PR #26358 完整报告

sgl-project/sglang

Revert "[perf][spec decoding] Skip full-vocab softmax in EAGLE draft when topk == 1 (#26235)"

合并时间：2026-05-26 17:47

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26358

执行摘要

- 一句话：回退 EAGLE topk==1 跳过 softmax 优化
- 推荐动作：该 PR 是紧急修复回退，建议所有使用 EAGLE + MTP 的部署立即合入。对于仅使用标准 EAGLE 且关注性能的团队，可关注后续优化的重新提交通常采用 topk==1 AND not is_mtp_path 的 guard 或数学恒等式 softmax 优化。PR 的讨论和 PR Body 中的根因分析方法值得精读，展示了精确的跨运行对比和手动验证流程。

功能与动机

PR #26235 的优化导致 DeepSeek-V3.2 + MTP 在 ROCm 7.2 上的 GSM8K 准确率从正常水平（>0.94）崩溃至 0.035，且 96% 输出无效。PR Body 中提供了详细的跨运行对比表格和手动验证数据，确认问题源自 sglang 侧 #26235。

实现拆解

1. 恢复完整 softmax 路径：在 *eagle_worker_v2.py* 的 *draft_forward* 和 *_draft_extend_for_decode* 方法中，删除原先对 self.topk == 1 的特殊分支（直接 argmax + hardcode topk_p=1.0），改为无条件执行完整 softmax 后再调用 fast_topk。
2. 同步更新 CUDA Graph 捕获逻辑：在 *eagle_draft_extend_cuda_graph_runner.py* 的 *run_once* 内部，删除相同的 topk==1 特殊分支，统一走 softmax + fast_topk 路径。
3. 回退验证：在 ROCm 7.2 mi35x 硬件上重新运行 MTP 精度测试，GSM8K 准确率恢复至 0.975（阈值 0.94），确认修复有效。
4. 后续指南：PR 建议未来的重新提交应在 topk==1 AND not is_mtp_path 条件下保留优化，避免影响 MTP 路径。

关键文件：

- python/sglang/srt/speculative/*eagle_worker_v2.py*（模块 推测解码；类别 source；类型 *core-logic*）：核心 EAGLE worker 逻辑，包含 *draft_forward* 和 *_draft_extend_for_decode* 两个关键方法中 topk==1 特殊分支的移除
- python/sglang/srt/speculative/*eagle_draft_extend_cuda_graph_runner.py*（模块 推测解码；类别 source；类型 *core-logic*）：CUDA Graph 捕获逻辑中 topk==1 特殊分支的同步移除，保证 CUDA Graph 路径与非 Graph 路径行为一致

关键符号：*draft_forward*, *_draft_extend_for_decode*, *run_once*

关键源码片段

python/sglang/srt/speculative/eagle_worker_v2.py

核心 EAGLE worker 逻辑，包含 draft_forward 和 _draft_extend_for_decode 两个关键方法中 topk==1 特殊分支的移除

```
# python/sglang/srt/speculative/eagle_worker_v2.py ( 回退后 )

def draft_forward(self, forward_batch):
    # ... 前面的循环和 forward 调用 ...
    logits_output = self.draft_runner.forward(
        forward_batch, skip_attn_backend_init=True
    ).logits_output
    maybe_detect_nan(logits_output.next_token_logits, f"draft_forward step {i}")
    # 回退：始终执行完整 softmax，不再区分 topk == 1
    probs = torch.softmax(logits_output.next_token_logits, dim=-1)
    topk_p, topk_index = fast_topk(probs, self.topk, dim=-1)
    # ... 后续处理 ...

def _draft_extend_for_decode(self, forward_batch):
    # ... 前面的 select_index 处理 ...
    # 回退：同样移除 topk==1 特殊分支
    probs = torch.softmax(draft_logits_output.next_token_logits, dim=-1)
    ret_topk_p, ret_topk_index = fast_topk(probs, self.topk, dim=-1)
    ret_hidden_states = draft_logits_output.hidden_states
    # ... 后续构建返回值 ...
```

python/sglang/srt/speculative/eagle_draft_extend_cuda_graph_runner.py

CUDA Graph 捕获逻辑中 topk==1 特殊分支的同步移除，保证 CUDA Graph 路径与非 Graph 路径行为一致

```
# python/sglang/srt/speculative/eagle_draft_extend_cuda_graph_runner.py ( 回退后 )

def run_once():
    # ... 前处理 ...
    ret = self.model_runner.model.forward(
        forward_batch.input_ids,
        forward_batch.positions,
        forward_batch,
    )
    # 回退：始终执行完整 softmax，不再区分 topk == 1
    probs = torch.softmax(ret.next_token_logits, dim=-1)
    ret.topk_p, ret.topk_index = fast_topk(probs, self.topk, dim=-1)
    # ... 后处理 ...
```

评论区精华

gemini-code-assist[bot] 在 Review 评论中建议，与其完全回退到完整 softmax，可以使用数学恒等式 $\text{softmax}(x)_{\text{argmax}} = 1 / \sum e^{x_i - \max(x)}$ 来优化 top-1 概

率计算，避免 materialize 完整 softmax tensor，在保留性能收益的同时避免精度崩溃。该建议未在本次 PR 中实施，但被 PR 作者引用为后续重新提交的可行方案。

- 替代优化方案：数学恒等式 softmax 避免精度崩溃 (performance): 未采纳，但作为后续重新提交的优化方向被记录。
- draft_forward 中 topk==1 优化替代方案 (performance): 未采纳，回退后统一使用完整 softmax。

风险与影响

- 风险：核心风险：回退本身是安全的，代码恢复到 PR #26235 合并前的基线状态（该基线已通过所有测试和精度验证）。潜在风险是回退后 EAGLE draft 在 topk==1 场景下的性能开销恢复（完整 softmax 增加 HBM 带宽和计算量），但这是正确的代价。无回归风险，因为回退代码已在合并前通过了 CI 和精度验证。
- 影响：影响范围：仅影响使用 EAGLE 推测解码且 topk==1 的推理路径。对于非 MTP 场景（如标准 EAGLE 或 EAGLE3），此回退会引入额外的 softmax 开销，影响性能但不影响正确性。对于 MTP 场景（如 DeepSeek-V3.2+MTP），回退修复了精度崩溃的严重缺陷，恢复可用性。整体上这是一个修正 bug 的积极影响，但存在性能回退的副作用。
- 风险标记：核心路径变更，缺少测试覆盖，性能回退

关联脉络

- PR #26235 [perf][spec decoding] Skip full-vocab softmax in EAGLE draft when topk == 1: 本 PR 正是对该 PR 的完整回退。该优化在 MTP 路径上导致精度崩溃，回退将其恢复至基线。