PR #6849 完整报告

verl-project/verl

[data, rollout, worker] feat: add Open-R1 multimodal and TinyLLaVA-Video-R1 preprocessing and training scripts

合并时间：2026-07-01 15:10

原文链接：http://prhub.com.cn/verl-project/verl/pull/6849

执行摘要

- 一句话：新增 Open-R1 多模态和 TinyLLaVA-Video 数据预处理及训练脚本
- 推荐动作：该 PR 提供了完整的多模态数据集集成范例，值得阅读和参考。特别是图像预处理中保留原始字节的方式，以及训练脚本中设备自动检测的写法。但注意奖励函数需用户自行实现，直接运行示例脚本前需创建对应的 reward score 文件。

功能与动机

为了在 verl 中支持多模态 GRPO 训练，需要提供数据预处理和训练配置示例。该 PR 基于社区需求，使得用户能够直接使用 Open-R1 多模态数据和 TinyLLaVA-Video-R1 数据集进行训练，无需自行编写数据转换逻辑。同时修复了多模态训练中出现的 position_ids 损坏问题。

实现拆解

1. 数据预处理脚本：新增 *examples/data_preprocess/openr1mm.py* 和 *examples/data_preprocess/tinyllava_video_r1.py*，分别处理图像和视频数据集。 - 图像脚本使用 *datasets* 库加载数据，通过 *cast_column(decode=False)* 保留原始字节，避免有损重新编码。 - 视频脚本从 JSONL 解析问题、答案和视频路径，构建标准的 think/answer 格式 prompt。 - 两个脚本最终输出统一的 verl parquet 格式。
2. 训练启动脚本：新增 *examples/grpo_trainer/run_qwen3_5_2b_openr1_fsdp.sh* 和 *examples/grpo_trainer/run_qwen3_5_2b_video_fsdp.sh*，基于 FSDP 的 GRPO 训练配置。 - 脚本自动检测 GPU 或 NPU 设备，设置数据路径、模型路径、奖励函数路径等参数。 - 奖励函数需用户自行提供（位于 verl/utils/reward_score/*.py）。
3. 引擎层修复：修改 verl/workers/engine_workers.py，在 train_mini_batch 循环起始位置添加 maybe_fix_3d_position_ids(mini_batch_td) 调用，修正多模态数据中可能错乱的 3D position_ids，防止 _ragged_idx 损坏。

关键文件：

- *examples/data_preprocess/tinyllava_video_r1.py*（模块 数据预处理；类别 source；类型 dependency-wiring；符号 build_prompt_text, make_map_fn, process_fn, load_jsonl）：新增的视频数据集预处理脚本核心文件，实现了从 JSONL 到 parquet 的完整转换流程，包括视频路径解析和 prompt 构建。

- *examples/data_preprocess/openr1mm.py*（模块 数据预处理；类别 source；类型 dependency-wiring；符号 make_map_fn, process_fn）：新增的图像数据集预处理脚本核心文件，演示了如何保留图像原始字节并构建 parquet 数据集。
- *examples/grpo_trainer/run_qwen3_5_2b_video_fsdp.sh*（模块 训练脚本；类别 other；类型 configuration）：视频 GRPO 训练启动脚本，展示了完整的数据、模型、奖励配置。
- *examples/grpo_trainer/run_qwen3_5_2b_openr1_fsdp.sh*（模块 训练脚本；类别 other；类型 configuration）：图像 GRPO 训练启动脚本，展示了完整的数据、模型、奖励配置。
- verl/workers/engine_workers.py（模块 引擎；类别 source；类型 core-logic；符号 maybe_fix_3d_position_ids）：核心引擎修改，修复了多模态训练中的 position_ids 损坏问题。

关键符号：build_prompt_text, make_map_fn, process_fn, load_jsonl, maybe_fix_3d_position_ids

关键源码片段

examples/data_preprocess/tinyllava_video_r1.py

新增的视频数据集预处理脚本核心文件，实现了从 JSONL 到 parquet 的完整转换流程，包括视频路径解析和 prompt 构建。

```
def make_map_fn(
    data_source: str,
    video_dir: str,
    split: str,
    video_fps: Optional[float] = None,
    video_max_frames: Optional[int] = None,
):
    """工厂函数，遵循 verl 的标准闭包模式。"""

    def process_fn(example, idx):
        problem = example["problem"]
        solution = example["solution"] # 答案已为 "<answer>X</answer>" 格式

        # 解析视频绝对路径：去掉 "./" 前缀并与 video_dir 拼接
        video_rel = example["video_filename"].lstrip("./")
        video_path = os.path.join(video_dir, video_rel)
        if not os.path.exists(video_path):
            print(f"[WARN] 视频文件不存在: {video_path}", file=sys.stderr)

        prompt_content = build_prompt_text(problem) # 生成 "<video>\n{problem}

{INSTRUCTION}"

        # 视频采样参数：默认 fps=1, max_frames=32
        video_entry = {"video": video_path}
        if video_fps is not None:
            video_entry["fps"] = video_fps
```

```
        if video_max_frames is not None:
            video_entry["max_frames"] = video_max_frames

        return {
            "data_source": data_source,
            "prompt": [{"role": "user", "content": prompt_content}],
            "videos": [video_entry],
            "ability": "video_qa",
            "reward_model": {"style": "rule", "ground_truth": solution},
            "extra_info": {
                "split": split,
                "index": idx,
                "question": problem,
                "answer": solution,
                "video_path": video_path,
            },
        }

    return process_fn
```

