PR #35205 完整报告

sgl-project/sglang

[Sampling] Restore finite top-k requirement for sampling masks

合并时间：2026-08-21 07:55

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35205

执行摘要

- 一句话：恢复有限 top_k 的采样掩码要求，拒绝 top-p-only 请求
- 推荐动作：值得快速精读，尤其是 scheduler.py 中采样掩码的请求校验分支，以及测试对拒绝路径的回归设计。它体现了“功能放宽必须匹配资源上界”的设计决策，对理解采样掩码的安全边界和后续 #35765 的通用支持有参考价值。

功能与动机

33593 出于 RL rollout 的常见配置（top_k=-1 配 top_p<1）放开了 return_sampling_mask 对有限 top_k 的要求。但 PR #35205 指出，top_p 约束的是概率质量而非保留 token 数，分布的尾部可能让掩码逼近全词表，采样掩码重建和返回元数据均无界。因此需要恢复有限 top_k 作为安全上限，通用方案在 #35765 中设计。

实现拆解

1. 恢复 scheduler.py 中 handle_generate_request 的请求校验：将原先“top_k != TOP_K_ALL 或 top_p < 1.0 即可放行”的放宽逻辑改为“req.return_sampling_mask 且 req.sampling_params.top_k == TOP_K_ALL 时拒绝”，并更新错误消息为 return_sampling_mask requires finite top_k; top_p-only sampling is valid but can return huge masks in the tail, blowing up metadata, so we need a safety cap. 该拒绝走 set_finish_with_abort、init_req_max_new_tokens、_add_request_to_queue 的标准错误路径，不影响调度主流程。
2. 同步调整 test/registered/sampling/test_sampling_mask.py：删除 test_generate_returns_top_p_only_sampling_mask 与 test_chat_completions_returns_top_p_only_sampling_mask；新增辅助方法 _assert_rejects_unbounded_sampling_mask 和测试 test_generate_rejects_unbounded_sampling_mask；将 Chat 测试 test_chat_completions_returns_sampling_mask 改为显式传 top_k 与 top_p，保持有限 top_k 下 Chat 路径的覆盖。

3. 无 schema、配置或部署配套改动；CI 注册不变，仍运行在 1-gpu-small 与 AMD 套件。

关键文件：

- python/sglang/srt/managers/scheduler.py（模块 调度器；类别 source；类型 core-logic；符号 handle_generate_request）：采样掩码请求校验的核心改动：恢复有限 top_k 强制要求，防止 top-p-only 采样导致掩码元数据无界膨胀。
- test/registered/sampling/test_sampling_mask.py（模块 采样掩码；类别 test；类型 test-coverage；符号 _assert_rejects_unbounded_sampling_mask, test_generate_rejects_unbounded_sampling_mask, test_chat_completions_returns_sampling_mask）：端到端回归覆盖：将 top-p-only 的预期从接受改为拒绝，并保留 Chat 路径在有限 top_k 下的可用性。

关键符号：handle_generate_request, _assert_rejects_unbounded_sampling_mask, test_generate_rejects_unbounded_sampling_mask, test_chat_completions_returns_sampling_mask

