PR #31575 完整报告

sgl-project/sglang

Fix rope config compatibility and VL/transformers-fallback weight loading

合并时间：2026-08-17 14:53

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/31575

执行摘要

- 一句话：修复 RoPE 配置与 VL/transformers 权重加载兼容性
- 推荐动作：值得精读。重点看三处设计：一是 get_rope_config() 对 v4/v5 配置的统一读取与 None 语义处理；二是 qwen.py 中“跳过逻辑必须覆盖所有索引路径”的 review 驱动加固；三是 transformers.py 中 5D 特征“延迟收集 + 批量最大零填充”的方案，它解释了为什么不能简单 flatten。若后续要扩展 transformers 回退路径或补齐 XPU 多模态支持，本 PR 是很好的参考模板。

功能与动机

PR body 开宗明义：Five independent, small fixes uncovered while enabling additional models。具体触发点包括：baidu/ERNIE-4.5-VL-28B-A3B-PT 与 allenai/Olmo-3-7B-Instruct 的 rope_parameters 存在但不含 rope_theta，get_rope_config() 直接 KeyError；Qwen/Qwen-VL-Chat 的 checkpoint 包含 transformer.visual.视觉权重，而该实现只覆盖文本主干导致加载崩溃；transformers v5 扁平化 SigLIP/CLIP 后旧 checkpoint 的 vision_tower.vision_model. 前缀失效；anyres 模型 5D pixel_values 直接 flatten 拼接会因各图 patch 数不同而崩溃；Phi-4-multimodal-instruct 的 dynamic-HD 预处理使并发请求图片 crop 数不同，跨请求 batched ViT 中 torch.cat 抛 Sizes of tensors must match except in dimension 0. Expected size 7 but got size 11。

实现拆解

第 1 步：统一并加固 RoPE 配置读取

- python/sglang/srt/utils/hf_transformers/common.py 的 get_rope_config()：当 config.rope_parameters 存在但缺 rope_theta 键时，回退到 config.rope_theta（默认 10000），不再抛 KeyError；当 rope_parameters 为 None 时返回 (config.rope_theta, None)，保持 v4 配置语义。
- python/sglang/srt/models/ernie45_moe_vl.py 的 Ernie4_5_VLMoeDecoderLayer.__init__ 与 python/sglang/srt/models/olmo2.py 的 Olmo2Attention.__init__改为调用 get_rope_config(config)，消除对 rope_parameters 键的直接索引依赖。
- 演进注意：commit 54db113e 显示早期实现曾用 or {} 把 None 转空 dict，导致 Qwen3-Next 等 v4 模型 RoPE 配置被破坏（CI logprobs 测试失败），后改为显式 None

判断，属于本 PR 内部的返工点。

第 2 步：Qwen 权重加载的防御式跳过

- QWenLMHeadModel.load_weights（python/sglang/srt/models/qwen.py）增加两层守卫：stacked_params_mapping 循环内先用 temp_name 检查 params_dict，缺失即 continue（防止 transformer.visual.mlp.w1 这类含 w1/w2 子串的视觉权重误入映射分支）；else 分支对 name 不在 params_dict 的 key 直接 continue。
- 第二层守卫是 review 阶段由 gemini-code-assist[bot] 指出后才补上的（commit 4465c049），体现"跳过逻辑必须覆盖所有索引路径，否则崩溃点会换个位置复活"。

第 3 步：transformers 通用回退路径的多模态修复

- MultiModalMixin.init：检测 self.model.vision_tower 是否缺少 vision_model 子模块，若缺少则将新增的 orig_to_new_prefix={"vision_tower.vision_model.": "model.vision_tower."} 与既有 weight_mapper 组合，兼容 transformers v5 扁平化结构下的旧 checkpoint。
- MultiModalMixin._collect_mm_kwargs：5D feature（num_images, num_patches, C, H, W）不再直接拼进 kwargs，先暂存到 pending_5d_features，收集完所有 item 后按 batch 最大 num_patches 零填充再 concat；模型侧 get_image_features 依据 image_sizes 自行还原真实 patch 数。

第 4 步：XPU 多模态调度路由调整

- python/sglang/srt/managers/mm_schedule.py 的 _get_chunked_prefill_embedding：路由条件从 _is_hip or _is_npu 扩展为 _is_hip or _is_npu or _is_xpu，XPU 与 ROCm/NPU 一样走 per-request 的 _get_chunked_embedding_by_item，结构上排除跨请求形状碰撞；代价是 XPU 上不再享受一次跨请求批量 ViT 的吞吐。

