PR #36020 完整报告

sgl-project/sglang

[docs] Split the Qwen3.8-27B NVFP4 cells by lm_head precision

合并时间：2026-08-25 01:50

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/36020

Qwen3.8-27B NVFP4 文档按 lm_head 精度拆分

执行摘要

本 PR 将 Qwen3.8-27B cookbook 部署面板中单一的 NVFP4 量化选项拆分为 NVFP4-BF16-Head 与 NVFP4-FP4-Head 两个变体，对应 RadixArk 发布的两份独立检查点（差异仅在 lm_head 是否打包为 FP4）。FP4-Head 单元格整体继承 BF16-Head 的实测参数，唯独 RTX 5090 的 fp32 单元格单独实测并解锁 3 项。改动全部落在文档站点配置（4 个文件），无运行时影响。

功能与动机

PR body 说明得很直接：NVFP4 导出不是"新版本替换旧版本"，而是 同时发布两份检查点——RadixArk/Qwen3.8-27B-NVFP4 把 lm_head 打包成 FP4，RadixArk/Qwen3.8-27B-NVFP4-BF16-LMHead 则保留密集 BF16 head。原先页面只有一个 NVFP4 选项，无法表达这种差异，因此拆成两个。

动机里还包含一个可复用的论证：两份导出包只在 lm_head 不同，密集 BF16 head 在磁盘上大约多 1.7 GB、运行时大约多 3.2 GB，因此"严格更难适配显存"。任何能把 BF16-Head 服务起来的参数组合，对 FP4-Head 只会更宽松——所以继承是"保守方向"（the conservative direction）。

实现拆解

1. 变更入口：docs/src/snippets/configs/Qwen/qwen3.8-27b.jsx 的 matchDims.quant 维度。nvfp4 选项被替换为 nvfp4-bf16-head 与 nvfp4-fp4-head 两个 id，每个覆盖 RTX PRO 6000、RTX 5090、DGX Spark、GB300 四类硬件，共 8 个 NVFP4 单元格；H200（SM90 无 FP4 tensor core）两个变体都保持置灰。
2. 可用性判断改前缀匹配：EAGLE、DSPARK、DFlash2 三个 overlay 在 RTX 5090 上的 disabled 条件从 sel.quant !== "nvfp4" 改为 !String(sel.quant).startsWith("nvfp4")，保证两个 NVFP4 变体共享同一套"32 GB 下草案模型只能叠加在 NVFP4 权重上"的约束。这是全 PR 最关键的一处机械改动——原等值判断会让新 id 全部失去匹配。
3. RTX 5090 fp32 单独实测：BF16-Head 导出包下 fp32 被置灰（密集 head 需约 3.2 GB，fp32 状态池放不下）；FP4-Head 导出包释放余量后，DSpark 两个单元格在 --mem-fraction-static 0.89、balanced ratio 6.63/6.12 下可服务，DFlash2 high-throughput 在 0.895 + --mamba-full-memory-ratio 10 下可服务，low-latency 仍置灰（0.8975/r14 下 KV 余量只有 7,752 / 9,216 tokens）。DSpark 的 mem-fraction 因

此按 ssmDtype 分支：fp32 用 0.89、bf16 用 0.88。

4. 配套同步：qwen3.8-27b-benchmarks.jsx 把 GB300 两条 NVFP4 benchmark 行的 quant 匹配键改为 nvfp4-fp4-head；_qwen38_mamba_ratio_calculator.jsx 把默认 quant 状态改为 nvfp4-bf16-head、KV dtype 回退判断改为前缀匹配；Qwen3.8-27B.mdx 检查点表格新增 BF16-Head 行并补充两变体差异说明。
5. 测试与 CI：无直接测试文件（文档变更）；依赖 Mintlify 预览与 check_cookbook_configs.mjs 对 dim.options 的结构校验。

