PR #40269 完整报告

vllm-project/vllm

[Bugfix][Spec Decode] Wire draft_probs into probabilistic draft_model rejection

合并时间：2026-05-14 09:04

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/40269

执行摘要

- 一句话：修复 V1 speculative decoding 中 draft_probs 未传递使 probabilistic rejection 失效
- 推荐动作：值得精读。本 PR 虽然改动量中等，但修复了一个重要的正确性问题，展示了 speculative decoding 中 draft_probs 的完整生命周期：从 proposer 采样时捕获，跨模块缓存，到 GPUModelRunner 按请求重新排列，最终传递给 rejection sampler。设计模式清晰，配套测试完善。尤其推荐关注 _get_spec_decode_draft_probs 中的请求顺序对齐逻辑。

功能与动机

关联 Issue #40149 指出，在 V1 中使用 draft_model 与 probabilistic 拒绝采样时，GPUModelRunner._sample() 传递给 RejectionSampler 的 draft_probs 参数始终为 None，导致 draft_prob 默认化为 1，偏离了 Leviathan et al. (2022) 定义的分布匹配逻辑（p(x)/q(x) 比率）。本 PR 旨在将 draft 模型的概率分布实际传入 rejection 流程，以恢复正确的 probabilistic 拒绝采样行为。

实现拆解

1. Proposer 侧采样与概率捕获：在 vllm/v1/spec_decode/llm_base_proposer.py 中新增 _sample_from_logits 和 _sample_draft_tokens 方法。当配置为 rejection_sample_method="standard" 且 draft_sample_method="probabilistic" 时，使用 compute_probs_and_sample_next_token 进行概率采样并返回分布；否则回退到贪婪采样。每次 propose() 调用末尾将 _last_draft_probs 缓存起来。
2. Runner 侧缓存与重新排列：在 vllm/v1/worker/gpu_model_runner.py 中新增 _draft_probs 和 _draft_prob_req_ids 属性，在 propose_draft_token_ids() 中从 proposer 的 take_last_draft_probs() 获取概率并关联当前请求 ID；在 sample_tokens() 开始时重置。
3. 按请求顺序提取：_get_spec_decode_draft_probs() 方法根据当前 input_batch.req_ids 和 num_draft_tokens，从缓存中按请求顺序提取对应的概率行和切片，处理缺失请求和零 draft 数量。
4. 传递有效概率：_sample() 中调用 _get_spec_decode_draft_probs() 并将返回值传入 rejection_sampler，替换原来的 None。

5. 配置重命名：vllm/config/speculative.py 中将 DraftSampleMethod 从 'greedy' | 'gumbel' 改为 'greedy' | 'probabilistic'，并更新文档字符串。
6. 测试配套：新增三个测试用例：test_propose_stores_probabilistic_draft_probs（验证 proposer 缓存概率）、test_sample_passes_reordered_draft_probs_to_rejection_sampler（验证 runner 重新排列并传递）、以及配置校验测试。

关键文件：

- vllm/v1/spec_decode/llm_base_proposer.py（模块 推测解码提议器；类别 source；类型 core-logic；符号 _sample_from_logits, _sample_draft_tokens, take_last_draft_probs）：核心提议器，新增 _sample_from_logits 和 _sample_draft_tokens 以在 probabilistic 模式下捕获 draft probabilities，并新增 take_last_draft_probs 供 GPUModelRunner 获取。
- vllm/v1/worker/gpu_model_runner.py（模块 模型运行器；类别 source；类型 data-contract；符号 _get_spec_decode_draft_probs）：模型运行器，新增缓存 _draft_probs 和 _draft_prob_req_ids，新增 _get_spec_decode_draft_probs 方法按请求顺序重新排列 draft 概率，并替换以前传递的 None。
- tests/v1/spec_decode/test_eagle.py（模块 Eagle 测试；类别 test；类型 test-coverage；符号 test_propose_stores_probabilistic_draft_probs, fake_compute_probs）：新增 test_propose_stores_probabilistic_draft_probs 验证 proposer 在 probabilistic 模式下正确缓存概率，并扩展了 _create_proposer 以支持新的配置参数。
- tests/v1/worker/test_gpu_model_runner.py（模块 模型运行器测试；类别 test；类型 test-coverage；符号 test_sample_passes_reordered_draft_probs_to_rejection_sampler）：新增 test_sample_passes_reordered_draft_probs_to_rejection_sampler 验证 GPUModelRunner 正确重新排序 draft_probs 并传递给 rejection sampler。
- tests/test_config.py（模块 配置测试；类别 test；类型 test-coverage；符号 test_draft_sample_method_probabilistic_is_accepted, test_draft_sample_method_gumbel_is_rejected）：新增 test_draft_sample_method_probabilistic_is_accepted 和 test_draft_sample_method_gumbel_is_rejected 测试验证配置接受 'probabilistic' 并拒绝 'gumbel'。
- vllm/config/speculative.py（模块 推测配置；类别 source；类型 core-logic；符号 DraftSampleMethod）：核心配置变更，将 DraftSampleMethod 从 'greedy' | 'gumbel' 改为 'greedy' | 'probabilistic'，并更新文档字符串。

