PR #26312 完整报告

sgl-project/sglang

[mtp] add rejection sampling for speculative decoding

合并时间：2026-06-21 06:10

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26312

执行摘要

- 一句话：为 speculative decoding 添加经典拒绝采样，提升接受长度和吞吐
- 推荐动作：建议架构师和服务化团队精读此 PR，特别是 reject_sampling.py 中 Triton 内核的两轮残差采样实现，该模式可复用于其他需要从修正分布采样的场景。同时，review 中关于缓冲区使用的对话也体现了 CUDA Graph 场景下组件间契约设计的良好实践。

功能与动机

当前 Eagle speculative decoding 的草稿 token 预测和目标验证均使用贪婪采样（argmax），导致实际 token 分布与目标分布差异较大，接受长度受限。经典拒绝采样通过从草稿分布 q 采样草稿 token 并以 min(1, p/q) 概率接受，可显著提升平均接受长度，进而提高整体吞吐。PR body 中的实验数据表明拒绝采样在多种模型和数据集上均有效果：吞吐最高提升 130%+，接受率提升 10%+。

实现拆解

1. 新增 Triton 拒绝采样内核（python/sglang/srt/speculative/reject_sampling.py）：
   - 实现 speculative_sampling_classic_kernel：对每个序列从候选 token 列表中循环验证，使用 coin*q < p 条件决定接受；全部接受时从 target probs 采样，否则从 max(0, p - q) 的残差分布做最终采样。
   - 实现 chain_speculative_sampling_triton 作为调用入口，封装批处理和 stride 计算。
   - 内核使用两轮词汇级循环（sum 和 binary search CDF）实现残差采样。
2. 修改 eagle_worker_v2.py 集成拒绝采样路径：
   - 构造函数添加断言 topk==1（拒绝采样仅支持链式 topk=1）。
   - alloc_memory_pool 中校验 draft 和 target 词汇表大小一致。
   - draft 方法现在返回 draft_probs（从 draft_forward 或 cuda_graph_runner 获取）。
   - draft_forward 方法在拒绝采样模式下收集每步的 draft_probs 并返回。
3. 在 spec_utils.py 添加辅助函数（fast_sample、renorm_draft_probs）：
   - fast_sample：对概率分布做 torch.multinomial 采样，返回采样概率和索引。
   - renorm_draft_probs：根据是否拒绝采样，对 draft logits 应用温度缩放 softmax 以匹配目标采样温度。
4. 修改 eagle_draft_cuda_graph_runner.py 支持 draft_probs 预分配：

- 新增 draft_probs 缓冲区（形状 [max_bs, vocab_size]），仅在拒绝采样启用时分配。
- 在 capture_one_shape 中创建 SamplingBatchInfo 传递 temperature，用于 CUDA Graph 捕获。

5. 修改 eagle_utils.py 中的 eagle_sample 函数：
   - 根据 speculative_use_rejection_sampling 全局标志选择调用 chain_speculative_sampling_triton 或原有的 tree_speculative_sampling_target_only。
   - 从 verify_input.draft_probs 获取草稿概率，非拒绝采样回退为全零张量。
6. 新增命令行参数 speculative_use_rejection_sampling（python/sglang/srt/server_args.py）：
   - 同步修改 overlap_utils.py、forward_batch_info.py、eagle_info.py 等以传递 draft_probs。
7. 新增端到端测试（test/registered/spec/eagle/test_eagle_reject_sampling.py）：
   - 启动 Qwen3.5-9B 服务并启用 NEXTN + 拒绝采样，在 GSM8K 子集上评估准确率和平均接受长度，断言准确率 > 0.8 且接受长度 > 2.5。

关键文件：

- python/sglang/srt/speculative/reject_sampling.py（模块 拒绝采样；类别 source；类型 core-logic；符号 speculative_sampling_classic_kernel, chain_speculative_sampling_triton）：新增核心 Triton 拒绝采样内核，实现验证循环和残差采样，是整个 PR 的技术核心。
- test/registered/spec/eagle/test_eagle_reject_sampling.py（模块 测试；类别 test；类型 test-coverage；符号 TestQwen35EagleRS, setUpClass, tearDownClass, test_a_gsm8k）：新增端到端测试，验证拒绝采样在真实模型上的准确率和接受长度，确保功能正确。
- python/sglang/srt/speculative/eagle_worker_v2.py（模块 推测工作器；类别 source；类型 core-logic）：核心工作器修改，集成拒绝采样路径，包括断言、draft_probs 收集与传递。
- python/sglang/srt/speculative/spec_utils.py（模块 推测工具；类别 source；类型 core-logic；符号 fast_sample, renorm_draft_probs）：新增 fast_sample 和 renorm_draft_probs 辅助函数，支持概率采样和温度缩放。
- python/sglang/srt/speculative/eagle_draft_cuda_graph_runner.py（模块 CUDA 图；类别 source；类型 dependency-wiring）：新增 draft_probs 缓冲区和 SamplingBatchInfo 传递 temperature，支持 CUDA Graph 下的拒绝采样。
- python/sglang/srt/speculative/eagle_utils.py（模块 推测工具；类别 source；类型 dependency-wiring）：在 eagle_sample 函数中新增拒绝采样分支选择，从 verify_input 获取 draft_probs。
- python/sglang/srt/arg_groups/speculative_hook.py（模块 配置；类别 source；类型 core-logic）：添加拒绝采样相关参数选项。
- python/sglang/srt/speculative/eagle_info.py（模块 数据结构；类别 source；类型 dependency-wiring）：新增 draft_probs 字段在 EagleVerifyInput 等数据结构中。