examples/data_preprocess/openr1mm.py

新增的图像数据集预处理脚本核心文件，演示了如何保留图像原始字节并构建parquet数据集。

```
def make_map_fn(split):
    def process_fn(example, idx):
        problem = example.pop("problem")
        solution = example.pop("solution")
        img = example.pop("image")

        # 构建 prompt ： 将 <image> 占位符嵌入问题前
        prompt_content = f"<image>\n{problem}

{instruction}"

        # 保持图像为原始字节形式，避免 Qwen VL Processor 再次编解码
        if isinstance(img, dict) and "bytes" in img:
            image_data = img
        elif isinstance(img, bytes):
            image_data = {"bytes": img}
        else:
            # 若已经为 PIL.Image，则直接使用（但很少执行到此）
            image_data = img

        return {
            "data_source": data_source,
            "prompt": [{"role": "user", "content": prompt_content}],
            "images": [image_data],
            "ability": "math",
            "reward_model": {"style": "rule", "ground_truth": solution},
```

```
            "extra_info": {
                "split": split,
                "index": idx,
                "question": problem,
                "answer": solution,
            },
        }
    return process_fn

# 阻止 datasets 自动解码图像
full_dataset = full_dataset.cast_column("image", datasets.Image(decode=False))
```

评论区精华

1. 奖励函数文件缺失：gemini-code-assist 指出训练脚本引用的奖励函数文件未包含在 PR 中。作者回应称，按照 verl 最新策略（wuxibin89 在 PR#6793 的指导），框架不再内置维护每个数据集的奖励函数，用户需自行提供。该讨论已关闭。
2. 图像自动解码问题：gemini-code-assist 指出 openr1mm.py 中 datasets.load_dataset 默认会解码图像，导致后续的 isinstance 检查失效。作者在最终代码中添加了 cast_column('image', datasets.Image(decode=False)) 以保留原始字节。
3. 列选择问题：同样在 openr1mm.py 中，gemini-code-assist 指出 Dataset.map 后未移除原始列，会导致已解码的图像数据写入 Parquet 文件。作者在 final 版本中添加了 select_columns(...) 仅保留必要列。

- 奖励函数文件缺失 (correctness): 作者回应称奖励函数文件有意不包含，遵循最新策略（wuxibin89 在 PR#6793 的指导），用户需自行提供。PR 审核通过。
- 图像自动解码问题 (correctness): 作者在最终代码中添加了 cast_column('image', datasets.Image(decode=False))，已修复。
- 未选择列导致图像数据残留 (performance): 作者在最终代码中添加了 select_columns(...) 仅保留必要列，已修复。

风险与影响

- 风险：
  - 奖励函数依赖：训练脚本引用 custom_reward_function.path 指向 verl/utils/reward_score/ 下的文件，但该目录未包含在 PR 中。用户直接运行脚本会因找不到模块而报错。需提醒用户自行编写奖励函数。
  - 图像解码性能：若未正确处理图像列（如未设置 decode=False），可能导致 Parquet 文件体积膨胀或损失图像质量。当前脚本已修复此问题。
  - position_ids 回退：maybe_fix_3d_position_ids 被无条件调用，若模型不支持 3D position_ids 可能引入额外开销或错误，但该函数内部应具有安全判断。
  - 配置严格性：训练脚本使用特定参数（如 data.image_patch_size=16），若用户数据集不匹配可能导致训练失败。
- 影响：

- 用户：新数据集用户可以直接使用提供的预处理脚本和训练脚本，快速启动多模态 GRPO 训练。
- 系统：engine_workers.py 的修改影响所有调用 train_mini_batch 的流程，但已通过 GPU 和 NPU 验证。
- 团队：该 PR 明确了奖励函数维护的策略（不内置，由用户提供），为未来数据集集成提供了范例。
- 风险标记：奖励函数脚本缺失，图像解码导致文件膨胀，position_ids 修复影响范围，训练脚本依赖路径正确

关联脉络

- 暂无明显关联 PR