PR #35436 完整报告

sgl-project/sglang

[docs] Add a fused-kernels page for SGLang Diffusion

合并时间：2026-08-19 16:32

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35436

执行摘要

- 一句话：新增 SGLang Diffusion 融合算子文档页
- 推荐动作：值得快速浏览：fused_kernels 页首次公开了融合算子的完整清单和双数值契约，对使用 SGLang Diffusion 的用户和内核开发者都有参考价值。该 PR 本身无代码，但作为文档 PR 的质量标杆，其“从代码树派生表格并执行验证”的做法值得借鉴。

功能与动机

在 PR#35114 的 ops/diffusion 重组之后，替代 *eager elementwise* 链的融合内核（adaLN modulate、residual gating、QK-norm、RoPE、norm epilogues、VAE layout）没有面向用户的页面。已有的 *package* README 是面向内核作者的选型矩阵，无法回答“哪些已被融合、输出是否变化、如何开启”的问题。本 PR 补充了这一页，并参照 Sol-Engine 的 kernel techniques 页面。

实现拆解

1. 新增文档页 *docs/docs/sglang-diffusion/fused_kernels.mdx*，系统记录 34 个已注册融合算子，按数值契约（bit-exact 与 request-gated）分类，并给出按算子域（normalization、adaLN、RoPE/QK-norm、activation、attention、data movement）的清单、按模型的覆盖表，以及如何查询注册表和固定后端。这是本 PR 的核心交付，为后续导航与交叉链接提供目标。
2. 在 *docs/docs/sglang-diffusion/performance-optimization.mdx* 中，向“performance lever”表格添加一行“Fused kernels”并链接到新页面，同时在文末的“Further reading”列表中加入 Fused Kernels 条目，让用户从性能优化入口能直达新文档。
3. 在 *docs/docs.json* 的 SGLang Diffusion 的 Performance Optimization 分组中注册 docs/sglang-diffusion/fused_kernels 路径，使新页面出现在侧边导航中；顺序放在 attention_backends 之后、parallelism 之前。
4. 验证工作：检查相对链接指向已有页面、docs.json 可解析、页面可从导航到达、标题遵循 docs/AGENTS.md 的 *sentence-case* 规则；由于本地 Node 版本不足，用脚本替代 mint broken-links 检查。所有内容表均从合并后的代码树派生，Python 示例在真实注册表上执行过。

关键文件：

- docs/docs/sglang-diffusion/fused_kernels.mdx（模块 文档页面；类别 docs；类型 documentation）：本次 PR 的核心交付：新增融合算子总览页，包含 34 个算子的双数值契约、按模型覆盖和启用方式
- docs/docs/sglang-diffusion/performance-optimization.mdx（模块 文档页面；类别 docs；类型 documentation）：在性能优化概述页的 lever 表和列表中添加指向新页面的链接，便于用户导航
- docs/docs.json（模块 导航配置；类别 config；类型 configuration）：在 SGLang Diffusion 的 Performance Optimization 分组中注册新页面路径，确保导航可达

关键符号：未识别

关键源码片段

docs/docs/sglang-diffusion/fused_kernels.mdx

本次 PR 的核心交付：新增融合算子总览页，包含 34 个算子的双数值契约、按模型覆盖和启用方式

```
<!-- 页面元信息：标题、描述与标签 -->
---
title: "Fused Kernels"
description: "The fused CUDA/Triton kernels SGLang Diffusion ships, what each one replaces, and which are on by default."
tag: "preserve"
---

<!-- 开头说明：为什么要融合、面向谁 -->
Diffusion transformers and VAEs spend a large share of their non-GEMM time on short elementwise chains — adaLN modulate, residual gating, QK-norm, RoPE, norm epilogues — each of which is a separate kernel launch and a separate HBM round trip in eager PyTorch. SGLang Diffusion replaces these chains with fused kernels under `sglang/kernels/ops/diffusion`.

<!-- 本文定位：清单而非调节开关 -->
This page is an inventory: what each kernel fuses, what its numerical contract is, and which models use it. It is not a lever you tune — most of these kernels are on by default and require no flag. The one switch is `--quality`, described below.

## Two numerical contracts

<!-- 核心概念：多步去噪放大逐步舍入差异，因此需要区分两种契约 -->
Multi-step denoising amplifies a per-step rounding difference into visible quality loss, so "close enough" and "bit-exact" are different products here. Every kernel in the package falls into one of two classes.

**Bit-exact — mounted unconditionally.** The kernel reproduces every rounding boundary of the eager chain, so `torch.equal` holds against the reference. Some go quite far to get there: the fused LayerNorm+modulate kernel replicates PyTorch's `vectorized_layer_norm_kernel` down to its Welford update order, guarded reciprocal, and warp-fold tree; the fused RMSNorm+scale/shift kernel replicates FlashInfer's CuTe-DSL `RMSNormKernel` fragment order and `shfl.
```

bfly` fold. Because the dispatch they replicate can change underneath them, each one still verifies itself against the live eager chain on first sight and falls back permanently on any mismatch.

Not bit-exact — request-gated. These differ from eager only at half-precision rounding-order level, but that is enough to matter, so they are mounted only for `quality="high"` requests, at batch boundaries, all-or-nothing per transformer. The default `quality="lossless"` runs the unmodified reference chain.

评论区精华

本 PR 没有公开的 review 评论或讨论线程。作者在 PR body 中说明：所有表格行均直接从合并后的代码树派生（注册表 _SPECS 有 34 个算子 id 对应 38 个实现），Python 示例已在真实注册表上执行；由于本地 Node 为 18，低于 Mintlify 所需的 20.17，mint broken-links 被等效脚本替代。

- 暂无高价值评论线程

风险与影响

- 风险：纯文档变更，无源代码或运行时影响。主要风险是文档内容随代码演化而过时：34 个算子清单、后端能力集来自 ops/diffusion/__init__.py 的 _SPECS，模型覆盖来自 multimodal_gen/runtime/models 的 import 位置，--quality 与 OpenAI quality 字段来自 SamplingParams.quality / _runtime_sampling_quality，这些引用点一旦变化，文档需要同步更新。另外，docs.json 中的路径若与页面不一致会导致导航断裂，但作者已通过脚本验证。无性能、安全风险。
- 影响：对用户：提供融合算子的透明清单和启用指南，帮助判断哪些 elementwise 链已被融合、输出是否变化以及如何通过 --quality high 开启请求门控的融合。对系统：无运行时影响。对团队：减少同类咨询，为后续融合算子开发提供索引基础；文档内容可作为未来 kernel 文档的模板。
- 风险标记：文档准确性，导航链接完整性

关联脉络

- PR #34680 [diffusion][Minimax H3]support subblock sparse attention on SM90: 涉及 diffusion 融合内核的算子演进，与本 PR 文档中的算子清单直接相关
- PR #34993 [diffusion] fix: make MiniMax-H3 AdaLN cache rebuild transactional: 涉及 adaLN modulate 相关融合逻辑，本 PR 文档记录了同类融合算子