PR #26877 完整报告

sgl-project/sglang

Fix Mamba2Metadata dropping has_mamba_track_mask

合并时间：2026-06-01 22:04

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26877

执行摘要

- 一句话：修复 Mamba2Metadata 丢失 has_mamba_track_mask 字段
- 推荐动作：值得精读。该 PR 展示了一个典型的"新字段添加但构造方法未同步"的缺陷模式，对维护多构造函数的数据类有警示意义。建议开发者在添加新字段时，同步更新所有 __init__ 调用点。

功能与动机

Fix https://github.com/sgl-project/sglang/pull/15829#issuecomment-4586094495. TestDisaggregationHybridAttentionMambaExtraBuffer.test_gsm8k flakes by dropping ~5pp below the 0.87 threshold. 根因是 MambaAttnBackendBase._track_mamba_state_extend 中的门控 if forward_metadata.has_mamba_track_mask: 读取到陈旧值 False，因为 Mamba2Metadata.prepare_decode / prepare_mixed 在 PR#20522 添加 has_mamba_track_mask 后未更新，新字段静默回退到 dataclass 默认值。

实现拆解

1. 在 python/sglang/srt/layers/attention/mamba/mamba2_metadata.py 的 prepare_decode 方法中，在返回 Mamba2Metadata 对象时添加参数 has_mamba_track_mask=forward_metadata.has_mamba_track_mask（第 179 行）。
2. 在同一个文件的 prepare_mixed 方法中，同样添加参数 has_mamba_track_mask=forward_metadata.has_mamba_track_mask（第 253 行）。
3. 在 test/registered/disaggregation/test_disaggregation_hybrid_attention.py 中，移除 TestDisaggregationHybridAttentionMambaExtraBuffer 类上的 @unittest.skipIf(is_in_ci(), "Temporarily disable the flaky test.") 装饰器，重新启用该测试用例。

关键文件：

- python/sglang/srt/layers/attention/mamba/mamba2_metadata.py（模块 Mamba 注意力；类别 source；类型 core-logic；符号 Mamba2Metadata.prepare_decode, Mamba2Metadata.prepare_mixed）：核心修复文件，在 prepare_decode 和 prepare_mixed 两个构造方法中添加了缺失的 has_mamba_track_mask 字段传递。
- test/registered/disaggregation/test_disaggregation_hybrid_attention.py（模块 解聚测试；类别 test；类型 test-coverage；符号 TestDisaggregationHybridAttentionMambaExtr