关键源码片段

python/sglang/srt/managers/scheduler.py

采样掩码请求校验的核心改动：恢复有限 top_k 强制要求，防止 top-p-only 采样导致掩码元数据无界膨胀。

handle_generate_request 中的采样掩码校验分支（整理后）：

```
    # ... 前置校验（dflash、disaggregation、spec 等）...

    # 核心变更：恢复有限 top_k 要求。top_p-only 采样虽然概率上合法，
    # 但掩码重建是逐 token 展开的元数据，长尾分布下可能接近全词表，
    # 造成 metadata 膨胀，因此必须强制有限 top_k 作为安全上限。
    if req.return_sampling_mask and req.sampling_params.top_k == TOP_K_ALL:
        error_msg = (
            "return_sampling_mask requires finite top_k; top_p-only sampling "
            "is valid but can return huge masks in the tail, blowing up "
            "metadata, so we need a safety cap."
        )
        req.set_finish_with_abort(error_msg)
        self.init_req_max_new_tokens(req)
        self._add_request_to_queue(req)
        return

    # 投机解码按接受的 token 输出，无法与掩码 1:1 对齐，拒绝。
    if req.return_sampling_mask and not self.spec_algorithm.is_none():
        error_msg = (
            "return_sampling_mask is not supported with speculative decoding."
        )
        req.set_finish_with_abort(error_msg)
        self.init_req_max_new_tokens(req)
        self._add_request_to_queue(req)
```

```
        return

    # ascend 后端直接从 logits 采样，不构建 top-k/top-p 支持，拒绝。
    if req.return_sampling_mask and get_exec().kernel.sampling_backend == "ascend":
        error_msg = (
            "return_sampling_mask is not supported with the ascend "
            "sampling backend."
        )
        req.set_finish_with_abort(error_msg)
        self.init_req_max_new_tokens(req)
        self._add_request_to_queue(req)
        return

    # ... 后续的多模态与入队逻辑 ...
```

test/registered/sampling/test_sampling_mask.py

端到端回归覆盖：将 top-p-only 的预期从接受改为拒绝，并保留 Chat 路径在有限 top_k 下的可用性。

测试侧的关键覆盖如下：

```
    _INVALID_SAMPLING_MASK_ERROR = (
        "top_p-only sampling is valid but can return huge masks in the tail"
    )


    def _assert_rejects_unbounded_sampling_mask(self, sampling_params):
        # 断言无有限 top_k 的采样掩码请求应被拒绝（400）。
        response = self._post_generate(sampling_params)
        self.assertEqual(response.status_code, 400, response.text)
        self.assertIn(_INVALID_SAMPLING_MASK_ERROR, response.text)


    def test_generate_rejects_unbounded_sampling_mask(self):
        # top_p < 1 但 top_k 不限：概率质量有界，但返回掩码长度无界。
        self._assert_rejects_unbounded_sampling_mask(
            {"temperature": 1.0, "top_p": _TOP_P,
             "max_new_tokens": _MAX_NEW_TOKENS, "ignore_eos": True}
        )
        # top_p = 1 且 top_k 不限：显式全词表请求，同样拒绝。
        self._assert_rejects_unbounded_sampling_mask(
            {"temperature": 1.0, "top_p": 1.0,
             "max_new_tokens": _MAX_NEW_TOKENS, "ignore_eos": True}
        )
```

评论区精华

本 PR 没有代码行级 review 评论，hnyls2002 直接 APPROVED。核心决策由 PR body 与关联 Issue 承载：top_p 不能约束掩码重建长度，有限 top_k 是当前的安全兜底，通用支持由

#35765 跟踪。CI 评论里 gongy 只触发了 /rerun-test test/registered/sampling/test_sampling_mask.py，属于验证动作，未产生进一步技术讨论。

- 暂无高价值评论线程

风险与影响

- 风险：行为兼容性风险：依赖 #33593 top-p-only 掩码的客户端（如 RL rollout）现在会收到 400，需要显式传入有限 top_k 或等待 #35765。错误消息从“cannot return the full vocabulary”变为“requires finite top_k...”，依赖文本匹配的客户端或测试可能失效。校验位于 handle_generate_request（scheduler.py），所有 /generate 请求都会经过，但仅在 return_sampling_mask 开启时生效，影响面可控。测试覆盖了拒绝与正常路径，但未对长尾分布下掩码长度做显式压测，作为安全回退可接受。
- 影响：影响范围集中在采样掩码功能的使用者（RL 训练、RL 推理客户端）：服务器端从返回巨型元数据改为快速失败拒绝，降低网络传输与内存占用。对团队而言，本 PR 与 #35765 通用支持形成明确递进，后续实现需要保持请求契约一致；对 OpenAI Chat 路径的功能暴露无影响，只要调用方提供有限 top_k。
- 风险标记：行为兼容性回退，核心请求校验路径，错误消息变更，后续依赖 #35765

关联脉络

- PR #33593 [RL] Expose top-p-only sampling masks: 本 PR 回退该 PR 放宽有限 top_k 要求的部分，恢复安全校验。
- PR #35765 Support sampling masks without finite top-k: 跟踪通用支持的设计，本 PR 的安全上限是通往该目标的临时约束。