PR #33023 完整报告

sgl-project/sglang

feat(inkling): migrate short convs onto the ShortConv attention backend

合并时间：2026-08-01 02:52

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/33023

执行摘要

- 一句话：Inkling 短卷积迁移到注意力后端，消除每步 4 倍元数据冗余
- 推荐动作：值得精读。重点看 3 个设计决策：① in-graph / out-graph hook 拆分背后的 decode 延迟悬崖（约 7%）；② 图静态缓冲 " 一次性分配、grow-only、切片 + assert" 的地址稳定契约；③ 用 AST 全树扫描守护跨硬件 hook 签名的测试思路。若你维护 mamba 系模型（Kimi-Linear、GDN、ZAYA1 等），本 PR 的 sidecar 迁移模式可直接复用；若维护 NPU 或其他硬件后端，注意 update_mamba_state_after_mtp_verify 的 req_pool_indices 已是契约参数。

功能与动机

动机来自 #29867 的 sidecar 架构未覆盖 Inkling：ShortConvolution 仍直接访问 pool，且所有权 keyed on layer_id == 0，而一个 Inkling decoder 层持有 4 个 ShortConvolution 模块（k_sconv、v_sconv、attn_sconv、mlp_sconv），导致同一套元数据准备每步重复执行4次。PR body 量化了收益：decode step 3342 vs 3561 kernels，_fused_decode_metadata_kernel 28→7、_fused_extend_metadata_kernel 20→5。另一个关键动机是 hook 放置的性能悬崖：元数据准备应放进 init_forward_metadata_in_graph 被 CUDA graph 录制并免费重放，放图外实测掉约 7% decode 吞吐。

实现拆解

1. 新增 backend sidecar：python/sglang/srt/layers/attention/linear/inkling_sconv_backend.py（+572 行）新增 InklingShortConvAttnBackend 与 InklingShortConvMetadata。后端单点拥有每步 conv 元数据：torch.index_select 直读 req_index_to_mamba_index_mapping 做 slot gather（无 PAD 哨兵，依赖 MambaSlotAllocator.clear 保留 slot 0）、_slot_gather_recordable 区分普通表查表与统一内存池的 translate（后者不能录进图）、fused_decode_sconv_metadata / fused_extend_sconv_metadata 一次 launch 产出 query_start_loc / has_initial_state / (cache_mask, safe_idx, cu, si)、extend 的 prefix-cache track_conv_indices。__init__ 中 _alloc_graph_buffers 按配置的捕获形状一次性分配图静态缓冲，_graph_metadata_out 只切片 + assert，init_cuda_graph_state 立即触发越界检查，避免 prefill 图先捕获后 decode runner 重分配导致地址失效。
2. 模型侧瘦身并移除 4 倍冗余：python/sglang/srt/models/inkling_common/sconv.py（+61/-342）删除模块级 _metadata_cache 与 _owns_extend_metadata、

_prepare_extend_common_metadata、_unfused_extend_common_metadata、_prepare_extend_sconv_metadata、_prepare_decode_sconv_metadata 等全部自管逻辑；ShortConvolution 改为 get_attn_backend().conv_state_metadata(self.layer_id,forward_batch) 纯读取句柄，_sconv_cache / _weight_2d 变成纯视图，同一步内 4 个卷积模块共享同一份元数据。

