

PR #2235 完整报告

radixark/miles

feat(fsdp): support rollout routing replay (R3) for fsdp backend

合并时间: 2026-08-08 03:42

原文链接: <http://prhub.com.cn/radixark/miles/pull/2235>

执行摘要

- 一句话: FSDP 后端新增 R3 路由重放, 覆盖三种 MoE 架构
- 推荐动作: 值得精读。设计文档质量很高: 适配器模式把架构差异收敛到 RoutingReplayAdapter, 按全局 decoder 层索引而非 MoE 序号注册流是正确的数据契约, 梯度检查点下 forward/backward 双游标嵌套是值得借鉴的模式。若后续要扩展新 MoE 架构或维护 FSDP 训练正确性, 本 PR 是必读参考; 同时建议关注 --ci-test 之外的重放校验手段。

功能与动机

PR body 明确指出 R3 的核心本身是后端无关的 (replay_base.py、replay_data.py 与 rollout 侧产出 routed_experts 的部分均未改动), Megatron 在其 fork 的 TopKRouter 内安装 hook, 而 FSDP 训练的是原生 HF modeling, 因此需要按模型实例安装 hook。动机是让 FSDP 训练时精确复现 rollout 中的专家路由选择, 从而压缩 train/infer 对数概率差; 实测 qwen3 系列约 2x 改善, 且 check_replay_result 从未触发 (失配 token 占比始终低于 1%)。

实现拆解

1. 新增通用适配层 `miles/backends/experimental/fsdp_utils/adaptations/routing_replay.py`: 定义 RoutingReplayAdapter (name/applies_to/module_cls_name/install) 与模块级注册表, 提供 enable (按参数设置 manager 状态)、discover_moe_modules (按模块类名发现 MoE 层, 并从模块路径正则解析全局 decoder 层索引)、install (为每层挂 hook 并以全局层索引注册 replay 流)、fill (把 rollout 的路由数据灌入每层队列)、log_prob_stage/stage/rewind/reset (stage 编排与游标控制)。选择按路径正则解析层索引而非沿属性树遍历, 是因为 Qwen3.5 文本 / 多模态变体把层列表放在不同深度。
2. 新增架构级 hook `miles/backends/experimental/fsdp_utils/models/replay_routers.py`: install_qwen3_router_replay 重绑 Qwen3MoeTopKRouter/Qwen3_5MoeTopKRouter 的 forward, 用 _miles_replay_topk 替换原始 torch.topk; install_glm4_moe_lite_router_replay 重绑 Glm4MoeLiteMoE.route_tokens_to_experts, 只替换最终专家选择 topk, 保留 group-limited 路由及其 topk(2) 与组选择, 因为回放组会把专家 id 塞进组编号空间; GLM 的权重仍从 sigmoid 后 logits 中 gather, 保证可微。
3. 在 specs 中注册适配器: adaptations/specs/qwen3_moe.py、qwen3_5.py、glm4_moe_lite.py 各自通过 register_routing_replay_adapter 注册 applies_to 判定与对应 hook; qwen3_5 的 _is_qwen3_5_moe 同时覆盖 qwen3_5_moe_text 与多模态变体。

