

PR #38611 完整报告

sgl-project/sglang

[docs] Add the NVIDIA NVFP4 export to the Qwen3.8-27B cookbook

合并时间: 2026-09-12 02:58

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/38611>

执行摘要

本 PR 为 Qwen3.8-27B cookbook 接入 NVIDIA 官方 ModelOpt NVFP4 导出 (nvidia/Qwen3.8-27B-NVFP4) : 新增 4 个启动 cell 并逐字复用 RadixArk FP4-head 配方; 顺带修复了 RTX 5090 上 DSPARK / DFLASH2 / MTP 行无法启动的内存池问题, 修正了 Mamba ratio 计算器对 auto KV 精度的前缀误判。全部 202 个 cell 在 v0.5.19 上用完整 GSM8K (1319 题) 逐个实测, 准确率区间 93.18%-95.15%。纯文档变更, 对运行时零影响, 但文档内嵌的配置逻辑复杂度和维护成本明显上升。

功能与动机

PR body 的 Motivation 段写明: NVIDIA 官方发布了 ModelOpt NVFP4 导出 nvidia/Qwen3.8-27B-NVFP4, 而 cookbook 页面已内置两个 RadixArk NVFP4 checkpoint, 导致拉取 NVIDIA 导出的读者没有可复制的启动命令 cell。作者通过逐字节对比两个导出的 config.json / hf_quant_config.json / safetensors index 证明它们是同一 W4A4 权重体 (401 层量化、2,194 张量、21.92 GB、FP4 打包 lm_head), 因此可以逐字复用现有配方而非另设操作点。

实现拆解

1. 新增量化选项与 cell: docs/src/snippets/configs/Qwen/qwen3.8-27b.jsx 的 matchDims.quant 中新增 nvfp4-nvidia, 4 个新 cell (RTX PRO 6000、RTX 5090、DGX Spark、GB300) 的 flags 与 nvfp4-fp4-head 对应 cell 逐字相同。overlay 的禁用逻辑基于 startsWith("nvfp4"), 新选项自动继承 32 GB 卡上的可用性包络 (spec 算法开放、fp32 GDN 状态开放)。
2. 修复 RTX 5090 内存池自洽: 实测发现 DSPARK 行原配方无法 boot——KV 池按 127,332 token 自举而单请求只需 9,216 token, draft 模型权重计入静态内存后 GDN 状态池拆分不足。DSPARK 行扩展 stripPrefixes 并新增 --max-total-tokens 16384 与按 quant x ssmDtype x tier 区分的 mem-fraction / chunk / ratio 固定值; DFLASH2 与 dense-lm_head 的 MTP fp32 组合也补上同类 pin。
3. 修正计算器 KV 精度推断: _qwen38_mamba_ratio_calculator.jsx 的 kvDtype 从 startsWith("nvfp4") 前缀测试改为显式匹配两个 RadixArk id, 避免 NVIDIA 导出 (无 kv_cache_scheme, auto 落 bf16) 被少算一半 kv_bytes_per_token; 同时删除 "fp8 KV 池携带校准 scale" 的错误陈述。
4. 文档与版本基线对齐: Qwen3.8-27B.mdx 新增 checkpoint 表格行, 部署说明升级为 202 cell 全量 GSM8K 实测口径; 安装命令经历 pin v0.5.19 → unpin 到 latest 的演进, 并同

步修正 DSPARK 0.89→0.91 等陈旧声明。

5. 验证配套：无自动化测试文件，依赖 check_cookbook_configs.mjs 结构校验 + 202 cell 全量 GSM8K 人工实测（5 checkpoint × 4 spec × 2 tier × 2 GDN dtype）。

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

本 PR 核心：新增 nvfp4-nvidia 量化选项与 4 个 cell，并重构 RTX 5090 上 EAGLE / DSPARK / DFLASH2 三行的 stripPrefixes 与 flags 逻辑，修复原配方无法启动的问题。

// matchDims 中的量化选项：新增 NVIDIA 官方导出，复用 FP4-head 配方。

```
{ 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" },
  // NVIDIA 的 ModelOpt 导出与 RadixArk FP4-head 是同一 W4A4 权重体：
  // 401 层量化、同样的张量集合、21.9 GB 磁盘占用、同样 FP4 打包的
  // lm_head，所以每条 cell 都逐字复用 FP4-head 配方的参数。
  // 唯一差异是它不声明 kv_cache_scheme, `--kv-cache-dtype auto`
  // 在这里会落到 bf16 而不是 fp8_e4m3；而页面所有 cell 都显式
  // pin 了 fp8_e4m3，所以最终启动命令与 KV 池完全一致。
  { id: "nvfp4-nvidia", label: "NVFP4-NVIDIA" },
] },
```

// DSPARK overlay：在 32 GB RTX 5090 上由本行完全接管内存池。

// 原先只 pin --max-running-requests 1，KV 池却按 127,332 token

// 自举（单请求 8192 in / 1024 out 只需 9,216），draft 模型

// 3.64 GB 计入 --mem-fraction-static 后 GDN 状态池又默认拆分不足，

// 启动直接报 max_mamba_cache_size=0 ... max_num_reqs=0。

stripPrefixes: (sel) =>

sel.hw === "rtx5090"

