

# PR #34973 完整报告

sgl-project/sglang

[AMD] DSV4: fuse the qk-norm-rope pair on the MTP target-verify path

合并时间: 2026-08-21 14:08

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/34973>

## 执行摘要

- 一句话: MTP verify 融合 qk-norm-rope, 提速约 2% 精度持平
- 推荐动作: 值得精读。核心看点: 1) “只融合一半”的边界分析——为什么 decode 可以融合 store (1-source 消费者) 而 verify 不能 (2-source 消费, 因果掩码来自 store 时机与 index streams); 2) 环境变量默认开启但可回退的渐进式上线方式; 3) review 中关于 unified-KV store 时序的纠错过程, 对理解 DSV4 验证路径的因果机制极有帮助。建议后续补齐 AMD DSV4/MTP target-verify 的自动化回归, 防止未来重构踩中 store 时序陷阱。

## 功能与动机

在 DeepSeek-V4 unified-KV 路径上, 融合的 qk-norm-rope 内核此前只接进 decode。MTP target-verify 每层回退到两次独立 launch (sglang::fused\_q\_norm\_rope 与 sglang::fused\_norm\_rope), 61 层合计每 verify step 122 次 launch。PR body 明确指出不能把 store 一并融合: "Target-verify is 2-source — attention reads history from the ring and takes the current chunk as k/v so the draft tokens stay causally masked against one another. Writing the current chunk into the ring ahead of attention makes those tokens indistinguishable from history... Fusing the store as well drops GSM8K from 0.950 to 0.027 while looking like a ~7% throughput win, because it skips exactly the work correctness depends on." 因此只融合 norm+RoPE 这一半, store 保持原状。

## 实现拆解

1. 环境开关: 在 python/sglang/srt/envron.py 的 Envs 类中新增 SGLANG\_OPT\_FUSED\_QK\_NORM\_ROPE\_VERIFY = EnvBool(True) (默认开启), 置 0 时 verify 路径回退到未融合的双 launch 链路; decode、prefill、非 unified 路径完全不受影响。
2. 门控重构: 在 python/sglang/srt/models/deepseek\_v4.py 的 MQALayer.\_forward\_prepare 中, 把 do\_fused\_store 改名 do\_fused\_qk\_norm\_rope, 并新增 fuse\_verify 判定 (环境开关 && forward\_batch.forward\_mode.is\_target\_verify())。改名回应 review: 门控控制的是“是否走融合 qk-norm-rope 内核”, 而 store 只是 decode 路径的子行为。
3. verify 分支: unified and fuse\_verify 成立时, 先对 kv 做 .contiguous() (它是 qkv\_a 的 strided slice, ring store 要求连续缓冲; 未融合路径在 \_compute\_kv\_bf16 内部付同样拷贝, 属 parity 开销), 再把 swa\_cache/swa\_loc 置 None 交给内核——内核据此只做

q-norm+RoPE 与 kv-norm+RoPE, 不写 ring, 且无需调用 get\_unified\_swa\_loc。

4. 返回契约: verify 路径上 kv 不再被置 None, 原样返回给调用方 forward。调用方既有逻辑 `attn_k = kv` 让 attention 以当前 chunk 为 k/v, `save_kv_cache = kv is not None` 触发后端在 attention 前做因果索引的 ring store, 与未融合路径行为完全一致。
5. 配套验证: 无自动化测试变更; 作者提供 GSM8K 精度实验 (基线 0.950 vs 补丁 0.949/0.945) 与 TP8 吞吐数据 (并发 4~32 下 +1.67%~+2.60%), 并明确说明精度测量未启用 `SGLANG_SIMULATE_ACC_LEN`, 避免绕过 accept/reject 路径。

关键文件:

- `python/sglang/srt/models/deepseek_v4.py` (模块 模型层; 类别 source; 类型 core-logic; 符号 `MQALayer._forward_prepare`): 核心改动文件, 在 `MQALayer._forward_prepare` 中把 fused qk-norm-rope 内核从仅 decode 扩展到 MTP target-verify 路径, 精确控制 store 行为与 kv 返回契约。
- `python/sglang/srt/envron.py` (模块 配置项; 类别 source; 类型 configuration; 符号 `SGLANG_OPT_FUSED_QK_NORM_ROPE_VERIFY`): 新增 `SGLANG_OPT_FUSED_QK_NORM_ROPE_VERIFY` 环境开关 (默认开启), 是 verify 路径默认行为变更的配置入口, 提供回退机制。

