PR #43125 完整报告

vllm-project/vllm

[BugFix] Use correct logprobs for `logprob_token_ids`

合并时间：2026-05-22 06:42

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/43125

执行摘要

- 一句话：修复 logprob_token_ids 使用错误 logits 而非 logprobs
- 推荐动作：值得阅读以了解 MRV1 sampler 中 logprobs 数据流的正确模式。关注点：gather 操作的输入选择（logits vs logprobs）以及 torch dynamo 优化技巧（mark_unbacked 虽未被采纳但值得借鉴）。

功能与动机

当使用 logprob_token_ids 选项时（用于 generative scoring），sampler 应使用正确的 logprobs 模式（raw_logprobs 或 raw_logits）以确保与 top-k logprobs 一致。原代码直接对 logits 进行 gather，未考虑 logprobs 模式，且不必要地重新计算 logprobs（来自 logits 的 softmax）。PR body 明确指明这是一个 MRV1 特有的 bug，MRV2 不受影响。

实现拆解

1. 调整 logprobs 计算触发条件：在 Sampler.forward 中，将 raw_logprobs 的计算条件从 num_logprobs is not None 扩展为 num_logprobs is not None or sampling_metadata.logprob_token_ids，确保即使未请求 top-k logprobs，只要存在 logprob_token_ids 就预先计算 raw_logprobs。
2. 修改 gather_specific_token_logprobs 的参数：将 logits 参数改为 logprobs（torch.Tensor），方法内部不再对 logits 做 log_softmax，而是直接 gather 传入的已计算 logprobs。
3. 替换核心计算逻辑：删除对 fused Triton kernel compute_token_logprobs 的调用，改用标准的 torch.gather 操作从 logprobs 中提取指定 token ids 的 logprob。这消除了对 vllm.v1.worker.gpu.sample.logprob 模块的导入依赖。
4. 添加断言保护：在调用 gather_specific_token_logprobs 前添加 assert raw_logprobs is not None，确保 logprobs 已正确计算。

关键文件：

- vllm/v1/sample/sampler.py（模块 采样器；类别 source；类型 core-logic）：核心单文件变更：修复了 gather_specific_token_logprobs 使用错误输入（logits 而非 logprobs）的 bug，并调整了 logprobs 计算触发条件。

关键符号：Sampler.forward, Sampler.gather_specific_token_logprobs

关键源码片段

vllm/v1/sample/sampler.py

核心单文件变更：修复了 gather_specific_token_logprobs 使用错误输入（logits 而非 logprobs）的 bug，并调整了 logprobs 计算触发条件。

```
# vllm/v1/sample/sampler.py

class Sampler(nn.Module):
    def forward(self, logits, sampling_metadata, ...):
        logprobs_mode = logprobs_mode_override or self.logprobs_mode
        num_logprobs = sampling_metadata.max_num_logprobs
        raw_logprobs: torch.Tensor | None = None
        # 修复：不仅当 num_logprobs 非空时计算 logprobs，
        # 也当 logprob_token_ids 非空时计算，确保 gather 时有正确的 logprobs。
        if num_logprobs is not None or sampling_metadata.logprob_token_ids:
            if logprobs_mode == "raw_logprobs":
                raw_logprobs = self.compute_logprobs(logits)
            elif logprobs_mode == "raw_logits":
                raw_logprobs = logits.clone() if logits.dtype == torch.float32 else logits.to(torch.
                float32)

        # ... logit 处理、采样 ...
        sampled, processed_logprobs = self.sample(logits, sampling_metadata)
        if processed_logprobs is not None:
            raw_logprobs = processed_logprobs
        sampled = sampled.long()

        logprob_token_ids_tensors = None
        if sampling_metadata.logprob_token_ids:
            # 添加断言确保 raw_logprobs 已初始化
            assert raw_logprobs is not None
            # 使用 logprobs 而非 logits 进行 gather
            logprob_token_ids_tensors = self.gather_specific_token_logprobs(
                raw_logprobs, sampling_metadata.logprob_token_ids, sampled
            )
        # ... 返回 SamplerOutput ...

    def gather_specific_token_logprobs(
        self,
        logprobs: torch.Tensor, # 参数名从 logits 改为 logprobs
        logprob_token_ids: dict[int, list[int]],
        sampled: torch.Tensor,
    ) -> LogprobsTensors | None:
        """Gather logprobs for specific token IDs requested per request.
        Used by generative scoring API to return logprobs for an explicit
        set of token ids rather than the top-k.
        """
        if not logprob_token_ids:
```

```
        return None
    batch_size = logprobs.shape[0]
    device = logprobs.device
    # ... 构建 token_ids_tensor ...

    # 替换：不再调用 fused Triton kernel (compute_token_logprobs)，
    # 而是直接对已计算的 logprobs 进行 gather。
    gathered_logprobs = logprobs.gather(-1, token_ids_tensor)
    # ... 掩码无效位置 ...
```

评论区精华

gemini-code-assist[bot] 指出两个潜在问题：

1. raw_logprobs 可能为 None：当 logprobs_mode 为 processed 模式且 max_num_logprobs 为 None 时，raw_logprobs 可能保持 None，导致断言失败。该 reviewer 建议在 logprob_token_ids 非空时主动计算 logprobs。此建议已在实现中被采纳。
2. TorchDynamo 重编译优化：建议在 gather 后的 logprobs 张量上使用 torch._dynamo.decorators.mark_unbacked 标记 batch 维度为非固定，避免因 batch size 变化导致不必要的重编译。该建议未被采纳（未在 patch 中体现）。

- raw_logprobs 可能为 None 的隐患 (correctness): 已修复：扩展了 raw_logprobs 的计算条件，使其在 logprob_token_ids 非空时也被计算。
- TorchDynamo 重编译优化建议 (performance): 未采纳。patch 中未添加 mark_unbacked 调用，可能因为实际性能影响不大或后续 PR 处理。

风险与影响

- 风险：低风险。变更局限在单文件（sampler.py），逻辑清晰且已通过 review 审批。主要风险是：
  - 当 logprob_token_ids 与 max_num_logprobs 均为 None 但 logprobs_mode 为 processed 时，raw_logprobs 可能仍为 None（依赖 sample 方法内部赋值路径）。但 assert 会捕获此情况，且 added commit 已扩展计算条件，降低了此风险。
  - 删除了 fused Triton kernel，可能对极端大 batch 场景下 gather 性能有轻微影响（但 PR 声称简化后避免了不必要的 log_softmax 计算，整体可能更高效）。
  - 影响：
  - 用户影响：修复了 generative scoring API 中 logprob_token_ids 返回错误 logprobs 的问题，使其行为与 top-k logprobs 一致。
  - 系统影响：减少一次不必要的 log_softmax 计算（从 logits 到 logprobs），可能轻微提升包含 logprob_token_ids 的推理延迟。
  - 团队影响：减少了一个 Triton kernel 的维护成本，代码更简洁。
  - 风险标记：单文件修改，回归风险低，依赖调整（删除导入）

关联脉络

- 暂无明显关联 PR