PR #40651 完整报告

vllm-project/vllm

[Model Runner V2] Fix rejection sampling acceptance rate gap vs MRV1

合并时间：2026-04-27 10:12

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/40651

执行摘要

- 一句话：MRV2 启用 one-hot 拒绝采样，修复接受率差距
- 推荐动作：建议所有使用 MRV2 推测解码的同学阅读此 PR，了解新的 draft_sample_method 配置及其对接受率和内存的影响。特别值得关注的设计模式是：通过 Triton 编译常量（HAS_DRAFT_LOGITS）在编译期分支内核逻辑，无运行时开销。同时关注后续可能的扩展：允许在不缓存 draft logits 的情况下使用随机采样。

功能与动机

MRV1 使用 argmax 采样 draft tokens，将 draft 概率分布视为 one-hot，改变了标准拒绝采样的概率测试 u < p(x)/q(x) 为 u < p(x)，从而获得更高效的接受率。而 MRV2 默认使用 strict acceptance（只接受完全匹配的 draft tokens），导致接受率较低。MRV2 的 probabilistic 方法虽然使用完整 draft 概率但需要存储 full draft logits，内存开销大。因此需要实现一个中间方案：在 MRV2 中启用与 MRV1 等价的 one-hot 拒绝采样，以修复接受率差距，同时避免额外内存消耗。

实现拆解

1. 配置模块（config/speculative.py）：将 RejectionSampleMethod 类型从 'strict' | 'probabilistic' | 'synthetic' 改为 'standard' | 'synthetic'，新增 DraftSampleMethod 类型 'greedy' | 'gumbel'，并添加 draft_sample_method 配置字段（默认 'greedy'）。同时更新了 rejection_sample_method 的默认值为 'standard'。
2. 概率拒绝采样工具（probabilistic_rejection_sampler_utils.py）：将核心内核重命名为 _compute_block_stats_kernel，新增编译常量 HAS_DRAFT_LOGITS。当 HAS_DRAFT_LOGITS 为 False 时，跳过 draft 统计量的计算，仅计算 target 的 argmax 或统计量。在 _probabilistic_rejection_kernel 中也加入 HAS_DRAFT_LOGITS 条件，避免加载和计算 draft logits。
3. 拒绝采样器（rejection_sampler.py）：删除 _strict_rejection_sample_kernel 和 strict_rejection_sample 函数。将 __call__ 方法中的 'strict' 和 'probabilistic' 分支合并为 'standard' 分支，始终调用 probabilistic_rejection_sample，该函数根据是否传入 draft_logits 自动选择行为（有 draft_logits 时使用完整概率，无时使用 one-hot 近似）。
4. Eagle 推测器（eagle/speculator.py）：新增 _sample_draft 方法，根据 self.draft_logits 是否分配（由 draft_sample_method 决定）来选择 gumbel 采样（分配时）或 argmax 采样（未分配时）。重构 prefill 和 generate_draft 方法，将重复的

gumbel_sample 调用替换为 _sample_draft，消除代码重复并集中控制 draft 采样策略。

关键文件：

- vllm/v1/worker/gpu/spec_decode/probabilistic_rejection_sampler_utils.py（模块 推测解码；类别 source；类型 core-logic；符号 _compute_block_max_and_sumexp_kernel, _compute_block_stats_kernel）：核心概率拒绝采样工具，重构内核以支持条件性 draft logits 计算，是接受率提升的关键。
- vllm/v1/worker/gpu/spec_decode/rejection_sampler.py（模块 推测解码；类别 source；类型 core-logic；符号 _strict_rejection_sample_kernel, strict_rejection_sample）：删除了 strict_rejection_sample，简化拒绝采样器控制流，统一使用 probabilistic_rejection_sample。
- vllm/v1/worker/gpu/spec_decode/eagle/speculator.py（模块 推测解码；类别 source；类型 core-logic；符号 _sample_draft）：新增 _sample_draft 方法封装 draft 采样逻辑，根据配置选择 gumbel 或 argmax，重构预填充与 decode 流程。
- vllm/config/speculative.py（模块 配置；类别 source；类型 core-logic）：定义 RejectionSampleMethod 和 DraftSampleMethod 新类型，新增 draft_sample_method 配置字段，驱动行为变更。

关键符号：_sample_draft, _compute_block_stats_kernel, _probabilistic_rejection_kernel, probabilistic_rejection_sample, RejectionSampler.call

