

PR #50958 完整报告

vllm-project/vllm

[Bugfix][Model] Gemma3n/Gemma4: pad variable-length audio batches

合并时间: 2026-08-05 12:46

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/50958>

执行摘要

- 一句话: 修复 Gemma3n/Gemma4 变长音频批处理崩溃
- 推荐动作: 值得精读。三个看点: (1) 这是典型的多模态输入管道 " 隐式契约断裂 " 案例——缓存解 pad、batched() 字段配置、TensorSchema dynamic_dims 三方各自合理但组合失效, 根因分析链条完整 (含日志证据与版本矩阵比对); (2) review 驱动测试升级 (单元测试 → e2e 回归测试) 与 helper 归属调整展示了 vLLM 维护者对测试价值与模块边界的偏好; (3) 修复手法 (zero-pad + mask 双返回、padding 不泄漏进 LM) 可作为后续多模态模型处理变长输入的模板。建议关注合并后 Gemma3n profile/dummy input 路径是否有回归。

功能与动机

Issue #50957 报告: 四个并发请求、每个携带一个音频 clip, 一旦时长不同就导致 `AttributeError: 'list' object has no attribute 'squeeze'`, 进而 `EngineDeadError` 让 `EngineCore` 宕机, 所有在途请求失败。PR body 明确指出根因链条: `_call_hf_processor` 为了多模态缓存条目自包含而按 item 解 pad, 但其注释声称 `_get_mm_fields_config` 会在 batch 时重新 padding, 实际上 `MultiModalFieldConfig.batched` 只 stack 形状一致的条目, 其余返回 list; `Gemma4AudioInputs` 通过 `dynamic_dims={"s"}` 放行该 list, `_process_audio_input` 却把它当 tensor 处理。Gemma3n 因 schema 缺少 `dynamic_dims` 会更早一步被 `TensorSchema` 以 `ValueError: inconsistent shapes` 拒绝, 该问题已被报告两次 (#24006、#38297) 仍未修复。

实现拆解

实现拆解 (共 4 步):

1. 梳理并确认根因链路: 定位到三个环节的契约断裂——`Gemma4MultiModalProcessor._call_hf_processor` 按 item 解 pad 音频特征 (vllm/model_executor/models/gemma4_mm.py); `MultiModalFieldConfig.batched` 仅在形状一致时 stack; `TensorSchema` 的 `dynamic_dims={"s"}` 放行不同长度的 list 进入模型前向。作者在 PR body 中给出日志证据: 形状 `[60,128]x2` 时正常 stack, `[51,128]/[54,128]` 混合时得到 python list。
2. 新增共享辅助函数 `batch_audio_features` (vllm/model_executor/models/gemma3n_mm.py): 输入为 Tensor 时走原 `squeeze(1)` 快路径 (等长 batch); 输入为 list 时按 batch 内最长 clip 的帧数分配 (bn, s_max, f) 零张量与 (bn, s_max) 的 bool mask, 逐条拷贝有效帧, 填充帧保持 0 且 mask 为 False (invalid)。设计要点是填充帧由 audio tower 消

费 mask、调用方只保留 mask 内位置，保证 padding 永不进入语言模型。

3. 修复 schema 与三个调用点：Gemma3nAudioInputs 的 input_features_padded 与 input_features_mask 字段补上 dynamic_dims={"s"}（与 Gemma4 现有配置对齐），否则在进入模型前就被 TensorSchema 拒绝；gemma3n_mm.py、gemma4_mm.py、gemma4_unified.py 三处 _process_audio_input 统一改调 batch_audio_features，并同步修正 gemma4_mm.py 中关于 "will re-pad all fields" 的误导性注释。
4. 测试与 review 演进配套：初始提交包含 audio_utils.py 单元测试，经 Isotr0py review 后改为 e2e 回归测试 tests/models/multimodal/generation/test_gemma_ragged_audio.py（参数化 Gemma3n/Gemma4 两个架构），helper 也从独立模块移入 gemma3n_mm.py 以避免新增公共模块和循环依赖；该测试将单条资产切片为 1.0/3.5/7.0/12.0s 四个 clip 并断言长度互异，单次 generate 调用保证同时进入同一 forward pass 从而必然触发 ragged 路径。