3. kernel 输出契约扩展：python/sglang/srt/models/inkling_common/kernels/sconv.py（+69/-20）新增 SconvMetadataOut 与 _metadata_out，fused_decode_sconv_metadata / fused_extend_sconv_metadata 增加 out= 参数以写入预分配静态目标；track 索引缓冲的 padded tail 必须清零（旧实现用 batch token 数 clamp 整个 padded 区间，stale 行会索引到当前 token 缓冲之外）。
4. 投机解码与注册接线：attention_registry.py 的 attn_backend_wrapper 对 Inkling 返回 InklingShortConvHybridAttnBackend(full, sidecar, full_attention_layer_ids)，并新增 attn_backend_wrapper_for_draft_extend 给 MTP 草稿的 draft-extend 后端同样挂 sidecar（其他模型 no-op）；draft_utils.py 对多步 draft decode 容器前置拒绝（_assert_draft_needs_no_conv_sidecar，指向 --enable-multi-layer-eagle）；MTP verify 提交从 inkling.py 的 update_conv_state_after_mtp_verify（-32 行）迁入 backend 的 commit_conv_state_after_mtp_verify，spec_utils.py / dflash_worker_v2.py 统一走 backend hook；NPU ascend_hybrid_linear_attn_backend.py 加宽 update_mamba_state_after_mtp_verify 签名接受 req_pool_indices。
5. 测试与回归防护：新增 test_inkling_sconv_metadata_once.py（每步只解析一次、图目标地址稳定、MTP commit 使用传入的 req_pool_indices）；test_ngram_mamba_verify_update.py 增加 TestMtpVerifyHookSignature，用 AST 全树扫描所有 update_mamba_state_after_mtp_verify 覆盖，防止任何硬件后端 hook 漏参；short_conv_backend.py 基类缓冲改 grow-only + assert，并跑通 run_suite（1829 个注册测试）。

关键文件：

- python/sglang/srt/layers/attention/linear/inkling_sconv_backend.py（模块 卷积后端；类别 source；类型 core-logic；符号 InklingShortConvMetadata, InklingShortConvAttnBackend, init, _alloc_graph_buffers）：新增的核心后端：单点拥有每步 conv 元数据、图静态缓冲分配与 in-graph/out-graph hook 拆分，是本 PR 全部设计决策的载体。
- python/sglang/srt/models/inkling_common/sconv.py（模块 模型卷积；类别 source；类型 data-contract；符号 _conv_state, _sconv_cache, _weight_2d, _prepare_extend_sconv_cache）：模型侧 342 行删除的重构核心：ShortConvolution 不再碰 pool，改为纯读 backend 句柄，4 倍冗余在此消除。
- python/sglang/srt/models/inkling_common/kernels/sconv.py（模块 卷积内核；类别 source；类型 data-contract；符号 SconvMetadataOut, _metadata_out, fused_decode_sconv_metadata, fused_extend_sconv_metadata）：kernel 侧输出契约扩展：SconvMetadataOut / _metadata_out 让 fused 元数据 kernel 可以写进图静态目标，是地址稳定契约的落点。

- python/sglang/srt/layers/attention/attention_registry.py（模块 后端注册；类别 source；类型 core-logic；符号 attn_backend_wrapper_for_draft_extend, attn_backend_wrapper）：接线点：attn_backend_wrapper 对 Inkling 装配 sidecar，新增 draft-extend 专用包装入口。
- python/sglang/srt/layers/attention/linear/short_conv_backend.py（模块 基类后端；类别 source；类型 core-logic；符号 _alloc_cache_indices_buf, _refresh_cache_indices, cache_indices_dtype, needs_extend_host_mirrors）：基类契约扩展：cache_indices_dtype / needs_extend_host_mirrors 可配置，cache 索引缓冲改 grow-only 并断言先于图捕获就位，影响所有 ShortConv 使用者。
- python/sglang/srt/speculative/draft_utils.py（模块 草稿解码；类别 source；类型 dependency-wiring；符号 _assert_draft_needs_no_conv_sidecar, create_decode_backend, create_draft_extend_backend）：投机解码配套：多步 draft decode 对含卷积的 Inkling 前置拒绝，draft-extend 后端挂上 sidecar。
- test/registered/unit/models/test_inkling_sconv_metadata_once.py（模块 回归测试；类别 test；类型 test-coverage；符号 TestInklingSconvMetadataOnce, TestInklingMtpVerifyCommit, _MockMambaPool, _MockReqToTokenPool）：核心回归测试：每步只解析一次、图目标地址稳定、MTP commit 使用传入的 req_pool_indices，直接钉住本 PR 的两大契约。
- test/registered/unit/spec/test_ngram_mamba_verify_update.py（模块 契约测试；类别 test；类型 test-coverage；符号 TestMtpVerifyHookSignature, test_all_overrides_accept_the_call_kwargs）：AST 全树扫描测试 TestMtpVerifyHookSignature：守护所有硬件后端 hook 签名，是本次 review 事故沉淀的通用性防护。

