大语言模型的强化学习训练
提升推理、数学与代码能力
无需人类标注的自我学习方法

许达

未来院三室

2025 年 12 月 5 日

目录

为什么需要强化学习训练 LLM?

传统方法的局限

- 监督微调 (SFT) 依赖大量人工标注数据
- 人类标注成本高、难以扩展
- 复杂推理任务的标注质量难以保证
- 模型只能学习到人类已知的解法

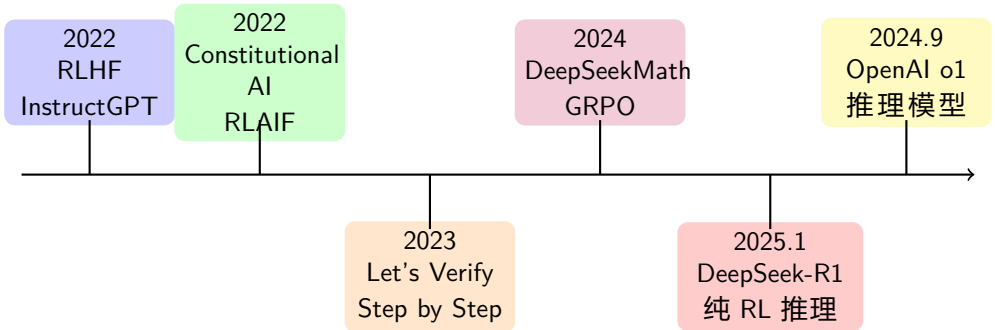
强化学习的优势

- 自动探索：模型自主发现解题策略
- 可扩展：无需人工标注
- 突破上限：可能发现人类未知的方法
- 自我改进：持续迭代优化

核心思想

通过**奖励信号**（如答案正确性、代码执行结果）引导模型自主学习推理能力

里程碑式进展


2022
RLHF
InstructGPT
2022
Constitutional
AI
RLAIF
2023
Let's Verify
Step by Step
2024
DeepSeekMath
GRPO
2025.1
DeepSeek-R1
纯 RL 推理
2024.9
OpenAI o1
推理模型

主要研究方向

数学推理	代码生成	逻辑推理
GSM8K, MATH	HumanEval, MBPP	多步推理
竞赛级问题	竞赛编程	常识推理
证明生成	代码调试	科学问答

关键论文:

- Constitutional AI (Anthropic, 2022)
- Let's Verify Step by Step (OpenAI, 2023)
- DeepSeekMath (DeepSeek, 2024)
- DeepSeek-R1 (DeepSeek, 2025)

PPO 算法回顾

Proximal Policy Optimization (PPO) 是 RL 训练 LLM 的主流算法

目标函数:

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

其中:

- $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ 是策略比率
- $\hat{A}_t$ 是优势函数估计
- ϵ 是裁剪参数（通常 0.1-0.2）

关键组件:

1. **Actor**：策略网络 π_θ（生成回答的 LLM）
2. **Critic**：价值网络 V_ϕ（评估状态价值）
3. **Reward Model**：奖励模型 R_ψ（评分）

GRPO 算法 - DeepSeek 的创新

Group Relative Policy Optimization：去除 Critic 网络，简化训练

1: **for** 每个问题 q **do**
2: 采样一组回答 $\{o_1, o_2, ..., o_G\}$ 从 $\pi_{\theta_{old}}$
3: 计算奖励 $\{r_1, r_2, ..., r_G\}$
4: 归一化优势: $\hat{A}_i = \frac{r_i - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r})}$
5: 更新策略最大化:
6: **end for**

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}_{q,\{o_i\}} \left[\frac{1}{G} \sum_{i=1}^{G} \min \left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)} \hat{A}_i, \text{clip}(\cdot) \hat{A}_i \right) \right]$$

$$-\beta \cdot \mathbb{D}_{KL}[\pi_\theta || \pi_{ref}]$$

GRPO vs PPO 对比

特性	PPO	GRPO
Critic 网络	需要	不需要
内存占用	高	低
训练稳定性	依赖 Critic 质量	组内相对比较
采样数量	单个/少量	多个（组采样）
优势估计	GAE	组内归一化
实现复杂度	高	低

