PR #34940 完整报告

sgl-project/sglang

[Diffusion] Fix H3 swap PEFT SwiGLU lora_B halves when loading FFN Lora

合并时间：2026-08-17 18:47

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34940

执行摘要

- 一句话：修复 MiniMax-H3 加载 LoRA 时 SwiGLU lora_B 顺序颠倒
- 推荐动作：值得精读。虽然只有 17 行改动，但 PR body 展示了高质量的权重布局验证方法论：先用 bit-exact 数值匹配证明两侧布局差异，再以最小化字符串守卫精准修复，最后用生成图像与音频 RMS 确认效果。这种 " 静默错误 + 布局交换 " 的排查思路对多格式权重 / LoRA 加载兼容问题有推广价值。

功能与动机

PR body 指出：MiniMax-H3 融合 SwiGLU 的 *gate/up* 为 (28672, 5376) 矩阵，两种官方布局行序相反（diffusers/LightX2V 的 ff.net.0.proj 为 [up; gate]，sglang 原生 mlp.fc1 为 [gate; up]）。加载 LightX2V LoRA 时已经重命名 ff.net.0.proj → mlp.fc1，却未交换 lora_B 行，形状匹配且不报错，导致 FFN 适配器被两半颠倒地应用。官方仓库同一 DiT 的两种导出（FL2VA/transformer 与 transformer/）证明该行序差异存在于官方权重本身，而非 sglang 独有约定。

实现拆解

变更入口为 python/sglang/multimodal_gen/runtime/pipelines_core/lora_pipeline.py，核心工作分如下几步：

1. 确认行序假设：PR body 通过两类证据锁定问题——官方仓库同一 DiT 的两种导出（FL2VA/transformer 的 mlp.fc1 与 transformer/ 的 ff.net.0.proj）在数值对比中，融合矩阵仅在交换两半后 52/52 逐块 bit-exact 匹配；LightX2V 两次导出的 lora_B 同样仅在交换后 50/50 精确，而 lora_A 与 fc2 天然匹配，说明差异只存在于 lora_B 的行序。
2. 新增守卫函数：在模块级新增 _swap_peft_swiglu_fc1_lora_b(source_name, target_name, weight)。守卫条件为 weight.dim() == 2、source_name 包含 " .ff.net.0.proj.lora_B"、target_name 以 ".mlp.fc1.lora_B" 结尾，三者同时满足才执行 weight.chunk(2, dim=0) 后按 [gate, value] 顺序 cat，否则原样返回。
3. 接入通用加载路径：在 load_lora_adapter 的参数映射循环中，完成 lora_param_names_mapping_fn 与 param_names_mapping_fn 的 key 映射、merge 逻辑之后、写入 self.lora_adapters 之前调用该函数，确保所有经 PEFT 重命名的 FFN LoRA 在落盘前布局已被修正。

4. 隔离影响范围：守卫刻意排除原生融合的 mlp.fc1 key 与其它模型（如 Flux）的 ff.net.0.proj；lora_A（输入侧）与 fc2（输出侧）天然不需要交换，均保持原样。由于函数对所有模型通用但绝大多数路径直接 return，属增量式修正。
5. 验证配套：PR 未新增自动化测试文件，但提供了双维验证——数值上 52/52、50/50 的 bit-exact 匹配，效果上 4 步 /8 步生成对比（毛皮涂抹消失、五官清晰，audio RMS 从 0.0063 升至 0.0097、0.0067 升至 0.0129），并确认延迟与峰值内存不变。

关键文件：

- python/sglang/multimodal_gen/runtime/pipelines_core/lora_pipeline.py（模块 LoRA 加载；类别 source；类型 core-logic；符号 _swap_peft_swiglu_fc1_lora_b）：唯一改动文件。在 LoRAPipeline 的通用加载入口 load_lora_adapter 中新增 _swap_peft_swiglu_fc1_lora_b 守卫函数，精准修复 MiniMax-H3 场景下 PEFT 格式 lora_B 行序颠倒的静默错误，同时通过字符串守卫隔离其它模型与原生 key。

