自我介绍

有关工作：PROGRESSIVE CURRICULUM SYNTHESIS MAKES LLMS BETTER MATHEMATICAL REASONERS
工作实习经历：美团基础研发部 M17 大模型算法实习生
腾讯机器学习平台部混元大模型 SFT 组算法实习生
2023 ATEC 大模型新闻检测赛道亚军&网络安全赛道冠军
2023 第四届 SEED 大赛医疗卫生赛道冠军
2024 综艺燃烧吧天才程序员 4 线下赛总冠军
2024 ICML Automated Math Reasoning Third Place Winer
2024 芒果 TV 大模型逻辑推理赛道冠军
2025 第三届电磁大数据非凡挑战赛亚军
2025 第三届世界科学智能大赛合成生物赛道亚军
2025 天翼云熙壤杯数学推理赛道冠军&线下赛亚军
2025 ICCV Visual Question Answering with Spatial Awareness First PLACE WINNER

Openseek-初赛技术方案

本人初赛总共就交了两次，主要是为了进复赛也没怎么做，第一次提交是选取提供数据集中的高质量合成 SFT 的数据从 36.6->36.7，第二次提交是发现默认学习率过大，采用余弦退火再用同一批数据进行训练，提升到 36.99 能进入到复赛节省时间就没认真做了。

Openseek-复赛技术方案

1. 踩坑

做 Math reasoning 原本是本人很擅长的题目，但是这个模型在做强化的时候就变得很奇怪。首先采用常规的 DAPO，因为 DAPO 上了一些 trick 所以在训练强化学习的时候会稳定很多，但是不出意外的 200steps 之后就开始坍缩，长度锐减模型 reward 降低，模型不输出来降低 penalty。无论是使用 boxed{}格式奖励和最终的 acc 奖励混合还是单使用 acc 奖励效果都不会很好。

然后接着尝试了 GSPO，因为对于 MoE 架构的强化学习训练这个方法效果是好一些，至少不会是路由坍缩，但是发现效果还不行，这时候就有点诡异了，因为觉得是路由坍缩导致的训练效果不好，但是还是会长度坍缩。

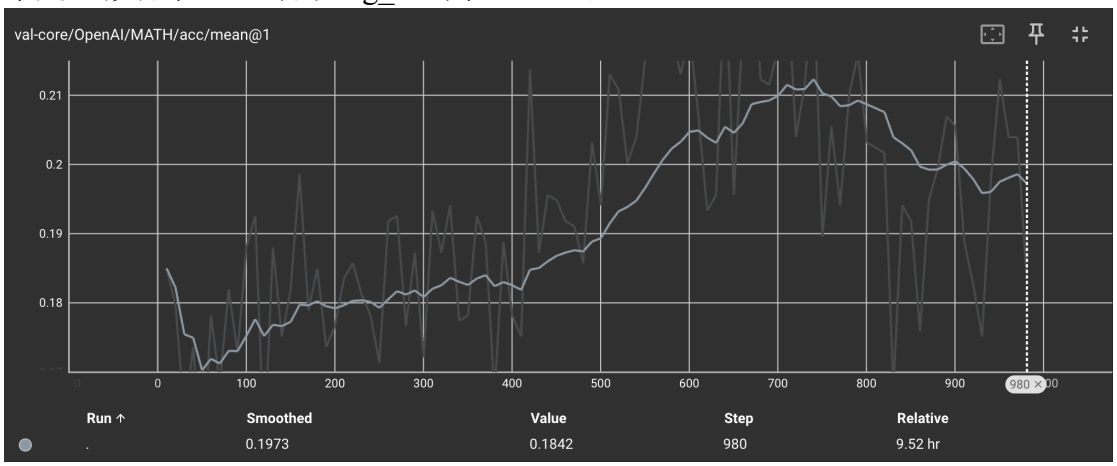
进而尝试最原始的 grpo 并将学习率降低但是效果还是不好，怀疑是模型能力不行，所以将 rollout 的 number 变大，效果有一定缓解，但是还是会导致坍缩的效果，所以在这里折腾了很多天。

接着怀疑是模型探索能力弱的前提下，rule reward 无法给很强烈的监督信号，进而转向 PPO 进行训练，并且使用 Qwen2.5-Math-7B-PRM 当作 reward。尝试训