关键源码片段

vllm/v1/worker/gpu/spec_decode/rejection_sampler.py

删除了 strict_rejection_sample，简化拒绝采样器控制流，统一使用 probabilistic_rejection_sample。

```
class RejectionSampler:
    def __call__(
        self,
        logits: torch.Tensor,
        input_batch: InputBatch,
        draft_logits: torch.Tensor | None = None,
    ) -> SamplerOutput:
        draft_sampled = input_batch.input_ids[input_batch.logits_indices]
        num_nans = get_num_nans(logits) if self.sampler.compute_nans else None

        if self.rejection_sample_method == "standard":
            # 'standard' 方法统一使用 probabilistic_rejection_sample
            # 该函数内部根据 draft_logits 是否为 None 自动选择合适的 rejection 策略
            pos = input_batch.positions[input_batch.logits_indices]
            processed_logits = self.sampler.apply_sampling_params(
                logits, input_batch.expanded_idx_mapping,
                input_batch.idx_mapping_np, pos, draft_sampled,
                input_batch.expanded_local_pos,
            )
            sampled, num_sampled = probabilistic_rejection_sample(
```

```
                processed_logits, draft_logits, draft_sampled,
                input_batch.cu_num_logits, pos,
                input_batch.idx_mapping, input_batch.expanded_idx_mapping,
                input_batch.expanded_local_pos,
                self.sampler.sampling_states.temperature.gpu,
                self.sampler.sampling_states.seeds.gpu,
                self.num_speculative_steps,
            )
            # 计算 logprobs 时使用 processed 或 raw logits
            logprobs_tensors = self._get_logprobs_tensors(...)
        elif self.rejection_sample_method == "synthetic":
            # synthetic 方法使用预定义的接受率，不依赖实际 logits
            ...
```

vllm/v1/worker/gpu/spec_decode/eagle/speculator.py

新增 _sample_draft 方法封装 draft 采样逻辑，根据配置选择 gumbel 或 argmax，重构预填充与 decode 流程。

```
def _sample_draft(
    self,
    logits: torch.Tensor,
    idx_mapping: torch.Tensor,
    pos: torch.Tensor,
    step: int,
) -> torch.Tensor:
    # 根据 draft_logits 是否分配（由 draft_sample_method 控制）
    # 选择采样策略
    if self.draft_logits is not None:
        # 'gumbel' 模式：使用 Gumbel 噪声随机采样，
        # 并将原始 logits 保存到 draft_logits 供 rejection sampling 使用
        return gumbel_sample(
            logits,
            idx_mapping,
            self.temperature,
            self.seeds,
            pos + 1, # 注意：位置加 1 以使 Gumbel 噪声与目标采样同步
            apply_temperature=True,
            processed_logits_out=self.draft_logits[:, step],
        )
    else:
        # 'greedy' 模式（默认）：直接取 argmax，
        # 等价于将 draft 分布视为 one-hot，与 MRV1 行为一致
        return logits.argmax(dim=-1)
```

评论区精华

benchislett评论：“Need to double-check the math but at a high level it looks good”，表明 reviewer 关注数学正确性，但大致认可方案。

gemini-code-assist[bot]自动审查指出：当 draft_sample_method 为 'greedy' 时强制使用 argmax 限制了随机采样选项，建议允许即使不缓存 draft logits 也能使用 gumbel 采样（processed_logits_out=None）。这是一个合理的设计权衡：当前实现以内存效率为重，牺牲了随机 drafted 的灵活性。

- 强制 argmax 限制随机采样 (design): PR 作者保留设计：greedy 模式以内存效率优先，后续可扩展。评论显示为设计权衡。
- 数学正确性检查 (correctness): 未进一步讨论，PR 被合并，大概率数学正确。

风险与影响

- 风险：
  1. 配置兼容性：rejection_sample_method 旧值 'strict' 和 'probabilistic' 被移除，用户需要更新配置。
  2. 核心路径重构：删除了 strict_rejection_sample Triton 内核，所有 MRV2 推测解码路径都改为 probabilistic_rejection_sample，可能引入回归。
  3. 默认行为变更：默认值从 'strict' 变为 'standard'（等价旧 'probabilistic' + 'greedy' draft），用户可能观察到不同的采样行为。
  4. Triton 内核正确性：新增的 HAS_DRAFT_LOGITS 编译常量可能导致条件分支未覆盖的边界情况。
  5. 作用域局限：draft_sample_method 当前仅影响 MRV2 的 Eagle 模型，其他推测方法或 MRV1 不受影响。 - 影响：用户：使用 MRV2 + speculative decoding 的用户将看到接受率提升（benchmark 显示从 17.55% 提升至 20.63%），输出吞吐量提升约 14-18%。需要更新 config 中的 rejection_sample_method 键名。

系统：默认不分配 draft logits 张量（draft_logits = None），减少了 GPU 内存占用；Triton 内核绕过了不必要的计算，提升效率。

团队：代码结构更清晰，删除了重复的 strict rejection 逻辑，后续维护和优化更简便。

- 风险标记：配置兼容性变更，核心路径重构，Triton 内核逻辑变更，默认行为变更

关联脉络

- 暂无明显关联 PR