PR #1794 完整报告

radixark/miles

feat(multi-lora): enable and validate MoE expert adapters

合并时间：2026-07-30 09:42

原文链接：http://prhub.com.cn/radixark/miles/pull/1794

执行摘要

- 一句话：启用 MoE 专家多 LoRA，扩展 (tp,pp,ep) 分片与校验
- 推荐动作：值得精读。三个设计决策有借鉴价值：(1) 用一次缓存的 gloo all-gather 获取 realized 坐标集合而非推导 tp × pp × ep 笛卡尔积，正确处理 ETP < TP 的稀疏拓扑；(2) rank 轴从张量末尾寻址，使同一函数同时正确服务 2-D dense 与 3-D/4-D packed 导出张量；(3) 把依赖 Megatron 解析结果的校验放在 provider.finalize() 之后，避免 CLI 层对 None 默认值的误判。配套测试（尤其 test_completeness_ignores_unrealized_coordinates）精准锚定了每个修复动机，适合作为回归测试范本；review 中 " 注释压缩到 1-2 行 " 的约定也体现了该仓库对可读性的要求。

功能与动机

MultiLoRA 此前完全跳过 MoE 专家线性层，导致 MoE 模型上最具可训练容量的部分——专家权重——静默地得不到任何适配器。PR body 明确目标："enable and validate MoE expert adapters"，把多适配器路径与单适配器 LoRA 在 grouped experts 上对齐，前置依赖 radixark/Megatron-Bridge#23。作者还通过 GLM-5.2_5layer 多 LoRA e2e（首个走通该路径的 MLA + dp-attention 模型）发现了两个配套缺陷：Megatron-Bridge#25（replicated base linear 下 token span 未收窄到 SP shard、缺失 output gather）与 sglang#32421（分布式 LoRA 加载是唯一仍断言 dp_size == 1 的端点），两者分别阻塞 MLA 模型训练与 dp-attention 场景的 serving，缺任一都表现为运行期失败或策略静默漂移。

实现拆解

1. 专家叶子目标识别（miles/utils/multi_lora.py）：新增 _EXPERT_LEAF_NAMES（linear_fc1/linear_fc2/gate_proj/up_proj/down_proj）、_ALL_MODULE_ALIASES 与 targets_expert_leaves。该函数把每个 target 条目映射到其叶子名（最后一个点分组件）并与专家叶子集合比对，bulk 别名（all/all-linear/all_linear）直接视为命中，兼容通配符路径。它是所有 MoE 特化处理（关闭 permute fusion、专家校验）的闸门，误判会静默跳过这些处理。
2. 启动期参数约束（同文件 validate_multi_lora_args）：新增三个断言——pipeline_model_parallel_size 必须为 1（pipelined schedule 下 adapter routing 不是 recompute-safe，且无单个 rank 持有完整 adapter 可推送）；--qkv-format 必须为 thd（per-slot token span 依赖按序列连续打包 sample，bshd 会交错）；拒绝 experts_shared_outer_loras（训练侧是 per-expert 布局，sglang 侧 shared-outer 布局

