

PR #39366 完整报告

sgl-project/sglang

fix: stop shadowing the DSpark shared-experts fusion guard

合并时间: 2026-09-15 13:24

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/39366>

执行摘要

- 一句话: 删除 DSpark 重复方法定义, 恢复 NPU 共享专家融合守卫
- 推荐动作: 值得精读, 但重点不在改动量 (只删 6 行), 而在于三点: 一是“类体内同名方法后定义者生效”这种极易被多 PR 合并掩盖的静默 bug 模式, 值得作为代码审阅与静态检查的检查项; 二是 `shared_experts_fusion_disable_reason` 这一层“模型类向框架暴露契约”的设计——单个 `classmethod` 返回值就能左右 `num_fused_shared_experts` 与权重加载路径, 任何改动都需要按契约级变更对待; 三是作者主动暴露的验证空白与 CI 维护模式叠加, 提示这类“恢复被遮蔽逻辑”的修复最好附带一个纯 Python 层面的单测 (例如 `monkeypatch _is_npu` 后断言返回值非空, 或断言类上只存在一份定义), 而不是依赖 NPU 实机。若团队内有 NPU 环境, 建议在合并后补跑 `test/registered/unit/models/test_deepseek_v4_shared_expert_fusion.py` 并做一次 DSpark draft 的精度 / 吞吐确认。

功能与动机

PR body 指出这是一个会静默改变 NPU 行为的死代码问题:

`DeepseekV4ForCausalLMDSpark` 定义了两次 `shared_experts_fusion_disable_reason`, 作者明确写道“The second definition wins, so the first one, the only one with the NPU branch, is dead code”。后果是“on NPU, shared-experts fusion is not disabled for the DSpark draft even though the guard was written to disable it”, 而守卫自带的提示语说明原因: “NPU ModelSlim weight loading cannot map shared experts into fused expert slots”, 于是 draft 会在一条不支持该映射的权重加载路径上执行专家融合。作者溯源认为带 NPU 分支的版本来自 PR#33676、无条件版本来自 PR#33312, “the overwrite looks like a merge artifact rather than an intentional revert”, 并且该问题是在审计 DSV4 启动流程时顺带发现的。

实现拆解

1. 锁定问题位置: 改动集中在 `python/sglang/srt/models/deepseek_v4_dspark.py` 的 `DeepseekV4ForCausalLMDSpark` 类体, 类中紧邻出现了两个同名 `@classmethod` `shared_experts_fusion_disable_reason`。Python 类体按顺序执行、后定义者覆盖先定义者, 因此带 `if _is_npu`: 分支、返回禁用原因字符串的那份实现从未被调用。
2. 删除遮蔽定义: 本 PR 只做删除 (0 additions / 6 deletions), 移除第二份无条件 `return DeepseekV4ForCausalLM.shared_experts_fusion_disable_reason(hf_config, quant_config)` 的实现。保留下来的是唯一一份带 NPU 判定的版本: NPU 上返回明确的

disable reason, 非 NPU 平台才回落基类实现。

3. 影响链路和数据契约：该 classmethod 是模型类向框架暴露的“是否禁用 shared experts 融合”契约入口，其返回值决定 `is_shared_experts_fusion_disabled()` 的判定结果，进而决定 `__init__` 中 `self.num_fused_shared_experts = 0 if is_shared_experts_fusion_disabled() else config.n_shared_experts`。守卫恢复后，NPU 上 DSpark draft 的该字段回到 0，不再把 shared experts 映射进 fused expert 槽位；类内其他初始化（`parse_dspark_draft_config`、`gamma`、`target_layer_ids`、`uses_own_vocab_modules = _is_npu`）完全未动。
4. 平台差异与配套改动：非 NPU 平台（CUDA/ROCm）两条定义都会转发到 `DeepseekV4ForCausalLM.shared_experts_fusion_disable_reason`，删除后行为逐字节等价，作者据此判定“Not applicable on CUDA ... behavior is byte-identical”。仓库内没有新增 / 修改测试、配置、文档或部署脚本；PR checklist 的“Add unit tests”一项未勾选。
5. 验证现状：reviewer 试图重跑 `test/registered/unit/models/test_deepseek_v4_shared_expert_fusion.py`，但 CI 处于维护模式、重跑未派发（bot 回复明确“Re-run was not dispatched”），而作者自述没有 NPU 硬件，因此恢复后的 NPU 分支既没有实机验证，也没有对应的单测执行记录。

