PR #34945 完整报告

sgl-project/sglang

[Diffusion] Native Qwen3-VL vision encoder

合并时间：2026-08-16 09:58

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34945

执行摘要

- 一句话：Qwen3-VL 视觉编码器去掉 Transformers 依赖，改为原生实现
- 推荐动作：值得精读。该 PR 展示了将 Transformers 模型组件替换为 SGLang 原生实现时需要注意的多个约束：checkpoint 参数名兼容、FP32 位置插值、计算布局对齐，以及 layerwise offload/FSDP 分片的接入方式。同时有明确的数值对拍方法和分层 offload 测试，可作为后续视觉编码器原生化的参考。

功能与动机

在 SGLang-Diffusion 中接入 Qwen3-VL 时，此前视觉编码器直接复用 Transformers 的 Qwen3VLVisionModel。PR body 的目的是将 Transformers 持有的视觉模型替换为原生 SGLang-Diffusion 实现，以便视觉块纳入 layerwise offload 与 FSDP 分片、避免对 Transformers 内部 vision 辅助函数的隐式依赖，同时保证 checkpoint 参数名不变、数值严格对齐。

实现拆解

1. 新增 python/sglang/multimodal_gen/runtime/models/encoders/qwen3vl_vision.py，实现 Qwen3VLVisionPatchEmbed、Qwen3VLVisionRotaryEmbedding、Qwen3VLVisionAttention、Qwen3VLVisionMLP、Qwen3VLVisionBlock、Qwen3VLVisionPatchMerger、Qwen3VLVisionTransformer 及 Qwen3VLVisionOutput、_PackedSequenceMetadata。
2. 修改 python/sglang/multimodal_gen/runtime/models/encoders/qwen3vl.py，将 Qwen3VLModel.visual 由 Qwen3VLVisionModel._from_config(config.vision_config) 改为 Qwen3VLVisionTransformer(config.vision_config)，删除 _get_flat_visual_features 对 Transformers vision_utils 的 CPU grid_thw 专用分支，并新增 layer_names = ['model.visual.blocks']。
3. 修改 python/sglang/multimodal_gen/configs/models/encoders/qwen3vl.py，在 _fsdp_shard_conditions 加入 is_block，使视觉块参与 FSDP 分片。
4. 修改 python/sglang/multimodal_gen/runtime/models/encoders/minimax_h3_qwen3vl.py，为 MiniMaxH3Qwen3VLEncoder 增加 layer_names。
5. 新增 python/sglang/multimodal_gen/test/unit/test_qwen3vl_vision.py，覆盖视觉布局与 merge 顺序、checkpoint 参数名保持、FP32 位置插值保持、offload 注册。

关键文件：

- python/sglang/multimodal_gen/runtime/models/encoders/qwen3vl_vision.py（模块 视觉编码器；类别 source；类型 core-logic；符号 Qwen3VLVisionOutput, _PackedSequenceMetadata, from_cu_seqlens, Qwen3VLVisionPatchEmbed）：新增原生 Qwen3-VL 视觉编码器实现，是本 PR 的核心源码文件；包含全部视觉子模块与打包序列元数据。
- python/sglang/multimodal_gen/runtime/models/encoders/qwen3vl.py（模块 模型接入；类别 source；类型 core-logic；符号 Qwen3VLModel, _get_flat_visual_features, Qwen3VLForConditionalGeneration）：将 Qwen3VLModel.visual 从 Transformers Qwen3VLVisionModel 切换为原生 Qwen3VLVisionTransformer，并清理 Transformers 专用辅助分支、登记 layer_names。
- python/sglang/multimodal_gen/test/unit/test_qwen3vl_vision.py（模块 视觉测试；类别 test；类型 test-coverage；符号 test_native_vision_layout_matches_qwen3_merge_order, test_native_vision_keeps_checkpoint_parameter_names, test_native_vision_keeps_position_math_in_fp32, test_qwen3_multimodal_encoders_layerwise_offload_vision_blocks）：新增 4 项针对性单测，覆盖布局对齐、checkpoint 参数名、FP32 位置数学、offload 注册，是验证原生视觉塔正确性的配套。
- python/sglang/multimodal_gen/configs/models/encoders/qwen3vl.py（模块 模型配置；类别 source；类型 configuration；符号 Qwen3VLArchConfig）：在 _fsdp_shard_conditions 中加入 is_block，使视觉塔 block 模块纳入 FSDP 分片。
- python/sglang/multimodal_gen/runtime/models/encoders/minimax_h3_qwen3vl.py（模块 模型接入；类别 source；类型 configuration；符号 MiniMaxH3Qwen3VLEncoder）：MiniMaxH3Qwen3VLEncoder 同步登记 model.visual.blocks 到 layer_names，保证视觉块也能参与 layerwise offload。

关键符号：Qwen3VLVisionTransformer.forward, Qwen3VLVisionAttention.forward, Qwen3VLVisionAttention._packed_attention, _apply_vision_rotary_embedding, _vision_position_ids, _vision_cu_seqlens, _interpolate_position_embeddings, Qwen3VLVisionPatchEmbed.forward, Qwen3VLVisionPatchMerger.forward, Qwen3VLModel._get_flat_visual_features

