

PR #1990 完整报告

THUDM/slime

[ci] fix kl check on R3

合并时间: 2026-05-30 10:31

原文链接: <http://prhub.com.cn/THUDM/slime/pull/1990>

执行摘要

- 一句话: 修复 R3 模式下 KL 检查逻辑
- 推荐动作: 该 PR 是必要的紧急修复, 值得快速合并。建议未来对 R3 与正常路由的数值差异进行更深入的分析, 并考虑将对应的 TODO 转化为 issue。

功能与动机

在启用 R3 (rollout routing replay) 时, actor 前向和 ref 前向使用不同的路由策略, 导致初始 actor/ref KL 不为零, 原有的 `assert log_dict["train/kl_loss"] < 1e-8` 会误报。PR body 和代码注释均说明了这一原因。

实现拆解

1. 修改 `slime/backends/megatron_utils/model.py` 中的训练 KL 检查: 在 `train()` 函数中, 将原有的无条件 KL 断言 `if accumulated_step_id == 0 and "train/kl_loss" in log_dict` 改为增加 `not getattr(args, "use_rollout_routing_replay", False)` 条件, 当启用 R3 时跳过该断言, 因为初始 KL 不再为零。
2. 修改 `slime/backends/megatron_utils/data.py` 中的 rollout 数据 KL 检查: 在 `log_rollout_data()` 函数中, 将原有的 `rollout_id == 0` 条件前增加 `not getattr(args, "ci_disable_kl_checker", False)` 条件, 并更新注释说明 R3 导致 log-probs 不一致。同时保留了 `not getattr(args, "use_rollout_routing_replay", False)` 条件作为另一层保护。
3. 新增配置开关: `data.py` 中引入了 `ci_disable_kl_checker` 配置项, 允许在 CI 中完全禁用 KL 检查, 提供更灵活的兜底手段。

关键文件:

- `slime/backends/megatron_utils/model.py` (模块 训练引擎; 类别 source; 类型 core-logic; 符号 train): 修改了训练步骤中 CI 的 KL 检查逻辑, 增加了 `use_rollout_routing_replay` 条件跳过断言。
- `slime/backends/megatron_utils/data.py` (模块 数据加载; 类别 source; 类型 core-logic; 符号 `log_rollout_data`): 修改了 rollout 数据记录中的 CI 检查, 新增 `ci_disable_kl_checker` 配置和 R3 条件。

关键符号: `train`, `log_rollout_data`

关键源码片段

slime/backends/megatron_utils/model.py

修改了训练步骤中 CI 的 KL 检查逻辑，增加了 `use_rollout_routing_replay` 条件跳过断言。

```
def train(...):
    # ... 训练循环内部
    if args.ci_test and not args.ci_disable_kl_checker:
        if step_id == 0 and "train/ppo_kl" in log_dict and "train/pg_clipfrac" in log_dict:
            # TODO: figure out why KL is not exactly zero when using PPO loss with KL clipping
            assert log_dict["train/ppo_kl"] < 1e-8, f"{log_dict=}"
            # R3 replays rollout routing for the actor path, while ref
            # log-probs are computed with normal routing. The initial
            # actor/ref KL is therefore not expected to be exactly zero.
            if (
                accumulated_step_id == 0
                and not getattr(args, "use_rollout_routing_replay", False)
                and "train/kl_loss" in log_dict
            ):
                assert log_dict["train/kl_loss"] < 1e-8, f"{log_dict=}"
```

slime/backends/megatron_utils/data.py

修改了 rollout 数据记录中的 CI 检查，新增 `ci_disable_kl_checker` 配置和 R3 条件。

```
def log_rollout_data(...):
    # ... 收集 rollout 指标
    reduced_log_dict = gather_log_data("rollout", args, rollout_id, log_dict)
    if args.ci_test and reduced_log_dict is not None:
        # This is an initial actor/ref zero-KL check. R3 replays rollout
        # routing for the actor forward, while the reference forward
        # intentionally falls through to normal routing, so their
        # log-probs are not expected to match bit-for-bit in CI.
        if (
            rollout_id == 0
            and not getattr(args, "ci_disable_kl_checker", False) # 新增全局开关
            and not getattr(args, "use_rollout_routing_replay", False) # R3 保护
            and "rollout/log_probs" in reduced_log_dict
            and "rollout/ref_log_probs" in reduced_log_dict
        ):
            # 允许极小误差
            assert abs(reduced_log_dict["rollout/log_probs"] - reduced_log_dict["rollout/ref_log_
            probs"]) < 1e-8
```

评论区精华

无 review 评论。代码注释中保留了 TODO，说明需要进一步理解 PPO KL 裁剪导致初始 KL 非零的原因。

- 暂无高价值评论线程

风险与影响

- 风险：
 - 回归风险：低。变更仅在 CI 测试条件下生效（args.ci_test），不影响正常训练流程。
 - 漏测风险：如果 R3 模式下确实存在 KL 异常，新增的 ci_disable_kl_checker 可能被误用从而隐藏 bug。但代码中同时保留了 use_rollout_routing_replay 的检查，双重保险。
 - 配置兼容性：新增的 ci_disable_kl_checker 是 getattr 安全读取，不存在旧配置不兼容问题。
- 影响：
 - 影响范围：仅影响 CI 测试流程，不涉及用户训练逻辑。
 - 直接影响：修复了 R3 模式下 CI 测试的误报，使 PR 验证能够正确通过。
 - 间接影响：无。
 - 风险标记：CI 专用变更，缺少测试覆盖

关联脉络

- PR #1987 [ci] don't compare ref_logprob and logprob when R3 is on: 相同的处理背景——R3 开启时跳过 CI 中的 logprob 对比检查，本 PR 是同一思路的延续，覆盖训练 KL 检查。