关键文件：

- `python/sglang/srt/models/deepseek_v4_dspark.py`（模块 模型定义；类别 source；类型 core-logic；符号 `shared_experts_fusion_disable_reason`，`DeepseekV4ForCausalLMDSpark`）：唯一改动文件。删除 `DeepseekV4ForCausalLMDSpark` 类体内被遮蔽的第二份 `shared_experts_fusion_disable_reason`，使带 `_is_npu` 分支的融合禁用守卫重新生效，进而决定 `num_fused_shared_experts` 的取值。

关键符号：`DeepseekV4ForCausalLMDSpark.shared_experts_fusion_disable_reason`，`DeepseekV4ForCausalLMDSpark.init`

关键源码片段

`python/sglang/srt/models/deepseek_v4_dspark.py`

唯一改动文件。删除 `DeepseekV4ForCausalLMDSpark` 类体内被遮蔽的第二份 `shared_experts_fusion_disable_reason`，使带 `_is_npu` 分支的融合禁用守卫重新生效，进而决定 `num_fused_shared_experts` 的取值。

`python/sglang/srt/models/deepseek_v4_dspark.py`（删除遮蔽定义后的形态）

```
class DeepseekV4ForCausalLMDSpark(nn.Module):
    # ModelSlim 的 NPU checkpoint 携带按 QuaRot 对齐、只属于 MTP 的 embedding/head 权重；
    # 原生 CUDA 路径保持原 DSpark 行为，直接复用目标模型的 vocab 模块。
    uses_own_vocab_modules = _is_npu

    # 修复点：类体内原本有两份同名 classmethod，后一份无条件转发基类，
    # 按 Python 类体语义“后定义者生效”，把下面这份带 NPU 分支的实现彻底遮蔽。
    # 删除重复定义后该守卫在 NPU 上重新生效：返回值会被框架用于判定是否禁用
    # shared experts 融合，而不是让调用方继续走到融合路径上。
```

```

@classmethod
def shared_experts_fusion_disable_reason(cls, hf_config, quant_config):
    if _is_npu:
        # NPU ModelSlim 权重加载无法把 shared experts 映射进 fused expert 槽位,
        # 所以这里必须给出 disable reason, 这正是被遮蔽掉的那段分支。
        return (
            "NPU DSpark ModelSlim weight loading does not support mapping "
            "shared experts into fused expert slots."
        )

    # 非 NPU (CUDA/ROCm 等) 仍回落到基类实现: 删除前后结果完全一致,
    # 因此该 PR 对这些平台是零行为变更, 回归风险也集中不到这里。
    return DeepseekV4ForCausalLM.shared_experts_fusion_disable_reason(
        hf_config, quant_config
    )

def __init__(
    self,
    config: DeepSeekV4Config,
    quant_config: Optional[QuantizationConfig] = None,
    prefix: str = "",
) -> None:
    super().__init__()
    self.config = config
    self.quant_config = quant_config
    # 守卫的最终落点: 生效时取 0, 即不把 shared experts 融合进 expert 槽位;
    # 守卫被遮蔽时 NPU 会退回 config.n_shared_experts, 走到不支持的融合路径。
    self.num_fused_shared_experts = (
        0 if is_shared_experts_fusion_disabled() else config.n_shared_experts
    )

    # 其余 DSpark draft 初始化 (markov 校验、gamma、num_stages、
    # target_layer_ids 等) 本次未改动, 此处省略。

```

评论区精华

Review 侧只有一个动作: [mmangkad](#) 给出 APPROVED, 并在 PR 上以评论形式要求 [/rerun-test test/registered/unit/models/test_deepseek_v4_shared_expert_fusion.py](#), 说明 reviewer 认可改动方向, 但希望用既有的 shared-expert fusion 单测把这次恢复的语义盖住。该重跑请求被 CI 维护模式拦下, [github-actions\[bot\]](#) 回复 “CI infrastructure is currently under maintenance … Re-run was not dispatched”, 并强调维护期间禁止合并非 CI 修复类 PR。作者在描述中主动标注了验证空白: “I do not have NPU hardware, so the restored branch is not exercised here. It restores the behavior #33676 intended; an NPU reviewer confirming that intent would be the right check before merge.” 这构成合并时唯一未闭环的点。

