PR #34788 完整报告

sgl-project/sglang

[Fix] Restore layer-level DSV4 RoPE policy

合并时间：2026-08-14 12:26

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34788

执行摘要

- 一句话：恢复 DSV4 层级 RoPE 策略，修复纯 SWA 层误用 YaRN
- 推荐动作：值得精读。该 PR 展示了如何根据官方参考实现修复层粒度配置回归，并通过精心构造的 stub 类实现高性价比的 CPU 单测隔离。对维护 DSV4 系列模型、或需要对齐官方 RoPE 策略的开发者有参考价值，重点关注 original_seq_len 的条件化处理和 active_rope_scaling 的传递方式。

功能与动机

PR body 明确指出：#25144 made original_seq_len unconditional, so pure sliding-window layers started applying YaRN even when compress_ratio == 0。DeepSeek V4 在层粒度选择 RoPE，纯 SWA 层使用主 unscaled RoPE，带压缩器的层对 Q、局部 SWA 分支和压缩 KV 统一使用压缩 YaRN RoPE。该行为需与官方 DeepSeek 推理实现及 HF Transformers 的 DeepSeek V4 实现保持一致。

实现拆解

1. 修正 MqaAttentionBase.__init__ 中的 RoPE 参数构造（python/sglang/srt/models/deepseek_v4.py 约 607-638 行）：将 self.rope_scaling 由共享引用改为 dict 拷贝，避免后续修改污染配置对象；original_seq_len 的默认值改为仅当 self.compress_ratio 非零时取 scaling["original_max_position_embeddings"]，纯 SWA 层（compress_ratio == 0）则置为 0，从而不再误触 YaRN 缩放。
2. 调整 MQALayer.__init__ 的 rotary_emb 构造（约 701-716 行）：引入 active_rope_scaling，仅对 compress_ratio in (4, 128) 的层在 scaling 中写入 rope_type: deepseek_yarn；纯 SWA 层传入 None，保证其 rotary_emb 使用主 RoPE。该变量同时被 Compressor 和 C4 Indexer 复用，确保 C4/C128 层的 Q、SWA 分支与压缩分支使用同一套旋转编码。
3. 简化 Compressor / C4Indexer 的注入：将原 getattr(self, "rotary_emb", None) 改为直接引用 self.rotary_emb，因为现在该属性在 MQALayer 中必定已赋值。
4. 新增 CPU 回归测试（test/registered/unit/models/test_deepseek_v4_rope_policy.py）：用 _ModuleStub、_RMSNormStub、_RoPEConsumerStub、_RotaryEmbeddingStub 隔离依赖，构造 MQALayer 三种配置（压缩比 0/4/128），断言 freqs_cis 与 precompute_freqs_cis 的期望输出一致，并验证 rotary_emb.rope_scaling、Compressor、C4Indexer 的引用关系与 None 情况。

5. 配套验证：除 CPU 单测外，还重跑了 B200/H200 上的 FP4/FP8 *e2e* 与 CP 测试，全部通过。

关键文件：

- python/sglang/srt/models/deepseek_v4.py（模块 模型实现；类别 source；类型 data-contract；符号 MqaAttentionBase.init, MQALayer.init, Compressor, C4Indexer）：核心修复所在。MqaAttentionBase.__init__ 与 MQALayer.__init__ 中的 RoPE 策略被调整为层粒度选择，直接影响所有 DSV4 层的注意力计算正确性。
- test/registered/unit/models/test_deepseek_v4_rope_policy.py（模块 模型测试；类别 test；类型 test-coverage；符号 _ModuleStub, _RMSNormStub, _RoPEConsumerStub, _RotaryEmbeddingStub）：新增的 CPU 回归测试，通过 stub 隔离覆盖纯 SWA、C4、C128 三种层类型的 RoPE 策略，防止 #25144 类回归再次出现。