关键符号：InklingShortConvAttnBackend.init, InklingShortConvAttnBackend._alloc_graph_buffers, InklingShortConvAttnBackend.init_forward_metadata_in_graph, InklingShortConvAttnBackend.init_forward_metadata_out_graph, InklingShortConvAttnBackend.conv_state_metadata, InklingShortConvAttnBackend.commit_conv_state_after_mtp_verify, ShortConvolution._conv_state, ShortConvolution._update_sconv_cache_for_draft_extend, fused_decode_sconv_metadata, fused_extend_sconv_metadata, _metadata_out, attn_backend_wrapper_for_draft_extend, _assert_draft_needs_no_conv_sidecar, TestMtpVerifyHookSignature.test_all_overrides_accept_the_call_kwargs

关键源码片段

python/sglang/srt/models/inkling_common/sconv.py

模型侧 342 行删除的重构核心：ShortConvolution 不再碰 pool，改为纯读 backend 句柄，4 倍冗余在此消除。

```
    def _conv_state(self, forward_batch: ForwardBatch):
        """当前 step 中本层的卷积状态句柄。

        InklingShortConvAttnBackend 在元数据准备阶段一次性解析出整套
```

```
        step 全局元数据，这里只是纯读取，同一步内 4 个卷积模块共享同一份；
        旧实现以 layer_id == 0 为所有权键，整套准备每步重复执行 4 次。
        """
        return get_attn_backend().conv_state_metadata(self.layer_id, forward_batch)

    def _sconv_cache(self, meta) -> torch.Tensor:
        # 从 backend 句柄里取本 SconvType（k/v/attn/mlp 之一）的池视图。
        return meta.layer_cache.conv[self.sconv_type.value]

    def _weight_2d(self) -> torch.Tensor:
        # 把 [d, 1, w] 参数重排成 Triton kernel 需要的 [d, w]。
        return rearrange(self.weight, "d 1 w -> d w")

    def _update_sconv_cache_for_draft_extend(
        self, forward_batch, sconv_cache, cache_indices, hidden_states
    ):
        """按接受的 token 数写回正确的卷积窗口。

        DRAFT_EXTEND_V2 中草稿模型把全部 num_draft_tokens 过了一遍 sconv，
        但只有 num_accept_tokens 个 token 应反映到最终 conv 状态；这里重建
        恰好接受 num_accept_tokens 个 token 后的滑动窗口并写进缓存，
        替代常规的 update_sconv_cache 调用。若开启 mamba 持久缓存且接受区间
        跨过 mamba_track_interval 边界，还会把该边界处的 conv 状态
        写入持久 ping-pong 缓存（mamba_track_indices）。
        """
        num_accept_tokens = forward_batch.spec_info.num_accept_tokens
        batch_size = forward_batch.batch_size
        # ...（后续按接受数重建窗口并 scatter/persist）
```

