

PR #53704 完整报告

vllm-project/vllm

[Bugfix] Handle empty FlatLogprobs slices and delta output

合并时间: 2026-08-26 21:00

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/53704>

执行摘要

- 一句话: 修复 FlatLogprobs 空切片与 delta 输出 -0 溢出的 bug
- 推荐动作: 该 PR 值得精读: 一是 Python 中 -0 与切片边界的经典陷阱, 二是序列协议实现中空切片处理的正确姿势。可作为后续 logprobs 相关维护的参考样例。建议关注 delta 输出路径与 FlatLogprobs 的交互测试是否覆盖更多切片边界 (如 step 非 1)。

功能与动机

Issue#37037 报告了两个相关问题: FlatLogprobs.__getitem__ 对空切片 (如 logprobs[0:0]、logprobs[5:5]) 无条件访问 start_indices[index][0] 和 end_indices[index][-1], 对空列表索引抛 IndexError, 违反 Python 序列协议 (标准 list 返回 []); 同时 vllm/v1/engine/output_processor.py 中 logprobs[-len(token_ids):] 在 token_ids 为空时 -0 等价于 0, 导致返回所有累积 logprobs, delta 流式模式下向客户端发送 stale/duplicate 数据。PR body 明确目标: 返回空 FlatLogprobs 而非抛异常, 并避免无新 token 时切片 -0。

实现拆解

1. 空切片修复 (vllm/logprobs.py): slice 分支先取 selected_starts = self.start_indices[index] 与 selected_ends = self.end_indices[index]; 若 selected_starts 为空, 直接 return FlatLogprobs(), 避免对空列表取 [0]/-1; 非空时仍以 min_index 归一化索引, 保持既有切片语义不变。
2. Delta 输出修复 (vllm/v1/engine/output_processor.py): RequestState._new_completion_output 中新增 num_new_tokens = len(token_ids); 有新增 token 时用 logprobs[-num_new_tokens:] 截取增量, 无新增 token 时用 logprobs[:0], 两种写法都保留具体容器类型 (list 或 FlatLogprobs) 且返回空内容; 注释说明避免 [-0:] 的原因。
3. 回归测试配套: tests/test_logprobs.py 对 slice(0, 0)、slice(3, 3)、slice(10, 10) 参数化断言返回空 FlatLogprobs 且 start_indices/end_indices 为空列表; tests/v1/engine/test_output_processor.py 新增 test_delta_output_without_new_tokens_returns_empty_logprobs, 用 MagicMock 构造带累积 logprobs 的 RequestState, 参数化 flat_logprobs 覆盖 list 与 FlatLogprobs 两种容器, 验证 delta 且无新 token 时输出 logprobs 长度为 0。
4. 配套说明: 无新增配置、schema 或部署变更, 改动仅在输出处理与测试路径。

关键文件：

- vllm/logprobs.py（模块 概率日志；类别 source；类型 core-logic；符号 FlatLogprobs.getitem）：核心修复文件：FlatLogprobs.__getitem__ 空切片分支直接返回空 FlatLogprobs，避免 IndexError，并将 selected_starts/selected_ends 提取为局部变量以统一处理空与非空切片。
- vllm/v1/engine/output_processor.py（模块 输出处理；类别 source；类型 core-logic；符号 RequestState._new_completion_output）：delta 输出路径修复：_new_completion_output 中避免 -0 切片，无新 token 时用 logprobs[:0] 返回空容器，保留具体类型。
- tests/v1/engine/test_output_processor.py（模块 输出测试；类别 test；类型 test-coverage；符号 test_delta_output_without_new_tokens_returns_empty_logprobs）：新增 delta 无新 token 场景的参数化回归测试，覆盖 list 与 FlatLogprobs 两种容器。
- tests/test_logprobs.py（模块 概率日志测试；类别 test；类型 test-coverage）：补充 FlatLogprobs 空切片回归测试，覆盖多个空切片位置并断言返回空 FlatLogprobs。

关键符号：FlatLogprobs.getitem, RequestState._new_completion_output, test_delta_output_without_new_tokens_returns_empty_logprobs