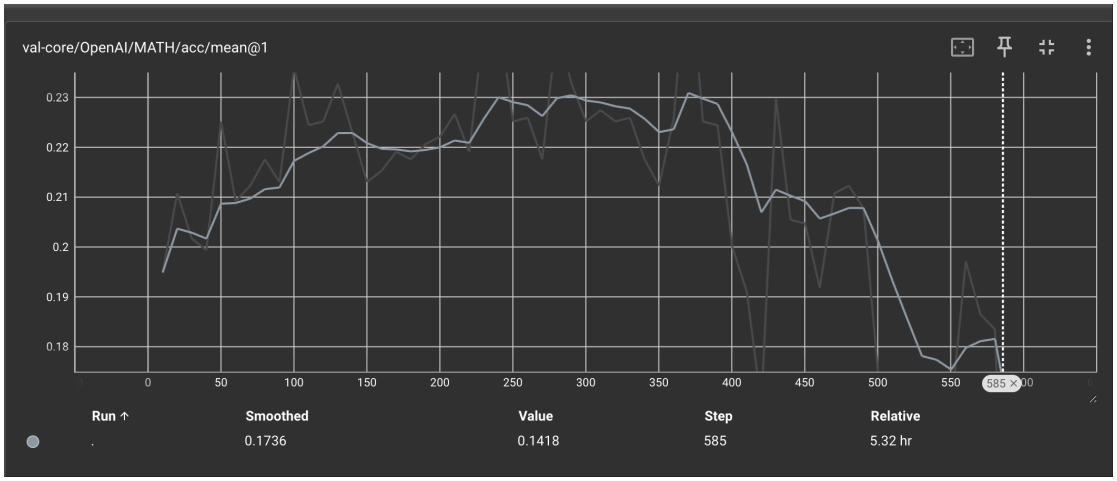
练后发现模型长度不会坍缩了，但是会学着去 hack PRM 了。模型会重复输出 boxed 和 Answer 这种结果去 hack PRM 拿到高分，因为自己探索拿不到高分的情况下，学着去 hack PRM，能拿到高分，就不会自己想去探索了。然后尝试修改比例调整 PRM 和 answer acc 为 2:8，才能够正常推理。此时能达到 avg_acc 大约 9 分左右的成绩后会坍缩。开始尝试加强 KL loss 约束以及降低 response_length 为 1024，不让过长重复输出结果影响过程，从而初步稳定训练。

2. 技术方案流程

Step1:首先采用 GSM8K 训练集和 MATH 训练集以及 BIG_MATH 等比例采样的 50K 训练集混合的训练集初步提升模型的数学能力，尝试过业界常用的 DAPO 的训练集但是太难了没有特别多的奖励信号，在 batchsize512 和 resposne_length 为 1024 的设置下 PPO 训练 15epochs，但是在 700steps 之后效果有所下降，大概率是过拟合了。此时的 avg_acc 为 11.7 左右



Step2:接着采用 skill 和 knowledge base 类似我论文里面的方法基于 GSM8K 和 MATH 的训练集合合成约 50K 的数据，并且使用 Rejection Sampling 利用 GPT5 进行过滤，并且混合之前上一轮 steps1 的 rollout 里同一个问题成功率高于 20% 的问题大约 11k 进行 in-domain 的训练。随着训练发现模型的指令遵循能力下降具体表现为不输出 boxed{}格式，PRM 打分下降，以及 validation 的 acc 下降，所以在 300steps 的时候保存 checkpoint 进行 SFT。此时的 avg_acc 为 12.9 左右



Step3:其实对于效果的提升 LongCoT 对于蒸馏模型的能力提升是最明显的但是为了后续 PPO 的探索能力，以及为了防止模型参数量突然改变造成的灾难性遗忘，采用冷启动数据集中的 MetaMath 和 GSM8K、MATH、合成数据训练 3 个 epoch。大头还是冷启动中的数据防止分布发生较大的偏移。此时的 avg_acc 为 13.5 左右。在此插一嘴，中途尝试过修改 verl，边进行 PPO 和 SFT，例如训练一个 step 的 PPO 再训练一个 step 的 SFT，相当于 loss 加权，但是会发生模型训练到 200steps 之后长度暴涨从而重复输出的情况，从日志分析怀疑是 actor 和 critic 不共享权重导致 actor 被 SFT 更新之后 critic 没有更新，还没来得及解耦尝试因为又要修改很多源码，因为比赛时间来不及了。

Step4:此时模型的基础数学能力上来之后，参考 DeepscaleR 论文的方式采用 response_length 增加到 2048 并且用相同的数据集进行 PPO 训练 1200steps，但是在 600steps 之后还是发生了坍缩，最终采用 450 的 checkpoint 作为最终的提交结果。此时的 avg_acc 为 13.9 左右。主要提升 MATH 的能力，事实上确实这一步在 GSM8K 的效果轻微变化，MATH 提升了 2pp。

3. 总结

   这个模型太诡异了，不过也能理解业界现在很多方法都是基于 Qwen 去做强化，换成其他模型基座能力不强的前提下，强化确实很难做，因为时间原因也没有探索很多其他方案。总结来说还是得 KL 强约束+response_length 不能过长造成异常的 rollout 轨迹影响训练的稳定性。但是有时候该崩还是会崩。。

运行代码

1. 环境

   Verl050/requirements_sglang.txt
2. 运行代码顺序

   examples/ppo_trainer 目录下都有 shell
3. Verl 修改

   爆改了 verl 了，修改总结在 md 文件下
4. 模型路径

https://www.modelscope.cn/models/Echoch/openseek-ppo/files
5. 数据路径

https://www.modelscope.cn/datasets/Echoch/openseek-dataset/files