PR #26117 完整报告

sgl-project/sglang

[VLM] Preserve preprocessed input ids

合并时间：2026-05-23 17:02

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26117

执行摘要

- 一句话：保留 VLM 预处理输入 ID 并优化 grid 处理
- 推荐动作：值得阅读，展示了如何通过提取通用方法和字段来优化 VLM 预处理路径，尤其 _is_preprocessed_input 系列方法设计可复用，适合在多模态输入验证场景推广。注意确认没有遗漏原有格式匹配逻辑。

功能与动机

PR 描述明确指出需要 'Preserve pre-tokenized VLM input ids through the base processor fast path'，使预处理后的 input ids 可以保留供后续使用，避免重复 tokenization。此外在 mm_utils.py 中，原有代码对 video_grid_thw 和 image_grid_thw 的处理存在重复的 torch.as_tensor 和 torch.prod 调用，且未优化设备间数据拷贝，需要重构以提升性能。

实现拆解

1. 新增 input_ids 字段：在 BaseMultiModalProcessorOutput dataclass 中添加 input_ids: Optional[Union[List[int], torch.Tensor]] = None，用于存储预 tokenized 的输入 ID。
2. 提取预处理判断方法：在 BaseMultimodalProcessor 中新增 _get_preprocessed_input_format、_is_preprocessed_input、_all_mm_data_is_preprocessed 三个类 / 静态方法，统一判断输入数据是否已预处理（processor_output 或 precomputed_embedding），替代原有零散的内联条件。
3. 重构 _load_single_item：将原来通过 isinstance 检查和 format 字符串比较的逻辑替换为 cls._is_preprocessed_input(data)，使代码更简洁且可复用。
4. 更新 validate_mm_data 与 _process_loaded_mm_data：使用 _is_preprocessed_input 代替硬编码的格式字符串判断，保持一致性。
5. 优化 mm_utils.py 中的 grid 处理：提取 _grid_rows_to_cpu_list 函数确保 grid tensor 移至 CPU 并转换为 Python list，提取 _prod_grid_values 计算元素乘积；在 get_new_expanded_mm_items 中统一使用这两个函数，避免重复的 torch.as_tensor 和 torch.prod 调用，减少设备同步。

关键文件：

- python/sglang/srt/multimodal/processors/base_processor.py（模块 处理器；类别 source；类型 core-logic；符号 _get_preprocessed_input_format, _is_preprocessed_input, _all_mm_data_is_preprocessed, _ensure_input_ids_is_tensor）

：核心变更：新增 input_ids 字段、预处理判断方法，并更新多个内部方法（_load_single_item、validate_mm_data、_process_loaded_mm_data），是整个 PR 的主体。

- python/sglang/srt/managers/mm_utils.py（模块 工具函数；类别 source；类型 core-logic；符号 _grid_rows_to_cpu_list, _prod_grid_values）：提取 _grid_rows_to_cpu_list 和 _prod_grid_values，优化 get_new_expanded_mm_items 中的 grid 处理，减少重复 tensor 转换和乘积计算。

关键符号：_get_preprocessed_input_format, _is_preprocessed_input, _all_mm_data_is_preprocessed, _ensure_input_ids_is_tensor, _grid_rows_to_cpu_list, _prod_grid_values, get_new_expanded_mm_items

关键源码片段

python/sglang/srt/multimodal/processors/base_processor.py

核心变更：新增 input_ids 字段、预处理判断方法，并更新多个内部方法（_load_single_item、validate_mm_data、_process_loaded_mm_data），是整个 PR 的主体。

```
@dataclasses.dataclass
class BaseMultiModalProcessorOutput:
    # input_text with all multimodality placeholder token expanded
    input_text: str

    # original pre-tokenized ids, useful for processor_output/precomputed inputs,
    # when they already carry the input ids
    input_ids: Optional[Union[List[int], torch.Tensor]] = None

    # frames loaded from image, in given order
    images: Optional[list[Union[Image.Image, dict]]] = dataclasses.field(default_factory=list)
    # ... other fields


class BaseMultimodalProcessor:
    # ...

    @staticmethod
    def _get_preprocessed_input_format(data):
        """
        判断输入数据是否已预处理，并返回具体格式枚举；若未预处理则返回 None。
        """
        if not isinstance(data, dict):
            return None
        data_format = data.get("format")
        if isinstance(data_format, MultimodalInputFormat):
            return data_format
        # 兼容字符串形式的格式名称
        if data_format in (
            MultimodalInputFormat.PROCESSOR_OUTPUT.name,
```

```
            "processor_output",
        ):
            return MultimodalInputFormat.PROCESSOR_OUTPUT
        if data_format in (
            MultimodalInputFormat.PRECOMPUTED_EMBEDDING.name,
            "precomputed_embedding",
        ):
            return MultimodalInputFormat.PRECOMPUTED_EMBEDDING
        return None

    @classmethod
    def _is_preprocessed_input(cls, data):
        """根据格式判断输入是否为已预处理数据"""
        return cls._get_preprocessed_input_format(data) is not None

    @classmethod
    def _all_mm_data_is_preprocessed(cls, *data_lists):
        """
        检查所有多模态数据列表中的所有元素是否都已预处理。
        如果任意一个元素不是预处理格式，则返回 False。
        """
        has_mm_data = False
        for data_list in data_lists:
            if not data_list:
                continue
            if not isinstance(data_list, list):
                data_list = [data_list]
            for item in data_list:
                if item is None:
                    continue
                has_mm_data = True
                if not cls._is_preprocessed_input(item):
                    return False
        return has_mm_data
```

评论区精华

PR 没有收到人工 review 评论，仅包含两条自动机器人消息（quota 限制和 /tag-and-rerun-ci 命令），无技术讨论。

- 暂无高价值评论线程

风险与影响

- 风险：主要风险来自逻辑替换的等价性：_is_preprocessed_input 方法需要严格覆盖原有所有预处理格式判断，否则可能漏判或误判导致加载异常。mm_utils.py 中 _grid_rows_to_cpu_list 假定 tensor 可安全 detach 并 tolist，对于无法直接 tolist 的 tensor（如稀疏张量）会报错，但实际使用场景均为连续稠密 tensor，风险较低。整体变更未引入测试，若后续修改对应判断逻辑可能被忽略。

- 影响：对用户透明，不改动 API，仅内部优化。多模态模型（VLM）在预处理路径中可保留 input_ids，减少重复 tokenization 开销；grid 处理优化减少了不必要的 GPU 同步和 tensor 创建。团队内部代码结构更清晰，便于后续扩展。
- 风险标记：缺少测试覆盖，核心路径变更

关联脉络

- PR #26116 [VLM] Reuse Qwen pretokenized ids: 相同功能线：都围绕 VLM 预处理 input ids 的保留与复用，本 PR 为基础组件通用化，PR#26116 为具体 Qwen 模型适配。
- PR #24144 [BugFix][EPD] adapt for qwen3.5-mtp & del duplicated logs: 涉及 Qwen VLM 处理器，与预处理逻辑相关，本 PR 的通用方法可能影响该模型。