训练永远不会产生）。ETP 刻意不在 CLI 层校验，因为其 None 默认值只有 Megatron 解析后才明确。

3. bridge 侧模型构建调整（miles/backends/megatron_utils/bridge_lora_helpers.py）：_setup_lora_model_via_bridge 在 provider.finalize() 之前，若 multi-LoRA 且目标命中专家叶子则强制 provider.moe_permute_fusion = False（fused permute 记录的是 TE 的 row_id_map，专家适配器无法回放该置换，只损失 fused kernel）。finalize 之后调用新增的 _validate_multi_lora_moe_support，在已解析的 Megatron 配置上校验 ETP == 1、moe_grouped_gemm、无 fp8/fp4、无 capacity padding、双侧专家投影（gate_proj/up_proj/down_proj 全在目标中，避免 sglang rollout 静默丢弃单侧目标）、无 permute fusion。
4. checkpoint 分片按 (tp, pp, ep) 键控（miles/backends/megatron_utils/multi_lora_utils.py）：megatron_shard_name 生成三坐标分片名，ep_size == 1 时省略 _ep 后缀保持旧布局可加载；adapter_shard_topology 用一次缓存的 gloo all-gather 汇总所有 rank 的 realized 坐标并选举每个分片唯一 writer（同坐标取最小 global rank）；find_latest_checkpoint 按 realized 坐标集合做完整性检查并保留 legacy 回退；save_multi_lora_checkpoints 的 writer 门控从 intra_dp_cp.rank == 0 改为 adapter_shard_topology 结果。关键洞察：ETP < TP 时 realized 坐标不是 tp × pp × ep 笛卡尔积，本地无法推导。
5. rank 裁剪修正（同文件 slice_lora_to_rank）：rank 轴改为从张量末尾寻址（lora_A 为 tensor.ndim - 2、lora_B 为 tensor.ndim - 1），对 2-D 稠密张量行为不变，对 packed grouped-expert 导出（(E, rank, in) / (E, out, rank)）正确落在 rank 轴上；padding 非零仍硬失败。exclude_modules 不再转发给 MultiLoRA（已在参数校验中减掉，ModuleMatcher 断言其为空）。
6. 测试配套：新增 tests/fast/backends/megatron_utils/test_multi_lora_checkpoint_naming.py（分片命名、unrealized 坐标完整性，含 TP=2/EP=2/ETP=1 场景）与 tests/fast/utils/test_targets_expert_leaves.py；扩展 test_slice_lora_to_rank.py（packed expert 四个用例）与 test_arguments.py（PP/bshd/shared-outer 拒绝、无 ETP 标志接受）。测试均在 torchrun 4 rank（TP=2、EP=2、ETP=1）下验证了 writer 选举：每 realized 坐标恰好一个 writer，2 个分片而非笛卡尔积要求的 4 个。

关键文件：

- miles/backends/megatron_utils/multi_lora_utils.py（模块 LoRA 工具；类别 source；类型 core-logic；符号 all_megatron_checkpoints_exist, megatron_shard_name, adapter_shard_topology, find_latest_checkpoint）：核心变更文件：checkpoint 分片从 (tp, pp) 扩展到 (tp, pp, ep)，新增 megatron_shard_name 与 adapter_shard_topology（gloo all-gather 选举 writer、realized 坐标集合），find_latest_checkpoint 增加 legacy 回退，slice_lora_to_rank 修正 packed expert 张量的 rank 轴裁剪。
- miles/backends/megatron_utils/bridge_lora_helpers.py（模块 桥接层；类别 source；类型 core-logic；符号 _validate_multi_lora_moe_support, _setup_lora_model_via_bridge）：新增 _validate_multi_lora_moe_support 校验函数，并在 provider.finalize 前后分别强制关闭 moe_permute_fusion 与执行 MoE 支持校验，是训练侧启动路径的关键闸门。

