PR #2040 完整报告

radixark/miles

[AMD] DeepSeek-V4: use the ROCm precision-parity norm path

合并时间：2026-08-01 06:09

原文链接：http://prhub.com.cn/radixark/miles/pull/2040

执行摘要

- 一句话：AMD DSv4 训练改走 ROCm 精度对齐 norm 路径
- 推荐动作：值得精读 PR body——这是一个典型的 " 环境变量静默失效导致数值漂移 " 调试案例：从指标回归（0.045 → 0.213）出发，层层追踪到上游 commit 对 producer/consumer 契约和 compressor 选择的改动，再用交叉验证（分别应用两个修复的中间值）证明双重根因。对 AMD 脚本维护者和依赖 SGLang 内部 env 的团队有借鉴价值；阅读时建议关注其验证矩阵的设计（同一 step 0、同权重、同配方的对照），以及精度与吞吐权衡的量化方法。

功能与动机

恢复 AMD MI355X 上 DeepSeek-V4-Flash FP8 RL 训练的 train-rollout logprob 数值一致性。PR body 指出：radixark/miles#1607 落地时 4 节点跑出的 train_rollout_logprob_abs_diff 为 0.045，sglang-miles 升级后同一配方变为 0.213。两个独立上游变更导致该回归：sgl-project/sglang#29275 翻转了 fused RMS FP8 activation scale 的 producer/consumer 契约但未同步 models/deepseek_v4.py（由 sgl-project/sglang#31727 修复）；sgl-project/sglang#28612 移除 pre-V2 compressor 选择后，脚本中原有的 SGLANG_OPT_USE_COMPRESSOR_V2=false 失效，而 CompressorBackendMixin.forward_unified 在 ROCm 上仍区分两条数值不等价的路径（_forward_unified_hip Triton parity 路径与 _forward_compress_all_in_one fused JIT 路径），SGLANG_OPT_USE_JIT_NORM 默认 true 导致 fused JIT 路径被静默启用。logprob 不一致会直接污染 PPO/GRPO 的 GAE 与优势估计，属于训练正确性问题。

实现拆解

1. 定位双重根因：PR body 通过交叉验证区分两个独立的上游问题——sgl-project/sglang#29275 的 FP8 activation scale 布局契约翻转（需上游 #31727 修复，本 PR 无法解决）与 compressor 路径选择。后者在 sgl-project/sglang#28612 删除 pre-V2 CompressorHip 选择逻辑后，仅剩 V2 内部 forward_unified 的两条 ROCm 分支，SGLANG_OPT_USE_JIT_NORM 默认 true 使 fused JIT 路径胜出。
2. 修改启动脚本配置：在 scripts/amd/run_deepseek_v4.py 的 _train 函数 extra_env_vars 字典中，删除已失效的 SGLANG_OPT_USE_COMPRESSOR_V2: "false"（上游已移除对应 pre-V2 实现），新增 SGLANG_OPT_USE_JIT_NORM: "false"，使 compressor 在 ROCm 上走 _forward_unified_hip 的 Triton 精度对齐路径。

3. 量化验证（4 节点 MI355X，TP8/PP4/EP8，response length 16384，任务 dapo_aime，step 0 同权重同配方）：

配置	abs_diff
当前代码树	0.21290
叠加 #31727 scale 修复	0.13108
再叠加本配置	0.04510
健康参考	0.04482

单独应用本配置（无 scale 修复）只能到 0.16284，证明两个修复缺一不可。

1. 吞吐代价测量：sglang.bench_one_batch 单节点 MI355X（TP4、batch 8、input 1024、output 64）显示 prefill -0.5%、decode -8.0%、端到端 -6.1%，原因是 parity 路径把一个 fused launch 拆成 Triton norm+rope、quant、scatter。
2. 配套与后续：无测试、文档或配置模板文件变更（纯单行 env 变更，且仅 ROCm 路径生效，CUDA 不受影响）。PR body 留下 TODO：修复 fused JIT 路径的数值漂移后移除该 parity fallback。

关键文件：

- scripts/amd/run_deepseek_v4.py（模块 训练脚本；类别 source；类型 configuration；符号 _train）：唯一变更文件。在 _train 的 extra_env_vars 字典中删除已失效的 SGLANG_OPT_USE_COMPRESSOR_V2=false（上游 sglang#28612 移除 pre-V2 compressor 后该开关无效果），新增 SGLANG_OPT_USE_JIT_NORM=false，使 DSv4 compressor 在 ROCm 上从 fused JIT 路径切换到 Triton precision-parity 路径，恢复 train_rollout_logprob_abs_diff 一致性。