docs/src/snippets/configs/Qwen/qwen3.8-27b.jsx

部署面板的核心配置：quant 维度拆分、全部 overlay 可用性判断与 RTX 5090 fp32 实测参数都集中在这个文件，是全 PR 的主改动（+162/-17）。

```
// 以下为 qwen3.8-27b.jsx 中拆分后量化维度与 RTX 5090 overlay 逻辑的整理片段，
// 省略了硬件与节点等无关维度。
// 量化维度把单一 NVFP4 选项拆成两个变体：两份导出包只在 lm_head 上不同，
// BF16-Head 保留密集 head（磁盘大约多 1.7 GB、运行时大约多 3.2 GB），
// FP4-Head 把 head 打包为 FP4。BF16-Head 严格更难适配显存，所以 FP4-Head
// 单元格直接复用 BF16-Head 配方，属保守继承方向。
{ id: "quant", title: "Quantization", options: [
  { id: "bf16", label: "BF16" },
  { id: "fp8", label: "FP8" },
  { id: "nvfp4-bf16-head", label: "NVFP4-BF16-Head" },
  { id: "nvfp4-fp4-head", label: "NVFP4-FP4-Head" },
] }

// EAGLE / DSPARK / DFlash2 在 RTX 5090 的可用性判断从等值比较改为前缀匹配，
// 让两个 NVFP4 变体共享同一约束：32 GB 显存下草案模型只能叠加在 NVFP4
// 权重上。前缀匹配是这次拆分能成立的关键——旧写法会漏掉新 id。
disabled: (sel) =>
  sel.hw === "rtx5090" && !String(sel.quant).startsWith("nvfp4"),
disableReason:
  "On the 32GB RTX 5090 the DSpark draft model only fits on top of the NVFP4 weights",

// RTX 5090 的 mem-fraction-static 是唯一没有直接继承的单元格：FP4-Head
// 导出包实测 fp32 在 0.89 时可服务（balanced ratio 下 pool 25,911 / K=6、
// 29,490 / K=5），BF16-Head 导出包下 fp32 则被 SSM dtype 行置灰。
// 因此按 ssmDtype 分支钉住 0.89 / 0.88 两个值。
...(sel.hw === "rtx5090"
  ? [sel.ssmDtype === "float32"
      ? "--mem-fraction-static 0.89"
      : "--mem-fraction-static 0.88"]
  : []),
```

docs/src/snippets/_qwen38_mamba_ratio_calculator.jsx

ratio 计算器的 KV dtype 回退判断必须跟着新 id 走；这里改成前缀匹配，否则选择 FP4-Head 时计算器会把 KV 池误判为 bf16，导致 ratio 建议错误。

```
// 以下为 _qwen38_mamba_ratio_calculator.jsx 中 KV dtype 推导的整理片段。
```

```
// 依据当前选中的量化选项推导默认 KV 精度：两个 NVFP4 检查点都在配置里声明
// kv_cache_quant_algo: FP8，所以默认 --kv-cache-dtype auto 会落到 fp8_e4m3；
// BF16 / FP8 检查点则保持 bf16 KV 池。显式传入的 --kv-cache-dtype 标志始终
// 优先于检查点声明，因此先检查标志，再回退到 quant 前缀判断。
const kvDtype =
  kvFlag === "fp8_e4m3"
    ? "fp8_e4m3"
    : kvFlag === "bfloat16" || kvFlag === "bf16"
      ? "bfloat16"
      : String(quant).startsWith("nvfp4")
        ? "fp8_e4m3"
        : "bfloat16";
```

评论区精华

本 PR 没有实质 review 讨论：zijiexia 直接批准（评审意见为空），唯一评论是 mintlify bot 的预览部署通知。技术论证全部沉淀在 PR body 里：

- 批量继承 + 定点补测：两份导出包只在 lm_head 不同，BF16-Head 严格更难适配，因此 FP4-Head 单元格继承 BF16-Head 参数是“保守方向”，能省掉 8 个单元格里 7 个的重复测量。
- 唯一例外被单独实测：RTX 5090 fp32 单元格若继承会“低估”（BF16-Head 下实测 fp32 不可用，但 FP4-Head 下可用），所以作者直接在这份导出包上逐格测量，给出 pool 25,911 / K=6 / 10.15 ms、ratio 6.12 / pool 29,490 / K=5 等完整数据，并说明测量复现了早前一轮独立结果。

风险与影响

风险集中在配置键迁移与数据溯源：

- quant id 是页面级匹配键：nvfp4 改为两个前缀 id 后，所有 disabled 判断、benchmark 行、ratio 计算器都要同步；本 PR 覆盖了 4 个文件，但若站内其它页面或脚本仍有 "nvfp4" 硬编码，会静默失去匹配，建议合入前全站 grep。
- GB300 benchmark 数据溯源存疑：两条行被重新键到 nvfp4-fp4-head，但原始备注只写 NVFP4 = RadixArk W4A4-0811 (private)，没有说明测量时用的是哪份导出包；若实测跑在 BF16-Head 上，归属就不准确。
- RTX 5090 fp32 是紧边界实测值：DFlash2 low-latency 在 0.8975/r14 下的 KV 余量只有 7,752 / 9,216 tokens，SGLang 后续显存布局变化后这些 0.89 / 0.895 / ratio 6.63 参数很可能失效，需要随版本滚动复测。
- 不涉及运行时代码，回归面限定在文档站点。

关联脉络

- 本 PR 直接构建在 #35825之上（PR body 明确写到 Builds on #35825 (merged)），后者提供了 RTX 5090 / RTX PRO 6000 / DGX Spark 的 BF16-Head 参数网格，本 PR 的所有继承参数都来自它。

- 与 #35957（修复 decode retraction 的 recurrent state 丢失）属同一功能线：Qwen3.8-27B 是 Mamba 混合架构模型，本页的 ratio 计算器直接镜像 kv_cache_configurator._calculate_mamba_ratio，运行时状态池大小与文档参数相互印证。
- 与 #36204一类"按量化与硬件验证状态维护文档"的实践同脉络：文档参数以实测为准、标注验证状态（GB300 保持 in-progress），把测量 provenance 写进注释。