? ["--mem-fraction-static", "--mamba-full-memory-ratio",
"--chunked-prefill-size", "--max-total-tokens"]

: [],

flags: (sel) => [

--speculative-algorithm DSPARK",

--speculative-draft-model-path RadixArk/Qwen3.8-27B-DSpark",

// 以下取值都在 v0.5.19 上以 ISL 8192 / OSL 1024、并发 1 实测。

// 两个 FP4-head 导出共用一组 pin；dense-lm_head 导出因运行时

// 多占约 3.2 GB，需要自己的 mem-fraction 与 chunk 组合。

...(sel.hw === "rtx5090"

? [

--max-total-tokens 16384",

...(sel.quant === "nvfp4-bf16-head"

? ["--mem-fraction-static 0.92",

--chunked-prefill-size 512",

sel.tier === "low-latency"

? "--mamba-full-memory-ratio 3.38"

: "--mamba-full-memory-ratio 2.71"]

```

: sel.ssmDtype === "float32"
  ? ["--mem-fraction-static 0.91",
    "--chunked-prefill-size 1024",
    sel.tier === "low-latency"
      ? "--mamba-full-memory-ratio 6.94"
      : "--mamba-full-memory-ratio 5.56"]
  : ["--mem-fraction-static 0.88",
    sel.tier === "low-latency"
      ? "--mamba-full-memory-ratio 3.38"
      : "--mamba-full-memory-ratio 3.12"]],
]
: []),
],

```

docs/src/snippets/_qwen38_mamba_ratio_calculator.jsx

修复 KV dtype 推断的前缀误判，避免新导出下计算器少算一半 kv_bytes_per_token；是整个 PR 中最具方法论意义的一处小改动。

```

// 推断 --kv-cache-dtype 未显式指定时 (auto) 实际落到的 KV 精度。
// 两个 RadixArk NVFP4 导出声明了 kv_cache_quant_algo: FP8, auto
// 会落到 fp8_e4m3; BF16 / FP8 以及 NVIDIA 导出都没有声明 KV 方案,
// auto 会留在 bf16。这里不能沿用 String(quant).startsWith("nvfp4")
// 前缀测试——NVIDIA 导出的 id 是 nvfp4-nvidia, 前缀相同但行为不同,
// 前缀测试会把它的 kv_bytes_per_token 少算一半。
// 显式 flag 永远优先于 checkpoint 声明。
const kvFlag = flagArg("--kv-cache-dtype");
const kvDtype =
  kvFlag === "fp8_e4m3"
    ? "fp8_e4m3"
    : kvFlag === "bfloat16" || kvFlag === "bf16"
      ? "bfloat16"
      : quant === "nvfp4-bf16-head" || quant === "nvfp4-fp4-head"
        ? "fp8_e4m3"
        : "bfloat16";

```

评论区精华

本 PR 没有实质性的 review 交锋：唯一评论是 mintlify bot 的 preview 部署通知，最终由 wisclmy0611 APPROVED（无评语）。值得注意的权衡都沉淀在 PR body 与 config 注释中：

Same footprint and same quantization layout, so the mem-fraction pins, the 32 GB RTX 5090 fit and the DGX Spark earlyoom headroom all hold.

The Mamba ratio calculator inferred the auto KV precision from a `String(quant).startsWith("nvfp4")` test, which would have reported half the real kv_bytes_per_token for this checkpoint.

风险与影响

- 计算器逻辑无测试保护：_qwen38_mamba_ratio_calculator.jsx 的 kvDtype 改为显式 id 枚举后仍无自动化测试，未来新增第三个 NVFP4 导出若忘记同步，会再次静默算错内存比例。
- 配置复杂度上升：DSPARK 的 pin 已按 quant × ssmDtype × tier 三轴嵌套出 12 种取法，stripPrefixes 与 flags 必须成对维护，对后续维护者要求高。
- 版本基线漂移：安装命令最终 unpin 到 latest，而全部测量数据基于 v0.5.19；新版本若改变内存布局，页面 pin 可能失效且无人感知。
- 行为语义差异：NVIDIA 导出在 KV Cache Precision 切到 Auto 时 KV 池跑 bf16，与其它 NVFP4 cell 的 fp8_e4m3 不同，文档已提示但用户仍可能困惑。

影响范围：仅限 Qwen3.8-27B cookbook 读者；对运行时系统零影响。

关联脉络

历史 PR 中无直接改动同批文件的记录。本 PR 的关联更多体现在对运行时源码的引用锚点：

`ModelOptMixedPrecisionConfig.from_config` (modelopt_quant.py)、

`configure_kv_cache_dtype` (kv_cache_dtype.py)、

`BaseKVCacheMethod.process_weights_after_loading` (kv_cache.py) ——文档的 dtype 推断逻辑正是对这些函数行为的复刻，但本 PR 未改动它们。这也提示：cookbook 配置正在成为运行时行为的一份 "可执行镜像"，其准确性依赖文档侧对引擎语义的持续跟踪。