

PR #31343 完整报告

sgl-project/sglang

Fix MiMo-V2 on Blackwell: FA3 fallback and TP-aware audio weight loading

合并时间: 2026-07-16 04:26

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/31343>

执行摘要

- 一句话: 修复 MiMo-V2 在 Blackwell 上的 FA3 回退与 TP 音频权重加载
- 推荐动作: 建议团队精读此 PR, 尤其是关注如何使用统一的 VisionAttention 接口解决硬件后端选择问题, 以及如何在 load_weights 中处理 TP 下的权重重映射。这些设计模式对后续支持新模型或新硬件具有参考价值。

功能与动机

Blackwell (sm103) 不支持 sgl-kernel 的 FA3, 导致 MiMo-V2 模型在加载时会因导入 flash_attn_varlen_func 失败而无法注册模型类, 抛出 ValueError。此外, TP 下音频编码器的权重加载不正确。Issue #30167 详细描述了此问题。

实现拆解

1. 替换 attention 后端: 在 mimo_audio.py 中移除硬编码的 flash_attn_varlen_func 导入, 改用 sglang.srt.layers.attention.vision.VisionAttention。VisionAttention 可根据 GPU 架构自动选择 FA3/FA4/flashinfer 等后端。同时新增 _audio_rope_applier 函数作为自定义 RoPE 应用器, 替换原有的 apply_rotary_pos_emb 方法。
2. TP 感知的音频权重重映射: 在 mimo_audio.py 中新增 _remap_audio_tokenizer_state_dict 函数, 用于 tensor parallelism 下正确重映射音频 tokenizer 权重。在 mimo_v2.py 的 load_weights 中新增音频投影的重映射逻辑: 将 q_proj/k_proj/v_proj 合并为 qkv_proj, 将 out_proj 重命名为 proj, 并跳过 input_local_transformer 层 (使用 Qwen2 原生 attention)。
3. 视觉编码器配置调整: 在 mimo_vl.py 中将 MiMoVisionBlock 的 attn_implementation 从 "flash_attention_3" 改为 None, 使视觉编码器也自动选择后端。
4. 清理 workaround: 在 mimo_v2_asr.py 中删除 _maybe_override_audio_attn_for_blackwell 函数及其调用, 因为新的架构不再需要运行时补丁。

关键文件:

- python/sglang/srt/models/mimo_audio.py (模块 音频编码器; 类别 source; 类型 data-contract; 符号 flash_attn_varlen_func, _audio_rope_applier, _rotate_half, apply_rotary_pos_emb) : 核心变更: 替换 attention 后端, 新增 RoPE applier 和 TP 权重重映射

- python/sglang/srt/models/mimo_v2_asr.py (模块 ASR 模型; 类别 source; 类型 cleanup; 符号 `_maybe_override_audio_attn_for_blackwell`) : 移除了 Blackwell workaround, 简化代码
- python/sglang/srt/models/mimo_v2.py (模块 MiMo 模型; 类别 source; 类型 data-contract) : 添加 TP 感知的音频权重重映射
- python/sglang/srt/models/mimo_vl.py (模块 视觉模块; 类别 source; 类型 configuration) : 将视觉编码器的 `attn_implementation` 改为 `None` 以启用自动后端选择

关键符号: `_audio_rope_applier`, `_remap_audio_tokenizer_state_dict`,
`AudioEncoderAttention.init`, `AudioEncoderAttention.forward`,
`MiMoV2ForCausalLM.load_weights`, `_maybe_override_audio_attn_for_blackwell`,
`MiMoVisionBlock.init`

