超越杯算法赛道--决赛技术报告

参赛单位：北京城建智控科技股份有限公司
团队名称：UCI001
提交账号：liximeng0824

一、训练数据集

modelscope 搜索 math + RL，选择 Big-Math-RL-Verified-Processed 数据集

1、处理路径：

Big-Math-》Big-Math-RL-Verified-》Big-Math-RL-Verified-Processed

2、选择原因：

数据集包含一个类似通过难度的参数（llama8b_solve_rate，Llama-3.1-8B 模型在 64 次尝试中成功的百分比），原计划根据成功率进行从易到难得课程学习，但最后由于时间不足没有进一步尝试。

prompt (Value)	solution (Value)	source (Value)	domain (Value)	llama8b_solve_rate (Value)
The units digits of powers of 7 follow a repeating pattern: 7, 9, 3, 1. This pattern repeats every 4 powers. What is the units digit of 7^62? Express your answer as a single digit.	9	big_math	["Mathematics -> Number Theory -> Other"]	0.984375

二、数据预处理

1、子集选择：

（1）根据基线模型的初始能力情况，提取数据集中的 Orca-Math、cn_k12、GSM8k（训练）三个子集，其难度均为中小学数学水平。

（2）上述三个子集为独立数据集，与评测集无关；同时团队进行了全量样本查询，核实两者没有任何交集。

2、答案类型筛选：

团队考虑优先对稀疏数值类型的答案，代码实现时限定只提取答案可转为 float 数值类型的题目。

3、Prompt 模板：

最初采用 gsm8k 的 prompt 模板（“### + 答案”）构造训练数据，发现效果不好；之后，团队分析认为 SFT 基线模型应对默认 openseek_sft 模板（“\\boxed{{}}”）已具备任务处理能力，延用该策略更稳妥（评测 prompt 不变）。代码见 gsm8k_lxm2_newprompt_trainval.py。

4、**train/val 划分：**

为了在 verl 的 reward 打分时能够反应真实的数学能力提升情况，对数据集进行 train/val 划分，其中验证集比例设 5%。处理后：

big-math-rl-verified-processed_orca_cnk12_gsm8k_newprompt_2_train.parquet

big-math-rl-verified-processed_orca_cnk12_gsm8k_newprompt_2_val.parquet

5、训练数据预处理（最终）： gsm8k_lxm2_newprompt_trainval.py

三、Verl 奖励函数设计

1、修改模版：

verl/verl/utils/reward_score/gsm8k.py

2、**格式解析：**

参考评测工具中 parse.py 对“\\boxed{{}}”的解析方式

3、奖励分布：

boxed 格式且答案正确得 1.0，否则如果 boxed 格式正确得 0.2
reward_score/gsm8k.py

四、训练策略

实验初期尝试过 kl_in_loss/ kl_in_reward 两种 GRPO 策略进行训练，但无论超参如何调整验证 reward 均在 0.4 以下，同时 checkpoint 的评测结果也与基线模 SFT 模型近似，无法有效提升数学推理能力。

团队分析认为 kl 对于基线模型学习起到限制作用，在答案仅按最终正确与否的稀疏设定下，模型往往需要显著偏离 SFT 分布（更长推理、更大胆的搜索）才会提升正确率；kl_loss 会“拉回”SFT 分布，和有效更新方向冲突，在长序列数学题上更明显。随后，我们尝试关闭 kl 仅采用 GRPO 的组内相对优势估计进行实验训练，指标能够明显提升；同时也从使用 FlagScale 框架调整为直接使用 verl。

五、代码修改记录

1、原始代码 commit

```
commit 8b33abd84f360473f05e5a750aef36e974340cce (HEAD)
Author: Blue Space <57280232+ETOgaosion@users.noreply.github.com>
Date:   Mon Jun 30 15:27:02 2025 +0800

    [megatron] feat: add megatron memory log (#2272)

    ### What does this PR do?

    Log memory footprints in wandb during running like FSDP does.

    ### Checklist Before Starting

    - [ ] Search for similar PRs. Paste at least one query link here: ...
    - [ ] Format the PR title as `[{modules}] {type}: {description}` (This
    will be checked by the CI)
    - `{modules}` include `fsdp`, `megatron`, `sglang`, `vllm`, `rollout`,
    `trainer`, `ci`, `training_utils`, `recipe`, `hardware`, `deployment`,
    `ray`, `worker`, `single_controller`, `misc`, `perf`, `model`, `algo`,
    `env`, `tool`, `ckpt`, `doc`, `data`
    - If this PR involves multiple modules, separate them with `,` like
    `[megatron, fsdp, doc]`
      - `{type}` is in `feat`, `fix`, `refactor`, `chore`, `test`
    - If this PR breaks any API (CLI arguments, config, function signature,
    etc.), add `[BREAKING]` to the beginning of the title.
      - Example: `[BREAKING][fsdp, megatron] feat: dynamic batching`

    ### Test

    > For changes that can not be tested by CI (e.g., algorithm
```

2、修改 status

```
HEAD detached at 8b33abd8
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   verl/trainer/config/ppo_trainer.yaml
        modified:   verl/trainer/ppo/ray_trainer.py
        modified:   verl/utils/reward_score/gsm8k.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        examples/data_preprocess/gsm8k_lxm.py
        examples/data_preprocess/gsm8k_lxm2.py
        examples/data_preprocess/gsm8k_lxm2_newprompt.py
        examples/data_preprocess/gsm8k_lxm2_newprompt_trainval.py
        examples/data_preprocess/gsm8k_lxm2_newprompt_trainval_omni-math.py
        examples/data_preprocess/gsm8k_lxm_difficulty.py

no changes added to commit (use "git add" and/or "git commit -a")
```