关键符号：_train

关键源码片段

scripts/amd/run_deepseek_v4.py

唯一变更文件。在 _train 的 extra_env_vars 字典中删除已失效的 SGLANG_OPT_USE_COMPRESSOR_V2=false（上游 sglang#28612 移除 pre-V2 compressor 后该开关无效果），新增 SGLANG_OPT_USE_JIT_NORM=false，使 DSv4 compressor 在 ROCm 上从 fused JIT 路径切换到 Triton precision-parity 路径，恢复 train_rollout_logprob_abs_diff 一致性。

```
# scripts/amd/run_deepseek_v4.py —— DeepSeek-V4-Flash FP8 RL 启动脚本（ROCm/gfx950 专用），
# _train 内构造注入 rollout 引擎的环境变量，决定 DSv4 compressor 在 ROCm 上的执行路径
extra_env_vars = {
    "SGLANG_SKIP_CHECKPOINT_LOAD_CHECK": "1",
    "SGLANG_DSV4_FP4_EXPERTS": "0",
    "SGLANG_HACK_FLASHMLA_BACKEND": "triton",
```

```
    "SGLANG_OPT_USE_TILELANG_INDEXER": "true",

    # 关键开关：显式关闭 fused JIT norm+rope，走 Triton 精度对齐路径。
    # 背景：上游 sgl-project/sglang#28612 移除 pre-V2 compressor 选择后，
    # 原脚本中的 SGLANG_OPT_USE_COMPRESSOR_V2=false 已失效（本 PR 删除）；
    # 而 SGLANG_OPT_USE_JIT_NORM 默认 true，会让 fused JIT 路径静默胜出，
    # 导致 train_rollout_logprob_abs_diff 从 0.045 漂移到 0.21 量级。
    # 置 false 后，ROCm 走 CompressorBackendMixin._forward_unified_hip，
    # 即 Triton precision-parity 路径：step 0 的 abs_diff 从 0.13108
    # 恢复到 0.04510，接近健康参考 0.04482；代价是 decode 吞吐约 -8%。
    "SGLANG_OPT_USE_JIT_NORM": "false",

    "SGLANG_OPT_USE_FUSED_COMPRESS": "true",
    "SGLANG_HEALTH_CHECK_TIMEOUT": "120",
    "AITER_BF16_FP8_MOE_BOUND": "0",
}
```

评论区精华

Review 过程极简：唯一评审者 guapisolo 直接批准（"LGTM."），无公开评论线程；gemini-code-assist 仅发布一条与本次改动无关的 " 服务已 sunset" 通知。值得注意的技术权衡都记录在 PR body 中而非评审对话里：Triton 分支源码注释明确其存在目的是 precision parity with V1；" 仅应用 compressor 改动只能到 0.16284，两个修复缺一不可 " 的交叉验证；以及 decode 吞吐 -8% 的量化代价与恢复精度的取舍结论。

- 评审与合并 (other): 直接合并；精度恢复被接受为优先目标，吞吐代价作为已知取舍记录在案，并留下 TODO 跟踪 fused JIT 路径的数值漂移修复。

风险与影响

- 风险：
  - 吞吐回退（已量化但需接受）：启用 Triton parity 路径后 decode 吞吐 552.3 → 508.0 token/s（-8.0%），端到端 -6.1%；对 16K response 长度的 RL rollout 会放大该代价，这是恢复精度的必要取舍。
  - 上游依赖未闭合：完整恢复依赖 sgl-project/sglang#31727（当前仍 open）的 fused-RMS scale 修复；若该修复变动或行为漂移，当前配置只能恢复到 0.16284，仍处于漂移区间。
  - 配置语义脆弱：SGLANG_OPT_USE_JIT_NORM 是 SGLang 内部 env，默认值随上游版本变化；本次改动无测试保护（scripts/amd 下脚本无单测覆盖），未来上游若移除 Triton parity 路径（PR body 的 TODO 也计划如此），logprob 漂移会以同样方式静默复发。
  - 影响面：仅 scripts/amd/run_deepseek_v4.py 的 ROCm 分支，CUDA 路径零改动，回归风险小。
  - 影响：对用户：仅影响在 MI355X/gfx950 上运行 DeepSeek-V4-Flash FP8 RL 训练的团队；恢复 train-rollout logprob 一致性直接保障 PPO/GRPO 优势估计与 GAE 的正确性，避免 RL 在错误数值下静默训练。对系统：rollout decode 吞吐下降约 8%，对纯推理

场景不友好，但对 RL 训练场景属于精度优先的合理取舍。对团队：为 AMD 支持线确立了一组经量化验证的 *env* 基线，并暴露了依赖上游 SGLang 内部开关的风险，后续需要跟踪 #31727 与 fused JIT 数值漂移修复。

- 风险标记：依赖上游未合并修复 sglang#31727，*decode* 吞吐下降约 8%，无测试覆盖的配置变更，仅 ROCm 路径生效

关联脉络

- PR #1607 [AMD] Enable DeepSeek-V4-Flash FP8 RL training on MI355X: 本 PR 的直接前身：其设定的 SGLANG_OPT_USE_COMPRESSOR_V2=false 在上游 sglang#28612 后失效，本 PR 将其替换为 SGLANG_OPT_USE_JIT_NORM=false，并复用同一 4 节点 MI355X 验证基线。
- PR #1733 [AMD] Drop inert DSv4 rollout knobs and add an MTP recipe: 同为 scripts/amd/run_deepseek_v4.py 的清理与调优，延续 " 清理失效 AMD 配置项 " 的主题，属于同一脚本维护线。