PR #28267 完整报告

sgl-project/sglang

[NPU] Add causal conv1d

合并时间：2026-08-05 22:22

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/28267

执行摘要

- 一句话：NPU GDN 后端改用原生 causal_conv1d 算子，TTFT 降低 16%
- 推荐动作：值得精读，尤其是 _get_conv_weights_t 的缓存设计和 run_mode 区分 decode/prefill 的接入方式。对要维护 NPU 后端的同学，建议关注三点：一是权重缓存与权重重载的失效机制；二是 activation_mode 硬编码是否应改为从 layer.activation 映射；三是补齐针对 conv states 更新的单测。整体是一份高质量的单文件 kernel 替换 PR，设计收敛、性能收益明确。

功能与动机

PR body 明确说明目标是“apply custom causal_conv1d on npu”，并依赖 sgl-kernel-npu 的 PR#592 提供 NPU 上的自定义 causal_conv1d 实现；原先的 sgl_kernel_npu 通用实现需要在 decode 时对 conv_states 做 transpose + clone，并在多个路径重复构造转置权重，带来不必要的显存开销。issue 评论中 Hexq0210 要求补充 qwen3-next 的准确性与性能测试（由后续 CI 覆盖）。

实现拆解

实现按以下步骤推进：

1. 统一算子入口：在 python/sglang/srt/hardware_backend/npu/attention/ascend_gdn_backend.py 中删除对 sgl_kernel_npu.mamba.causal_conv1d 三个函数的 import 与模块级别名（causal_conv1d_fn、causal_conv1d_update），全部改为调用 torch.ops.npu.causal_conv1d，按 run_mode（1 表示 decode/verify 更新，0 表示 prefill）区分语义；activation_mode=1 固定为 silu 激活。
2. 新增权重缓存：新增 _get_conv_weights_t(layer) 方法，首次调用时将 layer.conv_weights.transpose(0, 1).contiguous() 的结果挂到 layer._conv_weights_t 上，后续 decode / extend 复用，消除每步重复 transpose+contiguous 的开销。
3. 调整 conv states 布局处理：原 decode 路径先 conv_states.transpose(1, 2).clone() 再写回；新算子直接在原始布局上更新，省去一次 clone 与写回；prefill 路径也由原来的 transpose+contiguous 调整为直接对 conv_states[:, -(kernel_size - 1):, :] 切片做 contiguous 后传入。
4. 修正 track 掩码写入：在 mamba_track_mask 存在时，mixed_qkv_to_track 与 conv_states 的索引方式由带 transpose 的写法改为按新布局直接索引，保证 prefix

cache 场景下状态同步正确。

5. 配套说明：本 PR 未包含测试文件变更，准确性与性能数据仅在 PR body 中给出；外部依赖 sgl-kernel-npu PR#592 需先合入才能运行。

关键文件：

- python/sglang/srt/hardware_backend/npu/attention/ascend_gdn_backend.py（模块 NPU 后端；类别 source；类型 core-logic；符号 _get_conv_weights_t, forward_decode, forward_extend）：唯一变更文件，GDN 注意力后端在 NPU 上的核心路径：将 decode、extend verify、prefill 三条路径的 causal conv1d 全部切换为 torch.ops.npu.causal_conv1d，并新增 _get_conv_weights_t 权重缓存。

关键符号：_get_conv_weights_t, forward_decode, forward_extend

关键源码片段

python/sglang/srt/hardware_backend/npu/attention/ascend_gdn_backend.py

唯一变更文件，GDN 注意力后端在 NPU 上的核心路径：将 decode、extend verify、prefill 三条路径的 causal conv1d 全部切换为 torch.ops.npu.causal_conv1d，并新增 _get_conv_weights_t 权重缓存。

```
# ascend_gdn_backend.py：NPU GDN 后端核心改造片段

def _get_conv_weights_t(self, layer: RadixLinearAttention) -> torch.Tensor:
    # 将转置后的权重缓存到 layer 对象上，避免 decode 热路径中
    # 每次 forward 都执行 transpose + contiguous（会产生新显存分配）
    w = getattr(layer, "_conv_weights_t", None)
    if w is None:
        # NPU 自定义算子要求权重布局为转置后连续，转置顺序与
        # 原 sgl_kernel_npu 实现保持一致
        w = layer.conv_weights.transpose(0, 1).contiguous()
        layer._conv_weights_t = w
    return w

def forward_decode(self, layer, forward_batch, mixed_qkv, a, b, **kwargs):
    layer_cache = self.req_to_token_pool.mamba2_layer_cache(layer.layer_id)
    conv_states = layer_cache.conv[0]
    # 原生算子直接维护 conv_states 布局，省去旧实现中的
    # transpose(1, 2).clone() 与写回操作
    mixed_qkv = torch.ops.npu.causal_conv1d(
        mixed_qkv,
        self._get_conv_weights_t(layer),
        conv_states=conv_states,
        bias=layer.bias,
        query_start_loc=query_start_loc,
        cache_indices=cache_indices,
        activation_mode=1, # 固定 silu，原实现透传 layer.activation
        pad_slot_id=-1,
```

```
            run_mode=1, # 1 = decode 更新模式
        )
        query, key, value = torch.split(
            mixed_qkv, [layer.q_dim, layer.k_dim, layer.v_dim], dim=-1
        )
        # 后续交给 kernel_dispatcher.decode 完成 GDN 核心注意力计算
        return core_attn_out

    # extend prefill 分支（节选）：run_mode=0 对应 prefill 模式，
    # 直接对 conv_states 尾部窗口切片，不需要再整体 transpose
    kernel_size = layer.conv_weights.shape[-1]
    conv_states_for_prefill = conv_states[:, -(kernel_size - 1):, :].contiguous()
    mixed_qkv = torch.ops.npu.causal_conv1d(
        mixed_qkv,
        self._get_conv_weights_t(layer),
        conv_states=conv_states_for_prefill,
        bias=layer.bias,
        query_start_loc=query_start_loc,
        cache_indices=cache_indices,
        has_initial_state=has_initial_states,
        activation_mode=1,
        pad_slot_id=-1,
        run_mode=0, # 0 = prefill 模式
    )
    conv_states[:, -(kernel_size - 1):, :] = conv_states_for_prefill
```