关键符号：_sample_from_logits, _sample_draft_tokens, take_last_draft_probs, _get_spec_decode_draft_probs

关键源码片段

vllm/v1/spec_decode/llm_base_proposer.py

核心提议器，新增 _sample_from_logits 和 _sample_draft_tokens 以在 probabilistic 模式下捕获 draft probabilities，并新增 take_last_draft_probs 供 GPUModelRunner 获取。

```
# llm_base_proposer.py ( 新增字段和方法 )
# 在 __init__ 中启用条件
```

```
    self._enable_probabilistic_draft_probs = (
        self.speculative_config.rejection_sample_method == "standard"
        and self.speculative_config.draft_sample_method == "probabilistic"
    )
    self._last_draft_probs: torch.Tensor | None = None

    def _sample_from_logits(self, logits, sampling_metadata):
        """根据配置和采样元数据决定是否使用概率采样。
        Returns: (采样 token ids, 概率分布或 None)
        """
        if not self._enable_probabilistic_draft_probs:
            return logits.argmax(dim=-1), None # 贪婪模式
        if sampling_metadata.all_greedy:
            return logits.argmax(dim=-1), None # 全部贪婪则无需概率
        # 使用 compute_probs_and_sample_next_token 进行概率采样
        return compute_probs_and_sample_next_token(logits, sampling_metadata)

    def _sample_draft_tokens(self, hidden_states, sampling_metadata):
        """对 draft 模型隐藏状态进行采样。
        在 probabilistic 模式下计算 logits 后调用 _sample_from_logits。
        """
        if not self._enable_probabilistic_draft_probs or sampling_metadata.all_greedy:
            return self._greedy_sample(hidden_states), None
        logits = self.model.compute_logits(hidden_states)
        return self._sample_from_logits(logits, sampling_metadata)

    def take_last_draft_probs(self):
        return self._last_draft_probs

    # 在 propose() 中每次调用前重置
    def propose(self, ...) -> torch.Tensor:
        self._last_draft_probs = None
        ...
        # 调用 _sample_draft_tokens 获得 draft_probs
        draft_token_ids, draft_probs = self._sample_draft_tokens(sample_hidden_states, sampling_
        metadata)
        if draft_probs is not None:
            self._last_draft_probs = draft_probs.view(-1, self.num_speculative_tokens, draft_probs.
            shape[-1]).contiguous()
        return draft_token_ids.view(-1, self.num_speculative_tokens)
```

