

PR #6325 完整报告

verl-project/verl

[veomni] feat: add MoE router replay (R2/R3) support

合并时间: 2026-05-27 11:49

原文链接: <http://prhub.com.cn/verl-project/verl/pull/6325>

执行摘要

- 一句话: 为 VeOmni 引擎添加 MoE 路由器重放支持
- 推荐动作: 推荐阅读 `verl/utils/veomni/router_replay.py` 中的注释和设计决策 (id 键索引、R3 掩码、回退机制)。对于专注于 VeOmni 或 MoE 训练的研究者, 理解 RR 的工作原理有助于排查策略梯度偏差。PR 中的测试设计也值得参考, 特别是模拟 `recompute` 的测试。

功能与动机

在 MoE 模型中, rollout 和 actor 计算 `log_prob` 时路由器产生的 top-k 索引可能因浮点计算顺序而不同, 导致策略梯度的污染。通过记录 rollout 阶段的决策并在 actor 中重放 (R3), 或在 actor 计算 `log_prob` 时记录并重放 (R2), 可消除这种不匹配, 提高训练一致性和效率。PR 描述中强调: "Removes the floating-point drift between rollout and actor router decisions that otherwise contaminates the RL policy gradient on MoE models."

实现拆解

1. 核心控制器 (VeOmniRouterReplay) 在 `verl/utils/veomni/router_replay.py` 中实现状态机。
通过 id 键位置索引 (使用 `id(module)`) 分配层位置, 支持激活检查点下的反向重计算。提供 `begin_record/begin_replay/on_router_forward/collect_recorded/clear` 等生命周期方法。
2. 引擎集成 (`transformer_impl.py`): 在 `VeOmniEngineWithLMHead` 中集成控制器。在 `__init__` 中解析配置并初始化控制器; 在 `initialize()` 中执行防御性检查 (要求 `use_remove_padding=True`) 并安装钩子; 在 `prepare_model_inputs` 中记录侧通道信息 (`pad_size`、`cu_seqlens`) 并构建 R3 掩码; 在 `forward_backward_batch` 中组织 RECORD/REPLAY 生命周期, 使用 `try/finally` 确保状态清理。
3. 工作者配置 (`engine_workers.py`, `actor.py`): 在 `ActorRolloutRefWorker.__init__` 中读取 `actor.veomni.router_replay.mode`, 原只支持 `megatron`, 现扩展支持 `veomni`。在 `VeOmniActorConfig` 中添加配置字段。
4. 配置和示例: 修改 `veomni.yaml` 和 `_generated_ppo_veomni_trainer.yaml` 暴露 `router_replay` 配置块; 添加 `examples/router_replay/run_qwen3_5_35b_a3b_veomni.sh` 演示 R2/R3 使用。
5. 测试覆盖: 新增两个 CPU 测试文件 (共 29 个测试), 分别覆盖控制器状态机 (包括 `per-layer reverse recompute`) 和引擎胶水代码 (侧通道、`NestedTensor` 断言、严格错误

路径)。

关键文件:

- verl/utils/veomni/router_replay.py (模块 重放控制; 类别 source; 类型 entrypoint; 符号 RouterReplayAction, VeOmniRouterReplay, init, action) : 新增的核心控制器文件, 实现 VeOmniRouterReplay 状态机, 包含 RECORD/REPLAY 生命周期、id 键位置索引、跨 rank 聚合、安装 / 卸载等全部核心逻辑。
- tests/utils/veomni/test_router_replay_on_cpu.py (模块 控制器测试; 类别 test; 类型 test-coverage; 符号 ctrl, _FakeRouter, routers, _scores) : 新增的控制器状态机测试, 覆盖 RECORD 生命周期、per-layer 反向重计算、REPLAY 缺失目标错误路径等 15 个测试。
- tests/workers/test_router_replay_engine_helpers_on_cpu.py (模块 引擎集成测试; 类别 test; 类型 test-coverage; 符号 _make_jagged_input_ids, _make_jagged_routed_experts, _make_engine_with_controller, controller) : 引擎胶水代码的测试, 验证侧通道传递、NestedTensor 断言、R2/R3 掩码构造等 14 个测试。
- verl/workers/engine/veomni/transformer_impl.py (模块 引擎集成; 类别 source; 类型 dependency-wiring; 符号 _maybe_push_router_replay_state) : 引擎核心实现, 集成路由器重放钩子, 修改 forward_backward_batch 和 prepare_model_inputs 以支持 RECORD/REPLAY 生命周期。
- verl/utils/veomni/__init__.py (模块 包初始化; 类别 source; 类型 core-logic) : 新包初始化文件, 使 router_replay 成为 verl.utils.veomni 子模块。
- verl/workers/engine_workers.py (模块 工作者配置; 类别 source; 类型 core-logic) : 工作者工厂, 扩展路由器重放检查逻辑以支持 veomni 后端。
- verl/workers/config/actor.py (模块 配置定义; 类别 source; 类型 core-logic) : 配置定义, 为 VeOmniActorConfig 添加 router_replay 字段。
- examples/router_replay/run_qwen3_5_35b_a3b_veomni.sh (模块 示例脚本; 类别 other; 类型 entrypoint) : 使用示例脚本, 演示如何通过环境变量 ROUTING_REPLAY_MODE 启用 R2/R3。
- verl/trainer/config/engine/veomni.yaml (模块 引擎配置; 类别 config; 类型 configuration) : 配置文件, 暴露 router_replay 配置块。
- verl/trainer/config/_generated_ppo_veomni_trainer.yaml (模块 生成配置; 类别 config; 类型 configuration) : 生成的完整配置, 包含 router_replay 默认值。