GRPO 核心创新：

- 用**组内相对排名**代替绝对价值估计
- 同一问题的多个回答互相比较
- 显著降低计算成本，保持训练效果

奖励函数设计

可验证奖励（无需人类标注）：

数学问题

- 答案匹配：$r = \mathbf{1}[\text{answer} = \text{ground\_truth}]$
- 格式奖励：正确使用`\boxed{}`
- 中间步骤验证（可选）

代码问题

- 执行结果：$r = \frac{\#\text{passed tests}}{\#\text{total tests}}$
- 编译成功：额外奖励
- 效率奖励（可选）

复合奖励

$$R = \alpha \cdot R_{accuracy} + \beta \cdot R_{format} + \gamma \cdot R_{length}$$

长度惩罚（防止冗长）

$$R_{length} = -\lambda \cdot \max(0, L - L_{threshold})$$

格式奖励

$$R_{format} = \begin{cases} r_+ & \text{格式正确} \\ r_- & \text{格式错误} \end{cases}$$

DeepSeek-R1 概述

2025 年 1 月发布，首个公开的纯 RL 训练推理模型

核心发现

- 纯 RL 可以**涌现**推理能力
- 无需 SFT 冷启动
- 自发学会 Chain-of-Thought
- 出现"Aha moment" 顿悟现象

性能表现

- AIME 2024: **79.8%**（接近 o1）
- MATH-500: **97.3%**
- Codeforces: **2029** rating
- 超越多数闭源模型

重要突破

证明了**大规模 RL 训练**可以让模型自主学习复杂推理，而非仅仅模仿人类解法

DeepSeek-R1 训练流程

四阶段训练流程：

1. **Stage 1: 冷启动 SFT** - 少量长 CoT 数据（数千条）
2. **Stage 2: 推理 RL** - GRPO + 规则奖励
3. **Stage 3: 拒绝采样 SFT** - 用 RL 检查点生成数据
4. **Stage 4: 全场景 RL** - 扩展到更多任务

Base Model → 冷启动 SFT → 推理 RL → 拒绝采样 → 全场景 RL

关键点：

- Stage 2 使用**纯规则奖励**，无需奖励模型
- 奖励 = 准确性 + 格式正确性
- 模型自发产生长链推理

DeepSeek-R1-Zero：纯 RL 实验

无任何 SFT，直接从 Base 模型开始 RL 训练

惊人发现：

1. **自我验证**：模型学会检查自己的答案
2. **反思**：发现错误后重新思考
3. **长链推理**：自动产生详细推导步骤
4. **探索多种方法**：尝试不同解题路径

Aha Moment 示例

"Wait, let me reconsider this problem..."
"Hmm, I made an error in step 3. Let me redo..."
模型自发产生的反思性语言！

局限：可读性差、语言混杂，需要后续 SFT 改进

DeepSeek-R1 奖励设计

极简奖励函数（Stage 2 推理 RL）：

$$R = R_{accuracy} + R_{format}$$

- **准确性奖励**：
  - 数学：答案与标准答案匹配
  - 代码：通过测试用例
  - 使用规则验证，无需神经网络
- **格式奖励**：
  - 要求输出包含 `<think>...</think>` 推理过程
  - 最终答案在 `<answer>...</answer>` 中
  - 惩罚格式不正确的输出

为什么有效?

- 准确性奖励提供**稀疏但准确**的信号
- 模型被迫学习正确的推理过程来提高准确率
- 无需人类标注的推理链

蒸馏到小模型

DeepSeek-R1 的推理能力可以**蒸馏**到小模型

模型	**参数量**	**AIME 2024**	**MATH-500**
DeepSeek-R1	671B	79.8%	97.3%
R1-Distill-Qwen-32B	32B	72.6%	94.3%
R1-Distill-Qwen-14B	14B	69.7%	93.9%
R1-Distill-Qwen-7B	7B	55.5%	92.8%
R1-Distill-Qwen-1.5B	1.5B	28.9%	83.9%
OpenAI o1-mini	-	63.6%	90.0%

