PR #25130 完整报告

sgl-project/sglang

[NPU]Bugfix:Set default values for npu_wrapper_preprocess parameters

合并时间：2026-05-14 14:21

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25130

执行摘要

- 一句话：修复 NPU 上 Qwen2-VL 图像预处理与 GDN 闪退兼容性问题
- 推荐动作：建议阅读以了解 NPU 后端如何适配上游库变更，但该 PR 变更较小，团队可快速合并。后续应补充测试用例以确保预处理补丁的兼容性。

功能与动机

当前 Transformers 的 Qwen2-VL 处理器已弃用 Qwen2VLImageProcessorFast 及其快速处理器辅助 API，NPU 的预处理补丁仍针对旧版，导致运行时 API/ 签名不匹配而失败。同时 Ascend GDN 后端调用 torch.ops.npu.recurrent_gated_delta_rule 时使用了错误的关键字参数 num_accept_tokens（应为 num_accepted_tokens）。

实现拆解

1. 切换图像预处理补丁目标：在 npu_apply_qwen_image_preprocess_patch() 中将 apply_module_patch 的目标从 transformers.models.qwen2_vl.image_processing_qwen2_vl_fast.Qwen2VLImageProcessorFast 改为 transformers.models.qwen2_vl.image_processing_qwen2_vl.Qwen2VLImageProcessor。
2. 更新导入和参数名：将 group_images_by_shape 和 reorder_images 的导入源从 transformers.image_processing_utils_fast 迁移到 transformers.image_transforms；将 npu_wrapper_preprocess 和 npu_wrapper_video_preprocess 中的参数 interpolation 重命名为 resample，并更新其类型注解。
3. 调整字典访问 → 属性访问：在 smart_resize 调用中将 size["shortest_edge"]、size["longest_edge"] 改为 size.shortest_edge、size.longest_edge，匹配新版 SizeDict API。
4. 修复 GDN 后端关键字参数：在 ascend_gdn_backend.py 的 fused_recurrent_gated_delta_rule_update 函数中，将 num_accept_tokens=num_accept_tokens 改为 num_accepted_tokens=num_accept_tokens，与算子接口对齐。

关键文件：

- python/sglang/srt/hardware_backend/npu/modules/qwen_vl_processor.py（模块 NPU 模块；类别 source；类型 dependency-wiring；符号 npu_wrapper_preprocess, npu_wrapper_video_preprocess, npu_apply_qwen_image_preprocess_patch,

transform_patches_to_flatten）：核心修改文件：切换补丁目标、更新导入源、重命名参数、调整字典访问方式，共 +11/-15 行。

- python/sglang/srt/hardware_backend/npu/attention/ascend_gdn_backend.py（模块 NPU 模块；类别 source；类型 core-logic；符号 fused_recurrent_gated_delta_rule_update）：修复关键字参数拼写错误，共 +1/-1 行。

关键符号：npu_wrapper_preprocess, npu_wrapper_video_preprocess, npu_apply_qwen_image_preprocess_patch, fused_recurrent_gated_delta_rule_update

关键源码片段

python/sglang/srt/hardware_backend/npu/modules/qwen_vl_processor.py

核心修改文件：切换补丁目标、更新导入源、重命名参数、调整字典访问方式，共+11/-15行。

```
# python/sglang/srt/hardware_backend/npu/modules/qwen_vl_processor.py

# 导入来源从 fast 模块迁移到标准模块
from transformers.image_transforms import group_images_by_shape, reorder_images
# 原代码 : from transformers.image_processing_utils_fast import group_images_by_shape, reorder_images

def npu_wrapper_preprocess(func):
    def _preprocess(
        self,
        images: list["torch.Tensor"],
        do_resize: bool,
        size: SizeDict,
        resample: "PILImageResampling | tvF.InterpolationMode | int | None", # 原为 interpolation
        do_rescale: bool,
        rescale_factor: float,
        do_normalize: bool,
        image_mean: float | list[float] | None,
        image_std: float | list[float] | None,
        patch_size: int,
        temporal_patch_size: int,
        merge_size: int,
        disable_grouping: bool | None,
        return_tensors: str | TensorType | None,
        **kwargs,
    ):
        # ... 中间处理逻辑 ...
        if do_resize:
            resized_height, resized_width = smart_resize(
                height,
                width,
                factor=patch_size * merge_size,
                min_pixels=size.shortest_edge, # 原为 size["shortest_edge"]
                max_pixels=size.longest_edge, # 原为 size["longest_edge"]
```

```
            )
            stacked_images = self.resize(
                image=stacked_images,
                size=SizeDict(height=resized_height, width=resized_width),
                resample=resample, # 原为 interpolation=interpolation
            )
        # ... 后续处理 ...

# patch 目标从 Fast 类切换到标准类
def npu_apply_qwen_image_preprocess_patch():
    # ...
    apply_module_patch(
        "transformers.models.qwen2_vl.image_processing_qwen2_vl.Qwen2VLImageProcessor", #
        原为 ...qwen2_vl_fast.Qwen2VLImageProcessorFast
        "_preprocess",
        [npu_wrapper_preprocess],
    )
```

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

修复关键字参数拼写错误，共 +1/-1 行。

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

# 在 fused_recurrent_gated_delta_rule_update 方法中
attn_core_out = torch.ops.npu.recurrent_gated_delta_rule(
    mix_qkv,
    recurrent_state,
    beta=beta,
    scale=scale,
    actual_seq_lengths=actual_seq_lengths,
    ssm_state_indices=ssm_state_indices.view(batch_size, seq_len),
    nk=num_heads,
    nv=num_value_heads,
    intermediate_state=intermediate_state,
    cache_indices=cache_indices,
    num_accepted_tokens=num_accept_tokens, # 原为 num_accept_tokens=num_accept_tokens
    g=g,
)
```

评论区精华

审阅者 Hexq0210 提问：“添加默认值能解决什么问题？”（What problems can adding default values solve?），表明对变更必要性的质疑。但该问题未获得回复，PR 随后由 sglang-npu-bot 批准合并。

- 添加默认值的必要性 (question): 未获作者回复，PR 仍被批准合并。

风险与影响

- 风险：低风险。变更是对上游 Transformers 库 API 变更的适配，且已在 NPU 上通过 Qwen3.6-27B 模型验证（GSM8K 准确率 94.91%）。但测试覆盖不足：未添加对应的单元测试来验证预处理补丁兼容性，后续上游再次变更可能导致回归。
- 影响：影响范围限定于 NPU 后端使用 Qwen2-VL 图像处理器或 Ascend GDN 的前向路径。修复后 NPU 用户可正常使用 Qwen3-VL 系列模型，不涉及 CPU/GPU 等其他后端。
- 风险标记：缺少测试覆盖

关联脉络

- 暂无明显关联 PR