关键符号: VeOmniRouterReplay.init, VeOmniRouterReplay.install, VeOmniRouterReplay.uninstall, VeOmniRouterReplay.on_router_forward, VeOmniRouterReplay.begin_record, VeOmniRouterReplay.begin_replay, VeOmniRouterReplay.collect_recorded, VeOmniRouterReplay.set_microbatch_targets, VeOmniEngineWithLMHead._maybe_push_router_replay_state, VeOmniEngineWithLMHead.forward_backward_batch, VeOmniEngineWithLMHead.prepare_model_inputs

关键源码片段

verl/utils/veomni/router_replay.py

新增的核心控制器文件，实现 VeOmniRouterReplay 状态机，包含 RECORD/REPLAY 生命周期、id 键位置索引、跨 rank 聚合、安装 / 卸载等全部核心逻辑。

```
class VeOmniRouterReplay:
    """Router replay controller for VeOmni (FSDP2 + optional Ulysses SP)."""
    def __init__(self, sp_group: dist.ProcessGroup | None = None) -> None:
        self._sp_group = sp_group
        self._action: RouterReplayAction = RouterReplayAction.DISABLED
        # id(router_module) -> position. Populated lazily on first sight
        # of each router; stable across the lifetime of the controller
        # Key: uses `id(module)` so per-layer activation checkpointing
        # backward recompute (reverse order) still maps to the correct
        # position — monotonic cursor would fail here.
        self._id_to_pos: dict[int, int] = {}
        # RECORD: per-layer-position list of [local_nnz, topk] tensors
        # One entry per micro-batch; outer list grows as new routers fire
        # on the first micro-batch.
        self._recorded: list[list[torch.Tensor]] = []
        # REPLAY: per-layer-position list of target indices for current micro-batch
        self._targets: list[torch.Tensor] = []
        ...
    def install(self, model: nn.Module) -> None:
        """Set this controller as the active replay target in the VeOmni
        moe_router_replay module-level singleton."""
        import veomni.utils.moe_router_replay
        veomni.utils.moe_router_replay.set_active_replay(self)
```

评论区精华

审查中主要涉及三点：

- 性能优化: `gemini-code-assist[bot]` 建议在构造布尔掩码时使用 `torch.repeat` 替代 Python 列表，以减少转换开销。
- 配置解耦: `wuxibin89` 建议将 `router_replay` 的检查逻辑从 `engine_workers.py` 移至 `VeOmniActorConfig.__post_init__`，保持工作者层不感知后端细节。
- 依赖版本: `Luosuu` 指出可以升级 `veomni` 版本至 0.1.11，以支持必要的钩子。最终 PR 合并前应已解决这些问题（版本已更新）。
 - 布尔掩码构造优化 (performance): 未明确是否采纳，但在合并前应有机会优化。
 - 配置检查移至配置类 (design): PR 合并前已调整，当前代码中配置检查已部分移到 `VeOmniActorConfig`。
 - VeOmni 版本升级 (other): PR 中已更新 CI 配置使用 0.1.10，可能后续再升级。

风险与影响

- 风险：
 1. 核心逻辑复杂: 状态机在激活检查点下的正确性依赖 id 键索引，但任何未预期的模块对象创建（如 FSDP2 rebuild）可能破坏稳定性。

2. 配置约束: 要求 `use_remove_padding=True`, 若用户在非 `rmpad` 路径下启用 RR 会抛出运行时错误, 但文档说明不足可能导致困惑。
3. 外部依赖: 依赖 VeOmni 侧 `set_active_replay` 钩子, 若 VeOmni 版本不匹配将导致安装失败。
4. 性能开销: RECORD 需要跨 SP ranks 做 all-gather, 可能增加通信开销; R3 模式的内存占用随序列长度增加。
5. 测试覆盖局限: CPU 测试覆盖了状态机和引擎胶水, 但端到端 GPU 测试仅通过手动脚本验证, 可能在多卡 / 分布式环境下有未暴露的问题。 - 影响: 影响范围: 直接影响 VeOmni 引擎上的 `qwen3_moe/qwen3_5_moe` 模型训练, 通过减小策略梯度方差可能提升训练效果。程度: 中。由于引入可选功能, 默认 disabled, 不会影响现有工作流。但用户如需启用, 需满足 `use_remove_padding=True` 及特定 VeOmni 版本。团队需保持对 VeOmni 钩子演进的同步。 - 风险标记: 核心路径变更, 外部依赖 VeOmni, 配置约束 `use_remove_padding`, 缺少分布式端到端测试

关联脉络

- PR #6470 [veomni] feat: wire MoE load-balance monitor into VeOmni engine: 也为 VeOmni 引擎增加 MoE 相关监控, 属于同一技术栈, 可能共享部分配置或基础设施。
- PR #6453 [veomni] feat: add VeOmni-native critic support: 同样扩展 VeOmni 引擎功能, 涉及相似的引擎集成模式。