PR #42233 完整报告

vllm-project/vllm

[Bugfix] Fix scipy audio resampling ratio

合并时间：2026-05-13 18:52

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/42233

执行摘要

- 一句话：修复 scipy 音频重采样比率计算错误
- 推荐动作：这是一个经过充分验证的精准 bugfix，变更逻辑清晰、测试完备（从 xfail 到新增回归），建议快速合并。值得关注的设计决策是：使用 GCD 计算约分比率，比直接使用浮点比率或条件分支更精确且无精度损失。

功能与动机

PR body 指出：当原始和目标采样率不能整除时，之前的整数除法实现会导致比率错误，例如 44100 -> 16000 被当作 1/2 而不是 160/441，从而产生错误的输出长度。该功能被 Phi-4-MM 等多模态模型使用。

实现拆解

1. 修复比率计算（vllm/multimodal/audio.py）：将两路带整数除法的 if/elif 分支替换为统一的 GCD 方案。先将 orig_sr 和 target_sr 四舍五入为整数，若相等则直接返回。否则用 math.gcd 计算最大公约数，以约分后的整数作为 resample_poly 的上 / 下采样因子。
2. 指定重采样轴：在 scipy_signal.resample_poly 调用中新增 axis=-1 参数，确保多声道音频（(channels, samples)）沿时间轴重采样，保持声道数不变。
3. 启用并新增测试（tests/multimodal/test_audio.py）：移除 test_resample_audio_scipy_non_integer_ratio 上的 @pytest.mark.xfail，使其验证修复。新增 test_resample_audio_scipy_non_divisible_sample_rates（44100 -> 16000 转换）和 test_resample_audio_scipy_resamples_last_axis_for_multichannel（验证多声道输入输出形状正确）。并添加了 import math。

关键文件：

- vllm/multimodal/audio.py（模块 音频处理；类别 source；类型 core-logic）：核心 bugfix 所在，修改了 scipy 音频重采样比率的计算方式，并指定 axis=-1 正确支持多声道。
- tests/multimodal/test_audio.py（模块 音频处理；类别 test；类型 test-coverage；符号 test_resample_audio_scipy_non_divisible_sample_rates, test_resample_audio_scipy_resamples_last_axis_for_multichannel）：启用了已有的 xfail 测试，新增了两个回归测试覆盖不可整除比率和多声道场景。

关键符号：resample_audio_scipy

关键源码片段

vllm/multimodal/audio.py

核心 bugfix 所在，修改了 scipy 音频重采样比率的计算方式，并指定 axis=-1 正确支持多声道。

```
def resample_audio_scipy(
    audio: npt.NDArray[np.floating],
    *,
    orig_sr: float,
    target_sr: float,
) -> npt.NDArray[np.floating]:
    # 先将浮点采样率四舍五入为整数
    orig_sr_int = int(round(orig_sr))
    target_sr_int = int(round(target_sr))

    # 相等时直接返回原音频，避免无意义的计算
    if orig_sr_int == target_sr_int:
        return audio

    # 使用最大公约数计算最简重采样比率
    # 例如 44100 -> 16000：up = 16000//100 = 160，down = 44100//100 = 441
    gcd = math.gcd(orig_sr_int, target_sr_int)
    # axis=-1 确保多声道音频沿时间轴重采样，而不是声道轴
    return scipy_signal.resample_poly(
        audio,
        target_sr_int // gcd,
        orig_sr_int // gcd,
        axis=-1,
    )
```

tests/multimodal/test_audio.py

启用了已有的 xfail 测试，新增了两个回归测试覆盖不可整除比率和多声道场景。

```
def test_resample_audio_scipy_non_divisible_sample_rates():
    # 常见场景：44100Hz 重采样到 16000Hz，不可整除
    audio = np.arange(441, dtype=float)
    out = resample_audio_scipy(audio, orig_sr=44100, target_sr=16000)

    expected_len = math.ceil(len(audio) * 16000 / 44100)
    assert len(out) == expected_len
    assert isinstance(out, np.ndarray)
    assert np.isfinite(out).all()


def test_resample_audio_scipy_resamples_last_axis_for_multichannel():
    # 验证多声道音频（channels, samples）的形状保持
    audio = np.arange(2 * 441, dtype=float).reshape(2, 441)
    out = resample_audio_scipy(audio, orig_sr=44100, target_sr=16000)
```

```
    expected_len = math.ceil(audio.shape[-1] * 16000 / 44100)
    assert out.shape == (2, expected_len)
    assert np.isfinite(out).all()
```

评论区精华

AI 代码审查机器人（gemini-code-assist[bot]）指出 resample_poly 默认 axis=0，对多声道音频会错误地沿声道轴重采样，建议添加 axis=-1。PR 作者采纳该建议并添加了多声道回归测试。维护者 Isotr0py 要求额外对 Phi-4-MM 模型集成测试验证，作者运行后报告 4 passed / 3 failed（3 个失败与本次变更无关，是上游 transformers 兼容性问题）。

- 多声道音频的 axis 参数 (correctness): 作者采纳建议，在 resample_poly 调用中添加 axis=-1，并添加多声道回归测试验证形状正确。
- Phi-4-MM 集成测试验证 (testing): 作者运行测试，4 passed / 3 failed（3 个失败是由于上游 transformers 兼容性问题，与本变更无关）。

风险与影响

- 风险：
  1. 回归风险：低。核心变更从整数除法改为 GCD 约分，在可整除情况下（如 4->2）结果相同。新增参数 axis=-1 仅影响多声道输入，不影响单声道。
  2. 性能影响：可忽略。GCD 计算是 O(log n) 的极轻量操作。
  3. 兼容性：无。函数签名和返回值类型均未改变。
  4. 测试覆盖：已知的 xfail 测试已被启用，新增了不可整除比率和多声道测试，且通过了 Phi-4-MM 的全量集成测试（除上游不兼容部分）。 - 影响：影响范围：仅影响使用 resample_audio_scipy 的路径，目前只有 Phi-4-MM 等模型依赖此函数。影响程度：中。修复前不可整除的采样率转换（如常见的 CD 44.1kHz 转 16kHz）会产生错误长度的音频，可能导致下游任务异常或静默截断。用户体验：修复后音频预处理行为符合预期。维护成本：极低，代码行数减少，逻辑更简洁。 - 风险标记：暂无

关联脉络

- 暂无明显关联 PR