PR #25697 完整报告

sgl-project/sglang

[diffusion] Fix GLM-Image /v1/images/edits support

合并时间：2026-05-20 17:11

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25697

执行摘要

- 一句话：修复 GLM-Image 的 /v1/images/edits 支持
- 推荐动作：值得一读，特别是关注扩散模型编辑功能接入方式的读者。其中 image_path_to_list 和 pooled_image_features_to_tensor 的抽象模式、空张量保护前置检查都是稳健编码的好例子。此外，generate_prior_tokens 中的跨来源 Token ID 上采样逻辑展示了如何适配多图像输入。

功能与动机

根据 Issue #25579 的反馈，GLM-Image 原本只支持文生图（T2I），缺少图生图（TI2I）支持。通过将 task_type 标记为 TI2I，使得 /v1/images/edits 接口能够接受图像输入。此外，还需要解决图像加载、特征聚合和空张量崩溃等问题。

实现拆解

1. 配置变更：在 python/sglang/multimodal_gen/configs/pipeline_configs/glm_image.py 中将 task_type 从 ModelTaskType.T2I 改为 ModelTaskType.TI2I，这是 /v1/images/edits 接受图像输入的前提。
2. 工具函数：在 python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/glm_image.py 新增 image_path_to_list（统一图像路径为列表）和 pooled_image_features_to_tensor（将图像特征提取为张量），方便下游处理。
3. 多来源图像支持：重写 generate_prior_tokens 方法，从多图像来源分别提取特征并聚合，然后对每个来源的 Token ID 进行上采样（_upsample_token_ids），适配目标分辨率。
4. 编辑条件加载：forward 方法中从 batch.image_path 加载编辑条件图像，当存在图像时自动调整输出尺寸为输入图像尺寸，并带入生成流程。
5. 空张量保护：在 layernorm.py 的 forward_cuda 和 scale_shift.py 的 fuse_scale_shift_kernel 中增加空张量检查，当输入为空时降级到原生实现，防止 CUDA kernel 崩溃。

关键文件：

- python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/glm_image.py（模块 扩散管线；类别 source；类型 core-logic；符号 image_path_to_list, pooled_image_features_to_tensor, component_uses, generate_prior_tokens）：核心修改：新增辅助函数、调整 forward 和 generate_prior_tokens 以支持编辑输入

- python/sglang/multimodal_gen/configs/pipeline_configs/glm_image.py（模块 扩散管线；类别 source；类型 configuration；符号 task_type）：关键触发点：修改 task_type 使管线支持 TI2I
- python/sglang/multimodal_gen/runtime/layers/layernorm.py（模块 扩散管线；类别 source；类型 core-logic）：增加空张量保护，防止 CUDA kernel 崩溃
- python/sglang/jit_kernel/diffusion/triton/scale_shift.py（模块 JIT 内核；类别 source；类型 core-logic）：对应空张量保护，与 layernorm 改动配套

关键符号：image_path_to_list, pooled_image_features_to_tensor, generate_prior_tokens, forward_cuda, fuse_scale_shift_kernel

关键源码片段

python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/glm_image.py

核心修改：新增辅助函数、调整 forward 和 generate_prior_tokens 以支持编辑输入

```
import torch
from typing import List, Union

def image_path_to_list(image_path: Union[str, List[str]]) -> List[str]:
    # 统一图像路径为列表，方便后续遍历
    return image_path if isinstance(image_path, list) else [image_path]

def pooled_image_features_to_tensor(image_features) -> torch.Tensor:
    # 从图像特征中提取 pooler 输出，并统一为张量格式
    pooler_output = getattr(image_features, 'pooler_output', None)
    if pooler_output is not None:
        image_features = pooler_output
    if isinstance(image_features, torch.Tensor):
        return image_features
    # 如果 features 是 tuple 或 list，cat 之
    return torch.cat(tuple(image_features), dim=0)
```

评论区精华

本 PR 无实质讨论，由维护者 mickqian 直接批准。

- 暂无高价值评论线程

风险与影响

- 风险：主要风险在于 task_type 变更可能影响纯 T2I 流程，但 PR 作者提供了两种场景的准确性测试，结果正常。新增的图像加载和特征处理路径可能在异常输入（如无效路径）时抛出错误，需依赖 load_image 的错误处理。空张量保护仅在被触发时降级到原生实现，可能略微降低性能，但影响极小。缺少针对边缘用例的测试覆盖。

- 影响：对用户：支持 GLM-Image 的图生图功能（/v1/images/edits），扩展模型可用场景。对系统：引入新的辅助函数和上采样逻辑，提高了代码复杂度但增强了鲁棒性。对团队：开启了扩散模型编辑能力接入的方向，类似改动可能推广至其他模型。影响范围限于 diffusion 子模块，不涉及核心调度或推理路径。
- 风险标记：扩展任务类型影响 T2I，缺少测试覆盖，空张量检查依赖 fallback

关联脉络

- 暂无明显关联 PR