PR #2226 完整报告

radixark/miles

fix: preserve routing-replay state around MTP spec creation

合并时间：2026-08-07 06:29

原文链接：http://prhub.com.cn/radixark/miles/pull/2226

执行摘要

- 一句话：保存并恢复 routing-replay 状态，修复 critic 崩溃
- 推荐动作：值得精读：这是一个典型的进程级单例状态污染 bug，展示了隐式全局状态在 actor/critic 双角色下的陷阱。建议关注 model_provider 中状态括号的写法，并考虑后续补组合回归测试。

功能与动机

PR body 说明崩溃场景：Running PPO (--advantage-estimator ppo) with --use-rollout-routing-replay on a MoE model that has MTP layers (e.g. GLM-4.7-Flash with mtp_num_layers=1) crashes the critic on its first training step，报错 IndexError: pop_forward out of range: forward_index=0, len(top_indices_list)=0。根因是 routing_replay_manager 是进程级单例，critic 角色刻意保持禁用，但 get_model_provider_func 在 MTP block spec 构建后无条件将其重新启用，导致 critic 的 MoE router 注册 replay buffer 后前向 pop 空数据。

实现拆解

1. 定位根因：miles/backends/megatron_utils/model_provider.py 的 model_provider 内 MTP block spec 分支，在 use_rollout_routing_replay 下先设 routing_replay_manager.enabled = False，退出时无条件设 True，污染 critic 的全局单例状态。
2. 修改方案：进入分支前保存 prev_routing_replay_enabled = routing_replay_manager.enabled，退出时恢复该值而非强制 True，并补充注释说明 critic 角色需要保持禁用。
3. 行为验证：actor 原有状态是 True，恢复后不变；critic 保持 False，不再注册 replay buffer，崩溃消除且 routing 计算走 fresh 路径。
4. 测试配套：无新增测试；CI 现有 PPO 用例为 dense Qwen3-4B 且无 MTP，无法覆盖该组合路径，属于已知覆盖缺口。

关键文件：

- miles/backends/megatron_utils/model_provider.py（模块 模型构建；类别 source；类型 core-logic；符号 model_provider, get_model_provider_func）：核心修复文件，MTP block spec 构建括号内保存并恢复 routing_replay_manager.enabled，避免 critic 被误启

用 replay 模式。

关键符号：model_provider, get_model_provider_func

关键源码片段

miles/backends/megatron_utils/model_provider.py

核心修复文件，MTP block spec 构建括号内保存并恢复 routing_replay_manager.enabled，避免 critic 被误启用 replay 模式。

```
def model_provider(...):
    # 函数开头省略参数组装逻辑，重点关注 MTP block spec 构建分支

    if args.mtp_num_layers:
        from megatron.core.models.gpt.gpt_layer_specs import get_gpt_mtp_block_spec

        mtp_kwargs = {"use_transformer_engine": use_te}
        if vp_stage is not None:
            mtp_kwargs["vp_stage"] = vp_stage

        # routing_replay_manager 是进程级单例，critic 角色刻意保持禁用；
        # 构建 MTP block spec 期间临时关闭 replay 注册，之后必须恢复原状态，
        # 而不是强制 True，否则 critic 的 MoE router 会误注册 replay buffer
        if getattr(args, "use_rollout_routing_replay", False):
            prev_routing_replay_enabled = routing_replay_manager.enabled
            routing_replay_manager.enabled = False
            logger.warning(
                "Rollout routing replay is not applicable for MTP modules, so skipped replay
                registration"
            )

        mtp_block_spec = get_gpt_mtp_block_spec(config, transformer_layer_spec, **mtp_kwargs)
        kwargs["mtp_block_spec"] = mtp_block_spec

        if getattr(args, "use_rollout_routing_replay", False):
            # 恢复之前保存的状态：actor 原本为 True，行为不变；critic 保持禁用
            routing_replay_manager.enabled = prev_routing_replay_enabled

    with build_model_context(**build_model_context_args):
        model = GPTModel(**kwargs)

    if post_process and role == "critic":
        model.output_layer = LinearForLastLayer(input_size=config.hidden_size, output_size=1,
        config=config)

    return model
```

评论区精华

review 无实质异议，guapisolo 在 issue 评论中表示 'good fix!'，两位 reviewer（guapisolo、yueming-yuan）均 approve。没有未解决的 design 讨论。

- 修复确认 (other): 无异议，PR 获得两位 reviewer approve，已合并。

风险与影响

- 风险：回归风险低，改动仅恢复状态；但注意 routing_replay_manager 是进程级单例，若未来构建期间有其他路径修改状态，保存 / 恢复可能掩盖或覆盖。当前构建是同步的，风险可接受。主要风险是组合路径无测试覆盖：需要 critic + MoE + MTP + --use-rollout-routing-replay 才能复现，现有 CI 覆盖不到，未来回归可能再次发生。
- 影响：对用户：使 GLM-4.7-Flash 等 MoE + MTP 模型在 PPO + routing replay 下能正常训练，消除首步崩溃。对系统：critic 计算 routing 走 fresh 路径，性能与预期一致。对团队：修复极小但补测试成本高，需权衡后续补充组合回归测试。
- 风险标记：缺少测试覆盖，全局单例状态污染

关联脉络

- PR #2215 fix(mtp): double-shift GPT-path MTP labels: 同处 miles/backends/megatron_utils 下的 MTP 相关修复，与本次 routing-replay + MTP 交互同属 MoE/MTP 训练正确性链条。
- PR #1284 Nemotron RL support: 引入 Nemotron 混合架构 MTP 支持，MTP 层在 RL 训练中的行为持续被修复和约束，本 PR 是该方向上的后续修正。