

PR #38791 完整报告

sgl-project/sglang

test(lora): enable ROCm logprob accuracy coverage

合并时间: 2026-09-11 11:29

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/38791>

执行摘要

- 一句话: LoRA logprob 准确性测试接入 AMD ROCm CI
- 推荐动作: 此 PR 以测试为主, 不建议深入精读源码, 但值得快速浏览, 作为“为既有测试做跨平台移植”的参考案例。两个值得注意的设计决策: 一是用 `is_hip()` / `is_xpu()` 做平台条件化的后端选择, 而不是为不同平台复制整套测试; 二是按资源需求把测试挂到不同 CI suite (单卡 vs 4 卡), 体现了测试成本与覆盖面的平衡。对计划在 SGLang 中为其他硬件平台扩展测试覆盖的贡献者有直接借鉴意义。

功能与动机

PR body 明确动机为“Add ROCm regression coverage for Qwen3-8B and GPT-OSS-20B LoRA logprob accuracy”。此前这两个 LoRA logprob 准确性测试只通过 `register_cuda_ci` 注册到 CUDA CI, AMD/ROCm 平台上的 LoRA 数值正确性没有任何回归测试兜底。测试本身将 SGLang 推理产出的 logprob 与训练参考数据 (HuggingFace 数据集 `yushengsu/lora-diff-Qwen3-8B` 与 `lora-diff-gpt-oss-20b`) 计算 KL 散度, 并用 $5e-3$ 阈值判定是否回归。

实现拆解

变更入口是两个已存在的 LoRA logprob 回归测试文件, 实现分四步:

1. AMD CI 注册: 两个文件都从 `sglang.test.ci.ci_register` 引入 `register_amd_ci`, 并按资源需求挂到不同 suite。Qwen3-8B (单卡, TP=1) 注册到 `stage-b-test-1-gpu-small-amd`, 预估 120s; GPT-OSS-20B (TP=4) 注册到 `stage-c-test-4-gpu-amd`, 预估 180s。CUDA 侧原有注册保持不变。
2. Qwen3-8B 平台化后端选择: 在 `test_lora_qwen3_8b_logprob_diff.py` 的 `attention_backend_kwargs()` 中新增 `is_hip()` 分支, ROCm 下 `attention/prefill/decode` 三个后端全部设为 `triton`; XPU 分支维持 `intel_xpu`; CUDA 默认分支维持 `flashinfer + fa4`。
3. GPT-OSS-20B 常量重构: `test_lora_gpt_oss_20b_logprob_diff.py` 将原来硬编码的 `attention_backend="flashinfer"` 与 `fa4` 常量重构为模块级 `ATTENTION_BACKEND`、`PREFILL_ATTENTION_BACKEND`、`DECODE_ATTENTION_BACKEND`, 用 `is_hip()` 三元表达式在 `triton` 与 `flashinfer/fa4` 之间切换, `engine` 构造改为引用这些常量, 消除了重复的平台判断。

4. 依据与验证：fa4 依赖 CUTLASS CUTE DSL、flashinfer 同为 CUDA-only，在 ROCm 上只能用 Triton 作为统一 attention 后端。PR body 提供了 MI355X ROCm 7.2 实测：Qwen3-8B 单卡 $KL=2.818798e-04$ ，GPT-OSS-20B TP=4 $KL=1.429855e-03$ ，均通过 $5e-3$ 阈值，针对性 pre-commit 检查通过。

无配置、schema 或部署配套改动；测试数据与适配器仍通过 HuggingFace snapshot_download 获取。

关键文件：

- test/registered/lora/test_lora_qwen3_8b_logprob_diff.py（模块 LoRA 测试；类别 test；类型 test-coverage；符号 attention_backend_kwargs）：LoRA logprob 回归测试核心文件：新增 register_amd_ci 注册，并在 attention_backend_kwargs() 中为 ROCm 增加 Triton attention 分支，是平台适配的核心改动所在。
- test/registered/lora/test_lora_gpt_oss_20b_logprob_diff.py（模块 LoRA 测试；类别 test；类型 test-coverage；符号 ATTENTION_BACKEND, PREFILL_ATTENTION_BACKEND, DECODE_ATTENTION_BACKEND, test_lora_gpt_oss_20b_logprob_accuracy）：另一个被接入 AMD CI 的 LoRA logprob 测试文件：除新增 register_amd_ci 外，还把硬编码的 attention 后端重构为基于 is_hip() 的平台条件化常量，展示了一种更简洁的平台适配写法。