关键符号：MqaAttentionBase.init, MQALayer.init, precompute_freqs_cis

关键源码片段

python/sglang/srt/models/deepseek_v4.py

核心修复所在。MqaAttentionBase.__init__ 与 MQALayer.__init__ 中的 RoPE 策略被调整为层粒度选择，直接影响所有 DSV4 层的注意力计算正确性。

```
# deepseek_v4.py — MqaAttentionBase.__init__ 中的 RoPE 参数构造（恢复层粒度策略）

from sglang.kernels.ops.attention.deepseek_v4_rope import precompute_freqs_cis

rope_theta, rope_scaling = get_rope_config(config)
# 拷贝一份 scaling，避免后续对 self.rope_scaling 的修改污染共享配置对象
self.rope_scaling = dict(rope_scaling) if rope_scaling else None
scaling = self.rope_scaling or {}

# RoPE 在层粒度选择：纯 SWA 层用主 unscaled RoPE，C4/C128 层用压缩 YaRN RoPE
self.rope_base = (
    config.compress_rope_theta if self.compress_ratio else rope_theta
)
original_seq_len: int = (
    rope_original_seq_len
    if rope_original_seq_len is not None
    else (
        # 只有带压缩器的层才启用 YaRN 的 original_max_position_embeddings
        scaling["original_max_position_embeddings"]
        if self.compress_ratio
        else 0 # 纯 SWA 层不启用 YaRN 缩放，original_seq_len 置 0
    )
)
freqs_cis = precompute_freqs_cis(
    dim=self.qk_rope_head_dim,
    seqlen=config.max_position_embeddings,
    original_seq_len=original_seq_len,
```

```
    base=self.rope_base,
    factor=scaling.get("factor", 1.0),
    beta_fast=scaling.get("beta_fast", 32),
    beta_slow=scaling.get("beta_slow", 1),
)
self.register_buffer("freqs_cis", freqs_cis, persistent=False)

# MQALayer.__init__ 中构造 rotary_emb，仅压缩层启用 deepseek_yarn

active_rope_scaling = None
if self.compress_ratio in (4, 128):
    active_rope_scaling = dict(self.rope_scaling or {})
    active_rope_scaling["rope_type"] = "deepseek_yarn"
self.rotary_emb = get_rope_wrapper(
    head_size=self.rope_head_dim,
    rotary_dim=self.rope_head_dim,
    max_position=config.max_position_embeddings,
    base=self.rope_base,
    rope_scaling=active_rope_scaling, # 纯 SWA 层传入 None，使用主 RoPE
    is_neox_style=False,
    device=get_device().device,
)
```

test/registered/unit/models/test_deepseek_v4_rope_policy.py

新增的 CPU 回归测试，通过 stub 隔离覆盖纯 SWA、C4、C128 三种层类型的 RoPE 策略，防止 #25144 类回归再次出现。

```
# test_deepseek_v4_rope_policy.py — 核心断言：纯 SWA 层与压缩层使用不同 RoPE

def test_pure_swa_layer_uses_unscaled_main_rope(self):
    layer = self._make_layer(0)
    # 主 RoPE：base=10000，original_seq_len=0，不使用 YaRN 缩放
    expected = precompute_freqs_cis(
        dim=64,
        seqlen=128,
        original_seq_len=0,
        base=10_000,
        factor=16.0,
        beta_fast=32,
        beta_slow=1,
    )
    torch.testing.assert_close(layer.freqs_cis, expected)
    self.assertIsNone(layer.rotary_emb.rope_scaling) # 纯 SWA 层无缩放
    self.assertIsNone(layer.compressor)
    self.assertIsNone(layer.indexer)

def test_c4_and_c128_layers_share_yarn_compress_rope(self):
    for compress_ratio in (4, 128):
        with self.subTest(compress_ratio=compress_ratio):
```

```
        layer = self._make_layer(compress_ratio)
        # 压缩层：base=160000，original_seq_len=65536，启用 YaRN
        expected_compressed = precompute_freqs_cis(
            dim=64,
            seqlen=128,
            original_seq_len=65_536,
            base=160_000,
            factor=16.0,
            beta_fast=32,
            beta_slow=1,
        )
        torch.testing.assert_close(layer.freqs_cis, expected_compressed)
        self.assertIs(layer.compressor.freqs_cis, layer.freqs_cis)
        self.assertIs(layer.compressor.rotary_emb, layer.rotary_emb)
        self.assertEqual(
            layer.rotary_emb.rope_scaling["rope_type"], "deepseek_yarn"
        )
        if compress_ratio == 4:
            self.assertIs(layer.indexer.freqs_cis, layer.compressor.freqs_cis)
            self.assertIs(layer.indexer.rotary_emb, layer.compressor.rotary_emb)
```

评论区精华

该 PR 没有实质性的 review 评论；作者在 PR 评论中发起 /rerun-test 请求，机器人确认 CPU 单测、B200 *e2e*（FP4/FP8/CP）和 H200 *e2e* 共 6 项测试全部通过。

- GPU *e2e* 回归测试重跑 (testing): 重跑全部通过：1 个 CPU 单测、3 个 B200 *e2e*、2 个 H200 *e2e* 均绿。

风险与影响

- 风险：变更集中在 DeepSeek V4 注意力的 RoPE 配置路径，属于核心推理逻辑，影响所有使用 DSV4 模型（Flash/Pro）的输出正确性。修复后纯 SWA 层的旋转编码将切换回主 RoPE，与官方对齐，但可能改变此前版本基于错误配置产生的结果，需要模型质量回归确认。rope_scaling 由引用改为每层 dict 拷贝，该对象较小（仅几个 key），开销可忽略。GPU 侧依赖现有 *e2e* 覆盖，B200/H200 已验证；其它硬件（如 NPU/AMD）仅影响其 rotary 包装器的输入，风险较低。
- 影响：影响所有 DeepSeek V4 模型推理的注意力计算正确性，尤其修复了纯 SWA 层的 RoPE 缩放错误。对性能几乎无影响（仅多一次小字典拷贝），对配置兼容性无破坏（compress_rope_theta、original_max_position_embeddings 等字段仍按原语义使用）。测试侧新增了针对层粒度策略的确定性回归测试，后续可防止同类回归。
- 风险标记：核心推理路径变更，影响模型输出正确性，配置语义变更，依赖官方参考实现

关联脉络

- PR #25144 Made original_seq_len unconditional: PR body 明确指出该 PR 将 original_seq_len 无条件化，导致纯 SWA 层误用 YaRN，是本次修复的直接根因。