关键符号: `MQALayer._forward_prepare`

## 关键源码片段

### `python/sglang/srt/models/deepseek_v4.py`

核心改动文件, 在 `MQALayer._forward_prepare` 中把 fused qk-norm-rope 内核从仅 decode 扩展到 MTP target-verify 路径, 精确控制 store 行为与 kv 返回契约。

```
# 门控与分支: unified-KV 路径下, decode 与 target-verify 共享同一个 fused
# qk-norm-rope 内核; 区别仅在于 verify 不做内核自身的 cache store。
```

```
fuse_verify = (
    envs.SGLANG_OPT_FUSED_QK_NORM_ROPE_VERIFY.get()
    and forward_batch.forward_mode.is_target_verify()
)
do_fused_qk_norm_rope = (unified and (is_decode or fuse_verify)) or (
    not unified and self.use_fused_qk_norm_rope
)
```

```
if do_fused_qk_norm_rope:
```

```
    # 先完成 q 侧投影: gxf95 上走 fused rmsnorm + fp8 quant, 否则走普通 q_norm
```

```
    if _is_gfx95_supported:
```

```
        q_for_wqb, q_lora = _fused_rmsnorm_fp8_quant(
            q_lora, self.q_norm.weight, self.q_norm.variance_epsilon
        )
```

```
        q, _ = self.wq_b(q_for_wqb)
```

```
    else:
```

```
        q_lora = self.q_norm(q_lora)
```

```
        q, _ = self.wq_b(q_lora)
```

```

kv = (
    qkv_a[..., self.q_lora_rank :]
    if qkv_a is not None
    else self.wkv(x_linear)[0]
)

token_to_kv_pool = get_token_to_kv_pool()
if unified and fuse_verify:
    # target-verify 走 unified-KV decode 路径: 后端在 attention 之前通过
    # save_kv_cache = True 把当前 chunk 的 KV 写进 ring, 并用 per-token
    # causal index streams 保证每个 draft query 只 attend 到自身之前的位置,
    # 因果掩码来自这些索引流而非写入时机。因此这里只融合 norm + RoPE,
    # 跳过内核自己的 store, 返回 kv 让既有后端 store 原样执行; swa_loc
    # 只服务于内核 store, 此路径无需计算。
    kv = kv.contiguous() # qkv_a 的 strided slice, ring store 要求 contiguous
    swa_cache, swa_loc = None, None
    swa_page_size, bf16_store = 1, True
elif unified:
    # decode 是单源消费者, ring 持有全部历史, 必须由内核先写 store 再 attention
    swa_cache = token_to_kv_pool.get_unified_kv(self.layer_id)
    swa_loc = attn_backend.get_unified_swa_loc(forward_batch)
    swa_page_size, bf16_store = 1, True
else:
    # 非 unified 路径维持原有 SWA 缓存语义
    swa_cache = token_to_kv_pool.get_swa_raw_buffer(self.layer_id)
    swa_loc = attn_backend.get_swa_out_cache_loc(forward_batch)
    swa_page_size, bf16_store = token_to_kv_pool.swa_kv_pool.page_size, False

q = fused_qk_norm_rope_swa_store(
    q=q,
    kv=kv,
    q_norm_weight=None,
    kv_norm_weight=self.kv_norm.weight,
    q_rms_eps=self.eps,
    kv_rms_eps=self.eps,
    rope_head_dim=self.qk_rope_head_dim,
    cos_cache=self.cos_cache,
    sin_cache=self.sin_cache,
    positions=positions,
    swa_cache=swa_cache,
    swa_loc=swa_loc,
    swa_page_size=swa_page_size,
    q_out=q_out,
    dtype=x.dtype,
    bf16_store=bf16_store,
)
# verify 路径上内核已原地完成 norm + RoPE 且未写 cache, 把 kv 交回调调用方:
# attn_k = kv 让 attention 读到当前 chunk, save_kv_cache = kv is not None
# 触发后端在 attention 前做因果索引的 ring store, 与未融合路径完全一致。

```

```
if not (unified and fuse_verify):  
    kv = None
```

## python/sclang/srt/envIRON.py

新增 SGLANG\_OPT\_FUSED\_QK\_NORM\_ROPE\_VERIFY 环境开关（默认开启），是 verify 路径默认行为变更的配置入口，提供回退机制。

```
# DSV4 Aiter 相关开关  
SGLANG_OPT_USE_AITER_SILU_MUL = EnvBool(False)  
SGLANG_OPT_USE_FUSED_QK_NORM_ROPE = EnvBool(True)  
# unified-KV 此前把 fused qk-norm-rope 内核只接到 decode，MTP target-verify  
# 每层仍以两个独立 kernel 完成 norm + RoPE；置 0 可回退到未融合链路。  
SGLANG_OPT_FUSED_QK_NORM_ROPE_VERIFY = EnvBool(True)  
SGLANG_OPT_USE_AITER_INDEXER = EnvBool(False)
```

