PR #31624 完整报告

sgl-project/sglang

[Refactor] Move output logprob processing into the logprob_processor layer

合并时间：2026-07-18 13:33

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/31624

执行摘要

- 一句话：将输出 logprob 处理逻辑从 Sampler 移至 logprob_processor 层
- 推荐动作：建议仔细审核 OutputLogprobProcessor 的 attach_logprobs_to_output 与原有 _attach_logprobs_to_output 的一致性，以及 preprocess_fn 注入逻辑。适合对 logprob 处理感兴趣的开发人员阅读，了解重构模式。

功能与动机

PR #20071 引入了 logprob_processor 层处理输入 logprob，本 PR 完成输出侧的迁移，实现 logprob 处理的完整分离，提高代码组织性和可维护性。

实现拆解

1. 将 get_token_ids_logprobs_batch_optimized 函数从 sampler.py 移动到 logprob_processor.py，并保留其优化性能的向量化实现。
2. 在 logprob_processor.py 中添加 OutputLogprobsResult 数据类，提供类似 InputLogprobsResult 的结构。
3. 提取 OutputLogprobProcessor 类，包含 write_to 方法（将结果写入 LogitsProcessorOutput）和 attach_logprobs_to_output 方法，替代 Sampler 中原有的 _attach_logprobs_to_output 方法。
4. 在 Sampler 的 __init__ 中创建 OutputLogprobProcessor 实例，并在 forward 中调用其方法。同时将 _preprocess_logits 作为 preprocess_fn 注入，以便在处理器中使用。
5. 保留 Sampler.compute_logprobs_only 作为外层代理，保持外部调用兼容性。未增加测试用例，但现有测试应覆盖功能。

关键文件：

- python/sglang/srt/layers/logprob_processor.py（模块 logprob 层；类别 source；类型 core-logic；符号 get_token_ids_logprobs_batch_optimized, OutputLogprobsResult, write_to, OutputLogprobProcessor）：主要变更文件，新增 OutputLogprobProcessor 类、OutputLogprobsResult 数据类和 get_token_ids_logprobs_batch_optimized 函数，完成 output logprob 的集中处理。
- python/sglang/srt/layers/sampler.py（模块 采样器；类别 source；类型 core-logic；符号 _attach_logprobs_to_output, get_token_ids_logprobs_batch_optimized）：去除原生的 _attach_logprobs_to_output 和 get_token_ids_logprobs_batch_optimized，改为委托

OutputLogprobProcessor，简化 Sampler 代码。

关键符号：get_token_ids_logprobs_batch_optimized, OutputLogprobsResult, write_to, OutputLogprobProcessor, attach_logprobs_to_output, compute_logprobs_only

关键源码片段

python/sglang/srt/layers/logprob_processor.py

主要变更文件，新增 OutputLogprobProcessor 类、OutputLogprobsResult 数据类和 get_token_ids_logprobs_batch_optimized 函数，完成 output logprob 的集中处理。

```
def get_token_ids_logprobs_batch_optimized(
    logprobs: torch.Tensor,
    token_ids_logprobs: List[List[int]],
) -> Tuple[List, List]:
    """向量化批量处理 token ID logprobs 提取，使用单次 GPU gather 替代逐项
    gather，适合大批量。"""
    batch_size = len(token_ids_logprobs)
    device = logprobs.device

    # 计算每个请求的长度，将 None 视为空列表
    token_lengths = torch.tensor(
        [len(token_ids or []) for token_ids in token_ids_logprobs], device=device
    )
    total_tokens = int(token_lengths.sum().item())

    if total_tokens == 0:
        return [logprobs.new_empty(0) for _ in token_ids_logprobs], [
            [] for _ in token_ids_logprobs
        ]

    # 构建扁平索引：row_indices 和 col_indices
    row_indices = torch.repeat_interleave(
        torch.arange(batch_size, device=device), token_lengths
    )
    col_indices = torch.tensor(
        [
            token_id
            for token_ids in token_ids_logprobs
            for token_id in (token_ids or [])
        ],
        device=device,
        dtype=torch.long,
    )

    # 单次 gather 操作
    gathered_logprobs = logprobs[row_indices, col_indices]

    # 根据每个请求的长度 split 回去
```

```
    values = torch.split(gathered_logprobs, token_lengths.tolist())
    indices = torch.split(col_indices, token_lengths.tolist())
    return list(values), [split.tolist() for split in indices]
```

评论区精华

本 PR 无 review 评论或讨论线程。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低，因为 PR 声称字节级一致（除两处声明替换）。但潜在风险包括：对 sampling_info.device 的替换为 batch_next_token_ids.device 可能在不同设备场景下有差异；attach_logprobs_to_output 方法现在在 OutputLogprobProcessor 中，需要确保所有调用路径正确；由于涉及 Sampler 核心逻辑，若 preprocess_fn 注入有误可能影响 logprob 计算。回归风险可控，现有 CI 测试已通过。
- 影响：影响开发者：logprob 处理逻辑更集中，便于后续维护和扩展。对用户：功能无变化，API 保持兼容。对系统：性能理论上无变化，但未来优化 logprob 处理时 logprob_processor 成为唯一入口。
- 风险标记：核心路径变更，外部 API 兼容需注意

关联脉络

- PR #20071 refactor logprob processor layer: 本 PR 是 #20071 的后续，完成输出侧 logprob 处理的迁移，实现 logprob_processor 层的完整分离。