vllm/v1/worker/gpu_model_runner.py

模型运行器，新增缓存 _draft_probs 和 _draft_prob_req_ids，新增 _get_spec_decode_draft_probs 方法按请求顺序重新排列 draft 概率，并替换以前传递的 None。

```
    # gpu_model_runner.py ( 新增缓存和提取方法 )
    # 在 __init__ 中新增属性
    self._draft_probs: torch.Tensor | None = None
```

```
self._draft_prob_req_ids: list[str] | None = None

# 在 propose_draft_token_ids 和 sample_tokens 的开始重置
def propose_draft_token_ids(self, scheduler_output):
    self._draft_probs = None
    self._draft_prob_req_ids = None
    # ... 原有逻辑，在获得 draft_probs 后缓存
    draft_probs = proposer.take_last_draft_probs()
    if draft_probs is not None:
        self._draft_probs = draft_probs
        self._draft_prob_req_ids = list(self.input_batch.req_ids)

# 新方法：将缓存的 draft_probs 按当前批次的请求顺序和 draft 数量重新排列
def _get_spec_decode_draft_probs(self, spec_decode_metadata):
    if self._draft_probs is None or self._draft_prob_req_ids is None:
        return None
    # 构建 请求 ID -> 缓存行索引 的映射
    row_by_req_id = {req_id: idx for idx, req_id in enumerate(self._draft_prob_req_ids)}
    draft_probs_rows = []
    # 遍历当前批次的请求和对应的 draft 数量
    for req_id, num_draft in zip(self.input_batch.req_ids, spec_decode_metadata.num_draft_
    tokens):
        if num_draft == 0:
            continue # 跳过无 draft 的请求，不影响排列顺序
        row_idx = row_by_req_id.get(req_id)
        if row_idx is None:
            # 若请求未在缓存中找到（如新请求），回退到 None
            logger.warning("Missing cached draft probabilities for request %s; "
                           "falling back to legacy speculative rejection behavior.", req_id)
            return None
        draft_probs_rows.append(self._draft_probs[row_idx, :num_draft])
    if not draft_probs_rows:
        return None
    return torch.cat(draft_probs_rows, dim=0).contiguous()

# 在 _sample 中使用
def _sample(self, logits, spec_decode_metadata):
    ...
    draft_probs = self._get_spec_decode_draft_probs(spec_decode_metadata)
    sampler_output = self.rejection_sampler(
        spec_decode_metadata,
        draft_probs, # 之前是 None
        logits,
        sampling_metadata,
    )
    return sampler_output
```

评论区精华

主要讨论包括：

1. benchislett 建议将 draft_sample_method 从 'gumbel' 重命名为 'probabilistic'，因为 V1 使用的是常规概率采样而非 Gumbel 噪声，bedeks 采纳并重命名。
2. benchislett 对 _get_spec_decode_draft_probs 中 num_draft == 0 的情况处理提出疑问，bedeks 解释跳过零长度的条目不会打乱顺序，且有回归测试覆盖（请求 [a,b,c] 分别对应 draft 数量 [2,0,1] 时，输出应为 [a[:2], c[:1]]）。
3. benchislett 指出 draft_probs_list 初始化应为 None 而非 []，bedeks 修正。

- draft_sample_method 命名调整 (design): bedeks 采纳并重命名为 probabilistic，同时更新文档字符串。
- 处理请求无 draft token 的边界情况 (correctness): bedeks 解释跳过零长度条目不会影响顺序，且有回归测试验证请求顺序 [a,b,c] 对应 draft 数量 [2,0,1] 时输出正确。
- draft_probs_list 初始化值 (correctness): bedeks 确认并修正为 None。

风险与影响

- 风险：
  1. 仅当 rejection_sample_method='standard' 且 draft_sample_method='probabilistic' 时才启用新路径，默认行为不受影响。
  2. 新增的 _last_draft_probs 等缓存若不及时清理，可能导致内存占用缓慢增长；代码在 propose_draft_token_ids 和 sample_tokens 中都设置了显式重置为 None。
  3. 请求顺序重新映射逻辑假定 input_batch.req_ids 与 spec_decode_metadata.num_draft_tokens 顺序一致，若不符可能导致概率错位；测试验证了正确性。
  4. 若请求缺失缓存概率（如新的请求进入但概率尚未填充），代码会回退到 None（即 fallback 行为），不会崩溃，但可能短暂回归旧行为。 - 影响：影响范围限于 V1 speculative decoding 中采用 probabilistic rejection 的用户。该修复使 acceptance rate 从约 0.22 提升至 0.45，显著提高推理性能。对使用默认 greedy 采样的用户无影响。新增测试覆盖确保了核心路径的可靠性。配置变更要求用户将 'gumbel' 改为 'probabilistic'，但之前该设置仅适用于 MRV2，V1 本应使用 'probabilistic'，因此这是预期修复。 - 风险标记：核心路径变更，数据流顺序依赖，缓存生命周期

关联脉络

- PR #40149 [Feature]: Speculative Decoding using draft_model does not use draft_probs: 此 issue 报告了 draft_probs 未传递的问题，是本次 PR 的直接动机。