4. 接入 actor 训练循环: miles/backends/experimental/fsdp_utils/actor.py 在 init 中调用 routing_replay.enable 与 routing_replay.install (仅 actor, ref 模型不安装以避免重复注册); _train_core 中 ref 阶段包 FALLTHROUGH, actor logprob 阶段包 log_prob_stage (rollout 变体为 replay_forward, 非 rollout 变体为 record) 后 rewind, 训练阶段整体包 REPLAY_BACKWARD, _train_step 的 forward 包 REPLAY_FORWARD, 使梯度检查点重算从独立 backward 游标取数。
5. 参数与数据路径修复: miles/args/arguments.py 新增 resolve_fsdp_num_layers (优先取 text_config 的 num_hidden_layers) 并在有 hf_checkpoint 时填充 args.num_layers, 因为 sglang_rollout 要用它 reshape 路由 buffer; miles/backends/training_utils/replay_data.py 改为不直接读 args.sequence_parallel, 避免 FSDP 路径 AttributeError。
6. 测试与 CI 配套: tests/fast/backends/test_fsdp_routing_replay.py 与 test_fsdp_replay_routers.py 走 stage-a-cpu, 共用新增的 tests/fast/fixtures/replay_fixtures.py (CpuReplay 模拟真实 Replay 的双游标语义); tests/e2e/fsdp/r3/ 下每架构一个 e2e (qwen3-30B-A3B、qwen3-35B-A3B、GLM-4.7-Flash), 全部开启 --use-rollout-routing-replay --ci-test --gradient-checkpointing --colocate, 用 check_replay_result 兜底校验重放正确性。

关键文件:

- miles/backends/experimental/fsdp_utils/adaptations/routing_replay.py (模块 路由重放; 类别 source; 类型 dependency-wiring; 符号 RoutingReplayAdapter, register_routing_replay_adapter, resolve_routing_replay_adapter, uses_rollout_replay): R3 适配层核心: 适配器注册与解析、MoE 模块发现、hook 安装与 replay 流注册、stage 编排与数据填充, 是 FSDP actor 与后端无关 R3 核心之间的唯一桥梁。
- miles/backends/experimental/fsdp_utils/models/replay_routers.py (模块 路由钩子; 类别 source; 类型 data-contract; 符号 _qwen3_router_forward, install_qwen3_router_replay, _glm4_moe_lite_route_tokens_to_experts, install_glm4_moe_lite_router_replay): 架构级 hook: 重绑 qwen3/qwen3_5 的 TopKRouter.forward 与 GLM group-limited 路由的最终 topk, 保持可微, 是 R3 能在 stock HF 模型上工作的关键。
- miles/backends/experimental/fsdp_utils/actor.py (模块 训练循环; 类别 source; 类型 dependency-wiring): 训练循环编排: fill、ref fallthrough、actor replay_forward + rewind、训练 replay_backward/replay_forward 嵌套, 是 R3 生效的主路径, 也是唯一改动点被 review 的地方。
- miles/args/arguments.py (模块 参数解析; 类别 source; 类型 core-logic; 符号 resolve_fsdp_num_layers): 修复 FSDP 路径 num_layers 缺失: sglang_rollout 需要用 args.num_layers 把路由 buffer reshape 成 [tokens, num_layers, topk], 新增 resolve_fsdp_num_layers 并从 hf_checkpoint 填充。
- miles/backends/experimental/fsdp_utils/adaptations/specs/qwen3_5.py (模块 适配器; 类别 source; 类型 core-logic; 符号 _is_qwen3_5_moe): 为 qwen3_5_moe / qwen3_5_moe_text 注册 R3 适配器, hook 复用 qwen3 的 forward, 是架构扩展的样板实现。

- tests/fast/backends/test_fsdp_replay_routers.py (模块 单测覆盖; 类别 test; 类型 test-coverage; 符号 test_fallthrough_matches_stock_forward, test_replay_forward_returns_the_recorded_indices, test_forward_and_backward_cursors_are_independent, test_replay_can_select_experts_outside_the_recomputed_groups) : 最核心的 fast 单测 : fallthrough 与 stock forward 一致、replay 返回记录 indices、前后向游标独立、权重可微、GLM 可重放组外专家 (test_replay_can_select_experts_outside_the_recomputed_groups) 。

关键符号: enable, install, discover_moe_modules, fill, log_prob_stage, stage, rewind, reset, _qwen3_router_forward, install_qwen3_router_replay, _glm4_moe_lite_route_tokens_to_experts, install_glm4_moe_lite_router_replay, resolve_fsdp_num_layers

