

# PR #35672 完整报告

sgl-project/sglang

[AMD] Enable draft\_extend CUDA graph for HIP DSA backend

合并时间: 2026-08-25 17:18

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/35672>

## 执行摘要

- 一句话: HIP 使能 DSA draft\_extend CUDA graph 捕获
- 推荐动作: 值得精读, 尤其是 CUDA graph 捕获的允许列表设计。这是一个小幅但影响明确的性能优化, 展示了如何通过后端隔离和本地导入实现跨平台支持。建议关注后续对 DSA HIP 的测试覆盖和长期稳定性。

## 功能与动机

在 HIP 平台上, draft\_extend CUDA graph 捕获仅对 aiter 多步后端和 DSV4 radix 后端开放。GLM MI355 MTP 使用 DeepseekSparseAttnBackend 进行 draft\_extend, 导致捕获被跳过, 每次 extend 都退回 eager launch, 增加了启动开销。PR 目标是使 DSA 后端也能在 HIP 上捕获 graph, 从而提升 MTP 推理性能。

## 实现拆解

1. 修改 HIP 允许列表: 在 eagle\_worker\_v2.py 的 \_capture\_cuda\_graphs 方法中, supports\_hip\_draft\_extend\_graph 判断新增了 isinstance(self.draft\_extend\_attn\_backend, DeepseekSparseAttnBackend) 条件, 使得 DSA 后端也被视为支持 CUDA graph 捕获。
2. 本地导入 DSA 后端: 在 \_is\_hip 分支内通过 from sglang.srt.layers.attention.dsa\_backend import DeepseekSparseAttnBackend 进行本地导入, 确保非 HIP 环境不会因缺少 DSA 依赖而导入失败, 同时保持原有 aiter 和 DSV4 的判断逻辑。
3. 格式化调整: 应用 black 格式化, 将 isinstance 检查合并为单行, 修复 CI lint 失败。
4. 无测试配套: PR 未添加新测试, 但提供了详细的精度和性能基准数据验证。

关键文件:

- python/sglang/srt/speculative/eagle\_worker\_v2.py (模块 推理调度; 类别 source; 类型 dependency-wiring; 符号 \_capture\_cuda\_graphs): 该文件是本次变更的唯一文件, 在 HIP 的 draft\_extend CUDA graph 捕获允许列表中新增 DSA 后端支持。

关键符号: \_capture\_cuda\_graphs

## 关键源码片段

[python/sglang/srt/speculative/eagle\\_worker\\_v2.py](#)

该文件是本次变更的唯一文件，在 HIP 的 draft\_extend CUDA graph 捕获允许列表中新增 DSA 后端支持。

```
# python/sglang/srt/speculative/eagle_worker_v2.py
# _capture_cuda_graphs 方法内，HIP 平台下的 draft_extend graph 支持判断
if _is_hip:
    # 本地导入，避免非 HIP 环境加载不必要的依赖
    from sglang.srt.layers.attention.aiter_backend import AiterMultiStepDraftBackend
    from sglang.srt.layers.attention.deepseek_v4_backend_hip_radix import
    DeepseekV4HipRadixBackend
    # 新增：导入 DSA 后端，使 HIP 也能识别
    from sglang.srt.layers.attention.dsa_backend import DeepseekSparseAttnBackend

    # 允许列表：aiter 多步后端或 DSV4 radix 或 DSA 后端均可支持 graph 捕获
    supports_hip_draft_extend_graph = (
        isinstance(self.draft_attn_backend, AiterMultiStepDraftBackend)
        or isinstance(self.draft_extend_attn_backend, DeepseekV4HipRadixBackend)
        or isinstance(self.draft_extend_attn_backend, DeepseekSparseAttnBackend)
    )
```

## 评论区精华

review 讨论较少，核心争议在于：

- CI 完整性：amd-bot 指出 AMD PR CI 不完整，且 PR 变更的代码路径未被任何 PR-CI 测试覆盖，只能在 nightly 中验证。
- 无 review 评论线程，两个 reviewer（kkHuang-amd 和 HaiShaw）均直接批准。
- 暂无高价值评论线程

## 风险与影响

- 风险：
  1. 回归风险：修改了 CUDA graph 捕获的允许条件，可能导致非预期后端在 HIP 上尝试捕获 graph 而失败。但当前仅新增 DSA，且 DSA 在 CUDA 上已验证支持 graph 捕获，风险较低。
  2. 依赖风险：本地导入 DSA 后端，若 HIP 环境中 DSA 依赖（如 deep\_gemm）缺失，会在此处抛出 ImportError。但 PR 假设 HIP 构建已包含 DSA 支持。
  3. 性能回归：非 MTP 场景数据表明性能基本不变，但未能覆盖所有配置，特别是在长上下文和超高并发下。
  4. 测试覆盖缺失：无单元测试覆盖此变更，依赖 nightly/ 手动验证。
- 影响：
  1. 用户影响：AMD HIP 平台上使用 GLM-MI355 进行 MTP 推理的用户将获得显著的性能提升（Token TPUT 最高 +14.7%，TPOT 最高 -13.3%，ITL 最高 -12.8%），改善推理延迟和吞吐。
  2. 系统影响：变更仅涉及 code path 的启用，不改变现有逻辑，但对使用 DSA 的 HIP 系统，CUDA graph 捕获可能增加内存消耗（graph 捕获额外内存），需注意显存占用。

3. 团队影响：为 AMD 平台其他稀疏注意力模型（如 DeepSeek 变体）未来的 graph 优化提供了参考模式。 - 风险标记：缺少测试覆盖，HIP 特定路径

## 关联脉络

- PR #35472 NPU 测试修复：同属后处理器依赖的测试修复，但关联较弱。
- PR #34516 NPU 测试修复：同上，关联较弱。