## 评论区精华

1. 命名 nit (1am9trash)：do\_fused\_store 在 verify 分支里 store 实际关闭，命名有误导，触发 CHANGES\_REQUESTED。作者回应将门控改名 do\_fused\_qk\_norm\_rope，明确门控控制“是否走融合内核”，store 仅是 decode 路径的子行为，随后获得 APPROVED。
2. cache-timing 纠错 (kkHuang-amd, P2)：作者最初的注释声称 verify 是 post-attention store、写 ring 会让 draft 与历史混淆；kkHuang 指出 unified-KV verify 实际走 decode 路径，save\_kv\_cache=True 让后端在 runtime.decode 之前把所有当前 KV 行写入 ring，由 per-token causal index streams (build\_decode\_streams / verify\_store\_state\_slot) 保证因果性，属于 pre-attention、causally indexed store。作者追踪 store\_swa\_into\_unified 与 runtime.decode 顺序后确认并修正了 verify 块、返回处和调用方注释 (commit 60b7278)。
3. CI 门槛 (kkHuang-amd)：AMD stage-A waiter 失败跳过了相关下游 DSV4/MTP 任务，合并前需当前 head 的非跳过 AMD 自动化信号；人工 GSM8K 与 profiling 证据被认可，inline P2 修正不阻塞 approval。
- do\_fused\_store 命名歧义 (style)：作者将门控改名 do\_fused\_qk\_norm\_rope，明确门控控制“是否走融合 qk-norm-rope 内核”，store 仅是 decode（及非 unified 路径）的子行为，随后获得 APPROVED。
- verify 路径 cache store 时序注释不准确 (correctness)：作者追踪 store\_swa\_into\_unified 与 runtime.decode 顺序后确认是 pre-attention 且 causally indexed，在 60b7278 修正了 verify 块、返回处和调用方注释，避免未来重构误解。
- 合并前 AMD CI 信号缺失 (testing)：作者的人工 GSM8K 与 profiling 数据被认为有效，但自动化 AMD 信号仍缺失；inline P2 修正不阻塞 approval，合并仍需人工确认。

## 风险与影响

- 风险：正确性风险：verify 路径的 store 时序直接决定 draft token 的因果掩码，本 PR 通过“只融合 norm+RoPE、保留后端 causally indexed store”规避了风险（若连 store 一起融合，GSM8K 从 0.950 跌到 0.027，且表现为虚假的 ~7% 吞吐提升），但该路径对时序极其敏感，未来任何对后端 store 顺序或此处门控的重构都可能悄悄破坏 MTP 精度。测试缺

口：本次没有新增自动化测试；AMD CI 的 DSV4/MTP 任务在合并前因 stage-A waiter 失败被跳过，当前 head 的自动化信号缺失，只能依赖人工 GSM8K 与 profiling。默认行为变更：SGLANG\_OPT\_FUSED\_QK\_NORM\_ROPE\_VERIFY 默认开启，target-verify 默认走向融合内核；回退开关存在但需显式设置，用户行为隐含变化。性能与兼容性：

kv.contiguous() 拷贝与未融合路径等价，无新增开销；改动仅在 ROCm +

DSV4unified-KV + MTP target-verify 组合生效，decode、prefill、非 unified 路径不受影响。

- 影响：用户影响：AMD MI355X 上运行 DeepSeek-V4 + EAGLE (4 draft token) 的用户，target-verify 每步 kernel launch 从 122 降到 61，吞吐提升约 1.7%~2.6%，GSM8K 精度持平 (0.950 vs 0.949/0.945，在噪声范围内)。系统影响：verify 路径 GPU 时间下降 (CSA 层 -1.8%、HCA 层 -3.0%，均 excl. collective)，kernel launch 开销显著减少。团队影响：为后续把 fused qk-norm-rope 扩展到更多 forward mode 提供了门控与环境开关范式；同时暴露了 AMD CI stage-A waiter 对 DSV4/MTP 任务覆盖的缺口，需要后续补齐自动化回归。
- 风险标记：正确性敏感路径，缺少测试覆盖，默认行为变更，平台特定 (AMD)，CI 覆盖缺口

## 关联脉络

- PR #35499 [AMD] Improve K3 dspark draft attn kernel perf: 同属 AMD 上 speculative-decoding 注意力路径优化，涉及 verify 注意力与 MLA 后端选择，与本 PR 是同一优化脉络。
- PR #33166 [AMD] DeepSeek-V4 MI355X: eliminate bpreshuffle fp8-scale copies at producer sites (MoE down, MLA o\_proj bmm): 同为 DeepSeek-V4 + MI355X (gfx950) 平台上的性能优化序列，属于同一模型 / 硬件优化线。
- PR #35081 Using unified radix tree by default for all case: 本次修改依赖的 unified-KV 路径属于 unified radix tree / unified cache 体系，该 PR 将其默认化，是本改动的前提基础。