PR #23513 完整报告

sgl-project/sglang

[Score API] Hoist query placeholder scan and specialize PositionalEmbeds stacking

合并时间：2026-04-30 04:51

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23513

执行摘要

- 一句话：提升 Score API 查询占位符扫描并优化 PositionalEmbeds 堆叠
- 推荐动作：值得精读其设计权衡：如何通过提升不变计算和分派堆叠优化性能，以及保留更高层接口供测试调用的做法。

功能与动机

PR #23513 指出在 _build_token_id_inputs 中单项目路径每次迭代通过 _resolve_embed_overrides_for_request 对查询进行占位符扫描，导致 O(N•Q) 复杂度；同时 PositionalEmbeds.__post_init__ 强制对所有元素执行 unsqueeze(0) 再 cat，不必要地对已是 1-D 的张量增加额外操作。

实现拆解

1. 提升查询占位符扫描：在 _build_token_id_inputs (tokenizer_manager_score_mixin.py) 中，通过直接调用 _resolve_overrides_for_sequence 将查询的占位符扫描移到循环外，得到 q_embeds 和 q_positions。单项目和多项项目模式均复用该结果，消除原多项模式的 query if i == 0 不对称和单项目的重复扫描。同时修改多项模式的 embedding 聚合：之前在每个项目内部调用 _resolve_embed_overrides_for_request 并生成 PositionalEmbeds，再用 torch.cat 合并；现在直接累积 all_embeds 列表，最后只调用一次 PositionalEmbeds 完成堆叠。

2. 优化 PositionalEmbeds 堆叠：在 embed_types.py 的 __post_init__ 中，根据列表首个元素的维数选择堆叠方式：若为 1-D（最常见）则使用 torch.stack 直接添加维度；若为 2-D 则使用 torch.cat。避免旧实现中每个任务都需要判断并可能 unsqueeze(0) 的开销。

3. 配套与测试：无新增测试文件，但通过既有测试 test_embed_overrides.py 验证等价性。

关键文件：

- python/sglang/srt/managers/tokenizer_manager_score_mixin.py（模块 评分模块；类别 source；类型 core-logic；符号 _build_token_id_inputs, _resolve_overrides_for_sequence）：核心变更：实现查询占位符提升，统一单 / 多项模式，减少重复扫描。
- python/sglang/srt/managers/embed_types.py（模块 嵌入类型；类别 source；类型 core-logic；符号 PositionalEmbeds, post_init）：优化 PostionalEmbeds 堆叠方式，减少 per-tensor 操作。

关键符号：_build_token_id_inputs, PositionalEmbeds.post_init

关键源码片段

python/sglang/srt/managers/embed_types.py

优化 PostionalEmbeds 堆叠方式，减少 per-tensor 操作。

```
from dataclasses import dataclass
from typing import List, Union
import torch


@dataclass
class PositionalEmbeds:
    """Embeddings to place at specific token positions.

    Accepts either a list of [1, hidden_dim] tensors or a pre-stacked [N, hidden_dim] tensor.
    In both cases, __post_init__ stacks into a single [N, hidden_dim] tensor to reduce
    ZMQ serialization overhead.

    Attributes:
        embeds: Stacked tensor of shape [N, hidden_dim] after __post_init__.
        positions: List of positions where embeddings should be injected.
    """

    embeds: Union[List[torch.Tensor], torch.Tensor]
    positions: List[int]

    def __post_init__(self):
        # Normalize list of tensors into a single [N, hidden_dim] tensor.
        # Dispatch by element rank to avoid a per-element unsqueeze.
        if isinstance(self.embeds, list):
            if not self.embeds:
                # Empty list raises on cat — caller must ensure non-empty
                self.embeds = torch.cat(self.embeds, dim=0)
            elif self.embeds[0].dim() == 1:
                # [hidden_dim] elements -> stack adds the leading dim natively
                self.embeds = torch.stack(self.embeds, dim=0)
            else:
                # [1, hidden_dim] (already has leading dim) -> plain concat
                self.embeds = torch.cat(self.embeds, dim=0)
        if self.embeds.shape[0] != len(self.positions):
            raise ValueError(
                f"embeds length ({self.embeds.shape[0]}) != "
                f"positions length ({len(self.positions)})"
            )
```

评论区精华

PR 中无 review 讨论，但作者在备注中指出单项目路径中 item_position_offset=len(query) 在 item_first=True 时位置错误，该预存问题未在本 PR 处理，留待后续修复。

- 暂无高价值评论线程

风险与影响

- 风险：修改为核心逻辑，但严格保持语义不变，且通过既有测试覆盖。风险较低。但预存位置偏移错误在启用 item_first=True 时可能影响 embedding 覆盖功能，未来需修复。另外新代码假设 query_embed_overrides 和 item_embed_overrides 列表长度与 items 一致，否则可能触发索引错误（此为前提条件）。
- 影响：影响范围限于 Score API 的 token-ID 输入路径，涉及 embedding 覆盖功能。对用户无感知，但减少计算开销。团队需关注预存 item_first 位置错误后续修复。
- 风险标记：预存位置偏移问题未修复，输入格式假设依赖调用方

关联脉络

- 暂无明显关联 PR