

PR #30260 完整报告

sgl-project/sglang

[Fix] `--mm-process-config` crash when video config contains

合并时间: 2026-07-28 09:24

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/30260>

执行摘要

- 一句话: 修复 Qwen3-VL 视频配置参数错误传递 crash
- 推荐动作: ### 建议
- 值得精读: 该 PR 展示了框架集成 HuggingFace processor 时常见的配置键冲突问题, 以及如何通过干净的过滤层解耦。设计决策 (过滤 vs retry) 的演变值得参考。
- 关注点: 若未来新增更多 SGLang 内部视频配置键, 需同步更新 `QWEN_VIDEO_PREPROCESS_CONFIG_KEYS`。建议添加注释指向该集合, 提醒开发者维护。

功能与动机

在 Qwen3-VL 模型上使用 `--mm-process-config` 配置视频参数时, 报错 `merged_typed_dict.__init__() got an unexpected keyword argument 'max_frames'` (见 PR 错误栈)。根本原因是 SGLang 将 `mm_process_config` 中的 video 配置作为 `videos_kwargs` 直接传入 HuggingFace processor, 而 processor 不接受 `max_frames`、`min_pixels` 等 SGLang 自定义键。解决方案是仅传递 processor 可识别的键 (如 `do_normalize`) , SGLang 内部键应在预处理阶段自行消费。

实现拆解

实现拆解

1. 定义过滤常量与函数(`qwen_vl.py`): 新增 `QWEN_VIDEO_PREPROCESS_CONFIG_KEYS` frozenset, 包含 `fps`, `nframes`, `min_frames`, `max_frames`, `min_pixels`, `max_pixels`, `total_pixels`, `resized_height`, `resized_width` 等 SGLang 内部键。新增 `_get_processor_video_config(video_config, video_metadata)` 函数: 若视频已预处理 (`video_metadata` 非空且无 `None`) , 则返回过滤后的配置 (移除上述键); 否则返回 `None` , 表示使用原始配置。
2. 扩展 `base_processor.py`: `process_mm_data` 方法新增可选参数 `processor_video_config` , 当提供时, 视频的 `videos_kwargs` 使用该参数值而非 `self.video_config` , 使上层可灵活注入过滤后的配置。
3. 修改 `qwen_vl.py` 主流程: 在 `process_mm_data_async` 中, 预处理视频后调用 `_get_processor_video_config` 决定是否传递过滤配置, 并通过 `processor_video_config` 参数传入 `process_and_combine_mm_data_async` (最终到达 `process_mm_data`) 。统一

了 qwen3_vl 等模型与其他模型的调用路径，减少了分支重复。

4. 补充单元测试(test_mm_process_config.py): 新增

test_preprocessed_video_config_is_filtered_before_single_call 验证预处理视频时只传递 do_normalize 等非 SGLang 键; 新增 TestQwenVideoConfigRouting 类验证 _get_processor_video_config 在已处理 / 未处理视频下的返回。同时增加 test_processor_error_is_not_retried 确保 processor 内部错误不会被静默重试。

5. NPU 兼容性修复(npu/modules/qwen_vl_processor.py): 将

transform_patches_to_flatten 中的 patches.view 替换为 patches.reshape, 避免因输入张量不连续导致的 RuntimeError。

关键文件:

- python/sglang/srt/multimodal/processors/qwen_vl.py (模块 多模态处理; 类别 source; 类型 core-logic; 符号 _get_processor_video_config, process_mm_data_async): 核心逻辑: 新增过滤函数和常量, 修改视频处理主流程, 是修复的关键。
- test/registered/unit/managers/test_mm_process_config.py (模块 测试; 类别 test; 类型 test-coverage; 符号 test_preprocessed_video_config_is_filtered_before_single_call, test_processor_error_is_not_retried, TestQwenVideoConfigRouting, test_preprocessed_video_drops_sglang_owned_config): 新增单元测试覆盖过滤和路由行为, 验证核心逻辑正确性。
- python/sglang/srt/multimodal/processors/base_processor.py (模块 多模态框架; 类别 source; 类型 core-logic; 符号 process_mm_data): 框架层扩展, 新增 processor_video_config 参数, 支撑上层配置注入。
- python/sglang/srt/hardware_backend/npu/modules/qwen_vl_processor.py (模块 NPU 后端; 类别 source; 类型 bugfix; 符号 transform_patches_to_flatten): 修复 NPU 上 transform_patches_to_flatten 的张量连续性 bug。

