PR #23335 完整报告

sgl-project/sglang

Fix diffusion fallback guards and validation

合并时间：2026-05-07 00:05

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23335

执行摘要

- 一句话：修复 diffusion 回退路径与形状校验
- 推荐动作：改动干净、测试聚焦，适合快速合入。作为 kernel 防护最佳实践示例值得存档，但无需深入精读。若团队有 NPU 部署或 diffusion 自定义 kernel 开发，建议参考此模式在其他 kernel 中补充类似输入校验。

功能与动机

PR body 明确写出：native diffusion RoPE fallbacks 需要接受全宽交错 cos/sin 缓存，以匹配 Triton 路径；而 CuTe fused scale/shift 对 [B, F, 1, D] 张量的帧整除性校验过于宽松（仅在报错前标记 failed = True 而非直接拒绝），可能导致不合法的形状被静默接受。此 PR 来自内部清理审查，旨在提高 diffusion kernel fallback 的健壮性。

实现拆解

1. NPU 和 Torch fallback 统一处理交错 RoPE：在 npu_fallback.py 和 torch_fallback.py 的 apply_rotary_embedding_native 函数开头添加条件判断——当 interleaved=True 且 cos.shape[-1] == x.shape[-1] 时，将 cos/sin 沿最后一个维度减半（取偶数索引），确保后续拆分逻辑正确。
2. 收紧 CuTe scale/shift 守卫：在 scale_residual_norm_scale_shift.py 的 validate_scale_shift 中，将四维输入的校验逻辑从“统一标记失败再抛异常”改为“先检查前三维形状，若不匹配直接标记失败；仅当形状匹配时才检查 S % F != 0 并立即抛出特定异常”，避免因 failed = True 覆盖更具体的整除性错误信息。
3. 追加聚焦测试：在 test_fused_norm_scale_shift.py 中新增 test_validate_scale_shift_rejects_non_divisible_frames 函数，验证当 S=10, F=4 时 validate_scale_shift 正确抛出包含预期信息的 ValueError。
4. 剔除 CI 配置变动：从分支中移除原本的 diffusion case parser 改动（scripts/ci/utils/diffusion/diffusion_case_parser.py），确保此 PR 不改变 CI 分区或覆盖行为。

关键文件：

- python/sglang/jit_kernel/diffusion/cutedsl/scale_residual_norm_scale_shift.py（模块 CuTe DSL；类别 source；类型 core-logic；符号 validate_scale_shift）：核心校验函数 validate_scale_shift 的逻辑修复：将四维张量的形状检查与整除性检查分离，优先检查前

三维，形状合法时才进一步校验 S % F，使异常信息更准确。

- python/sglang/jit_kernel/diffusion/triton/npu_fallback.py（模块 NPU 回退；类别 source；类型 core-logic；符号 apply_rotary_embedding_native）：NPU 回退函数 apply_rotary_embedding_native 新增对交错全宽 cos/sin 的处理：当 interleaved=True 且 cos 宽度与 x 宽度相等时，将 cos/sin 减半（取偶数索引），以匹配 Triton 路径的拆分格式。
- python/sglang/jit_kernel/tests/diffusion/test_fused_norm_scale_shift.py（模块 单元测试；类别 test；类型 test-coverage；符号 test_validate_scale_shift_rejects_non_divisible_frames）：新增 test_validate_scale_shift_rejects_non_divisible_frames 验收测试，确保非法帧组合被正确拒绝。同时导入 validate_scale_shift 函数。