关键源码片段

python/sglang/srt/models/mimo_audio.py

核心变更: 替换 attention 后端, 新增 RoPE applier 和 TP 权重重映射

```
# path: python/sglang/srt/models/mimo_audio.py (head)
# 核心变更: 移除硬编码 FA3 导入, 使用 VisionAttention 并新增 RoPE 应用器
```

```
from sglang.srt.layers.attention.vision import VisionAttention
```

```
def _audio_rope_applier(
    q: torch.Tensor,
    k: torch.Tensor,
    position_embeddings: Tuple[torch.Tensor, torch.Tensor],
    x_shape,
) -> Tuple[torch.Tensor, torch.Tensor]:
    """音频专用的半旋转位置编码实现"""
    cos, sin = position_embeddings
    cos = cos.unsqueeze(1)
    sin = sin.unsqueeze(1)
    x1_q, x2_q = q[..., : q.shape[-1] // 2], q[..., q.shape[-1] // 2 :]
    x1_k, x2_k = k[..., : k.shape[-1] // 2], k[..., k.shape[-1] // 2 :]
    q_embed = q * cos + torch.cat((-x2_q, x1_q), dim=-1) * sin
    k_embed = k * cos + torch.cat((-x2_k, x1_k), dim=-1) * sin
    return q_embed, k_embed
```

```
class AudioEncoderAttention(nn.Module):
    def __init__(self, embed_dim, num_heads, window_size=(-1,-1), causal=False):
        super().__init__()
        # 使用 VisionAttention 替代原来的四个独立线性层
        # 该实例会根据当前 GPU 计算能力自动选择最佳 attention 后端
        self.attn = VisionAttention(
            embed_dim=embed_dim,
            num_heads=num_heads,
            projection_size=embed_dim,
```

```

        use_qkv_parallel=True, # 并行计算 QKV
        qkv_bias=True,
        proj_bias=True,
        flatten_batch=True,
        window_size=window_size,
        customized_position_embedding_applier=_audio_rope_applier,
        prefix="attn",
    )

def forward(self, hidden_states, cu_seqlens, max_seqlen, rope_position_embeddings=None):
    # 委托给 VisionAttention, 自动完成 QKV 投影、位置编码和 attention 计算
    out = self.attn(
        hidden_states,
        cu_seqlens=cu_seqlens,
        position_embeddings=rope_position_embeddings,
        max_seqlen=max_seqlen,
    )
    return out

```

python/sglang/srt/models/mimo_v2.py

添加 TP 感知的音频权重重映射

path: python/sglang/srt/models/mimo_v2.py (head) — load_weights 新增的音频重映射
 # 在原有音频权重加载分支中, 增加了对 TP 下投影合并的支持

```

if self._is_multimodal and is_audio_weight:
    if name.startswith("audio_encoder."):
        name = name[len("audio_encoder."):]
    name = self.remap_audio_weight_name(name)
    # 跳过 input_local_transformer (该层使用 Qwen2 原生 attention, 无需重映射)
    if "input_local_transformer" not in name:
        # out_proj -> proj
        name = name.replace("self_attn.out_proj", "self_attn.attn.proj")
        audio_stacked = False
        # 合并 q/k/v 投影为 qkv_proj 并切片加载
        for param_name, weight_name, shard_id in [
            ("self_attn.attn.qkv_proj", "self_attn.q_proj", "q"),
            ("self_attn.attn.qkv_proj", "self_attn.k_proj", "k"),
            ("self_attn.attn.qkv_proj", "self_attn.v_proj", "v"),
        ]:
            if weight_name in name:
                name = name.replace(weight_name, param_name)
                if name not in params_dict:
                    break
                param = params_dict[name]
                weight_loader = param.weight_loader
                weight_loader(param, loaded_weight, shard_id)
                audio_stacked = True
                break

```

```
        if audio_stacked:
            continue
# 其他音频权重加载逻辑保持不变
if name not in params_dict:
    logger.warning(f"Audio param {name} not found, skipping")
    continue
param = params_dict[name]
weight_loader = getattr(param, "weight_loader", default_weight_loader)
if self._AUDIO_WEIGHT_SUBSTRING in name:
    weight_loader(param, loaded_weight[: param.shape[0], :])
else:
    weight_loader(param, loaded_weight)
continue
```

评论区精华

PR 仅有一个 APPROVED review，未提出任何评论或问题。所有变更由作者自行推进，经 reviewer Fridge003 批准后直接合入。

- 暂无高价值评论线程

风险与影响

- 风险：

1. 兼容性风险：VisionAttention 的自动后端选择可能在非主流 GPU 上产生意外行为，但该模块已在视觉模型中广泛使用，风险较低。
2. 权重映射风险：mimo_v2.py 中新增的音频权重重映射对 checkpoint 格式有一定假设，若遇到不同格式的 checkpoint，可能导致权重静默跳过或加载失败。
3. 回归风险：mimo_v2_asr.py 移除了 workaround，若未来出现其他 FA3 不可用场景，新架构应已解决此类问题，回归可能性低。
4. 测试覆盖：未添加自动化测试，仅依赖手动 benchmark 验证，缺少 CI 测试保护。- 影响：影响范围限于 MiMo-V2 系列模型（包括 ASR 变体）。正面影响：使 MiMo-V2 可在 Blackwell GPU（如 GB300）上正常启动和推理，TP 下音频权重加载正确。对非 Blackwell 用户无影响，对视觉模块和其他模型无影响。- 风险标记：缺少自动化测试覆盖，权重映射依赖 checkpoint 格式，VisionAttention 后端选择兼容性

关联脉络

- 暂无明显关联 PR