- python/sglang/srt/managers/overlap_utils.py（模块 重叠工具；类别 source；类型 core-logic）：支持 draft_probs 在 overlap 流水线中的传递。
- python/sglang/srt/model_executor/forward_batch_info.py（模块 前向批次；类别 source；类型 data-contract）：在 ForwardBatch 中添加 spec_info.draft_probs 字段传递。
- python/sglang/srt/server_args.py（模块 配置；类别 source；类型 core-logic）：新增 --speculative-use-rejection-sampling 命令行参数。

关键符号：speculative_sampling_classic_kernel, chain_speculative_sampling_triton, fast_sample, renorm_draft_probs, TestQwen35EagleRS, setUpClass, tearDownClass, test_a_gsm8k

关键源码片段

python/sglang/srt/speculative/spec_utils.py

新增 fast_sample 和 renorm_draft_probs 辅助函数，支持概率采样和温度缩放。

```
def fast_sample(probs: torch.Tensor, num_samples: int = 1):
    # 从概率分布 probs 中采样 num_samples 个 token，返回采样概率和索引
    sample_index = torch.multinomial(probs, num_samples=num_samples)
    sample_p = probs.gather(1, sample_index)
    return sample_p, sample_index


def renorm_draft_probs(
    next_token_logits: torch.Tensor,
    sampling_info,
    use_rejection_sampling: bool,
) -> torch.Tensor:
    """Draft-side next-token distribution.
    Plain softmax, except under rejection sampling where logits are
    temperature-scaled so the draft proposal q tracks the target sampling
    temperature (higher acceptance; correctness holds for any q).
    """
    if not use_rejection_sampling or not next_token_logits.size(0):
        return torch.softmax(next_token_logits, dim=-1)
    # 拒绝采样时对 logits 进行温度缩放，使 draft 分布 q 与目标 temperature 对齐
    return torch.softmax(next_token_logits / sampling_info.temperatures, dim=-1)
```

评论区精华

Review 中有一个讨论线程：Qiaolin-Yu 在 eagle_draft_cuda_graph_runner.py 中询问新增的 self.temperatures 等缓冲区的用途（评论 #3345123269）。liyucheng09 回复指出 temperature 用于 draft prob renorm（见 spec_utils.py 的 renorm_draft_probs），并说明 top_p/top_k 冗余即将在后续提交中移除。会话清晰，无重大争议，且冗余缓冲区已在后续提交清理。

- 缓冲区用途及冗余清理 (question): liyucheng09 解释 temperature 用于 draft prob renorm（见 spec_utils.py renorm_draft_probs），并指出 top_p/top_k 冗余即将在后续

提交中移除。

风险与影响

- 风险：
  1. 核心路径变更：Rejection sampling 修改了 speculative decoding 的核心验证路径，可能影响未启用拒绝采样的分支（通过条件分支隔离，风险较低）。
  2. Triton 内核稳定性：新增 Triton 内核需要 GPU 支持 Triton，且内核中数值逻辑（如 sum 和 CDF 搜索）在极端情况（norm_sum == 0）可能回退到最后一个 token，但注释说明此情况近不可逆。
  3. topk=1 限制：拒绝采样仅支持 topk=1，用户启用时需保证该配置，否则断言失败，不会静默错误。
  4. 显存开销：新增 draft_probs 缓冲区增加了显存占用，但仅在启用时分配。
  5. 兼容性：新参数默认 False，不影响现有行为和配置。
- 影响：
  1. 用户：启用拒绝采样后，在 temperature>0 的场景下吞吐提升显著（实验显示最高 130%+），但要求 topk=1 和共享词汇表。
  2. 系统：增加少许显存占用和计算开销（Triton 内核），但整体吞吐受益。
  3. 团队：需维护一个新的 Triton 内核文件，但内核逻辑经典且优化空间有限。
  4. 测试：添加了 CI 端到端测试，覆盖准确率和接受长度。 - 风险标记：核心路径变更，Triton 内核，仅支持 topk=1

关联脉络

- 暂无明显关联 PR