关键源码片段

python/sglang/multimodal_gen/runtime/models/encoders/qwen3vl.py

将 Qwen3VLModel.visual 从 Transformers Qwen3VLVisionModel 切换为原生 Qwen3VLVisionTransformer，并清理 Transformers 专用辅助分支、登记 layer_names。

```
class Qwen3VLModel(nn.Module):
    def __init__(self, config, *, use_tensor_parallel=False):
        super().__init__()
        # 视觉塔从 Transformers Qwen3VLVisionModel._from_config 换成原生实现，
        # 参数名保持一致，因此 checkpoint 可以直接复用。
        self.visual = Qwen3VLVisionTransformer(config.vision_config)
```

```
        self.language_model = Qwen3VLTextModel(config.text_config, use_tensor_parallel=use_
        tensor_parallel)
        self.rope_deltas = None
        self.config = config

    def _get_flat_visual_features(self, pixel_values, grid_thw):
        # 原生视觉塔内部自行处理 padding 与位置编码，
        # 不再依赖 transformers.vision_utils 的 get_vision_cu_seqlens/get_vision_position_ids。
        visual_out = self.visual(pixel_values, grid_thw=grid_thw)
        return visual_out.pooler_output, visual_out.deepstack_features
```

python/sglang/multimodal_gen/test/unit/test_qwen3vl_vision.py

新增 4 项针对性单测，覆盖布局对齐、checkpoint 参数名、FP32 位置数学、offload 注册，是验证原生视觉塔正确性的配套。

```
def test_native_vision_layout_matches_qwen3_merge_order():
    grid_thw = torch.tensor([[1, 4, 6], [2, 2, 4]])
    position_ids = _vision_position_ids(grid_thw, spatial_merge_size=2)
    cu_seqlens = _vision_cu_seqlens(grid_thw)
    # 锁定 position_ids 的 block-major 顺序：先 2x2 merge 块内（行，列）展开，
    # 再按块从左到右、从上到下推进；任何布局改动都会破坏这段断言。
    assert position_ids.shape == (40, 2)
    assert position_ids[:8].tolist() == [[0, 0], [0, 1], [1, 0], [1, 1], [0, 2], [0, 3], [1, 2], [1, 3]]
    assert cu_seqlens.tolist() == [0, 24, 32, 40]


def test_native_vision_keeps_checkpoint_parameter_names():
    config = SimpleNamespace(
        hidden_size=16, intermediate_size=24, hidden_act="gelu_pytorch_tanh", num_heads=2,
        depth=0, patch_size=2, temporal_patch_size=1, in_channels=3,
        num_position_embeddings=16, spatial_merge_size=2, out_hidden_size=12,
        deepstack_visual_indexes=[],
    )
    model = Qwen3VLVisionTransformer(config)
    # 参数名必须与 Transformers checkpoint 完全一致，否则权重加载会失败。
    assert set(model.state_dict()) == {
        "patch_embed.proj.weight", "patch_embed.proj.bias", "pos_embed.weight",
        "merger.norm.weight", "merger.norm.bias",
        "merger.linear_fc1.weight", "merger.linear_fc1.bias",
        "merger.linear_fc2.weight", "merger.linear_fc2.bias",
    }
```

评论区精华

提交只有 1 个 commit，review 评论为空，无公开讨论线程。PR body 给出了验证结论：远程 H100 单测 4 passed；真实 MiniMax-H3 27 层 checkpoint 与 Transformers 5.12.1 对拍 exact（max_abs=0）；2xH100 端到端 minimax_h3_t2va_2gpu_h100 通过，46.97s vs 47.54s 基线。

- 未检索到 review 讨论 (question): 无已记录的设计争论或未解决疑虑。

风险与影响

- 风险：风险点主要在于视觉塔数值对齐：虽然对拍实验显示 max_abs=0，但对拍仅覆盖 MiniMax-H3 27 层场景，若其它使用 Qwen3-VL 视觉塔的模型（如 JoyImage/Qwen3VLT2VA）采用不同配置，需重新对拍验证。_PackedSequenceMetadata 是基于 cu_seqlens 的打包序列假设，若遇到空序列或 delta 计算变化，attention 的 forward_varlen 返回逻辑可能受影响。FSDP/offload 分片新增 is_block 条件后，视觉块会被纳入分片和 offload，若 deepstack merger 或其它自定义模块的 layer_names 未同步登记，可能出现参数未分片或未被 offload 的情况。
- 影响：影响范围为 SGLang-Diffusion 原生 Qwen3-VL 视觉编码路径，主要使用者是 MiniMax-H3 和 JoyImage（Qwen3-VL-8B-Instruct）相关 pipeline。对用户来说，视觉编码行为、数值、输出结构均保持不变，因此是低感知的兼容性替换；对团队来说，去掉对 Transformers v5 内部函数（get_vision_cu_seqlens/get_vision_position_ids）的耦合，后续可自由优化视觉块。
- 风险标记：视觉塔数值对齐依赖对拍验证，packed varlen 后端回退路径，FSDP/offload 登记面扩展，依赖 Transformers 内部行为去除

关联脉络

- PR #34949 [Diffusion] Route MiniMax H3 VAE attention through native backends: 同属 diffusion 原生化的 refactor 系列，都把 Transformers/ 本地实现迁到 SGLang 统一的 attention 后端。
- PR #34980 [Diffusion] Native Hunyuan3D Paint and Delight models: 同一时期将外部实现迁移为 SGLang 原生模型系列，涉及相同 multimodal_gen runtime 与 offload 机制。
- PR #34951 [Diffusion] Native ERNIE prompt enhancer: 原生 encoder 接入 SGLang-Diffusion 的同类工作，涉及 encoder loader 与分层 offload。