关键符号: _get_processor_video_config, process_mm_data_async, process_mm_data, transform_patches_to_flatten

关键源码片段

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

框架层扩展, 新增 processor_video_config 参数, 支撑上层配置注入。

```
def process_mm_data(
    self,
    input_text,
    images=None,
    videos=None,
    audios=None,
    processor=None,
    processor_video_config: Optional[Dict[str, Any]] = None, # 新增参数
    **kwargs,
) -> dict:
    """处理多模态数据, 支持注入自定义视频配置。"""
```

```

processor, tokenizer = self._resolve_processor(processor)

if images:
    kwargs["images"] = images
    if self.image_config:
        kwargs.setdefault("images_kwargs", {}).update(self.image_config)
if videos:
    kwargs["videos"] = videos
    # 如果外部提供了 processor_video_config, 则优先使用, 否则回退到 self.video_config
    video_config = (
        self.video_config
        if processor_video_config is None
        else processor_video_config
    )
    if video_config:
        kwargs.setdefault("videos_kwargs", {}).update(video_config)
# ... 其余部分不变

```

评论区精华

评论区精华

- gemini-code-assist指出早期 retry 方案的 Bug: 直接在 kwargs 上迭代会引发 ValueError, 且建议递归过滤嵌套配置 (如 videos_kwargs)。该评论推动了最终过滤方案的确立。
- McZyWu给出简洁的过滤实现方向: for k in slang_video_keys: kwargs["videos_kwargs"].pop(k, None), 体现了在 base 层轻量过滤的意图。
- ping1jing2询问 view 改为 reshape 的原因, longxin9715解释: 输入张量不连续时 .view 会抛出 RuntimeError, .reshape 可自动处理连续性。
- Retry 逻辑的字典遍历 Bug 与嵌套过滤建议 (correctness): 最终方案未采用 retry, 而是直接过滤配置键, 避免了该问题。
- 在 base_processor 中直接 pop 非 processor 键 (design): 最终方案在 qwen_vl 中实现过滤, 传递过滤后的配置至 base, 而非在 base 中 pop。
- view 替换为 reshape 的原因 (question): 作者确认替换是修复运行时错误, .reshape 自动处理连续性。

风险与影响

- 风险: ### 风险分析
- 回归风险 (低): 仅 Qwen 视频处理路径受影响, 其他多模态处理器 (如 Qwen2-VL) 不受影响, 因为 _get_processor_video_config 只在 qwen_vl.py 中调用。base_processor.py 的改动兼容旧路径 (不传 processor_video_config 时行为不变)。
- 配置兼容性 (低): QWEN_VIDEO_PREPROCESS_CONFIG_KEYS 的白名单方式可能会遗漏某些未来新增的 SGLang 内部键, 导致 crash 重现。建议持续维护该集合, 或采用更通用的黑名单机制。

- NPU reshape 风险 (极低): .reshape 的行为可能在某些边缘情况下与 .view 不同 (如复制数据), 但根据提交者描述, 这是修复已有 bug, 且 reshape 是更安全操作。
- 影响: ### 影响分析
- 用户: Qwen3-VL 等模型用户现在可以安全使用 --mm-process-config 自定义视频处理参数, 而不会遇到服务启动或请求错误。
- 系统: 改动集中在多模态处理器内部, 不影响调度、注意力、推理性能。额外单元测试增强了配置路由的正确性。
- 团队: 维护开销极小, 新增的过滤逻辑清晰且可测。NPU 的 .reshape 修复提升特定硬件的稳定性。
- 风险标记: 多模态处理路径变更, 配置键白名单可能遗漏, NPU 兼容性修复

关联脉络

- 暂无明显关联 PR