s_t)}$， N为采样的轨迹数量，T_n为每条轨迹长度（采样方法如拒绝采样、MCTS、ETS等，会影响策略多样性、时间复杂度/速度、缓存需求）
  - $\nabla log\pi_\theta(a_t|s_t)$的形式相当于在每一个动作的梯度下除以采样到这个动作的概率，这样避免了在采样过程中采到很多奖励值很低但是出现频次高的动作，造成模型对这种**低奖励值高频次动作的偏好**
  - 同时为了避免某些好的动作没有被采样到但是采样到的坏的动作梯度永远为正造成模型训练失败，一条轨迹的累积奖励R_{τ_n} 不应该总为正值，可以考虑统一减去baseline，因此R_{τ_n}的优化空间很多，有很多刻画单步回报的项可供选择，因此后续采用**critic的价值函数**$V_\pi(s)$或$Q_\pi(s,a)$修正

补充细节2

- **Baseline计算**：
  - 通过状态价值函数、状态-动作价值函数的定义推导可得，两者的联系为$V_\pi(s_t) = E_{a_t\sim\pi(\cdot|s_t,a_t)}[Q_\pi(s_t,a_t)] = \sum_{a_t\epsilon \mathrm{A}}\pi(a_t|s_t)Q_\pi(s_t,a_t)$
  - **Temporal Difference**定义：advantage表示为在状态s_t下，某个动作a_t相对于平均水平有多好。具体来说，它是Q值(动作价值)减去V值(状态价值)：
    - 用$A_\pi(s_t,a_t)$代替R_τ可得actor的优化目标
    - 其中$V_\pi(s_t) - \gamma V_\pi(s_{t+1})$可以视为baseline
  - Critic的优化目标为最小化TD error
  - 由于TD error是无偏估计，因此有了优化方法GAE
    - 通过结合多步 TD 估计来平衡偏差和方差

TD Advantage/baseline推导:

$$\begin{aligned} A_\pi(s_t,a_t) &= Q_\pi(s_t,a_t) - V_\pi(s_t) \\ &= E_{s_{t+1}\sim P(.|s_t,a_t)}[r_t + \gamma V_\pi(s_{t+1})] - E_{s_{t+1}\sim P(.|s_t,a_t)}[V_\pi(s_t)] \\ &= E_{s_{t+1}\sim P(.|s_t,a_t)}[r_t + \gamma V_\pi(s_{t+1}) - V_\pi(s_t)] \\ &= E_{s_{t+1}\sim P(.|s_t,a_t)}[TD\_error] \end{aligned}$$

Critic优化目标:

$$\nabla J(\pi_\theta) \approx \frac{1}{N}\sum_{n=1}^{N}\sum_{t=0}^{T_n-1}(r_t + \gamma V_\pi(s_{t+1}) - V_\pi(s_t))\nabla log\pi_\theta(a_t|s_t)$$

$$arg\max_{\pi_\theta} J(\pi_\theta) = E_t[A_\pi(s_t,a_t)log\pi_\theta(a_t|s_t)]$$

Actor优化目标:

$$arg\min_{V_\pi} L(V_\pi) = E_t[(r_t + \gamma V_\pi(s_{t+1}) - V_\pi(s_t))^2]$$

- **轨迹生成方式**：
  - On-policy 方法，生成数据的策略必须是当前优化的策略
  - **Off-policy 方法**允许从与目标策略不同的行为策略中采样数据

  （降低采样成本，提升训练效率，减少$A_\pi(s_t,a_t)$的多次计算）
    - 用π_{old}和环境交互，得到一批经验数据（状态价值、优势、回报）
    - 将把这一批回合数据重复使用k次：即我们先把这批数据喂给π_{old}，更新得到π_{θ_0}；我们再把同一批数据喂给π_{θ_0}，更新得到π_{θ_1}；以此类推，做k次更新后，我们得到π_θ。这个过程叫off-policy（产出数据的策略和用这批数据做更新的策略不是同一个）
    - 训练的过程中，由于策略已经发生了改变($q(x)$ 到 $p(x)$)，采样出来的分布已经变了,由于无法从$p(x)$中采样。因此奖励的期望修正为$J_{\pi_\theta} = E_{\tau\sim\pi_{\theta_{old}}}\left[\frac{p(x)}{q(x)}A_{\tau\sim\pi_\theta}(s_t,a_t)\right] = E_{\tau\sim\pi_{\theta_{old}}}\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}A_\pi(s_t,a_t)\right]$
- **采样方法**：
  - Reject Sampling：Best-of-N，use verifier including PRM and ORM，轨迹多样性低，imbalance distribution
  - Tree Search: Beam search、MCTS、DVTS、**FastMCTS**，PRM is used to provide reward score，搜索空间大，缓存需求大