aBuffer）：移除 @unittest.skipIf 装饰器，重新启用之前被禁用的 flaky 测试。

关键符号：Mamba2Metadata.prepare_decode, Mamba2Metadata.prepare_mixed

关键源码片段

python/sglang/srt/layers/attention/mamba/mamba2_metadata.py

核心修复文件，在 prepare_decode 和 prepare_mixed 两个构造方法中添加了缺失的 has_mamba_track_mask 字段传递。

```
# 文件：python/sglang/srt/layers/attention/mamba/mamba2_metadata.py
# 变更：在 prepare_decode 和 prepare_mixed 中传递 has_mamba_track_mask

@staticmethod
def prepare_decode(
    forward_metadata: ForwardMetadata,
    seq_lens: torch.Tensor,
    *,
    is_target_verify: bool,
    draft_token_num: int,
) -> "Mamba2Metadata":
    """Decode 路径，在 CUDA 图捕获期间运行。"""
    return Mamba2Metadata(
        query_start_loc=forward_metadata.query_start_loc,
        mamba_cache_indices=forward_metadata.mamba_cache_indices,
        retrieve_next_token=forward_metadata.retrieve_next_token,
        retrieve_next_sibling=forward_metadata.retrieve_next_sibling,
        retrieve_parent_token=forward_metadata.retrieve_parent_token,
        track_conv_indices=forward_metadata.track_conv_indices,
        track_ssm_h_src=forward_metadata.track_ssm_h_src,
        track_ssm_h_dst=forward_metadata.track_ssm_h_dst,
        track_ssm_final_src=forward_metadata.track_ssm_final_src,
        track_ssm_final_dst=forward_metadata.track_ssm_final_dst,
        # 新增行：传递 has_mamba_track_mask，否则默认为 False
        has_mamba_track_mask=forward_metadata.has_mamba_track_mask,
        num_decodes=len(seq_lens),
        num_prefills=0,
        num_prefill_tokens=0,
        is_target_verify=is_target_verify,
        draft_token_num=draft_token_num,
    )

@classmethod
def prepare_mixed(
    cls,
    forward_metadata: ForwardMetadata,
    chunk_size: int,
    forward_batch: ForwardBatch,
) -> "Mamba2Metadata":
```

```
    """Mixed 路径（包含 extend 请求），不能运行 CUDA 图。"""
    # ... 省略中间代码 ...
    return Mamba2Metadata(
        query_start_loc=query_start_loc,
        mamba_cache_indices=forward_metadata.mamba_cache_indices,
        retrieve_next_token=forward_metadata.retrieve_next_token,
        retrieve_next_sibling=forward_metadata.retrieve_next_sibling,
        retrieve_parent_token=forward_metadata.retrieve_parent_token,
        track_conv_indices=forward_metadata.track_conv_indices,
        track_ssm_h_src=forward_metadata.track_ssm_h_src,
        track_ssm_h_dst=forward_metadata.track_ssm_h_dst,
        track_ssm_final_src=forward_metadata.track_ssm_final_src,
        track_ssm_final_dst=forward_metadata.track_ssm_final_dst,
        # 新增行：同样需要传递 has_mamba_track_mask
        has_mamba_track_mask=forward_metadata.has_mamba_track_mask,
        num_prefills=num_prefills,
        num_prefill_tokens=num_prefill_tokens,
        num_decodes=num_decodes,
        is_target_verify=forward_batch.forward_mode.is_target_verify(),
        draft_token_num=draft_token_num,
        mixed_metadata=cls.MixedMetadata(
            has_initial_states=has_initial_states,
            prep_initial_states=prep_initial_states,
            chunk_size=chunk_size,
            seq_idx=seq_idx,
            chunk_indices=chunk_indices,
            chunk_offsets=chunk_offsets,
            extend_seq_lens_cpu=forward_batch.extend_seq_lens_cpu,
        ),
    )
```

test/registered/disaggregation/test_disaggregation_hybrid_attention.py

移除 @unittest.skipIf 装饰器，重新启用之前被禁用的 flaky 测试。

```
# 文件：test/registered/disaggregation/test_disaggregation_hybrid_attention.py
# 变更：移除 TestDisaggregationHybridAttentionMambaExtraBuffer 上的跳过装饰器

# 删除前：
# @unittest.skipIf(is_in_ci(), "Temporarily disable the flaky test.")
class TestDisaggregationHybridAttentionMambaExtraBuffer(PDDisaggregationServerBase):
    @classmethod
    def setUpClass(cls):
        super().setUpClass()
        cls.model = "nvidia/NVIDIA-Nemotron-Nano-9B-v2"
        # ... 其余代码不变 ...
```

评论区精华

无 review 评论。PR 作者 ispobock 通过 issue 评论和多个 /rerun-test 命令验证修复效果，并在 PR body 中提供了 20 轮迭代的准确率数据，证明修复后两个测试类均稳定通过 0.87 阈值。

- 暂无高价值评论线程

风险与影响

- 风险：低风险。变更仅在两处构造函数中添加了一个字段的传递，该字段已存在于 ForwardMetadata 中，且 Mamba2Metadata dataclass 已定义该字段（默认值为 False）。修复后不会破坏现有逻辑，因为之前的错误行为（跳过 SSM 状态跟踪）被纠正，仅影响混合注意力模型的准确率。
- 影响：直接影响范围：修复了 nvidia/NVIDIA-Nemotron-Nano-9B-v2 模型在拆分部署 + extra_buffer 策略下的准确率回退问题。间接影响：所有使用 Mamba2Metadata 的混合注意力模型在 decode 和 mixed 场景下都能正确传递 has_mamba_track_mask，避免 SSM 状态被错误跳过。影响程度较低，仅涉及一个字段的传递。
- 风险标记：低风险，仅字段传递遗漏

关联脉络

- PR #20522 Eliminate hot-path D2H sync by adding has_mamba_track_mask: 引入 has_mamba_track_mask 字段但未更新 Mamba2Metadata 构造方法，是本 PR 修复的根因。
- PR #15829 相关 issue 中描述的 flaky 测试问题：本 PR 修复的 flaky 测试在 issue 讨论中提及。