PR #31985 完整报告

sgl-project/sglang

[Perf] Fold dspark dense draft embedding into the draft graph via forward_embed

合并时间：2026-07-22 08:00

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/31985

执行摘要

- 一句话：将 DSpark 密集草稿嵌入查询移入草稿图
- 推荐动作：这是一个小而精准的性能优化 PR，值得相关开发者阅读了解如何通过 forward_embed 机制消除 *eager* 阶段拷贝。建议精读 dspark.py 和 dflash.py 中的改动以理解设计模式。

功能与动机

密集 DSpark 草稿模型在 attach_shared_modules 中丢弃了共享嵌入（del embed_tokens），导致运行器每次草稿图回放前都必须急切地准备 input_embeds。而 deepseek_v4_dspark 已经使用了 forward_embed 机制（保留 embed_tokens，暴露 forward_embed），使得 CUDA 图运行器和草稿提议者可以跳过 *eager* 阶段的复制。本 PR 让密集 DSpark 模型采用相同方式，将嵌入查找移入草稿图内，消除不必要的 *eager* 阶段拷贝。

实现拆解

1. dspark.py – 保留 embed_tokens 并添加 forward_embed 方法：将 DSparkDraftMixin.attach_shared_modules 中的 del embed_tokens 改为 self.embed_tokens = embed_tokens，新增 forward_embed(self, input_ids) 方法，通过调用 self.embed_tokens(input_ids) 完成嵌入查找，并附带注释说明这样可以让运行器跳过 *eager* 阶段的 input_embeds 准备。
2. dflash.py – 为 DFlashDraftModel.forward 添加 forward_embed 回退逻辑：当 input_embeds 为 None 时，先检查模型是否有 forward_embed 方法（通过 hasattr），若有则调用 self.forward_embed(input_ids) 获取嵌入；否则保持原有行为，抛出 ValueError。这样 base 类不会破坏已有模型，而实现了 forward_embed 的模型可以自动利用该优化。
3. 无需修改运行器代码：CUDA 图运行器和草稿提议者已经支持 forward_embed 机制，只需模型实现该接口即可自动生效。

关键文件：

- python/sglang/srt/models/dspark.py（模块 模型定义；类别 source；类型 data-contract；符号 forward_embed）：核心变更文件：保留 embed_tokens 并新增 forward_embed 方法，将嵌入查找移入草稿图。

- python/sglang/srt/models/dflash.py（模块 模型定义；类别 source；类型 data-contract）：修改 forward 方法以支持 forward_embed 回退，使得基类可以自动利用优化。

关键符号：DSparkDraftMixin.attach_shared_modules, DSparkDraftMixin.forward_embed, DFlashDraftModel.forward

关键源码片段

python/sglang/srt/models/dspark.py

核心变更文件：保留 embed_tokens 并新增 forward_embed 方法，将嵌入查找移入草稿图。

```
# python/sglang/srt/models/dspark.py 中 DSparkDraftMixin 的修改

def attach_shared_modules(
    self, *, embed_tokens: nn.Module, lm_head: nn.Module
) -> None:
    self.embed_tokens = embed_tokens # 之前是 del embed_tokens，现在保留
    self.lm_head = lm_head

def forward_embed(self, input_ids: torch.Tensor) -> torch.Tensor:
    # 使用共享的目标嵌入在草稿图内部进行嵌入查找
    # 当草稿模型暴露 forward_embed 时，运行器会跳过 eager 阶段的 input_embeds 准备
    return self.embed_tokens(input_ids)
```

python/sglang/srt/models/dflash.py

修改 forward 方法以支持 forward_embed 回退，使得基类可以自动利用优化。

```
# python/sglang/srt/models/dflash.py 中 DFlashDraftModel.forward 的修改

@torch.no_grad()
def forward(
    self,
    input_ids: torch.Tensor,
    positions: torch.Tensor,
    forward_batch: ForwardBatch,
    input_embeds: Optional[torch.Tensor] = None,
    get_embedding: bool = False,
    pp_proxy_tensors=None,
) -> LogitsProcessorOutput:
    if input_embeds is None:
        if hasattr(self, "forward_embed"):
            # 如果模型实现了 forward_embed，则直接在图中获取嵌入
            input_embeds = self.forward_embed(input_ids)
        else:
            raise ValueError(
                "DFlashDraftModel requires `input_embeds` (use the target "
                "embedding)."
            )
    hidden_states = input_embeds
```

```
    # ... 后续层处理不变
```

评论区精华

无 review 讨论。

- 暂无高价值评论线程

风险与影响

- 风险：本 PR 风险极低。核心改动仅涉及两处： 1) DSparkDraftMixin 保留 *embed_tokens* 而非丢弃，添加 forward_embed 方法； 2) DFlashDraftModel.forward 增加 forward_embed 回退分支。不会影响现有行为，因为只有模型存在 forward_embed 时才会走新路径。未看到直接对应的测试变更，但运行了 DSPark 和 DFlash 的 sanity 测试并通过。潜在的兼容性风险：若其他子类依赖 attach_shared_modules 中丢弃 *embed_tokens* 的行为，可能受影响，但当前仅 DSparkDraftMixin 使用该混入。
  - 影响：影响范围局限于 DSPark 密集草稿模型的推理流程。用户无需任何配置更改即可自动获得性能提升（减少一次 *eager* 阶段的数据拷贝，将嵌入计算纳入 CUDA 图）。对系统其他模块无影响。
  - 风险标记：暂无

关联脉络

- 暂无明显关联 PR