蒸馏方法：用 R1 生成的长链推理数据对小模型进行 SFT

OpenAI o1 概述

2024 年 9 月发布，开启" 推理模型" 新范式

官方披露

- 使用强化学习训练
- 产生长链" 思考" 过程
- 思考过程对用户隐藏
- 性能随计算量 scaling

性能亮点

- AIME 2024: **83.3%** (13.4/15)
- Codeforces: **89 percentile**
- GPQA Diamond: **78%**
- PhD 级科学问答

核心理念

"Learning to reason with reinforcement learning"
通过 RL 学习如何进行有效的推理

o1 技术推测

基于公开信息和研究社区分析

可能的训练流程:

1. **过程奖励模型 (PRM)** 训练
   - 参考"Let's Verify Step by Step" 论文
   - 人工标注推理步骤的正确性
   - 训练模型评估每个步骤
2. **大规模 RL 训练**
   - 使用 PRM 作为奖励信号
   - 可能结合结果奖励 (ORM)
   - 大量计算资源
3. **搜索/采样策略**
   - 推理时多次采样
   - 可能使用树搜索
   - Best-of-N 或 MCTS 变体

过程奖励模型 (PRM)

Let's Verify Step by Step (OpenAI, 2023)

核心思想：奖励每个推理步骤，而非只看最终答案

结果奖励 (ORM)

$$R = \begin{cases} +1 & \text{最终答案正确} \\ 0 & \text{最终答案错误} \end{cases}$$

- 信号稀疏
- 难以定位错误
- Credit assignment 问题

过程奖励 (PRM)

$$R = \sum_{t=1}^{T} r_t$$

$$r_t = P(\text{step } t \text{ is correct})$$

- 密集信号
- 精确定位错误步骤
- 需要人工标注

实验结论：PRM 在数学推理上显著优于 ORM

o1 Scaling 特性

推理时计算的 scaling law

关键发现:

- 性能随推理时间/token 数量提升
- " 思考" 越多，答案越准确
- 存在新的 scaling 维度

(推理时间 vs 准确率)

更多思考时间 $\rightarrow$ 更高准确率

类似人类:
难题需要更多思考

两种 scaling:

1. **训练时 scaling**: 更多数据、更大模型
2. **推理时 scaling**: 更多思考、更多搜索

范式转变

从" 更大的模型" 到" 更多的思考"

Constitutional AI (Anthropic)

RLAIF：用 AI 反馈代替人类反馈

训练流程：

1. **Self-Critique**：模型评估自己的回答
2. **Revision**：根据" 宪法" 原则修改回答
3. **RL 训练**：用 AI 评分作为奖励

" 宪法" 示例：

- 选择最有帮助、诚实、无害的回答
- 选择最不具有欺骗性的回答
- 选择最尊重用户自主权的回答

优势：

- 大幅减少人类标注需求
- 可扩展到更多场景
- 原则可编程、可调整

Google 的推理研究

主要工作:

1. **Chain-of-Thought Prompting**
   - 提示模型展示推理步骤
   - Zero-shot CoT: "Let's think step by step"
   - 显著提升数学推理能力

2. **Self-Consistency**
   - 采样多个推理路径
   - 投票选择最一致的答案
   - 简单有效的提升方法

3. **Gemini 系列**
   - 多模态理解
   - 长上下文处理
   - 推理能力持续提升

训练数据构造

高质量问题来源:

数学

- GSM8K (小学数学)
- MATH (竞赛数学)
- AIME/AMC 历年真题
- 合成数学问题
- 网络爬取的数学问答

代码

- LeetCode 题目
- Codeforces 竞赛题
- HumanEval/MBPP
- GitHub 代码库
- 合成编程问题

数据增强:

- 问题改写/变体生成
- 难度渐进
- 多种解法生成

采样与探索策略

RL 训练中的采样:

1. **温度采样**

$$P(token) \propto \exp(\text{logit}/T)$$

   - 训练时 $T > 1$ 增加探索
   - 推理时 $T < 1$ 提高质量

2. **Top-p (Nucleus) 采样**
   - 只考虑累积概率达到 p 的 tokens
   - 平衡多样性和质量

3. **Best-of-N**
   - 生成 N 个候选
   - 用奖励模型选择最佳
   - 推理时常用策略

KL 散度约束

防止策略偏离太远:

$$\mathcal{L} = \mathcal{L}_{RL} - \beta \cdot \mathbb{D}_{KL}[\pi_\theta || \pi_{ref}]$$

为什么需要 KL 约束?