第 5 步：测试配套

- 新增 test/manual/models/test_transformers_collect_mm_kwargs.py（126 行）：以 SimpleNamespace 伪造 forward_batch/item，覆盖等 patch 数无 padding、不等 patch 数补零到 batch max、单 item 多图、decode 跳过收集、视频模态 key 五个场景；因面向 LLaVA 等特定 5D 回退模型，作者有意放 test/manual，不进 CI。

关键文件：

- python/sglang/srt/models/transformers.py（模块 回退模型；类别 source；类型 data-contract；符号 MultiModalMixin.init, MultiModalMixin._collect_mm_kwargs）：通用 transformers 回退路径的核心修复：新增 legacy vision_tower 键重映射与 5D pixel_values 批量最大 patch 零填充，是本 PR 改动量最大、影响面最广的文件。
- python/sglang/srt/models/qwen.py（模块 模型加载；类别 source；类型 data-contract；符号 QWenLMHeadModel.load_weights）：Qwen-VL-Chat 等 VL checkpoint 加载崩溃的根因修复，且 review 中 gemini-code-assist[bot] 指出后补了 stacked 映射分支的二层防御。
- python/sglang/srt/managers/mm_schedule.py（模块 多模态调度；类别 source；类型 dependency-wiring；符号 _get_chunked_prefill_embedding）：将 XPU 纳入

per-request 多模态编码路径，修复 Phi-4-multimodal 动态 HD 形状冲突崩溃，代价是放弃跨请求批量 ViT 吞吐。

- python/sglang/srt/utils/hf_transformers/common.py（模块 配置工具；类别 source；类型 core-logic；符号 get_rope_config）：get_rope_config() 对缺失 rope_theta 的 fallback 是 ERNIE-4.5-VL 与 Olmo-3 加载修复的地基，且 review 中确认了 v4/v5 的 None 语义差异。
- python/sglang/srt/models/ernie45_moe_vl.py（模块 模型实现；类别 source；类型 data-contract；符号 Ernie4_5_VLMoeDecoderLayer.init）：从直接索引 config.rope_parameters["rope_theta"] 改为统一 get_rope_config(config)，消除 ERNIE-4.5-VL 的 KeyError。
- python/sglang/srt/models/olmo2.py（模块 模型实现；类别 source；类型 data-contract；符号 Olmo2Attention.init）：Olmo-3 系列通过 get_rope_config 的 fallback 恢复正常加载，与 ERNIE 修复同源。
- test/manual/models/test_transformers_collect_mm_kwargs.py（模块 单元测试；类别 test；类型 test-coverage；符号 TestCollectMmKwargs5DPadding, test_equal_patch_counts_no_padding, test_different_patch_counts_padded_to_batch_max, test_multi_image_item_with_different_patch_counts_within_one_item）：新增 5D pixel_values padding 专项单元测试，覆盖等 / 不等 patch 数、单 item 多图、decode 跳过与视频 key；置于 test/manual 不进 CI。

关键符号：get_rope_config, MultiModalMixin._collect_mm_kwargs, MultiModalMixin.init, QWenLMHeadModel.load_weights, _get_chunked_prefill_embedding, Ernie4_5_VLMoeDecoderLayer.init, Olmo2Attention.init, TestCollectMmKwargs5DPadding

关键源码片段

python/sglang/srt/models/transformers.py

通用 transformers 回退路径的核心修复：新增 legacy vision_tower 键重映射与 5D pixel_values 批量最大 patch 零填充，是本 PR 改动量最大、影响面最广的文件。

```
# python/sglang/srt/models/transformers.py 中 MultiModalMixin._collect_mm_kwargs 的核心改造
# 背景：anyres 模型（如 LLaVA-OneVision）的 5D pixel_values 形状为
# (num_images, num_patches, C, H, W)，每个图片的 patch 数随分辨率 / 宽高比变化。
# 若直接沿 dim 0 flatten 拼接，不同 patch 数的 item 会形状不一致，导致
# torch.cat 抛错（例：Expected size 7 but got size 11），或残留错位 padding 行。

pending_5d_features: dict = {}

for batch_idx in range(len(mm_inputs or [])):
    mm_input = mm_inputs[batch_idx]
    if mm_input is None:
        continue
    for item in mm_input.mm_items or []:
        # model_specific_data 键值对沿用原有拼接逻辑
        for key, value in (item.model_specific_data or {}).items():
```

```
            if isinstance(value, torch.Tensor):
                value = value.to(device=target_device)
            if key not in kwargs:
                kwargs[key] = value
            elif isinstance(value, torch.Tensor) and isinstance(kwargs[key], torch.Tensor):
                kwargs[key] = torch.cat([kwargs[key], value], dim=0)

        if item.feature is not None:
            feature_key = self._mm_feature_kwarg.get(
                item.modality.name.lower(), "pixel_values"
            )
            feature = item.feature
            if isinstance(feature, torch.Tensor):
                feature = feature.to(device=target_device)
                # 5D feature 先暂存，等全部 item 收集完再统一处理
                if feature.dim() == 5:
                    pending_5d_features.setdefault(feature_key, []).append(feature)
                    continue
            # 非 5D feature 保持原有 concat 路径（batch 维拼接）
            if feature_key not in kwargs:
                kwargs[feature_key] = feature
            elif isinstance(feature, torch.Tensor) and isinstance(kwargs[feature_key], torch.Tensor)
            :
                kwargs[feature_key] = torch.cat([kwargs[feature_key], feature], dim=0)

    # 按 batch 内最大 patch 数零填充后再 concat，避免形状冲突；
    # 模型侧 get_image_features 会依据 image_sizes 自行切回真实 patch 数。
    for feature_key, tensors in pending_5d_features.items():
        max_patches = max(t.shape[1] for t in tensors)
        padded = []
        for t in tensors:
            if t.shape[1] < max_patches:
                pad = t.new_zeros((t.shape[0], max_patches - t.shape[1], *t.shape[2:]))
                t = torch.cat([t, pad], dim=1)
            padded.append(t)
        combined = torch.cat(padded, dim=0)
        if feature_key in kwargs:
            kwargs[feature_key] = torch.cat([kwargs[feature_key], combined], dim=0)
        else:
            kwargs[feature_key] = combined
```