- 恢复后的 NPU 分支没有实机验证, 目标单测也未能重跑 (testing): 改动方向被 reviewer 认可并合并, 但 NPU 目标路径在合并时既无实机运行也无单测结果, 属于明确的验证空白,

需要在 CI 恢复后补跑。

- 重复定义是合并产物还是有意回退 (question): 按“合并 artifact”处理, 删除被遮蔽的转发实现以恢复 #33676 的原始意图; reviewer APPROVED 表示接受该判断。
- CUDA 路径零行为变更的论证是否充分 (correctness): 以静态等价性论证代替测试, CUDA/ROCm 侧未追加回归测试; 考虑到两份定义文本确实等价, 该结论风险较低。

风险与影响

- 风险:

1. 验证缺口 (主要风险): 被恢复的 NPU 分支既无实机运行, 也无单测结果, test/registered/unit/models/test_deepseek_v4_shared_expert_fusion.py 的重跑被 CI 维护模式阻断; 如果 #33676 的意图被误读, NPU 上可能出现“融合能力被关掉但调用方另有假设”的新问题。
2. 行为回退面收窄且明确: 改动仅影响 _is_npu 为真的 DSpark draft; CUDA/ROCm 路径两条定义等价, 删除后逐字节一致, 回归风险低。
3. 性能特征变化 (NPU): 恢复禁用融合后, NPU 上 DSpark draft 不再把 shared experts 并入 fused expert 槽位, MoE 计算形态随之变化; 这是守卫本意, 但可能带来吞吐差异, 需要 NPU 侧 benchmark 确认可接受。
4. 复发风险: 同一类体内重复定义同名方法无法被普通代码审阅轻易发现, 本 PR 只删代码、未加防护 (注释、断言或 lint 规则)。后续若有人在类尾再次追加转发实现, 会再次遮蔽 NPU 分支。
5. 无 schema / 部署 / 兼容性影响: 不涉及配置项、API 契约或状态字典变更, 不同版本间的模型加载兼容性不受影响。- 影响: 用户侧: 在 NPU (Ascend) 上部署 DeepSeek V4 DSpark 投机解码的用户会回到“shared experts 不融合”的加载路径, 避免在 ModelSlim 权重布局下走不支持映射的融合路径; CUDA/ROCm 用户完全无感。系统侧: 影响面被限制在单个模型类的 MoE 融合开关, 不涉及调度、KV 缓存或分布式路径。团队侧: 留下一个需要 NPU 维护者确认的语义问题 (是否确实要禁用融合) 以及一条未执行的单测, 后续应由具备 NPU 资源的同学在 CI 恢复后补跑, 并考虑补一个不依赖 NPU 硬件的类属性级测试来防止再次遮蔽。- 风险标记: NPU 分支未实机验证, 目标单测未重跑 (CI 维护模式), 类体重重复定义可能复发, NPU 侧融合关闭或影响吞吐, CUDA 路径无行为变更 (低风险)

关联脉络

- PR #38833 [NPU][CI] Add CANN 9.1.0 and Ascend a5 nightly suites: 同属 NPU + DeepSeek 测试线: 本 PR 恢复的 NPU 分支正需要这类 Ascend 夜间套件才能在真机上被覆盖, 两者叠加说明 NPU 侧验证通路仍是瓶颈。
- PR #39403 [NPU][CI] Fix sglang.test.ascend import failure in multi-node e2e pods: 同为 NPU 测试基础设施修复, 与本 PR 依赖的 NPU 单测 / 实机执行链路 (如 test/registered/unit/models/test_deepseek_v4_shared_expert_fusion.py 的执行环境) 同源。

- PR #38409 [Fix] Wait for PDL before reading DeepSeek V4 K cache locations: 同为 DeepSeek V4 模型的正确性修复，说明 DSV4（含 DSpark draft）启动与推理链路近期持续在清理此类静默正确性问题，本 PR 属于同一波审计产物。