评论区精华

review 中最有价值的交锋集中在性能与正确性：

- gemini-code-assist[bot] 多次提出：在 decode 与 extend 的每个 forward 中都调用 layer.conv_weights.transpose(0, 1).contiguous() 会在热路径引入大量显存分配与拷贝，建议把转置后权重缓存在 layer 对象上。该建议被采纳，最终实现了 _get_conv_weights_t 缓存。
- gemini-code-assist[bot] 指出 torch.fx.node.has_side_effect 只是查询函数，不能完成副作用注册，需要把 op 加入 torch.fx.node._side_effectful_ops。该讨论针对 npu/utils.py 的 _mark_op_side_effectful，但最终合入版本未包含 utils.py 改动，此建议未落地。
- iridiumine 提问“Why not just import causal_conv1d?”以及“Why is contiguous used here? Is it necessary?”，作者 zhaozx-cn 回应“it is necessary for this op.”，说明 contiguous 是 NPU 算子对输入布局的硬性要求。
- 另有清理未使用 import（dataclass、List）的 medium 级建议。
- decode 热路径重复 transpose+contiguous 的开销 (performance): 作者采纳建议，新增 _get_conv_weights_t 并在首次调用后缓存为 layer._conv_weights_t。
- _mark_op_side_effectful 的副作用注册方式 (correctness): 该评论针对 npu/utils.py 中新增的辅助函数，但最终合入版本未包含 utils.py 改动，此建议未落地。
- contiguous 是否必要 (question): 确认 contiguous 为算子必需，保持现状。

- 未使用 import 清理 (style): 该建议属于代码整洁性调整，合入版本已清理 import（最终仅保留需要的类型导入）。

风险与影响

- 风险：
  1. 激活函数硬编码：新代码在 *decode*、*extend* 的 verify 与 prefill 三条路径都固定 activation_mode=1，而原实现透传 layer.activation。若未来接入非 silu 激活的 GDN 模型，会静默产生错误结果，缺少显式校验。
  2. conv weights 缓存失效风险：_get_conv_weights_t 把转置权重缓存在 layer 对象上，若权重在运行中被重新加载或 offload/swap，缓存不会自动失效，可能导致 kernel 使用陈旧权重。
  3. 缺少单元测试：PR 变更未附带任何测试文件，conv states 布局变化与 track 掩码索引调整的正确性只靠 PR body 中的 CEVAL 数据间接验证，回归风险较高。
  4. 外部依赖未合入：依赖 sgl-kernel-npu PR#592，若该 kernel 行为变化或未发布，本 PR 在 NPU 上会直接失败。
  5. run_mode / pad_slot_id 硬编码：run_mode=0/1 与 pad_slot_id=-1 的含义依赖 NPU 算子协议，若算子接口演进需要同步修改。 - 影响：影响范围为 Ascend NPU 硬件后端上的 GDN 线性注意力模型（如 Qwen3.5 系列）的 prefill、*decode*、MTP verify 三条路径。性能收益显著：Qwen3.5-35B-A3B Prefill TTFT 降低约 16%、*Decode* TPOT 降低约 11.4%，397B 模型 CEVAL 0.922 与官方 0.93 基本持平。对团队而言，该改法确立了 NPU 后端优先使用 torch.ops.npu 原生算子的技术路线，后续其他后端（如 *Mamba*）可以复用同一种接入模式；但需要注意激活函数与状态的布局约定是 NPU 特有的，通用后端无法直接共享。 - 风险标记：缺少测试覆盖，激活函数硬编码，依赖外部 kernel 未合入，conv 权重缓存失效风险

关联脉络

- PR #29027 [NPU] Adding a fast layernorm for diffusion models and fix BSA: 同属 NPU 硬件后端接入自定义 kernel 的演进方向，展示 NPU 后端 kernel 本地化与性能优化的一贯策略。
- PR #33523 [npu] [bugfix] Fix PD-disaggregation error: 同为 NPU 后端关键路径修复，说明 NPU 后端活跃维护，存在生态关联。
- PR #33607 [ci] add qwen 3.5 mtp + replayssm + flashinfer gdn test: 为 Qwen3.5 MTP 与 GDN 相关路径补充端到端测试，与本 PR 覆盖的模型与算子路径高度相关。
- PR #31865 [XPU] DeepSeek V4: use sgl-kernel-xpu implemetation of flash_mla_sparse_fwd for prefill: 硬件后端（XPU）用专用 kernel 替换通用实现的同类实践，体现跨 NPU/XPU 的 kernel 化趋势。