python/sglang/srt/models/qwen.py

Qwen-VL-Chat 等 VL checkpoint 加载崩溃的根因修复，且 review 中 gemini-code-assist[bot] 指出后补了 stacked 映射分支的二层防御。

```
# python/sglang/srt/models/qwen.py 中 QWenLMHeadModel.load_weights 的关键改动
# 问题：Qwen-VL-Chat 等 VL checkpoint 与 QWenLMHeadModel 共用架构名，
# 但实现只覆盖文本主干，transformer.visual.* 视觉权重在 params_dict 中不存在，
# 原实现直接索引 params_dict[name] 会抛 KeyError。
```

```
def load_weights(self, weights):
    stacked_params_mapping = [
        # (param_name, shard_name, shard_id): gate_up_proj 由 w1/w2 拼接而来
        ("gate_up_proj", "w2", 0),
        ("gate_up_proj", "w1", 1),
    ]
    params_dict = dict(self.named_parameters())
    for name, loaded_weight in weights:
        if "rotary_emb.inv_freq" in name:
            continue
        for param_name, weight_name, shard_id in stacked_params_mapping:
            if weight_name not in name:
                continue
            # 先查 params_dict 再改名：视觉层 key 也可能命中 "w1"/"w2" 子串
            #  （如 transformer.visual.mlp.w1），缺失时直接跳过而非改名后崩溃
            temp_name = name.replace(weight_name, param_name)
            if temp_name not in params_dict:
                continue
            name = temp_name
            if name.endswith(".bias") and name not in params_dict:
                continue
            param = params_dict[name]
            weight_loader = param.weight_loader
            weight_loader(param, loaded_weight, shard_id)
            break
        else:
            # 非 stacked 权重：GPTQ bias 与视觉编码器权重都跳过
            if name.endswith(".bias") and name not in params_dict:
                continue
            if name not in params_dict:
                continue
            param = params_dict[name]
            weight_loader = getattr(param, "weight_loader", default_weight_loader)
            weight_loader(param, loaded_weight)
```

python/sglang/srt/managers/mm_schedule.py

将 XPU 纳入 per-request 多模态编码路径，修复 Phi-4-multimodal 动态 HD 形状冲突崩溃，代价是放弃跨请求批量 ViT 吞吐。

```
# python/sglang/srt/managers/mm_schedule.py 中 _get_chunked_prefill_embedding 的路由条件
# 背景：_batch_encode_per_image_misses 把所有并发请求的 per-image cache miss 合并进
# 一次跨请求 ViT 调用，前提是各项 feature 张量除 batch 维外形状一致。
# Phi-4-multimodal-instruct 的 dynamic-HD 预处理按图片自身分辨率决定 crop 数，
# 两个不同请求的图片 crop 数不同时，torch.cat 会抛形状不匹配错误。
# ROCm/NPU 早已改走 per-request 路径，此处把 XPU 并入同一安全路径。

_is_hip = is_hip()
_is_npu = is_npu()
_is_xpu = is_xpu()
```

```
# ...（循环内的核心分支）...
is_per_image = all(len(item.offsets) == 1 for item in embedding_items_per_req)
if is_per_image:
    if _is_hip or _is_npu or _is_xpu:
        # per-request 路径只处理本请求自己的 item，结构上排除跨请求形状碰撞
        chunk = _get_chunked_embedding_by_item(
            data_embedding_func,
            embedding_items_per_req,
            items_offset,
            extend_prefix_len,
            extend_seq_len,
            device,
        )
        if chunk is not None:
            all_chunks.append((i, chunk))
    else:
        # CUDA 继续走跨请求批量 ViT 的高吞吐路径
        per_image_requests.append(req_info)
```