关键源码片段

vllm/logprobs.py

核心修复文件：FlatLogprobs.__getitem__ 空切片分支直接返回空 FlatLogprobs，避免 IndexError，并将 selected_starts/selected_ends 提取为局部变量以统一处理空与非空切片。

```
def __getitem__(self, index: int | slice):
    """按位置或切片取出 logprobs"""
    if isinstance(index, int):
        # 单个位置：从 start_indices[index] 到 end_indices[index] 收集该位置所有候选 token
        return {
            self.token_ids[i]: Logprob(
                logprob=self.logprobs[i],
                rank=self.ranks[i],
                decoded_token=self.decoded_tokens[i],
            )
            for i in range(self.start_indices[index], self.end_indices[index])
        }
    elif isinstance(index, slice):
        selected_starts = self.start_indices[index]
        selected_ends = self.end_indices[index]
        # 空切片没有可归一化的起始偏移，直接返回空容器，避免 IndexError
        if not selected_starts:
            return FlatLogprobs()
        min_index = selected_starts[0]
        max_index = selected_ends[-1]
        return FlatLogprobs(
            # 将子序列的 start/end 偏移归一到 0 基，便于上层按位置访问
```

```

        start_indices=[i - min_index for i in selected_starts],
        end_indices=[i - min_index for i in selected_ends],
        token_ids=self.token_ids[min_index:max_index],
        logprobs=self.logprobs[min_index:max_index],
        ranks=self.ranks[min_index:max_index],
        decoded_tokens=self.decoded_tokens[min_index:max_index],
    )
else:
    raise TypeError(f"Invalid index type: {type(index)}")

```

vllm/v1/engine/output_processor.py

delta 输出路径修复：_new_completion_output 中避免 -0 切片，无新 token 时用 logprobs[:0] 返回空容器，保留具体类型。

```

# 按 delta 模式准备 logprobs
logprobs = self.logprobs_processor.logprobs
if delta and logprobs:
    num_new_tokens = len(token_ids)
    # 避免 [-0:]: 当 delta 无新 token 时，-0 会被当作 0，
    # 导致返回完整累积历史；[:0] 保留具体容器类型 (list 或 FlatLogprobs)
    # 同时返回空内容。
    logprobs = logprobs[-num_new_tokens:] if num_new_tokens else logprobs[:0]

```

评论区精华

NolenLiang 在 CHANGES_REQUESTED 中指出早期提交的 `if delta and logprobs and token_ids` 条件会在 token_ids 为空时跳过切片，导致 logprobs 仍指向完整累积历史并传给 CompletionOutput，issue #37037 中的 stale/duplicate 问题未解决。他建议改为显式计算 num_new_tokens，并在为 0 时使用 logprobs[:0] 以保留具体容器类型。作者采纳该建议并补充了针对 list 与 FlatLogprobs 的回归测试；最终由 aoshen02 批准、ZJY0516 合入。

- delta 空 token 时旧修复仍泄漏完整 logprobs (correctness): 作者采纳建议，改为 `logprobs = logprobs[-num_new_tokens:] if num_new_tokens else logprobs[:0]`，并补充针对 list 与 FlatLogprobs 的回归测试。最终获得批准并合入。

风险与影响

- 风险：
 1. 行为变更：FlatLogprobs 空切片从不抛异常改为返回空容器，与 list 对齐，依赖旧异常捕获的调用方需检查。
 2. Delta 输出语义修正：无新 token 时不再返回累积 logprobs，可能影响依赖旧（错误）行为的客户端。
 3. 作用面：vllm/logprobs.py 是通用 logprobs 容器，被 prompt/sample logprobs 多处引用，空切片路径属于通用路径，回归面值得关注。
 4. 测试覆盖：新增测试覆盖 list 与 FlatLogprobs 两种容器及多个空切片位置，但未覆盖带 step 的切片等更复杂边界。- 影响：影响所有使用 v1 引擎且开启 logprobs + DELTA 输出模式的用户，特别是多模态或 spec decode 场景下可能出现无新 token 的迭代；同

时修复任何对 FlatLogprobs 做空切片操作的调用方。影响范围中等，但都是修正错误行为，对客户端而言流式输出不再收到重复或过期的 logprobs 数据。 - 风险标记：核心路径小改动，delta 输出语义修正，空切片协议行为变更

关联脉络

- 暂无明显关联 PR