- 防止**奖励黑客**（reward hacking）
- 保持模型的语言能力
- 稳定训练过程

KL 计算（近似）:

$$\mathbb{D}_{KL}[\pi_\theta || \pi_{ref}] \approx \mathbb{E}_{\pi_\theta}\left[\log \frac{\pi_\theta(a|s)}{\pi_{ref}(a|s)}\right]$$

β **选择**:

- 太大：限制学习
- 太小：不稳定/奖励黑客
- 典型值：0.01 - 0.1

训练稳定性技巧

大规模 RL 训练的挑战与解决:

1. **梯度裁剪**

$$g \leftarrow \min\left(1, \frac{c}{||g||}\right) \cdot g$$

2. **学习率调度**
   - Warmup 阶段
   - Cosine decay
   - 比 SFT 更小的学习率
3. **奖励归一化**

$$r_{norm} = \frac{r - \mu_r}{\sigma_r}$$

4. **多次 PPO epochs**
   - 每批数据更新多次
   - 但需要控制更新幅度

DeepSeek 报告的失败尝试

重要的负面结果:

1. **过程奖励模型 (PRM) 效果不佳**
   - 训练 PRM 本身很困难
   - 需要大量步骤级标注
   - 泛化能力有限
   - 最终放弃，使用结果奖励
2. **Monte Carlo Tree Search (MCTS) 未成功**
   - 搜索空间太大
   - 价值估计不准确
   - 计算成本过高
3. **直接从 Base 模型 RL 的问题**
   - 可读性差
   - 语言混杂
   - 需要少量 SFT 冷启动

常见问题与解决

奖励黑客

- 问题：找到漏洞获取高奖励
- 解决：KL 约束、多样化奖励

模式崩塌

- 问题：输出变得单一
- 解决：熵正则化、温度控制

训练不稳定

- 问题：损失震荡、发散
- 解决：小学习率、梯度裁剪

遗忘问题

- 问题：失去原有能力
- 解决：混合训练数据

长度控制

- 问题：输出过长或过短
- 解决：长度惩罚/奖励

格式问题

- 问题：不遵循指定格式
- 解决：格式奖励、SFT 预训练

研究前沿

值得关注的方向:

1. **更高效的 RL 算法**
   - 减少样本复杂度
   - 更稳定的训练
   - 更低的计算成本
2. **更好的奖励设计**
   - 自动发现奖励函数
   - 多目标优化
   - 可解释的奖励
3. **推理时 scaling**
   - 更高效的搜索算法
   - 自适应计算量
   - 推理成本优化
4. **多模态推理**
   - 视觉推理
   - 跨模态 reasoning

开放问题

尚未解决的挑战:

- **泛化性**：如何让推理能力迁移到新领域?
- **可解释性**：模型真的在" 推理" 还是在模式匹配?
- **长程规划**：如何进行更长 horizon 的推理?
- **世界模型**：是否需要显式的世界知识?
- **安全性**：推理模型可能更擅长欺骗?
- **效率**：如何降低推理成本?

核心要点回顾

1. **RL 是提升 LLM 推理能力的有效方法**
   - 无需大量人类标注
   - 可以超越人类示范
2. **GRPO 简化了训练流程**
   - 无需 Critic 网络
   - 组内相对比较
3. **可验证奖励是关键**
   - 数学：答案匹配
   - 代码：执行验证
4. **推理时 Scaling 开辟新维度**
   - 更多思考 = 更好结果
   - Test-time compute

参考资料

- DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via RL (2025)
- DeepSeekMath: Pushing the Limits of Mathematical Reasoning (2024)
- Let's Verify Step by Step (OpenAI, 2023)
- Constitutional AI: Harmlessness from AI Feedback (Anthropic, 2022)
- Training Language Models to Follow Instructions with Human Feedback (OpenAI, 2022)
- OpenAI o1 System Card (2024)
- Scaling Laws for Reward Model Overoptimization (2022)
- Chain-of-Thought Prompting Elicits Reasoning in LLMs (Google, 2022)

谢谢！欢迎提问讨论