评论区精华

review 中最有价值的交锋集中在两点：一是 gemini-code-assist[bot] 指出 qwen.py 初版修复不完整，视觉权重如 transformer.visual.mlp.w1 会在 stacked_params_mapping 循环内被 w1 子串匹配并改名后仍触发 KeyError，作者随后提交 4465c049 补上映射分支内的守卫；二是 mingfeima 质疑 transformers.py 中每次多模态 forward 后调用 torch.xpu.empty_cache() 会在服务热路径反复驱逐分配器缓存、伤害 XPU 性能，作者通过 A100 对照实验证明是 torch-xpu 缓存分配器碎片问题，最终移除 SGLang 侧 workaround 并计划向 torch-xpu 提 JIRA。另外 mingfeima 询问新增 UT 为何放 test/manual，作者说明其仅针对 LLaVA 等 5D 回退模型、适用范围窄，故不注册进 CI。

- stacked 映射分支内的 KeyError 遗漏 (correctness): 作者提交 4465c049 在映射循环内先用 temp_name 检查 params_dict，缺失即 continue，并在回复中确认 Handled the key error。
- torch.xpu.empty_cache 是否适合放在服务热路径 (performance): 作者移除 SGLang 侧 workaround（含 mistral.py 中同类改动），承诺向 torch-xpu 提 JIRA 跟进；该问题不再阻塞合并。
- 新增 UT 为何不进 CI (testing): 维持 manual 目录，不注册到任何 CI 套件。

风险与影响

- 风险：
  1. 兼容性回归：get_rope_config() 的返回语义在 v4/v5 配置间必须严格区分（None 与 {} 的差异），曾有 Qwen3-Next logprobs 测试失败的先例，后续新增模型配置时需回归验证。
  2. 静默跳过权重：qwen.py 对 params_dict 中不存在的 key 静默 continue，若未来模型新增参数或映射表拼写错误，加载期不再报错而是悄悄丢权重，排障成本上升。

3. 5D 零填充的假设：_collect_mm_kwargs 的 padding 依赖模型端 get_image_features 依据 image_sizes 自行切回真实 patch 数，若某模型不按此约定处理会产生精度误差；同时 batch 最大 patch 数填充会放大极端分辨率混合场景的显存与带宽占用。
4. XPU 性能回退：mm_schedule.py 将 XPU 移出跨请求批量 ViT 路径，图片并发高时 ViT 调用次数增加，吞吐下降。
5. 测试盲区：核心 padding 逻辑的 UT 放在 test/manual，常规 CI 无法拦截该逻辑的回归。 - 影响：用户侧：新增可加载并服务的模型包括 baidu/ERNIE-4.5-VL-28B-A3B-PT、allenai/Olmo-3-7B-Instruct、Qwen/Qwen-VL-Chat、llava-hf/llava-onevision-qwen2-0.5b-ov-hf、microsoft/Phi-4-multimodal-instruct（XPU 实测）。系统侧：transformers 通用回退路径成为更稳健的多模态兜底；XPU 多模态调度行为与 ROCm/NPU 对齐。团队侧：明确了 torch-xpu 缓存分配器碎片属于上游缺陷并计划提 JIRA，同时形成“特定硬件 / 模型场景的测试放 manual 目录”的取舍先例。影响程度中等，主要覆盖多模态与 XPU 场景，纯文本推理路径不受影响。 - 风险标记：XPU 多模态改 per-request 路径，吞吐回退，get_rope_config 语义变更有过 CI 回归先例，权重跳过策略可能掩盖模型不匹配，5D 零填充测试未进 CI

关联脉络

- PR #34995 [VLM] Avoid synchronizing multimodal placeholder counts: 同改 python/sglang/srt/managers/mm_schedule.py，同属 VLM 多模态调度热路径的持续优化与加固。
- PR #34988 [Diffusion] Reuse SRT SigLIP vision model: 涉及 vision tower（SigLIP）模型结构演进，与 transformers.py 中针对 transformers v5 扁平化 SigLIP 的权重重映射同主题。
- PR #35002 Support model-defined prefill input embedding width: 同为多模态 prefill 路径的兼容性扩展，可对照阅读 5D 特征处理与 input_embeds 宽度的配套改动。