- miles/utils/multi_lora.py（模块 LoRA 工具；类别 source；类型 core-logic；符号 targets_expert_leaves, validate_multi_lora_args）：新增 targets_expert_leaves 专家叶子识别函数，并扩展 validate_multi_lora_args 的启动期断言（PP=1、qkv-format=thd、拒绝 experts_shared_outer_loras），是所有 MoE 特化处理的闸门。
- tests/fast/backends/megatron_utils/test_multi_lora_checkpoint_naming.py（模块 分片命名；类别 test；类型 test-coverage；符号 _names, test_shard_name_omits_ep_suffix_without_expert_parallelism, test_shard_name_is_unique_per_expert_parallel_rank, test_completeness_check_requires_every_realized_shard）：新增测试覆盖分片命名与 realized 坐标完整性检查，其中 test_completeness_ignores_unrealized_coordinates 精确锚定了 ETP < TP 时笛卡尔积检查会永不恢复的核心缺陷。
- tests/fast/utils/test_targets_expert_leaves.py（模块 目标模块；类别 test；类型 test-coverage；符号 test_mlp_leaf_names_target_experts, test_expert_scoped_wildcards_target_experts, test_attention_only_targets_do_not, test_bulk_aliases_target_experts）：新增测试覆盖 targets_expert_leaves 的叶子名、通配符、bulk 别名、裸字符串与空输入行为，防止闸门函数静默误判。
- tests/fast/backends/megatron_utils/test_slice_lora_to_rank.py（模块 切片逻辑；类别 test；类型 test-coverage；符号 test_packed_expert_lora_a_is_sliced_on_the_rank_dim, test_packed_expert_lora_b_is_sliced_on_the_rank_dim, test_packed_expert_nonzero_padding_is_rejected, test_packed_expert_fewer_experts_than_rank_is_not_confused）：扩展测试覆盖 packed grouped-expert 张量的 rank 轴裁剪、非零 padding 拒绝与 " 专家数少于 rank 不混淆 " 场景，直接印证 slice_lora_to_rank 的维度修正。
- tests/fast/utils/test_arguments.py（模块 参数校验；类别 test；类型 test-coverage；符号 test_rejects_pipeline_parallelism, test_rejects_bshd_qkv_format, test_rejects_shared_outer_expert_loras, test_accepts_expert_leaf_targets_without_expert_tp_flag）：扩展测试覆盖新增的启动期断言：PP 拒绝、bshd 拒绝、shared-outer 拒绝，以及无 ETP 标志时 expert leaf 目标被接受（验证校验放在 post-finalize 而非 CLI 层的决策）。

关键符号：megatron_shard_name, adapter_shard_topology, all_megatron_checkpoints_exist, find_latest_checkpoint, slice_lora_to_rank, save_multi_lora_checkpoints, targets_expert_leaves, _validate_multi_lora_moe_support, validate_multi_lora_args

关键源码片段

miles/backends/megatron_utils/bridge_lora_helpers.py

新增 _validate_multi_lora_moe_support 校验函数，并在 provider.finalize 前后分别强制关闭 moe_permute_fusion 与执行 MoE 支持校验，是训练侧启动路径的关键闸门。

```
def _validate_multi_lora_moe_support(args: Namespace, provider) -> None:
    """拒绝多槽 grouped-expert adapter 无法服务的 MoE 配置。

    在 provider.finalize 之后检查，因为这些约束依赖 Megatron 解析后的配置
```

```
    而非 CLI 原始值，提前在 CLI 层校验会误伤省略默认参数的合法启动。
    """
    if not getattr(provider, "num_moe_experts", None):
        return
    if not targets_expert_leaves(args.target_modules):
        logger.info("[multilora] MoE model with no expert leaves in --target-modules; experts stay
        frozen")
        return

    # --expert-tensor-parallel-size 默认 None，只有 Megatron 解析后才明确。
    expert_tp = getattr(provider, "expert_tensor_parallel_size", 1) or 1
    assert expert_tp == 1, (
        f"Multi-LoRA on MoE experts requires expert_tensor_parallel_size=1 (resolved to "
        f"{expert_tp}); set --expert-tensor-parallel-size 1."
    )
    assert getattr(provider, "moe_grouped_gemm", False), (
        "Multi-LoRA on MoE experts requires moe_grouped_gemm=True (SequentialMLP expert "
        "linears are skipped, so the experts would train no adapter)."
    )
    # fp8/fp4 的量化 padding 会打乱 dispatcher 的 token 顺序，专家适配器无法对齐行号。
    assert not getattr(provider, "fp8", None) and not getattr(provider, "fp4", None), (
        "Multi-LoRA on MoE experts does not support fp8/fp4 experts (quantization padding "
        "desynchronizes the dispatched token order)."
    )
    # sglang 只在 gate_up 和 down 两个投影同时被目标时包装 FusedMoE 层；
    # 单侧目标会在 rollout 时被静默丢弃，必须在启动期拦截。
    served = set(convert_target_modules_to_hf(list(args.target_modules)))
    expert_pair = {"gate_proj", "up_proj", "down_proj"}
    if served & expert_pair:
        assert expert_pair <= served, (
            f"Multi-LoRA on MoE experts requires all of {sorted(expert_pair)} in "
            f"--target-modules (got {sorted(served & expert_pair)}); a one-sided expert "
            f"target is dropped at rollout time."
        )
    assert not getattr(provider, "moe_pad_expert_input_to_capacity", False), (
        "Multi-LoRA on MoE experts does not support --moe-pad-expert-input-to-capacity."
    )
    assert not getattr(provider, "moe_permute_fusion", False), (
        "Multi-LoRA on MoE experts requires moe_permute_fusion=False."
    )
```