关键源码片段

miles/backends/experimental/fsdp_utils/models/replay_routers.py

架构级 hook: 重绑 qwen3/qwen3_5 的 TopKRouter.forward 与 GLM group-limited 路由的最终 topk, 保持可微, 是 R3 能在 stock HF 模型上工作的关键。

```
# qwen3_moe 与 qwen3_5_moe 共用的重绑版 router forward:
# 把原始 torch.topk 换成 manager 提供的可重放 topk 函数。
def _qwen3_router_forward(self, hidden_states):
    hidden_states = hidden_states.reshape(-1, self.hidden_dim)
    router_logits = F.linear(hidden_states, self.weight)
    router_probs = F.softmax(router_logits, dtype=torch.float, dim=-1)
    router_top_value, router_indices = self._miles_replay_topk(router_probs, self.top_k)
    if getattr(self, "norm_topk_prob", True):
        router_top_value = router_top_value / router_top_value.sum(dim=-1, keepdim=True)
    router_top_value = router_top_value.to(router_logits.dtype)
    return router_logits, router_top_value, router_indices
```

```
def install_qwen3_router_replay(router: nn.Module) -> None:
    """两个架构的 forward 一致 (仅 qwen3_5 恒重归一化), 共用一个 hook."""
    router._miles_replay_topk = routing_replay_manager.get_topk_fn(
        lambda scores, k: torch.topk(scores, k, dim=-1), return_probs=True
    )
    router.forward = types.MethodType(_qwen3_router_forward, router)
```

```
# GLM group-limited 路由的重绑版: 只替换最终专家选择 topk。
# 前半段 group 选择保持原样, 因为回放 group 会把专家 id 塞进组编号空间;
# topk_weights 仍从 sigmoid 后的 router_logits 中 gather, 重放保持可微。
def _glm4_moe_lite_route_tokens_to_experts(self, router_logits):
    router_logits = router_logits.sigmoid()
    router_logits_for_choice = router_logits + self.gate.e_score_correction_bias
    group_scores = (
```

```

        router_logits_for_choice.view(-1, self.n_group, self.n_routed_experts // self.n_group)
        .topk(2, dim=-1)[0]
        .sum(dim=-1)
    )
    group_idx = torch.topk(group_scores, k=self.topk_group, dim=-1, sorted=False)[1]
    group_mask = torch.zeros_like(group_scores)
    group_mask.scatter_(1, group_idx, 1)
    score_mask = (
        group_mask.unsqueeze(-1)
        .expand(-1, self.n_group, self.n_routed_experts // self.n_group)
        .reshape(-1, self.n_routed_experts)
    )
    scores_for_choice = router_logits_for_choice.masked_fill(~score_mask.bool(), float("-inf"))
    topk_indices = self._miles_replay_topk(scores_for_choice, self.top_k)
    topk_weights = router_logits.gather(1, topk_indices)
    if self.norm_topk_prob:
        denominator = topk_weights.sum(dim=-1, keepdim=True) + 1e-20
        topk_weights = topk_weights / denominator
    topk_weights = topk_weights * self.routed_scaling_factor
    return topk_indices, topk_weights

def install_glm4_moe_lite_router_replay(block: nn.Module) -> None:
    """专家选择发生在 Glm4MoeLiteMoE 块内而非 gate，hook 安装到块上。"""
    block._miles_replay_topk = routing_replay_manager.get_topk_fn(
        lambda scores, k: torch.topk(scores, k, dim=-1, sorted=False)[1], return_probs=False
    )
    block.route_tokens_to_experts = types.MethodType(_glm4_moe_lite_route_tokens_to_experts,
    block)