关键符号：attention_backend_kwargs, test_lora_gpt_oss_20b_logprob_accuracy

关键源码片段

test/registered/lora/test_lora_qwen3_8b_logprob_diff.py

LoRA logprob 回归测试核心文件：新增 register_amd_ci 注册，并在 attention_backend_kwargs() 中为 ROCm 增加 Triton attention 分支，是平台适配的核心改动所在。

```
# 注册 CI: CUDA 沿用原配置，AMD 侧进入 stage-b 单卡套件。
register_cuda_ci(est_time=56, stage="extra-a", runner_config="1-gpu-large")
register_amd_ci(est_time=120, suite="stage-b-test-1-gpu-small-amd")
```

```
def attention_backend_kwargs():
    """Engine attention-backend kwargs for the current platform.

    fa4 与 flashinfer 都是 CUDA-only: fa4 派发到 CUTLASS CUTE DSL 内核，
    离开 CUDA 后无法导入。ROCm 上改用 Triton 作为统一 attention 后端，
    XPU 则继续走 intel_xpu 融合路径。
    """
    from sglang.srt.utils import is_hip, is_xpu

    if is_xpu():
        return {"attention_backend": "intel_xpu"}
    if is_hip():
        # ROCm 没有 fa4/flashinfer 内核，prefill 与 decode 全部落到 Triton。
        return {
```

```

        "attention_backend": "triton",
        "prefill_attention_backend": "triton",
        "decode_attention_backend": "triton",
    }
    # CUDA 默认路径保持不变: flashinfer + fa4。
    return {
        "attention_backend": "flashinfer",
        "prefill_attention_backend": PREFILL_ATTENTION_BACKEND,
        "decode_attention_backend": DECODE_ATTENTION_BACKEND,
    }

```

test/registered/lora/test_lora_gpt_oss_20b_logprob_diff.py

另一个被接入 AMD CI 的 LoRA logprob 测试文件：除新增 register_amd_ci 外，还把硬编码的 attention 后端重构为基于 is_hip() 的平台条件化常量，展示了一种更简洁的平台适配写法。

平台条件化的 attention 后端选择：ROCm 用 Triton，CUDA 保持 fa4/flashinfer。

ATTENTION_BACKEND = "triton" if is_hip() else "flashinfer"

PREFILL_ATTENTION_BACKEND = "triton" if is_hip() else "fa4"

DECODE_ATTENTION_BACKEND = "triton" if is_hip() else "fa4"

CUDA 继续跑 4 卡 B200，AMD 侧进入 stage-c 4 卡套件。

register_cuda_ci(est_time=96, stage="extra-b", runner_config="4-gpu-b200")

register_amd_ci(est_time=180, suite="stage-c-test-4-gpu-amd")

engine 构造改为引用平台化常量，替代原先硬编码的 "flashinfer"。

```

engine = sgl.Engine(
    model_path=BASE_MODEL,
    tp_size=TP_SIZE, # TP=4, 与 stage-c 的 4 卡 suite 匹配
    enable_lora=True,
    max_lora_rank=MAX_LORA_RANK,
    lora_paths={"my_lora": adapter_path},
    lora_backend=LORA_BACKEND,
    attention_backend=ATTENTION_BACKEND,
    moe_runner_backend=MOE_RUNNER_BACKEND,
    experts_shared_outer_loras=EXPERTS_SHARED_OUTER_LORAS,
    prefill_attention_backend=PREFILL_ATTENTION_BACKEND,
    decode_attention_backend=DECODE_ATTENTION_BACKEND,
)

```

评论区精华

该 PR 没有实质性的 review 讨论：HaiShaw 直接 approve，review 评论数为 0。唯一互动是作者 Arist12 在评论中 @ 了 yushengsu-thu（即测试所依赖 HF 数据集 lora-diff-Qwen3-8B 与 lora-diff-gpt-oss-20b 的作者），推测为确认参考数据来源，未形成技术辩论。作者在 PR body 中用 MI355X ROCm 7.2 上的实测 KL 指标（两个模型分别为 $2.8e-4$ 与 $1.4e-3$ ）主动给出后端切换的数值安全性证据，降低了 reviewer 的验证成本。

- 数据集维护者确认 (question): 未形成实质技术讨论；HaiShaw 直接 approve，无 review 评论。

风险与影响

- 风险：

1. CI 资源与耗时：两个测试以 120s、180s 预估时长加入 AMD CI（其中 GPT-OSS-20B 需要 TP=4 的 4 卡资源），会加长 AMD 测试矩阵总运行时间；若 CI 池多卡配额紧张，可能出现排队或间歇失败。
2. 外部依赖：测试运行时通过 snapshot_download 从 HuggingFace 拉取数据集与适配器，HF 资源缺失或网络抖动会直接导致 CI 失败，这是该类回归测试固有的外部脆弱性。
3. 跨平台数值基线：ROCm 上采用 Triton attention，数值路径与 CUDA 的 fa4/flashinfer 不同， $5e-3$ 阈值目前仅由一次实测支撑；后续 Triton 内核或 LoRA 计算路径演进时，阈值可能需要重新校准。
4. 回归面：改动仅限测试文件，不触碰任何源码主路径，对生产代码无回归风险。
 - 影响：对 AMD/ROCm 用户：LoRA 推理的 logprob 数值正确性从此进入持续回归守护范围，Qwen3-8B（稠密）与 GPT-OSS-20B（MoE）两种代表性模型均有覆盖。对 CI 系统：AMD 测试矩阵新增两个重型模型级测试（含一个 4 卡 TP=4 用例），CI 总耗时与资源占用上升。对团队协作：后续任何涉及 LoRA 计算路径、Triton attention 或 ROCm 内核的改动都会更快暴露数值回归，形成 AMD 平台的质量闭环。
 - 风险标记：AMD CI 耗时增加，依赖外部 HF 数据下载，跨平台数值基线待观察

关联脉络

- PR #38767 [AMD][CI] Retire the ROCm 7.0 kernel wheel: 同为 AMD/ROCm CI 与平台策略演进：该 PR 退役 ROCm 7.0 kernel wheel 并收敛构建矩阵，本 PR 在 ROCm 7.2（MI355X）上新增 LoRA 回归测试，体现 AMD 平台测试覆盖扩展的同一方向。