

PR #2855 完整报告

radixark/miles

Restrict NVFP4 RL to routed experts

合并时间: 2026-09-11 08:59

原文链接: <http://prhub.com.cn/radixark/miles/pull/2855>

执行摘要

- 一句话: NVFP4 量化范围收紧至路由专家, 共享专家保持高精度
- 推荐动作: 值得精读, 重点不在 diff 体量 (+25/-51), 而在于它把一条隐式契约变成默认行为: 建议仔细阅读 `_is_routed_expert_weight_name` 的判定顺序 (共享专家否决必须早于 `.experts.` 匹配)、`quantize_params_nvfp4` 的分发结构, 以及三条路径的对齐关系, 这是理解 miles NVFP4 RL 现状的关键。

需要额外关注的决策与后续动作: 一是 `--ci-disable-kl-checker` 属于「绕过而非修复」, 建议单独跟进 signed `train/ppo_kl < 1e-8` 门禁的判据缺陷 (应改为绝对值或幅值阈值); 二是 `tests/fast/test_nvfp4_scope.py` 被删除后没有替代守卫, 建议补回一个纯 CPU 的名称判定单测, 锁住单数 / 复数 / 嵌套共享专家三种命名; 三是把 FlashInfer 0.6.18 固化到镜像 / 依赖, 避免 E2E 长期依赖手工 wheel 覆盖; 四是给用户一条明确的迁移提示——旧 NVFP4 checkpoint 需要重新转换导出。

功能与动机

PR body 给出的动机是让 NVFP4 RL 的目标范围收敛为「只量化路由 MoE 专家」: `make NVFP4 RL target routed MoE experts only. Shared experts now preserve their source/high precision by default across offline HF conversion, Megatron-to-HF checkpoint export, and live weight updates.`。改动前三条路径并不一致——训练侧 TE matcher 早已是 routed-only, 而离线转换脚本的 `EXPERT_NAME_MARKERS` 里含 `.shared_experts.`、导出侧 `quantize_params_nvfp4` 还有独立的 `mlp.shared_experts` 分支, 导致共享专家在转换 / 导出时被量化为 NVFP4。PR 明确把共享专家定位为默认高精度, 并把 `--extra-high-precision-layers-*` 降级为「recipe 默认范围之外的额外例外」。关联 Issue #4486 记录了 FlashInfer `flashinfer_trtllm_routed` 的 NVFP4 MoE 在 Blackwell (SM100/SM103) 上的 NaN 回归, 其复现环境同样设置 `SGLANG_FP4_IGNORED_LAYERS=shared_expert,...`, 与「共享专家保持高精度」的社区实践一致; 本 PR 的 GLM-5.2 E2E 正好落在该路径上, 因此验证时手动覆盖到修复版 FlashInfer 0.6.18。

实现拆解

本 PR 的本质是把「NVFP4 只作用于路由专家」从训练侧的 matcher 约定, 升级为离线转换、训练、在线导出三条路径的统一默认值, 改动入口是名称判定函数与导出分发逻辑。

1. HF→NVFP4 离线转换范围收口 (`tools/convert_hf_to_nvfp4.py`, +13/-6)

- 把原 EXPERT_NAME_MARKERS (含 .experts.、.shared_experts.、block_sparse_moe.experts.、.moe.experts.) 拆成 ROUTED_EXPERT_NAME_MARKERS 与新增的 SHARED_EXPERT_NAME_MARKERS (.shared_expert. 单数、.shared_experts. 复数)。
- _is_moe_expert_weight_name 重命名为 _is_routed_expert_weight_name, 判定顺序变为: 先要求 .weight 结尾, 再在通用 .experts. 匹配之前用 SHARED_EXPERT_NAME_MARKERS 一票否决, 最后才走路由专家标记与 EXPERT_WEIGHT_SUFFIXES 后缀匹配。这个顺序是本次修复的关键——嵌套在共享专家路径内部的 experts 子串不会再被误判为路由专家。
- should_quantize 调用点同步改名; 用户传入的 --extra-high-precision-layers-hf (skip_weight_substrings) 依旧优先级最高。
- 模块 docstring 从 for MoE expert GEMMs only 改为 for MoE routed-expert GEMMs only. Shared experts and dense linear layers are left unmodified., 把语义写进工具契约。

2. Megatron→HF 导出与在线权重更新范围收口 ([miles/backends/megatron_utils/megatron_to_hf/processors/quantizer_nvfp4.py](#), +1/-12)

- quantize_params_nvfp4 删除 shared_expert_pattern = r"mlp.shared_experts\.(.+) " 整段分支, 只保留 mlp.experts\.(.+)\.weight(\d+) 且 rest ∈ {linear_fc1, linear_fc2} 时调用 _quantize_moe_params。
- 前置换名短路逻辑 (--extra-high-precision-layers-megatron、--first-last-layers-bf16)、_quantize_moe_params 的 gate/up 成对共享 global amax 逻辑均未改动, 注释由 # experts 改为 # routed experts。
- 影响面: checkpoint 导出与 live weight update 不再产出 NVFP4 共享专家张量, 与训练侧 matcher 对齐; 这是「共享专家默认高精度」在导出侧的落地点。

