大模型原理与应用

11：RL 微调路线二

多奖励 GRPO 与数学推理训练

看什么

- 为什么单一 reward 往往不够
- 多奖励训练怎样同时约束格式、答案与行为
- 高级版 GRPO 为什么更像“系统设计”

主线

1. 先承认“答对”不是唯一目标
2. 再把 reward 拆成多个可解释维度
3. 最后用多奖励系统塑造结构化推理输出

为什么从基础版走到高级版

基础版 GRPO 已经能跑起来。
但真实问题很快出现：

- 只奖励最终正确答案，模型可能格式混乱
- 只奖励格式正确，模型可能内容空洞

所以更成熟的训练会继续问：

- 什么叫“好答案”

多奖励系统在约束什么

这份材料把 reward 分成 4 类：

- format compliance
- approximate matching
- answer correctness
- number extraction

这意味着训练目标不再是单点打分， 而是把“结构”和“内容”一起纳入优化。

为什么这很像工程而不只是算法

你需要同时决定：

- 输出格式长什么样
- 如何识别 reasoning 段
- 如何抽取最终数值答案
- 哪些错误值得部分奖励

所以高级版 GRPO 的重点，已经不只是“会调用 `GRPOTrainer`”，而是“会设计 reward system”。

结构化输出为什么重要

这份 notebook 用了明确的标签：

- `<start_working_out> ... <end_working_out>`
- `<SOLUTION> ... </SOLUTION>`

这样做的好处是：

- 训练时更容易检查格式
- 推理时更容易解析结果
- 奖励函数更容易自动评估

这说明结构化输出本身就是训练目标的一部分。

这条路线仍然建立在参数效率之上

虽然它更“高级”，但底层工程判断没有变：

- 4-bit quantization
- LoRA
- 小到中等规模 instruct 模型

所以 RL 微调并没有脱离第 3 课前半段内容， 它只是把那些能力继续向“奖励驱动优化”推进。

多奖励比单奖励难在哪里

- 奖励之间可能互相冲突
- 奖励权重影响训练方向
- 自动解析答案本身可能出错
- 局部 reward 容易诱导模型投机

因此，多奖励训练的难点不是“多写几个函数”， 而是：

- 你是否真的知道自己想奖励什么

从这份材料该如何理解推理模型

推理模型并不只是“会输出更长的 thought”。 更准确地说，它往往是被系统性地训练成：

- 按特定格式思考
- 按特定格式作答
- 在多个 reward 下同时满足结构与正确性

这比普通 instruction tuning 更进一步。

一个总判断

高级版 GRPO 代表的不是一个单独技巧， 而是一种训练观：

- 模型行为可以被拆成多个维度
- 不同维度可以分别奖励
- 最终通过组合奖励塑造出更稳定的推理模式

这部分你该带走什么

- 多奖励训练是在回答“什么样的推理过程才算好”
- reward design 已经是模型设计的一部分
- 结构化输出、自动评测、参数效率微调在这里完全连到一起了

11 收束

- 对应材料：Advanced GRPO with Multi-Reward Training
- 对应 notebook：`13/2-prog/trl_grpo_reasoning_advanced_reward.ipynb`
- 建议用途：作为理解 R1 类推理模型训练细节的进阶材料