python/sglang/srt/models/inkling_common/kernels/sconv.py

kernel 侧输出契约扩展：SconvMetadataOut / _metadata_out 让 fused 元数据 kernel 可以写进图静态目标，是地址稳定契约的落点。

```
class SconvMetadataOut(TypedDict):
    """预分配的输出目标，供 fused 元数据 kernel 写入。

    需要地址跨 CUDA graph replay 稳定的调用方，把已按本 step 的 B / T 切片
    的静态缓冲传进来，kernel 直接写这些地址而不是重新分配。
    """

    query_start_loc: torch.Tensor # [B + 1] int32
    has_initial_state: torch.Tensor # [B] bool
    cache_mask: torch.Tensor # [B, 1, 1] bool
    safe_idx: torch.Tensor # [B] int64
    cu: torch.Tensor # [B + 1] int64
    si: torch.Tensor # [T] int32


def _metadata_out(
```

```
    out: "SconvMetadataOut | None", *, B: int, T: int, device: torch.device
) -> SconvMetadataOut:
    """元数据目标：要么现分配，要么对预分配缓冲做形状校验后复用。

    CUDA graph 路径必须传 out：被捕获内核持有这些地址，replay 时靠图内
    回填保持新鲜；临时分配会让捕获节点读写死地址。
    """
    spec = (
        ("query_start_loc", (B + 1,), torch.int32),
        ("has_initial_state", (B,), torch.bool),
        ("cache_mask", (B, 1, 1), torch.bool),
        ("safe_idx", (B,), torch.int64),
        ("cu", (B + 1,), torch.int64),
        ("si", (T,), torch.int32),
    )
    if out is None:
        # eager 路径没有捕获约束，直接分配即可。
        return SconvMetadataOut(
            **{name: torch.empty(shape, dtype=dtype, device=device)
               for name, shape, dtype in spec}
        )
    for name, shape, dtype in spec:
        t = out[name]
        assert (
            tuple(t.shape) == shape and t.dtype == dtype and t.is_contiguous()
        ), f"{name}: got {tuple(t.shape)}/{t.dtype}, want {shape}/{dtype} contiguous"
    return out
```

评论区精华

核心交锋集中在 Codex 自动 review 提出的 5 个 P1 与作者的处置：

- Ascend 签名缺失（真 bug，已修复）：Codex 指出 spec_utils.py 以关键字 req_pool_indices 调用验证 hook，而 AscendHybridLinearAttnBackend 的 override 未接收该参数，会在提交任何 mamba 状态前 TypeError。ch-wan 承认根因——"my sweep for the hook's definitions was grep ... | grep -v hardware_backend, which filtered out precisely the file that broke"，修复方式为加宽 Ascend override 接受并忽略该参数，并新增 AST 全树扫描测试 TestMtpVerifyHookSignature 兜底。
- 多步 draft decode 未挂 sidecar（真但窄，已前置拒绝）：Codex 指出 create_decode_backend() 返回容器而非 AttentionBackend，sidecar 无法挂载，首个内层 forward 会 AttributeError。ch-wan 判定不可达——所有发布的 Inkling 检查点都设置 mtp_local_layer_ids，banded MTP head 已先抛 NotImplementedError——因此改为 _assert_draft_needs_no_conv_sidecar 前置拒绝并提示 --enable-multi-layer-eagle，另加 test_inkling_draft_backend_guard 回归。
- 填充槽位掩码（false positive）：Codex 担心 decode 图补齐 batch 时 padded req_pool_indices 经 index 0 解析可能污染真实请求。ch-wan 反驳：MambaSlotAllocator.clear 保留 slot 0 作为 dummy 写入目标，零填充的填充行恰好落进

slot 0，且 torch.index_select 直出无需 PAD 哨兵，并已在代码注释记录缘由避免再争论。

- breakable prefill 与 writable metadata（未在可见评论中单独确认）：Codex 另提两个 P1——breakable prefill 捕获前后若 on_graph_path=False 会重新分配张量导致捕获节点读死地址；eager idle batch 时 EagerRunner._execute_idle 置 forward_metadata = None 可能撞上 getter-only property。从最终 head 代码看，init_forward_metadata_out_graph 对非录制阶段已统一走 on_graph_path=True 的静态缓冲路径，且 forward_metadata 仍是普通属性赋值；但这两项的具体结论未在可见讨论中完整记录。
  - Ascend NPU 的 MTP verify hook 签名缺失 req_pool_indices (correctness): 加宽 Ascend override 接受并忽略该参数；并新增 AST 全树扫描测试 TestMtpVerifyHookSignature 防止任何后端再次漏参。
  - Inkling 多步 draft decode 后端未装配 conv sidecar (correctness): 在 DraftBackendFactory.create_decode_backend 增加 _assert_draft_needs_no_conv_sidecar 前置拒绝并提示 --enable-multi-layer-eagle，另加 test_inkling_draft_backend_guard 回归。
  - 图回放时填充请求槽位是否需要掩码 (correctness): 判定为 false positive，代码注释记录原因避免再次争论。
  - breakable prefill 图的元数据地址稳定性 (correctness): 该 finding 的单独回应未在可见评论中完整记录；从最终 head 代码看，init_forward_metadata_out_graph 对非录制阶段已统一走 on_graph_path=True 的静态缓冲路径，建议以最终代码为准。
  - delegated forward_metadata 的可写性 (correctness): 未在可见评论中单独回应；从最终 head 代码看，forward_metadata 在 _prepare_slot_indices 中仍是普通实例属性赋值，未出现 getter-only 限制。