3. 测试配套收敛 ([tests/e2e/megatron/test_glm5_2_744b_a40b_5layer_nvfp4.py](#), +3/-28)

- 删除 EXTRA_HIGH_PRECISION_LAYERS_HF、EXTRA_HIGH_PRECISION_LAYERS_MEGATRON 常量与 _extra_high_precision_layers_hf_args、_extra_high_precision_layers_megatron_args 两个辅助函数, prepare() 与 mixed_precision_args 不再拼接这两个参数——因为排除共享专家已成为默认行为, 原参数是冗余 workaround。
- TE_PRECISION_CONFIG 删除 shared_experts_fc1_bf16 / shared_experts_fc2_bf16 matcher, 只留 routed_experts_fc1_nvfp4、routed_experts_fc2_nvfp4 与 default_bf16 兜底。
- ci_args 新增 --ci-disable-kl-checker (+1 行), 依据是 signed train/ppo_kl < 1e-8 断言在未改动基线 2799fe386 上同样失败, 属于既有问题。

4. 文档同步 ([docs/advanced/low-precision.md](#), +8/-5)

- 明确写入 NVFP4 applies only to the routed-expert FC1 and FC2 GEMMs; shared experts and all other unmatched tensors stay in BF16 across checkpoint conversion, training, and live weight export.

- 把 `--extra-high-precision-layers-*` 的定位从「常见例外：共享专家」改为「超出 recipe 默认范围的额外例外」，并说明 NVFP4 已自动排除共享专家。

5. 兼容性保留与非目标

- 首尾层 (`--first-last-layers-bf16`)、MLA 投影等模型特例仍可经两个 `--extra-high-precision-layers-*` 开关处理；NVFP4 量化数学、scale 布局、gate/up 配对 amax、MXFP8 与 dense 张量均未改动 (PR 列为 non-goals)。

契约路径	变更前	变更后	关键位置
离线 HF→NVFP4 转换	路由专家 + 共享专家都被量化	仅路由专家，共享专家显式排除	<code>_is_routed_expert_weight_name</code>
Megatron 训练 (TE matcher)	已是 routed-only，但 E2E 额外挂 shared expert bf16 matcher	保留 routed-only，删除冗余 matcher	<code>TE_PRECISION_CONFIG</code>
Megatron→HF 导出 / live update	路由专家 + 共享专家	仅路由专家	<code>quantize_params_nvfp4</code>

关键文件：

- `tools/convert_hf_to_nvfp4.py` (模块 权重转换；类别 source；类型 core-logic；符号 `EXPERT_NAME_MARKERS`, `ROUTED_EXPERT_NAME_MARKERS`, `SHARED_EXPERT_NAME_MARKERS`, `_is_moe_expert_weight_name`)：NVFP4 离线转换的范围判定入口。新增 `SHARED_EXPERT_NAME_MARKERS` 并把 `_is_moe_expert_weight_name` 重命名为 `_is_routed_expert_weight_name`，在通用 `experts`. 匹配之前显式否决单数 / 复数 / 嵌套共享专家名，是「共享专家默认保持高精度」的第一处落地点。
- `miles/backends/megatron_utils/megatron_to_hf/processors/quantizer_nvfp4.py` (模块 权重导出；类别 source；类型 core-logic；符号 `quantize_params_nvfp4`, `shared_expert_pattern`)：Megatron→HF checkpoint 导出与在线权重更新路径的量化分发点。`quantize_params_nvfp4` 删除了 `mlp.shared_experts.linear_fc1/2` 的独立分支，使导出产物与训练侧 routed-only matcher 对齐，直接影响 SGLang 侧加载的权重集合。
- `tests/e2e/megatron/test_glm5_2_744b_a40b_5layer_nvfp4.py` (模块 端到端测试；类别 test；类型 test-coverage；符号 `TE_PRECISION_CONFIG`, `EXTRA_HIGH_PRECISION_LAYERS_HF`, `EXTRA_HIGH_PRECISION_LAYERS_MEGATRON`, `_extra_high_precision_layers_hf_args`)：覆盖 NVFP4 RL 全链路的 8 卡 B200 E2E。删除冗余的 shared expert matcher 与 `--extra-high-precision-layers-*` 参数，并新增 `--ci-disable-kl-checker`；后者是本 PR 中最需要审慎对待的 CI 语义变化。
- `docs/advanced/low-precision.md` (模块 文档；类别 docs；类型 documentation)：低精度训练指南。把「NVFP4 只覆盖 routed-expert FC1/FC2、共享专家与其他未匹配张量保持 BF16」写入正文，并把 `--extra-high-precision-layers-*` 的定位从「常见例外」改为「默认范围之外的额外例外」，是用户理解新契约的唯一文档入口。