策略优化算法：DPO

- **DPO算法：**不需要奖励模型，直接利用人类偏好构建损失函数，使得受偏好的输出概率更高，未受偏好的输出概率更低

① **定义策略**：策略模型$\pi_\theta(y|x)$， y表示输出， x表示输入

② **构建人类偏好数据**：表示为(x, y^+, y^-)，y^+表示在人类偏好中胜出的回复，y^-表示不被偏好的回复

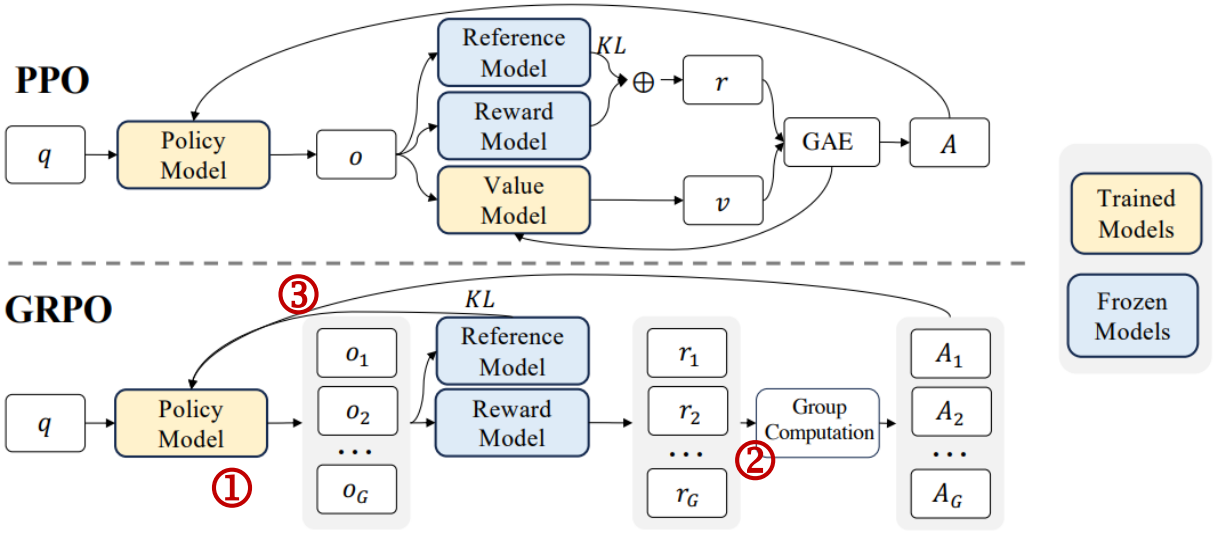
③ **计算偏好概率**：代表偏好y^+胜过y^-的概率$P_\theta(y^+ \succ y^-|x) = \frac{\pi_\theta(y^+|x)}{\pi_\theta(y^+|x)+\pi_\theta(y^-|x)}$

④ **最大化偏好概率**：将偏好概率带入目标函数可得策略的目标为$J(\theta) = E_{(x,y^+,y^-)}[\log(\frac{\pi_\theta(y^+|x)}{\pi_\theta(y^+|x)+\pi_\theta(y^-|x)})]$

⑤ **引入正则化项**：使训练更稳定，$J(\theta) = E_{(x,y^+,y^-)}\left[\log\left(\frac{\pi_\theta(y^+|x)}{\pi_\theta(y^+|x)+\pi_\theta(y^-|x)}\right)\right] + \beta KL(\pi_\theta||\pi_{ref})$

DeepSeekMath创新：关键算法GRPO

- **GRPO (Group Relative Policy Optimization)**：验证了纯强化学习在 LLM 中增强推理能力的可行性
  - **核心方法：分组采样与相对奖励、无需Value Model的高效PO**
    - ① **采样输出**：在一个问题上，生成多个候选输出O（**reject sampling方式**），以及对应的奖励R（**ORM方式**）
    - ② **计算评分**：对所有R进行对比，根据排名或分数差，给出每个输出的相对评分A，替代优势函数计算（GAE等）
    - ③ **更新策略**：舍弃VM，基于分组相对评分更新策略，显著减少训练资源（PPO中VM与PM大小相当，对每个token计算对应的value并更新VM，带来显著的显存和计算负担）



[1]. DeepSeekMath. https://arxiv.org/pdf/2402.03300, 2024. DeepSeek.

DeepSeekMath创新：关键算法GRPO

- **PPO与GRPO损失函数对比：**
  - **PPO**：梯度更新，同时通过clip操作**限制策略更新的幅度**，避免新旧策略之间出现过大的偏移
  - **GRPO**：同样是包含clip操作的梯度更新方法，区别在于
    - 价值函数A_t被**分组相对奖励**$\tilde{A}_{i,t}$取代
    - 需要对多组输出(G)计算**平均奖励**
    - 引入**KL散度**，额外限制策略与参考策略（如初始SFT模型）间的差异，防止训练爆炸

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

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

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

新策略相对于旧策略在相同状态、动作上的概率比率
即新动作/旧动作的调整幅度

- q：输入问题
- π_{old}：旧策略
- o：在旧策略上的输出
- $|o|$：输出序列长度
- A_t：优势函数
- $\tilde{A}_t$：分组相对奖励
- G：采样输出样本数量
- π_θ：当前策略
- π_{ref}：参考策略
- ε：策略偏移约束参数，如0.1