关键文件：

- vllm/model_executor/models/gemma3n_mm.py（模块 多模态模型；类别 source；类型 data-contract；符号 batch_audio_features, Gemma3nAudioInputs, _process_audio_input）：核心修复文件：新增共享辅助函数 batch_audio_features（Tensor 快路径 + list 零填充 + mask 生成），为 Gemma3nAudioInputs 的 input_features_padded/input_features_mask 补上 dynamic_dims={"s"} schema 配置，并改造 _process_audio_input 调用新 helper。Gemma4 系列两个文件都从这里导入 helper。
- tests/models/multimodal/generation/test_gemma_ragged_audio.py（模块 回归测试；类别 test；类型 test-coverage；符号 ragged_clips, test_variable_length_audio_batching）：新增 e2e 回归测试：把单条音频资产切片为四个长度互异的 clip，单次 generate 调用强制同时进入同一 forward pass，参数化覆盖 Gemma3n 与 Gemma4 两个架构；此前该路径只能靠并发时序碰巧触发，无任何回归保护。
- vllm/model_executor/models/gemma4_mm.py（模块 多模态模型；类别 source；类型 core-logic；符号 Gemma4MultiModalProcessor._call_hf_processor, Gemma4ForConditionalGeneration._process_audio_input）：崩溃实际发生点：_process_audio_input 对 list 调用 .squeeze(1) 导致 AttributeError。改为调用共享 helper，并修正 _call_hf_processor 中声称 batched() 会重新 padding 的误导性注释（实际从未发生）。
- vllm/model_executor/models/gemma4_unified.py（模块 多模态模型；类别 source；类型 core-logic；符号 Gemma4UnifiedForConditionalGeneration._process_audio_input）：无音频 tower 的 Unified 变体同样存在该 bug（对 list .squeeze(1)），调用点改为 batch_audio_features；作者明确此路径仅经单元测试和代码审查验证，无 e2e 覆盖。

关键符号：batch_audio_features, Gemma3nAudioInputs, gemma3n_mm._process_audio_input, gemma4_mm._process_audio_input, gemma4_unified._process_audio_input, test_variable_length_audio_batching, ragged_clips

关键源码片段

tests/models/multimodal/generation/test_gemma_ragged_audio.py

新增 e2e 回归测试：把单条音频资产切片为四个长度互异的 clip，单次 generate 调用强制同时进入同一 forward pass，参数化覆盖 Gemma3n 与 Gemma4 两个架构；此前该路径只能靠并发时序碰巧触发，无任何回归保护。

```
# tests/models/multimodal/generation/test_gemma_ragged_audio.py
# 单条资产切片成 4 个不同时长 clip，保证只有长度变化，
# 且必然走到 ragged 路径（等长 clip 会正常 stack，测不到 bug）。
@pytest.fixture(scope="module")
def ragged_clips():
    audio, sample_rate = AudioAsset("mary_had_lamb").audio_and_sample_rate
    clips = []
    for duration in CLIP_DURATIONS: # (1.0, 3.5, 7.0, 12.0)
        num_samples = int(duration * sample_rate)
        if num_samples > len(audio):
            pytest.skip(f"asset is shorter than {duration}s")
        clips.append((audio[:num_samples], sample_rate))
    # 显式断言四个 clip 长度互不相同，避免测试因巧合而空转
    assert len({len(clip) for clip, _ in clips}) == len(clips)
    return clips

@pytest.mark.core_model
@pytest.mark.parametrize("arch", sorted(MODELS)) # Gemma3n / Gemma4
@pytest.mark.parametrize("max_tokens", [8])
def test_variable_length_audio_batching(vllm_runner, ragged_clips, arch, max_tokens):
    # 从 processor 读取音频占位符，两个架构的拼写不同，不能硬编码
    processor = AutoProcessor.from_pretrained(model_name)
    prompt = processor.apply_chat_template(
        [
            {
                "role": "user",
                "content": [
                    {"type": "audio", "audio": None},
                    {"type": "text", "text": "Transcribe this audio."},
                ],
            }
        ],
        tokenize=False,
        add_generation_prompt=True,
    )
    with vllm_runner(
        model_name,
        dtype="bfloat16",
        enforce_eager=True,
        limit_mm_per_prompt={"audio": 1},
    ) as vllm_model:
        # 单次调用让所有 clip 被调度进同一 forward pass;
```

```
# 分开提交永远到不了 ragged 路径，测试也就测不到这个 bug。
outputs = vllm_model.generate_greedy(
    [prompt] * len(ragged_clips),
    max_tokens,
    audios=[[clip] for clip in ragged_clips],
)
assert len(outputs) == len(ragged_clips)
for output in outputs:
    assert len(output[1]) > 0, "Expected non-empty output"
```

评论区精华

Isotr0py 提出两条实质性 review 意见，作者均在第二个提交中落实：

Isotr0py (tests/models/multimodal/processing/test_audio_utils.py) : “Can you add an e2e test for gemma3/gemma4 instead?”

