PR #27798 完整报告

sgl-project/sglang

[AMD] Add transpose_scale arg for o_proj to fix GLM accuracy issue

合并时间：2026-06-17 16:01

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/27798

执行摘要

- 一句话：修复 GLM-5 在 AMD GPU 上的精度问题
- 推荐动作：该 PR 改动小但定位精准，是典型的生产环境精度回退修复案例。值得关注的是其根因分析过程以及如何通过参数透传解决 kernel fusion 遗留问题。对于维护 AMD 后端的工程师有参考价值。

功能与动机

2026-06-10 的 Nightly CI 发现 GLM-5 精度降至接近 0（对应 Issue #99）。PR#27289 融合了 scale-transpose 逐元素 kernel，但遗漏了 o_proj 的 scale 参数传递，导致 GLM-5.1 精度问题。本 PR 目的就是补救这一遗漏。

实现拆解

1. 在 python/sglang/srt/models/deepseek_common/attention_forward_methods/forward_mla.py 的 forward_absorb_core 函数中，定位到两处 fused_flatten_fp8_group_quant 调用（处理 _bmm_buf 和 attn_bmm_output 进入 o_proj 之前）。
2. 为这两处调用新增 transpose_scale=_use_aiter_bpreshuffle_gfx95 关键字参数，确保在 AMD gfx95x 平台上 scale 矩阵被正确转置，从而修复精度。
3. 该参数依赖 aiter 的新版本支持（PR aiter#3041 已合并），因此 PR body 中注明需要先升级 aiter 依赖。
4. 变更仅涉及一个文件，共 8 行新增和 2 行删除。

关键文件：

- python/sglang/srt/models/deepseek_common/attention_forward_methods/forward_mla.py（模块 MLA 注意力；类别 source；类型 data-contract）：核心修复文件，在 forward_absorb_core 函数中两处 fused_flatten_fp8_group_quant 调用新增 transpose_scale 参数，修复 GLM-5 精度问题。

关键符号：forward_absorb_core

关键源码片段

python/sglang/srt/models/deepseek_common/attention_forward_methods/forward_mla.py

核心修复文件，在 forward_absorb_core 函数中两处 fused_flatten_fp8_group_quant 调用新增 transpose_scale 参数，修复 GLM-5 精度问题。

```
# python/sglang/srt/models/deepseek_common/attention_forward_methods/forward_mla.py
# 以下两段代码展示了新增 transpose_scale 参数的关键变更。
# 当 o_proj 权重为 float8_e4m3fn 时，调用 fused_flatten_fp8_group_quant
# 对中间结果进行量化和 flatten，新参数 transpose_scale 控制是否
# 在量化前转置 scale 矩阵，修复 GLM-5 精度问题。

# 第一处：处理 _bmm_buf（当 _bmm_buf 不为 None 时）
elif self.o_proj.weight.dtype == torch.float8_e4m3fn:
    attn_bmm_output = fused_flatten_fp8_group_quant(
        _bmm_buf,
        group_size=128,
        dtype_quant=torch.float8_e4m3fn,
        transpose_scale=_use_aiter_bpreshuffle_gfx95, # 新增参数
    )

# 第二处：处理 attn_bmm_output（当 _bmm_buf 为 None 且未走前面的分支时）
elif self.o_proj.weight.dtype == torch.float8_e4m3fn:
    attn_bmm_output = attn_bmm_output.transpose(0, 1)
    attn_bmm_output = fused_flatten_fp8_group_quant(
        attn_bmm_output,
        group_size=128,
        dtype_quant=torch.float8_e4m3fn,
        transpose_scale=_use_aiter_bpreshuffle_gfx95, # 新增参数
    )
```

评论区精华

Gemini Code Assist 机器人指出一个问题：当 _use_aiter_gfx95 为 False 但权重为 float8 时，进入该分支会导致 NameError，因为 fused_flatten_fp8_group_quant 仅在 _use_aiter_gfx95 为 True 时被导入。作者 1am9trash 回应“只有 gfx950 才会进入 self.o_proj.weight.dtype == torch.float8_e4m3fn: 分支”，表明该分支已在其他逻辑中保证了前提条件，因此无需额外 guard。最终 PR 获得 HaiShaw 批准。

- 分支条件缺少 _use_aiter_gfx95 guard 可能导致 NameError (correctness): 作者回应“只有 gfx950 才会进入该分支”，认为了现有逻辑已保证前提，无需额外修改。

风险与影响

- 风险：该 PR 风险极低：仅新增一个可选参数，逻辑上如果 _use_aiter_bpreshuffle_gfx95 为 False，传递 False 与原先行为一致（等价于不传）。但需警惕：若 aiter 版本未及时升级，用户可能得到错误精度。PR 标记了依赖版本升级前置条件，合并后应确保 CI 中依赖正确更新。
- 影响：影响范围仅限 AMD gfx95x 平台上的 GLM-5 模型推理。修复后，GSM8K 准确率恢复至 v4-pro: 0.950 / GLM-5.1: 0.942。对其他模型和其他硬件无影响。
- 风险标记：依赖外部库（aiter）版本升级

关联脉络

- PR #27289 fuse V4 scale-transpose element-wise kernel into preceding quant kernel: 本 PR 修复了 PR#27289 引入的 o_proj scale 参数遗漏问题。