大模型原理与应用

10：RL 微调路线一

用 GRPO 做推理型后训练

看什么

- 为什么推理模型训练开始引入 RL post-training
- GRPO 相比 PPO 想简化什么
- 一条最小可运行的 GRPO 流程包含哪些组件

主线

1. 先把“会生成”变成“会推理”
2. 再用 GRPO 替代更重的 PPO 训练形态
3. 最后把数据、格式、奖励和训练器接起来

为什么这里开始讲 RL 微调

到第 3 课前面为止，我们看到的主要还是：

- supervised fine-tuning
- LoRA
- 数据格式化

但推理模型的关键问题是：

- 只靠模仿数据，未必足够学出稳定的长链条推理

所以后训练开始引入 RL。

GRPO 想解决什么问题

这份材料把 GRPO 放在 DeepSeekMath、DeepSeek-R1 的脉络里理解：

- 它是一种 RL post-training 技术
- 目标是增强复杂任务上的推理能力
- 特别适合数学题、多步推理、test-time compute 扩展

这里的关键词不是“更会聊天”， 而是“更会一步一步做题”。

GRPO 和 PPO 的直观差别

材料强调的一点是：

- GRPO 去掉了 value model

这意味着：

- 训练流程更简化
- 资源压力更小
- 更适合在现有大模型后训练里快速试验

课堂上最应该先记住的是这个结构性变化。

一条基础版 GRPO 流程长什么样

这份 notebook 的最小闭环是：

1. 选一个轻量 instruct 模型做 baseline
2. 准备数学推理数据
3. 把样本整理成带 system prompt 的对话格式
4. 定义 reward
5. 用 `GRPOTrainer` 跑后训练

这说明 RL 微调并不是和前面的 Hugging Face 流程断开， 而是在它之上再加一层奖励优化。

数据为什么要改成“推理 + 答案”双段式

材料用了类似 DeepSeek-R1 的格式：

- `<think> ... </think>`
- `<answer> ... </answer>`

这不是装饰性标签， 而是在明确告诉模型：

- 哪一段是推理过程
- 哪一段是最终答案

所以 RL 在这里奖励的，不只是答对， 还包括按指定结构进行推理。

为什么这里仍然用 LoRA

即便进入 RL 后训练， 工程约束并没有消失：

- 显存仍然有限
- 大模型仍然不适合全参数更新

所以这一条路线仍然延续前面的判断：

- 量化负责装载
- LoRA 负责参数效率
- GRPO 负责奖励驱动的能力塑形

这条路线最值得观察什么

- baseline 模型为什么选轻量版本
- 数据如何改造成 conversation prompt
- reward 如何定义“答得好”
- RL 后训练和 SFT 到底是衔接关系还是替代关系

这四个问题比背库函数名更重要。

一个总判断

基础版 GRPO 材料真正想展示的是：

- 推理模型并不是凭空出现的
- 它是在已有 instruct 模型之上，用 RL 后训练继续塑形出来的

所以它是“第二阶段强化”， 不是“重新训练一个模型”。

这部分你该带走什么

- GRPO 是推理型 post-training 的代表路线之一
- 它把大模型训练从“监督模仿”推向“奖励优化”
- 格式设计、reward 设计、训练器设计必须一起看

10 收束

- 对应材料：GRPO in TRL
- 对应 notebook：`13/2-prog/fine_tuning_llm_grpo_trl.ipynb`
- 建议用途：作为理解 DeepSeek-R1 类训练思路的第一步