

PR #6227 完整报告

verl-project/verl

[trainer] fix: dump all outputs in validation in main_ppo_sync

合并时间: 2026-05-06 15:53

原文链接: <http://prhub.com.cn/verl-project/verl/pull/6227>

执行摘要

- 一句话: 修复验证阶段多输出 dump 缺失问题
- 推荐动作: 建议在排序前增加防御性处理, 例如对 parts[1] 和 parts[2] 进行格式验证, 或使用 try-except 避免异常崩溃。该 PR 的设计思路 (分别获取所有输出和最终元数据) 值得学习, 但需要在健壮性上做补强。

功能与动机

在 `_validate` 中, 当 `validation_data_dir` 设置后, dump 只包含每个 session 的最终输出。对于多轮或多输出 rollout, 所有中间和替代输出被静默丢弃。此 PR 修复了 `_validate`, 使其获取所有 key 的 prompts/responses, 并在 dump 中包含每个输出, 同时将每个条目映射回其 session 的最终分数 /ground-truth。

实现拆解

1. 新增 dump_all_inputs、dump_all_outputs、dump_all_keys 三个列表, 用于跨 batch 累积所有输入、输出和 key。
2. 新增 session_to_sample_idx 字典, 将每个 session key 映射到其在 per-session 列表中的位置, 以便在写入 dump 时正确查找分数 /ground-truth。
3. 修改 session_max 构建逻辑: 不再直接取 final_indices = sorted(pos for _, pos in session_max.values()), 而是构建 sorted_sessions 保留 session 顺序, 并计算 base_offset 更新 session_to_sample_idx。
4. 新增独立的 kv_batch_get 调用获取所有 key 的 prompts 和 responses (不再通过最终 key 的 kv_batch_get 获取), 然后解码为文本。
5. 从最终 key 的 kv_batch_get 的 select_fields 中移除 prompts/responses, 保留 uid、rm_scores、num_turns、reward_model、data_source、extra_fields。
6. 在 _dump_generations 之后, 将 dump_all_inputs、dump_all_outputs、dump_all_keys 按 (uid, session_id, index) 排序后写入 dump 文件。

关键文件:

- verl/trainer/main_ppo_sync.py (模块 训练器; 类别 source; 类型 core-logic; 符号 _validate): 这是 PR 唯一修改的文件, 核心变更在 _validate 方法中, 包括新增数据收集列表、变更 kv_batch_get 调用逻辑以及排序和 dump 逻辑。

关键符号: `_validate`

关键源码片段

`verl/trainer/main_ppo_sync.py`

这是 PR 唯一修改的文件，核心变更在 `_validate` 方法中，包括新增数据收集列表、变更 `kv_batch_get` 调用逻辑以及排序和 `dump` 逻辑。

```
# 新增的列表和映射，用于跨 batch 累积所有输出
dump_all_inputs: list[str] = []
dump_all_outputs: list[str] = []
dump_all_keys: list[str] = []
session_to_sample_idx: dict[str, int] = {}

for batch_dict in self.val_dataloader:
    # ... 生成序列和采样 batch 的代码不变 ...

    # 4. 构建 session_max 时保留 session 顺序
    sorted_sessions = sorted(session_max.items(), key=lambda x: x[1][1])
    final_indices = [pos for _, (_, pos) in sorted_sessions]
    final_keys = [batch.keys[i] for i in final_indices]
    base_offset = len(sample_scores)
    # 记录每个 session_key 在 per-session 列表中的索引
    session_to_sample_idx.update(
        {session_key: base_offset + j for j, (session_key, _) in enumerate(sorted_sessions)}
    )

    # 单独获取所有 key 的 prompts 和 responses（不限于 final keys）
    text_data = tq.kv_batch_get(
        keys=batch.keys, partition_id=batch.partition_id, select_fields=["prompts",
        "responses"]
    )
    text_data["prompts"] = text_data["prompts"].to_padded_tensor(padding=self.tokenizer.
    pad_token_id)
    text_data["responses"] = text_data["responses"].to_padded_tensor(padding=self.
    tokenizer.pad_token_id)
    all_inputs = [self.tokenizer.decode(ids, skip_special_tokens=True) for ids in text_
    data["prompts"]]
    all_outputs = [self.tokenizer.decode(ids, skip_special_tokens=True) for ids in text_
    data["responses"]]

    # 最终 key 的 kv_batch_get 不再包含 prompts/responses
    fields = ["uid", "rm_scores", "num_turns", "reward_model", "data_source", "extra_fields"]
    data = tq.kv_batch_get(keys=final_keys, partition_id=batch.partition_id, select_fields=
    fields)

    sample_uids.extend(data.pop("uid").tolist())
    # 使用 all_outputs 和 all_inputs 通过 final_indices 索引
    sample_outputs.extend(all_outputs[i] for i in final_indices)
```

```

sample_inputs.extend(all_inputs[i] for i in final_indices)
# ... 其余数据收集不变 ...

# 在 dump 之前，累积所有输出和 key
# 排序逻辑：假设 key 格式为 {uid}_{session_id}_{index}
# 注意：gemini-code-assist 指出需要防御性处理避免 ValueError
sorted_indices = sorted(
    range(len(dump_all_keys)),
    key=lambda i: (
        dump_all_keys[i].rsplit("_", 2)[0], # uid
        int(dump_all_keys[i].rsplit("_", 2)[1]), # session_id
        int(dump_all_keys[i].rsplit("_", 2)[2]) # index
    )
)
# ... 写文件 ...

```

评论区精华

gemini-code-assist[bot] 提出了一个高优先级问题：排序逻辑假设 `parts[1]` (`session_id`) 和 `parts[2]` (`index`) 总是可以转换为整数，如果 key 格式异常会导致 `ValueError`，建议使用 `try-except` 或更健壮的解析方法。该评论未得到回复，但 PR 已被 wuxibin89 批准合并。

- 排序逻辑中 `int` 转换可能抛出 `ValueError (correctness)`：未在 PR 中得到答复或修复，但 PR 已被批准合并。该风险仍存在于已合并代码中。

风险与影响

- 风险：核心风险在于排序逻辑对 key 格式的假设：如果 key 格式不符合 `{uid}_{session_id}_{index}` 模式（例如包含非数字的 `session_id` 或 `index`），`int(parts[2])` 会抛出 `ValueError`，导致训练在验证结束时崩溃。此外，该路径无 CI 覆盖，回归问题难以被自动检测。
- 影响：影响范围限定于设置了 `trainer.validation_data_dir` 且使用多输出 rollout 的场景。对不设置该配置的用户无影响。变更提升了验证 dump 的完整性，便于调试和分析多轮对话或 beam search 输出。
- 风险标记：缺少测试覆盖，核心路径变更，已知未解决讨论

关联脉络

- PR #6101 [trainer] fix: include uid and sort by uid in validation generation dumps in `main_ppo_sync`: 此 PR 是 PR#6101 的跟进，PR#6101 添加了 uid 和排序，但未包含所有输出；此 PR 解决的是同一个验证 dump 路径的完整性问题。