关键符号：_swap_peft_swiglu_fc1_lora_b, load_lora_adapter

关键源码片段

python/sglang/multimodal_gen/runtime/pipelines_core/lora_pipeline.py

唯一改动文件。在 LoRAPipeline 的通用加载入口 load_lora_adapter 中新增 _swap_peft_swiglu_fc1_lora_b 守卫函数，精准修复 MiniMax-H3 场景下 PEFT 格式 lora_B 行序颠倒的静默错误，同时通过字符串守卫隔离其它模型与原生 key。

```
# python/sglang/multimodal_gen/runtime/pipelines_core/lora_pipeline.py

def _swap_peft_swiglu_fc1_lora_b(
    source_name: str, target_name: str, weight: torch.Tensor
) -> torch.Tensor:
    # 仅命中 PEFT -> 原生 H3 FFN 的重写路径：
    # LightX2V / diffusers 的 ff.net.0.proj 按 [up; gate] 存储，
    # 而 sglang 原生 mlp.fc1 期望 [gate; up]，两半顺序恰好相反。
    # 原生融合的 mlp.fc1 以及其它模型的 ff.net.0.proj（如 Flux）
    # 不会同时满足下面两个字符串条件，因此直接原样返回，避免误伤。
    if (
        weight.dim() != 2
        or ".ff.net.0.proj.lora_B" not in source_name
        or not target_name.endswith(".mlp.fc1.lora_B")
    ):
        return weight
    # 按第 0 维将 14336 行切成两半，再以 [gate; up] 顺序拼回；
    # lora_A 作用在输入侧，不需要行交换，fc2 不受影响。
    value, gate = weight.chunk(2, dim=0)
    return torch.cat([gate, value], dim=0)


# load_lora_adapter 中的接入点：完成 key 映射与 merge 后、
# 写入 lora_adapters 之前统一修正布局，保证所有重命名路径都生效。
for name, weight in lora_state_dict.items():
```

```
... # 省略 key 归一化与 merge 分支
weight = _swap_peft_swiglu_fc1_lora_b(name, target_name, weight)
if target_name in self.lora_adapters[lora_nickname]:
    raise ValueError(...)
self.lora_adapters[lora_nickname][target_name] = weight.to(self.device)
```

评论区精华

Review 过程非常简短：mickqian 仅评论 "/tag-and-rerun-ci" 触发带标签的 CI 重跑，随后直接 APPROVED，无 review 评论和未解决疑虑。技术论证几乎全部集中在 PR body 中，包括两种布局的 forward 实现对比、官方双导出的证据链，以及 52/52、50/50 的 bit-exact 数值验证表。

- CI 重跑与快速合入 (other): CI 通过后合入，无未解决的技术疑虑。

风险与影响

- 风险：
  1. 字符串守卫的脆弱性：_swap_peft_swiglu_fc1_lora_b 依赖 source_name 与 target_name 的字符串模式判断，若未来其它模型恰好同时满足两个条件会被误交换；若命名规则调整导致守卫漏判，则该静默错误不会报错，回归隐蔽。
  2. 缺少自动化测试：本次没有配套单测覆盖该路径（如构造 [up; gate] 布局的 lora_B 并断言加载后被交换），后续回归依赖手工验证与 CI 冒烟。
  3. 通用加载路径改动：函数被挂在 load_lora_adapter 的通用循环中，对所有模型的 LoRA 加载都会执行一次守卫判断，虽然多数路径直接 return，仍属核心加载路径的改动，需要回归关注。
  4. 多卡 / merge 分支覆盖不足：weight 在 merge 分支中可能被 stack 成 3 维，dim() != 2 会跳过交换；当前 MiniMax-H3 的 FFN LoRA 不涉及 merge，但该组合未显式验证。
  - 影响：用户侧：修复了使用 LightX2V / diffusers 训练 LoRA 并在 sglang MiniMax-H3 FL2VA 上推理的用户——此前 FFN 适配器静默失效（两半颠倒），修复后生成质量明显提升；Attention LoRA 与原生 mlp.fc1 LoRA 不受影响。 系统侧：仅加载期多一次 chunk + cat 的轻量张量操作，延迟与峰值内存不变；变更局限于 lora_pipeline.py 单文件，无 schema 或配置改动。 团队侧：为 diffusion 权重布局兼容提供了新范例与可复用的验证方法论，与 FP8 权重归一化（#35111）构成同一演进主线。
- 风险标记：字符串守卫启发式，缺少自动化测试，通用加载路径改动

关联脉络

- PR #34988 [Diffusion] Reuse SRT SigLIP vision model: 同属 multimodal_gen runtime 层（diffusion 管线与加载路径）的近期改动，说明该层正在持续收敛与重构。
- PR #35068 [Docs] Feature MiniMax-H3 in the popular-models banner: 同为 MiniMax-H3 相关 PR，MiniMax-H3 是当前 diffusion 主推模型，本 PR 修复其 LoRA 加载正确性。

- PR #35111 [AMD] diffusion: normalize ModelOpt-FP8 weights to e4m3fnuz on gfx942: 同为 diffusion 权重格式适配问题，表明 diffusion 权重布局 / 格式兼容是活跃演进方向。