作者据此删除单元测试，新写 e2e 测试 `test_variable_length_audio_batching`，并模仿既有 `test_ultravox.py` 同名测试的模式（`vllm_runner` + skip-guard + 非空输出断言）。作者特别说明与 ultravox 测试的刻意差异：不依赖两个资产“碰巧”时长不同，而是把单条资产切片成 4 个长度确定互异的 clip，并显式 `assert len({len(clip) ...}) == len(clips)`，避免等长 clip 正常 stack 导致测试空转。

Isotr0py (vllm/model_executor/models/audio_utils.py) : “Let's move to `gemma3n_mm.py` and import it for gemma4”

作者回应：helper 移入 `gemma3n_mm.py` 后无循环依赖（`gemma3n_mm` 不回引用 `gemma4` 模块），三个模型最终持有同一函数对象。另外作者在评论中说明一次 DCO 修复的 force-push 禁用了已排队的 auto-merge，树内容与获批版本 byte-identical，CI 全绿（89 项 Buildkite 检查）。两条 review 意见均已在合并前解决，无未解决疑虑。

- 单元测试替换为 e2e 回归测试 (testing): 作者删除单元测试，新增 `tests/models/multimodal/generation/test_gemma_ragged_audio.py`，参数化 Gemma3n 与 Gemma4，模仿 `test_ultravox.py::test_variable_length_audio_batching` 的骨架，并刻意用同源资产切片保证四个 clip 时长互异、必现 ragged 路径。
- 共享 helper 的放置位置 (design): 作者将 `batch_audio_features` 移入 `gemma3n_mm.py`，`gemma4_mm.py` 与 `gemma4_unified.py` 从该处导入；作者确认无循环依赖，三个模型持有同一函数对象，`audio_utils.py` 删除。
- DCO 修复导致 auto-merge 被禁用 (other): 流程性事件，无技术争议；最终 PR 由 Isotr0py 合并。

风险与影响

- 风险：
 1. 显存与性能: ragged 路径按 batch 内最长 clip 分配 (bn, s_max, f) 零填充张量，极端长短混合场景峰值显存与最差 padding 量成正比；不过 padding 通过 mask 被 tower 消费后即被剪掉，不会进入语言模型。该模式与 #31223 为 Qwen2Audio 等模型引入的

dynamic_dims 机制一致，属既有路径的推广，风险可控。

2. 数据契约变更：Gemma3nAudioInputs 新增 dynamic_dims={"s"} 属于 schema 级变更，dummy inputs / profile 阶段同样经过 TensorSchema 校验，存在 profile 与运行时行为不一致的潜在回归面，需关注后续 profile 相关测试。
3. 跨模块耦合：gemma4_mm.py 与 gemma4_unified.py 从 gemma3n_mm.py 导入 helper，模型文件间新增依赖；若未来 batch_audio_features 签名或语义变化，将同时影响三个模型。作者已确认无循环依赖。
4. 测试盲区：e2e 测试只参数化 Gemma3n 与 Gemma4，Gemma4Unified 不在其中；作者在 PR body 中明确 Gemma3n 与 Gemma4Unified 仅经单元测试和代码审查验证 ("by inspection only")，无 serving 环境实测。
 - 影响：用户侧：修复了音频多模态服务在并发不同时长音频请求下的硬崩溃，服务不再因单个混合时长 batch 整体宕机，直接提升生产可用性；影响模型为 google/gemma-3n 系列与 google/gemma-4 系列（含 Unified 变体）。系统侧：改动集中在 3 个模型文件 + 1 个新增测试，共 165 行，用单一共享 helper 取代原先散落各处的约 15 行 padding 逻辑（避免三份拷贝），并把 "zero-pad + mask 标记 invalid" 确立为 Gemma 系列音频路径的统一约定。团队侧：该修复为后续多模态模型处理变长输入提供了可复用的参考模式，也与 Qwen2Audio / AudioFlamingo3 / MiniCPM-V 已有的 dynamic_dims 路径对齐。总体影响范围中等偏正面：修复的是服务崩溃级 bug，但仅涉及音频多模态场景，不触碰核心调度与执行路径。
 - 风险标记：核心路径变更（多模态输入管道），跨模块导入耦合，schema 契约变更（dynamic_dims），部分模型缺少 e2e 验证

关联脉络

- PR #50957 [Bug]: Gemma4/Gemma3n audio — concurrent requests with different clip durations kill EngineCore: 本 PR 直接修复的 issue，提供复现脚本、崩溃栈与根因分析（v0.19.1 与 main 字节级一致）。
- PR #31223 （标题未在材料中提供）为 Qwen2Audio / AudioFlamingo3 / MiniCPM-V 引入 dynamic_dims 机制的 PR: PR body 明确指出 Gemma4AudioInputs 放行 list 所依赖的 dynamic_dims={"s"} 机制源自 #31223，本修复是该机制的推广与补齐。
- PR #44546 （标题未在材料中提供）同主题 gemma4_mm.py 音频 padding 修复 PR: 作者自述该 PR 已获批且 gemma4_mm.py 改动与本 PR 等价；作者表示若维护者倾向合并它，愿将本 PR rebase 到其未覆盖的部分。
- PR #45887 （标题未在材料中提供）同主题 gemma4_mm.py 音频 padding 修复 PR: 六个同主题未合并 PR 中唯一附带回归测试的，作者将其作为本 PR 测试方案的参照之一。