评论区精华

1. 注释与文档的克制（style）：yushengsu-thu 与 Zhichenzzz 在多轮 review 中反复要求压缩注释——"remove the comments or reduce to 1 or 2 lines"、"maybe squash those comments?"、"reduce to 1 sentence"。作者用 5f8ca7b、658e1349、8176f5da 三个提交把 docstring 与多行注释收敛到 1-2 行，保留约束性要点、删去推导过程。

2. targets_expert_leaves 的实现方式（design）：Zhichenzzz 建议 "could we define a more intuitive mapping here, instead of search leaves?"；作者在 7d907ac6 改为把每个条目映射到最后一个点分组组件并与 frozenset 比对，回复 "reworked: each entry maps to its leaf name (last dotted component), checked against a frozenset"。
3. get_gloo_group 导入方式（design/style）：Zhichenzzz 问 "should we use lazy imports?"；作者确认无循环依赖后改为顶层导入（"moved to a top-level import; no circular dep"）。
4. 全局变量与命名（style）：_shard_topology 移到文件顶部、my_rank 重命名为 current_rank，均按 review 意见落实。
5. CI 失败 triage（question）：作者在 Issue 评论中详细论证两个 CI 失败均为 cuDNN 环境问题（base image 的 libcudnn9-cuda-13 9.13.0.50 遮蔽 pip cuDNN，nightly 同样失败，由 #1808/#1824 独立报告与根因定位）；失败路径日志显示 multi_lora = False、multi_lora_n_adapters = 0，未触及本 PR 任何代码分支。最终 Zhichenzzz 审批通过。

- 注释与文档冗长，要求压缩 (style): 通过 5f8ca7b、658e1349、8176f5da 三个提交把所有 docstring 与多行注释压缩到 1-2 行或 1 句，保留约束性要点、删除推导过程。
- targets_expert_leaves 的实现方式 (design): 作者改版为把每个条目映射到最后一个点分组件的叶子名并与 frozenset 比对（7d907ac6），回复 "reworked: each entry maps to its leaf name (last dotted component), checked against a frozenset"。
- get_gloo_group 是否使用 lazy import (design): 作者确认顶层导入无循环依赖后改为顶层导入，回复 "moved to a top-level import; no circular dep"。
- 全局变量 _shard_topology 的位置 (style): 作者已将 _shard_topology 移到文件顶部定义。
- 变量命名 my_rank → current_rank (style): 作者回复 "renamed"，已按建议重命名。
- CI 失败是否为本 PR 引入 (question): 确认为环境问题而非本 PR 缺陷，阻塞于 #1824 的镜像修复；作者同时指出该类 RuntimeError 在 CI 中不可重试（tests/ci/ci_utils.py:94-108）。

