

PR #1882 完整报告

THUDM/slime

fix ppo value offload bugs

合并时间: 2026-05-06 12:39

原文链接: <http://prhub.com.cn/THUDM/slime/pull/1882>

执行摘要

- 一句话: 修复 PPO 使用 critic 时的 offload 训练 bug
- 推荐动作: 建议精读。本 PR 修复了 PPO 多角色 offload 训练中的关键状态管理问题, 展示了异步训练中 actor 与 rollout 引擎连接维护的典型解决方案: sleep 前断开、wake 后重建。design_pattern 值得参考, 特别是 disconnect/connect 的显式化。另外参数强制绑定的策略也值得讨论——究竟是缺陷修复还是约束收紧, 不同团队可能有不同偏好。

功能与动机

PR 标题和 description 为空, 但从代码变更推断, 原有 PPO 训练逻辑在 use_critic=True 搭配 offload_train 时存在 actor 进程休眠后未正确断开 rollout 引擎 NCCL 通信组、唤醒后未重建连接的问题, 导致权重更新失败或程序崩溃。本修复旨在确保 critic 存在时 actor 的 offload 行为符合预期, 避免通信状态不一致。

实现拆解

1. 强制 offload_train: 在 slime/utils/arguments.py 的 slime_validate_args 中, 当 use_critic=True 时自动设置 offload_train=True, 保证后续逻辑前提条件。
2. 添加 disconnect_rollout_engines 方法: 在 slime/backends/megatron_utils/update_weight/update_weight_from_distributed.py 中新增公开方法, 用于显式断开 rollout 引擎的 NCCL 连接并清空 model_update_groups。
3. 调整 actor 休眠与唤醒: 在 slime/backends/megatron_utils/actor.py 的 sleep 方法中, 增加条件: 当角色为 actor、使用 critic、非 colocate 且 weight_updater 有 disconnect_rollout_engines 时, 在销毁进程组前主动断开连接; 在 train 入口增加条件: 当 offload_train 或 use_critic 时先唤醒; 在 train 结尾增加相应条件睡眠; 在 update_weights 中也增加重连逻辑: 如果需重连则先 wake_up, 连接 rollout 引擎, 完成更新后睡眠。
4. 简化 GPU 偏移计算: 在 slime/backends/sglang_utils/sglang_engine.py 的 get_base_gpu_id 中移除已废弃的 use_critic 分支, 因为 actor 和 critic 不再混排 GPU 分配。
5. 调整训练入口: 在 train.py 中确保 update_weights 与 offload_rollout 解耦, 使其在 offload_rollout 之外也能正确执行。

6. 新增集成测试: tests/test_qwen3_4B_ppo_disaggregate.py 涵盖 Qwen3-4B 在 disaggregate 模式下使用 critic 的端到端 PPO 训练流程, 通过条件变量 ENABLE_EVAL 和 TIGHT_HOST_MEMORY 控制评估和内存压力。

关键文件:

- slime/backends/megatron_utils/update_weight/update_weight_from_distributed.py (模块 权重同步; 类别 source; 类型 core-logic; 符号 disconnect_rollout_engines) : 新增 disconnect_rollout_engines 方法, 提供显式断开 rollout 引擎 NCCL 通信组的功能, 是修复连接管理的关键。
- slime/backends/megatron_utils/actor.py (模块 Actor 模块; 类别 source; 类型 core-logic; 符号 init, sleep, wake_up, update_weights) : 核心训练文件, 修改 sleep/wake_up/update_weights 等方法的条件控制, 确保 actor 在 critic 模式下正确管理 rollout 引擎连接。
- tests/test_qwen3_4B_ppo_disaggregate.py (模块 PPO 解耦; 类别 test; 类型 test-coverage; 符号 prepare, execute) : 新增 PPO disaggregate 模式端到端测试, 验证修复后的 critic+offload 训练正确性。
- slime/backends/sglang_utils/sglang_engine.py (模块 SGLang 引擎; 类别 source; 类型 core-logic; 符号 get_base_gpu_id) : 移除已废弃的 use_critic GPU 偏移计算分支, 简化逻辑。
- slime/utils/arguments.py (模块 参数验证; 类别 source; 类型 core-logic; 符号 slime_validate_args) : 强制 offload_train 与 use_critic 绑定, 保证后续逻辑的前提条件。
- train.py (模块 训练入口; 类别 source; 类型 core-logic) : 调整 update_weights 调用时机, 确保在 offload_rollout 以外的场景也能执行。
- .github/workflows/pr-test.yml (模块 CI 配置; 类别 infra; 类型 infrastructure) : CI 配置的小调整, 增加新测试触发。
- .github/workflows/pr-test.yml.j2 (模块 CI 配置; 类别 infra; 类型 infrastructure) : CI 模板调整, 配合新增测试。

关键符号: disconnect_rollout_engines, sleep, wake_up, update_weights, get_base_gpu_id, slime_validate_args, prepare, execute