关键符号：validate_scale_shift, apply_rotary_embedding_native

关键源码片段

python/sglang/jit_kernel/diffusion/cutedsl/scale_residual_norm_scale_shift.py

核心校验函数 validate_scale_shift 的逻辑修复：将四维张量的形状检查与整除性检查分离，优先检查前三维，形状合法时才进一步校验 S % F，使异常信息更准确。

```
# python/sglang/jit_kernel/diffusion/cutedsl/scale_residual_norm_scale_shift.py
# 变更集中在 validate_scale_shift 函数的 4 维分支

def validate_scale_shift(t: torch.Tensor, B: int, S: int, D: int):
    if t.dtype not in (torch.float16, torch.bfloat16, torch.float32):
        raise ValueError(f"Validate failed: unsupported dtype: {t.dtype}")
    failed = False
    if t.ndim == 1 and (t.shape[0] not in (1, D)):
        failed = True
    elif t.ndim == 2 and ((t.shape[0] not in (1, B)) or t.shape[1] != D):
        failed = True
    elif t.ndim == 3 and (
        (t.shape[0] not in (1, B)) or (t.shape[1] not in (1, S) or t.shape[2] != D)
    ):
        failed = True
    elif t.ndim == 4:
        # 先检查前三维 shape，不匹配则标记 failed（报通用错误）
        # 仅当前三维正确时才检查帧整除性并抛出具体异常
        F = t.shape[1]
        if t.shape[0] != B or t.shape[2] != 1 or t.shape[3] != D:
            failed = True
        elif S % F != 0:
            raise ValueError(f"Validate failed: S({S}) must be divisible by F({F}).")
    if failed:
        raise ValueError(f"Validate failed: unsupported tensor shape: {t.shape}.")
    if t.stride()[-1] != 1:
        raise ValueError(f"Validate failed: not contiguous on dim D.")
```

python/sglang/jit_kernel/diffusion/triton/npu_fallback.py

NPU 回退函数 apply_rotary_embedding_native 新增对交错全宽 cos/sin 的处理：当 interleaved=True 且 cos 宽度与 x 宽度相等时，将 cos/sin 减半（取偶数索引），以匹配 Triton 路径的拆分格式。

```
# python/sglang/jit_kernel/diffusion/triton/npu_fallback.py
# 函数开头新增 3 行，处理全宽交错缓存

def apply_rotary_embedding_native(
    x: torch.Tensor, cos: torch.Tensor, sin: torch.Tensor, interleaved: bool = False
) -> torch.Tensor:
    # 如果 cos 和 x 最后一维宽度相同（即 cos 尚未被截半），且需要交错，
    # 则手动取偶数索引，使 cos/sin 变为 half-width，与后续 x[..., ::2] 匹配
    if interleaved and cos.shape[-1] == x.shape[-1]:
        cos = cos[..., ::2]
        sin = sin[..., ::2]
    cos = cos.unsqueeze(-2).to(x.dtype)
    sin = sin.unsqueeze(-2).to(x.dtype)
    # 后续保持不变 ...
```

评论区精华

PR 无实质性 Review 评论，仅由 author @BBuf 多次触发 /tag-and-rerun-ci 以重跑 CI，最终 reviewer @mickqian 给予 APPROVAL。初期分支曾包含 diffusion 用例解析器改动，但在最后commit中移除（commitmessage:"Dropdiffusionparserchangesfromcleanup PR"），确保职责单一。

- CI 重试与 parser 分离 (other): CI 通过后由 @mickqian 直接 approve，无异议。

风险与影响

- 风险：改动的四个文件均为 fallback/valiation 路径，不涉及主 Triton 或 CuTe kernel 执行流。在 npu_fallback.py 和 torch_fallback.py 中新增的 if 分支仅在 interleaved=True 且 cos 宽度与 x 宽度相同时触发，对非交错或宽度不匹配的场景无影响。validate_scale_shift 的调整改变了异常触发顺序，但最终仍会拒绝不合规形状，且新增的测试覆盖了核心场景。主要风险在于 NPU fallback 路径依赖 torch_npu 硬件库，若实际部署中 cos/sin 形状与预期不符可能导致新的错误；但原逻辑未处理该情况，本次修复属于严格化。
- 影响：影响范围限定在使用 diffusion kernel NPU 回退或 CuTe 融合 scale/shift 的用户。[B, F, 1, D] 中 S 不能被 F 整除的非法输入现在会被明确拒绝，避免静默数值错误。RoPE 交错回退行为与 Triton 路径一致化，消除因回退路径不同导致的精度差异。团队层面，此 PR 清理了已知的 validate_scale_shift 逻辑缺陷，并留下一份可复用的守卫测试。
- 风险标记：NPU 硬件依赖，手动 CI 重试，历史分支残留

关联脉络

- PR #27041 [diffusion] Optimize Cosmos3 lossless hot paths: 同属 diffusion kernel 优化方向，修改了相近的 diffusion fallback 模块，体现了对 diffusion 路径的持续完善。
- PR #27023 [diffusion] Optimize LingBot realtime transformer path: 同样优化 diffusion 回退路径（如 RoPE 缓存），与本 PR 修复的 fallback 函数有重叠关注点。