```

评论区精华

唯一一条针对代码的 review 讨论是 Rockdu 在 [miles/backends/experimental/fsdp_utils/actor.py](#) 第 463 行提出的疑问：'Not about this PR specifically, but I just realized we're forcing FSDP to use the recomputed logprob; is it as expected?', 即 FSDP 在 actor logprob 阶段默认使用重算 logprob 是否符合预期。PR body 的设计说明实际上已回答：actor logprob 阶段以 [replay_forward](#) 重放 rollout 的路由、训练 forward 同样重放，因此重算 logprob 与 rollout 采样分布一致，这正是 R3 的目标；合并前未见作者对该评论的公开回复。另外 Rockdu 在 issue 评论给出 LGTM，并贴出 Qwen3.5-35B-A3B 在 DAPO 8k 上下文上的训练曲线正常；因本 PR 触及 utils 目录 ([arguments.py](#))，Rockdu 要求 guapisolo 批准，guapisolo 以 'approve to unblock' 放行，yueming-yuan 也 approve。

- FSDP 强制使用重算 logprob 是否符合预期 (question): 未看到作者公开答复；R3 设计本身说明 actor logprob 与训练 forward 都会 [replay_forward](#) 重放 rollout 路由，故重算 logprob 与 rollout 分布一致，属于预期行为。PR 随后被 approve 并合并。

风险与影响

- 风险：核心训练循环编排改动集中在 actor.py 的 _train_core/_train_step: stage 的嵌套与游标回卷（rewind）一旦错位，在梯度检查点重算时会产生静默错误的梯度；该场景主要依赖 --ci-test 才开启的 check_replay_result 兜底，线上关闭该开关时缺少自动校验。hooks 依赖 transformers 类名与 forward 签名（Qwen3MoeTopKRouter、Glm4MoeLiteMoE 等），transformers 升级可能使 discover_moe_modules 找不到模块或行为漂移；install 对找不到适配器 / 层有显式报错，但类名相同而语义变化的场景仍难发现。GLM 只重放最终 topk、group 选择仍重算，可能出现重放专家与重算 group 冲突的情况，实测其提升也仅 1.17x。验证本身是跨 batch 对比（rollout temperature 1 导致两次运行数据不同），并非同数据 A/B。qwen3_5_moe_text 未列入 VERIFIED_MODEL_TYPES，该变体的 CI 覆盖尚未发生。
- 影响：影响范围集中在实验性 FSDP 后端：Qwen3-30B-A3B、Qwen3.5-35B-A3B、GLM-4.7-Flash 三种 MoE 架构的 GRPO 训练将可启用 R3，训练 / 推理路由一致性显著改善，qwen3 系列对数概率差降低约 2x。对用户而言，新增 --use-rollout-routing-replay 开关默认关闭，不改变现有行为；启用后每 rollout 需要额外的 replay 队列内存 / 显存开销。对系统而言，args.num_layers 在 FSDP 路径被正确填充，属于对既有缺失的修复，降低其他依赖该字段模块的隐性风险。对团队而言，FSDP 后端从此拥有与 Megatron 对齐的 R3 能力，但同时增加了对 transformers 版本结构的维护约束；CI 新增 3 个 GPU e2e 与 2 个 CPU fast 套件。
- 风险标记：核心训练循环编排变更，hooks 依赖 transformers 类名，升级易失效，验证基于跨 batch 对比而非同数据 A/B，qwen3_5_moe_text 未列入 VERIFIED_MODEL_TYPES，check_replay_result 仅 --ci-test 开启

关联脉络

- PR #2226 fix: preserve routing-replay state around MTP spec creation: 同一 routing_replay_manager 状态管理脉络：Megatron 侧保住状态修复，本 PR 在 FSDP 侧首次接入同一重放机制，两者共同定义 R3 的跨后端契约。
- PR #2213 fix(fsdp): apply the GDN packing patch to dense qwen3_5 and fix patch: 同为 FSDP 后端对 qwen3_5 架构的适配工作，触及 adaptations/specs/qwen3_5.py 与 packing 补丁，本 PR 复用同一 spec 注册表。
- PR #2215 fix(mtp): double-shift GPT-path MTP labels: 同为训练正确性修复；MTP 标签偏移与 R3 都影响 MoE 架构的梯度正确性，属于同一正确性主题。