关键源码片段

slime/backends/megatron_utils/update_weight/update_weight_from_distributed.py

新增 disconnect_rollout_engines 方法, 提供显式断开 rollout 引擎 NCCL 通信组的功能, 是修复连接管理的关键。

```
class WeightUpdater:
    # ... 其他方法 ...

    def disconnect_rollout_engines(self) -> None:
        """如果当前是PP源角色且存在model_update_groups, 则安全断开与rollout引擎的NCCL连接"""
        if not getattr(self, "_is_pp_src_rank", False) or self._model_update_groups is None:
```

```

        return
    # 调用底层函数断开连接
    disconnect_rollout_engines_from_distributed(
        self.args, self._group_name, self._model_update_groups, self.rollout_engines
    )
    self._model_update_groups = None # 置空标志, 防止重复断开

@torch.no_grad()
def update_weights(self) -> None:
    """权重更新主循环: 暂停rollout引擎、清缓存、广播非专家参数、广播专家参数、恢复引擎"""
    self.weight_version += 1
    # ... 具体广播逻辑 ...

```

slime/backends/megatron_utils/actor.py

核心训练文件, 修改 sleep/wake_up/update_weights 等方法的条件控制, 确保 actor 在 critic 模式下正确管理 rollout 引擎连接。

```

@timer
def sleep(self) -> None:
    """挂起actor, 释放GPU资源; 若使用critic则先断开rollout引擎连接"""
    assert self.args.offload_train
    clear_memory(clear_host_memory=True)
    print_memory("before offload model")
    # 需要断开连接的条件: 是 actor、使用 critic、非 colocate、weight_updater 支持断开
    if (
        self.role == "actor"
        and self.args.use_critic
        and not self.args.colocate
        and hasattr(self.weight_updater, "disconnect_rollout_engines")
    ):
        self.weight_updater.disconnect_rollout_engines()
    destroy_process_groups()
    torch_memory_saver.pause()
    print_memory("after offload model")

# 在 update_weights 中的关键分支 (在更新权重前后)
# 判断是否需要完整的重连流程
reconnect_rollout_engines = self.args.offload_train and self.args.use_critic and not self.args.colocate

if reconnect_rollout_engines:
    self.wake_up() # 唤醒 actor 进程组
elif self.args.offload_train:
    reload_process_groups() # 仅重载进程组

# 如果有新引擎加入或需要重连, 则连接 rollout 引擎
if num_new_engines > 0 or reconnect_rollout_engines:
    self.weight_updater.connect_rollout_engines(

```

```
        rollout_engines, rollout_engine_lock,
    )

# ... 权重更新步骤 ...

# 更新完成后，如果之前唤醒过则再次休眠
if reconnect_rollout_engines:
    self.sleep()
elif self.args.offload_train:
    destroy_process_groups()
```

评论区精华

Review 中 Copilot bot 指出两个 assert 安全问题：`sleep()` 和 `wake_up()` 内部 `assert self.args.offload_train`，但调用条件已成为 `offload_train or use_critic`，若 `use_critic=True` 而 `offload_train=False`（尽管参数校验已强制绑定，但逻辑上仍可能被绕过），`assert` 会失败。作者通过参数校验的强制绑定解决了该隐患，但 `assert` 仍保留，依赖上层校验保障安全性。

- `sleep/wake_up` 断言与调用条件不一致 (correctness): 作者通过 `slime_validate_args` 强制在 `use_critic` 时设置 `offload_train=True`，避免了断言触发的可能性，但 `assert` 仍存在，依赖参数校验的约束。

风险与影响

- 风险：
 - 强制修改 `use_critic` 时 `offload_train=True`，可能改变用户习惯，但实际训练中模型卸载是常见需求，风险较低。
 - `sleep/wake_up` 中的 `assert` 仍保留，若有人绕过参数校验直接调用底层接口，仍可能触发。
 - 新添加的 `disconnect_rollout_engines` 在 `colocate` 模式下不会被调用，但 `colocate` 与 `critic` 共存场景未明确测试。
 - 测试仅覆盖了 `disaggregate` 场景，`colocate` 或纯 `actor` 场景未新增测试。
- 影响：
 - 用户：开启 `critic` 训练的 PPO 用户在 `offload_train` 场景下不再遇到连接丢失崩溃；强制开启 `offload_train` 可能增加显存压力，但原本 `critic` 训练通常也需要 `offload`。
 - 系统：修改了核心训练循环（`actor.py train`），影响所有使用 `critic` 的 PPO 训练。
 - 团队：需要关注后续版本中 `critic` 与 `offload` 的耦合关系是否需解耦。
 - 影响程度：中，涉及训练流程正确性。
 - 风险标记：`critic` 与 `offload` 强制绑定，`sleep/wake_up` 断言隐患，`colocate` 模式未测试，仅 `disaggregate` 测试覆盖

关联脉络

- PR #1856 refactor/ppo: PPO 训练架构重构，解耦 actor-critic 配置与通信，本 PR 可能依赖于该重构引入的配置和通信机制。
- PR #1878 fix ppo value head load bugs: 同一系列的 PPO bug 修复，涉及 value head 加载，与本 PR 的 offload bug 修复同属 PPO 训练稳定性改进。