风险与影响

- 风险：
  1. CUDA graph 地址稳定契约：short_conv_backend.py 的 _alloc_cache_indices_buf 从可重分配改为 grow-only + assert，_graph_metadata_out 越界即断言。任何未来改动若违反 " 缓冲先于任何图捕获就位 "，会复现 illegal memory access；已有 test_graph_destinations_are_address_stable 覆盖。
  2. 基类契约变化影响其他 ShortConv 使用者：cache_indices_dtype（默认 int64，Inkling 用 int32，省每层窄化 cast）与 needs_extend_host_mirrors（默认 True，Inkling 用 False，省 device->host 同步）是基类新增的可配置点，ZAYA1 等同样基于 ShortConvAttnBackend 的模型需要回归其 extend 主机镜像路径。
  3. MTP verify hook 契约扩散：req_pool_indices 已成为 update_mamba_state_after_mtp_verify 的约定参数，NPU / DFlash 路径均已加宽；未来新硬件后端若漏参会在 verify 时 TypeError，AST 测试可兜底但依赖测试被执行。
  4. 未覆盖路径：NVFP4 量化（需 Blackwell，MTP 验证只在 bf16 检查点完成）、--enable-scattered-sconv、single-layer EAGLE（mtp_local_layer_ids 已 NotImplementedError）——这些路径的回归风险未在本 PR 内验证。

5. 删除模型侧提交分支：模型方法 update_conv_state_after_mtp_verify 被移除后，spec 路径只依赖 backend hook；若未来某模型重新需要在模型侧提交，需重新引入分发逻辑。 - 影响：对用户：Inkling 系列模型（含 Inkling-Small 248B TP8）decode 每步 CPU launch 与 kernel 数下降，元数据准备从每步 4 次降为 1 次；greedy 输出与 per-token logprobs 与 main 逐位一致（8/9 项检查，第 9 项在 main 本身不确定），GSM8K / AIME26 / MTP accept length 与吞吐（bs=32: 20964 vs 20887 tok/s）均持平或更优。对系统：conv 状态管理从模型代码收敛进 attention backend 统一契约，模型不再直接访问 pool；spec verify 提交统一走 backend hook，为 CUDA graph 录制和多硬件后端复用铺路。对团队：确立了 " 图内录制 vs 图外 replay" 的 hook 放置原则、图静态缓冲地址契约、以及用 AST 源码扫描守护跨硬件 hook 签名的做法；对非 Inkling 路径影响为 no-op（attn_backend_wrapper_for_draft_extend 其他模型原样返回）。 - 风险标记：核心路径变更，CUDA graph 地址契约，NPU 后端契约变动，MTP 验证路径耦合，NVFP4 路径未验证

关联脉络

- PR #29867 ShortConv attention backend: PR body 明确引用：本 PR 就是把 #29867 引入的 ShortConvAttnBackend sidecar 落地到 Inkling。
- PR #32971 [unified-memory] Support MLA-hybrid-Mamba (Kimi-Linear) on the Triton backend: 同一 MambaPool / 混合线性注意力后端家族，且统一内存池的 translate 契约正是本 PR _slot_gather_recordable 分支要区分的目标。
- PR #32972 [unified-memory] Let Kimi-Linear use the paged MLA attention backends: 同属统一内存池与混合注意力后端演进，修改 multi_ended_allocator / server_args 与本 PR 的 slot translate 路径相关。
- PR #32920 [Spec] Compact the target-verify mask when nothing reads it: 同属 speculative verify 路径的性能与契约优化，改动多个 attention backend 的 verify 元数据结构，与本 PR 的 target_verify 元数据路径相关。