PR #24874 完整报告

sgl-project/sglang

Reject repetition_penalty=0 in SamplingParams.verify()

合并时间：2026-05-13 12:25

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24874

执行摘要

- 一句话：拒绝 repetition_penalty=0 避免 GPU 崩溃
- 推荐动作：值得精读。这是一个典型的“输入验证防止内核崩溃”的 bugfix，展示了如何通过早期验证避免 GPU 级别的灾难性失败。设计决策值得在其他除法相关参数验证中复用。

功能与动机

PR body 明确指出：repetition_penalty=0 会令 kernel apply_scaling_penalties 将 logits 除以 0，产生 inf → NaN，导致所有 TP rank 因 device-side assert 崩溃，引擎须完全重启。此问题由上游 wrapper 将 0.0 作为未指定值的 fallback 而触发。

实现拆解

1. 收紧验证范围：在 python/sglang/srt/sampling/sampling_params.py 的 verify() 方法中，将第 142 行条件从 0.0 <= self.repetition_penalty <= 2.0 改为 0.0 < self.repetition_penalty <= 2.0，并更新错误信息为 "repetition_penalty must be in (0, 2] (1.0 = no penalty)"。
2. 调整单元测试：在 test/registered/unit/sampling/test_sampling_params.py 中：
   - 删除原 test_repetition_penalty_boundaries_valid（接受 0.0 和 2.0）。
   - 新增 test_repetition_penalty_zero_raises 验证 repetition_penalty=0.0 时抛出 ValueError。
   - 新增 test_repetition_penalty_boundary_two_valid 仅验证 2.0 边界有效。
   - 新增 test_repetition_penalty_small_positive_valid 验证 1e-3 等小正值仍被接受。
   - 更新 test_repetition_penalty_negative_raises 的 docstring 以反映新范围。
3. 更新文档：在 docs/basic_usage/sampling_params.md 中将有效范围从 [0, 2] 改为 (0, 2]。

关键文件：

- python/sglang/srt/sampling/sampling_params.py（模块 采样参数；类别 source；类型 core-logic；符号 verify）：核心验证逻辑所在文件，将 repetition_penalty 范围从 [0, 2] 收紧为 (0, 2]，防止 0 值传播到 GPU kernel。
- test/registered/unit/sampling/test_sampling_params.py（模块 采样参数；类别 test；类型 test-coverage；符号 test_repetition_penalty_zero_raises,

test_repetition_penalty_boundaries_valid, test_repetition_penalty_boundary_two_valid, test_repetition_penalty_small_positive_valid）：测试覆盖了新的拒绝逻辑、边界值和极小正值，确保验证正确。

- docs/basic_usage/sampling_params.md（模块 文档；类别 docs；类型 documentation）：文档同步更新有效范围，保持与代码一致。

关键符号：SamplingParams.verify

关键源码片段

python/sglang/srt/sampling/sampling_params.py

核心验证逻辑所在文件，将 repetition_penalty 范围从 [0, 2] 收紧为 (0, 2]，防止 0 值传播到 GPU kernel。

```
# python/sglang/srt/sampling/sampling_params.py 第 142-146 行 （head 版本）
# 关键变更：将 <= 改为 <，排除 0.0
if not 0.0 < self.repetition_penalty <= 2.0:
    raise ValueError(
        "repetition_penalty must be in (0, 2] (1.0 = no penalty), "
        f"got {self.repetition_penalty}."
    )
```

test/registered/unit/sampling/test_sampling_params.py

测试覆盖了新的拒绝逻辑、边界值和极小正值，确保验证正确。

```
# test/registered/unit/sampling/test_sampling_params.py （head 版本）
# 新增测试：验证 repetition_penalty=0 引发 ValueError
def test_repetition_penalty_zero_raises(self):
    """Test that verify() rejects repetition_penalty=0.

    A value of 0 makes the sampling kernel divide logits by 0, producing
    inf/NaN in the probability tensor and crashing every TP rank with a
    device-side assert.
    """
    sp = self._make(repetition_penalty=0.0)
    with self.assertRaises(ValueError):
        sp.verify(self.VOCAB_SIZE)

# 新增测试：验证小正值（如 1e-3）仍被接受
def test_repetition_penalty_small_positive_valid(self):
    """Test that a small positive repetition_penalty (e.g. 1e-3) is accepted."""
    self._make(repetition_penalty=1e-3).verify(self.VOCAB_SIZE)
```

评论区精华

reviewer Ratish1 建议同步更新文档 docs/basic_usage/sampling_params.md，作者随后完成修改。Ratish1 和 mergrer hzh0425 均 approve，无争议。

- 文档更新建议 (documentation): 作者已同步更新文档。

风险与影响

- 风险：范围收紧可能拒绝先前被认为合法的 repetition_penalty=0.0 请求，但该值本身在语义上无意义（0 惩罚相当于取消所有 token 的概率），且内核中必然导致崩溃。不存在回归风险。
- 影响：影响范围：仅影响提交了 repetition_penalty=0.0 的请求，这些请求现在会提前返回 ValueError 而非让引擎崩溃。用户可改用极小正值如 1e-3 趋近于无惩罚，或使用 1.0 表示无惩罚。这符合更健壮的防御性编程实践。
- 风险标记：暂无

关联脉络

- 暂无明显关联 PR