风险与影响

- 风险：
  1. checkpoint 布局变化（回归风险）：EP > 1 时分片文件名新增 _ep 后缀。已有 ep_size == 1 省略后缀 + find_latest_checkpoint 中 legacy 回退双保险，但混合版本集群或手工改名仍可能误配，发布说明应强调旧 checkpoint 读路径已验证。
  2. gloo all-gather 依赖（运行风险）：adapter_shard_topology 在首次保存 / 恢复时走一次 all-gather，要求所有 rank 同步参与；dist.is_initialized() 已有守卫，但 gloo 组不可用的环境会在启动期首次失败——作为一次性调用，风险可控。
  3. slice_lora_to_rank 行为变更（正确性风险）：2-D dense 张量行为不变，但 3-D/4-D packed 张量此前被静默错切（丢专家），现在会正确裁剪或 assert。对确实存在非零 padding 的异常老 checkpoint，导出会硬失败而非静默丢权重——这是有意的 fail-fast，但可能暴露此前被掩盖的数据问题。
  4. 强制关闭 moe_permute_fusion（性能风险）：仅影响 multi-LoRA 且目标命中专家叶子的运行，代价是 fused permute kernel 的损失；目标不含专家叶子的 multi-LoRA 不受

影响。

5. 启动期断言收紧（兼容性风险）：PP > 1 或 bshd 的既有 multi-LoRA 运行现在会被直接拒绝，而非像之前那样在训练中途暴露错误，需要用户在 CLI 层面适配。
6. 跨仓库版本耦合：正确运行依赖 Megatron-Bridge#23（MultiLoRAGroupedExpertLinear）与 sglang#32376（已合并）；MLA + dp-attention 模型还需 bridge#25 与 sglang#32421，缺任一都表现为运行期失败或策略静默漂移。 - 影响：用户影响：MoE 模型（Qwen3-MoE 等）的 MultiLoRA 训练能力从 " 密层 + 路由层 " 扩展到专家本身，是训练容量覆盖的实质提升；GLM-5.2 / DeepSeek 类 MLA 模型需配合两个额外修复。系统影响：MultiLoRA checkpoint 分片布局在 EP > 1 时变化，但旧布局可加载，影响面收敛在 multi-LoRA 训练路径。团队影响：这是依赖明确 merge 顺序的三仓库协作（megatron-bridge → miles → sglang），后续维护者需同步升级依赖才能获得完整功能，也意味着跨仓库 bug 的定位和修复需要联动。 - 风险标记：checkpoint 布局变更，跨仓库依赖，启动期行为收紧，核心训练路径变更

关联脉络

- PR #23 feat(peft): support multi-LoRA on grouped MoE expert linears: 前置依赖（Megatron-Bridge 仓库）：提供 MultiLoRAGroupedExpertLinear 与 install_moe_slot_routing，本 PR 的验证与分片逻辑建立在其之上，PR body 明确合并顺序为 bridge#23 → 本 PR。
- PR #25 fix(peft): correct dense MultiLoRALinear on replicated base linears: 配套修复（Megatron-Bridge 仓库）：GLM-5.2_5layer e2e 发现，replicated base linear（MLA q/kv down-projection）下 token span 未收窄到 SP shard 且缺失 output gather；MLA 模型（GLM-5/DeepSeek）必需。
- PR #32376 [sglang-miles] Warn instead of silently dropping one-sided MoE expert LoRA targets: 配套 PR（sglang 仓库）：与本 PR 的 " 双侧专家投影校验 " 形成同一条约束的两端——训练侧启动期硬校验，serving 侧日志警告，均已合并。
- PR #32421 [sglang-miles] Allow load_lora_adapter_from_distributed under dp attention: 配套 PR（sglang 仓库）：解除分布式 LoRA 加载端点对 dp_size == 1 的断言，dp-attention serving 场景（GLM-5.2 标准配置）必需，由同一 GLM-5.2 e2e 发现。
- PR #1803 [miles] 作者声明需要的依赖 PR: PR 讨论中作者明确 "Need this: https://github.com/radixark/miles/pull/1803"，是本 PR 功能落地前需要合入的 miles 侧依赖。
- PR #2012 router: enable dp-aware routing under dp-attention: 同一功能线的后续演进（本仓库）：dp-attention 开启时使用 dp-aware 路由，与 sglang#32421 放开的 dp-attention 分布式 LoRA 加载相互配合，共同支撑 dp-attention serving 场景下的多 LoRA 训练闭环。