关键符号: `_is_routed_expert_weight_name`, `should_quantize`, `quantize_params_nvfp4`, `_quantize_moe_params`, `_extra_high_precision_layers_hf_args`, `_extra_high_precision_layers_megatron_args`

关键源码片段

tools/convert_hf_to_nvfp4.py

NVFP4 离线转换的范围判定入口。新增 `SHARED_EXPERT_NAME_MARKERS` 并把 `_is_moe_expert_weight_name` 重命名为 `_is_routed_expert_weight_name`，在通用 `.experts`。匹配之前显式否决单数 / 复数 / 嵌套共享专家名，是「共享专家默认保持高精度」的第一处落地点。

```
# tools/convert_hf_to_nvfp4.py (按 PR 变更整理后的实现片段)
# 把「路由专家」与「共享专家」的名称标记彻底分开:
# 前者参与 NVFP4 量化, 后者一律保持 BF16。
ROUTED_EXPERT_NAME_MARKERS = (
    ".experts.", # DeepSeek / GLM 等主流 MoE 布局
    "block_sparse_moe.experts.", # Mixtral 风格布局
    ".moe.experts.", # 部分 Qwen-MoE 变体
)

# 共享专家标记同时覆盖单数 (.shared_expert.) 与复数 (.shared_experts.) 两种命名;
# 只要名字里出现其中任意一个, 即使内部还嵌套了 ".experts." 子串也一律排除。
SHARED_EXPERT_NAME_MARKERS = (
    ".shared_expert.",
    ".shared_experts.",
)

def _is_routed_expert_weight_name(name: str) -> bool:
    """判定一个 HF 权重名是否属于「路由专家」, 即是否允许被 NVFP4 量化。"""
    # 只处理权重张量本身, scale / bias 等附属张量直接跳过
    if not name.endswith(".weight"):
        return False
    # 关键顺序: 共享专家的否决必须发生在通用 ".experts." 匹配之前,
    # 否则嵌套在共享专家路径里的 experts 子串会被误判成路由专家
    if any(marker in name for marker in SHARED_EXPERT_NAME_MARKERS):
        return False
    if not any(marker in name for marker in ROUTED_EXPERT_NAME_MARKERS):
        return False
    # 仅接受 expert GEMM 权重后缀 (w1/w2/w3、gate/up/down 等)
    return any(name.endswith(suffix) for suffix in EXPERT_WEIGHT_SUFFIXES)

def should_quantize(
    name: str,
    weight: torch.Tensor,
    skip_weight_substrings: tuple[str, ...] = (),
```

```

) -> bool:
    # 用户显式指定的额外高精度子串（如 MLA 投影、首尾层）优先级最高
    if any(substr in name for substr in skip_weight_substrings):
        return False
    # 由 _is_moe_expert_weight_name 改为 routed-only 判定:
    # 共享专家从这里开始就退出 NVFP4 路径
    if not _is_routed_expert_weight_name(name):
        return False
    if weight.dtype not in (torch.float16, torch.bfloat16, torch.float32):
        return False
    if weight.dim() < 2:
        return False
    # NVFP4 采用 1D block scaling, 要求最后一维能被 group size 16 整除
    if weight.shape[-1] % NVFP4_GROUP_SIZE != 0:
        raise ValueError(
            f"Last dim {weight.shape[-1]} must be divisible by {NVFP4_GROUP_SIZE} "
            f"for NVFP4 quantization ({name})."
        )
    return True

```

[miles/backends/megatron_utils/megatron_to_hf/processors/quantizer_nvfp4.py](#)

Megatron→HF checkpoint 导出与在线权重更新路径的量化分发点。

[quantize_params_nvfp4](#) 删除了 [mlp.shared_experts.linear_fc1/2](#) 的独立分支, 使导出产物与训练侧 routed-only matcher 对齐, 直接影响 SGLang 侧加载的权重集合。

```

# miles/backends/megatron_utils/megatron_to_hf/processors/quantizer_nvfp4.py
# 以下为按 PR head 版本整理的 quantize_params_nvfp4 实现,
# 省略了与本变更无关的中间细节。
def quantize_params_nvfp4(args, megatron_name, converted_named_params, quantization_
config):
    # 显式高精度名单（首尾层、MLA 投影等）仍然最高优先级:
    # 命中即整批原样返回, 不做任何 NVFP4 量化
    if getattr(args, "extra_high_precision_layers_megatron", False):
        for layer_name in getattr(args, "extra_high_precision_layers_megatron", ()):
            if layer_name in megatron_name:
                return converted_named_params

    ignore_rules = _get_ignore_rules(quantization_config)

    # 先剥掉 decoder.layers.<idx>. / mtp.layers.<idx>. 前缀, 得到层内相对名字 rest
    decoder_layers_pattern = r"decoder\.layers\.(\d+)\.(\.+)"
    match = re.search(decoder_layers_pattern, megatron_name)
    if not match:
        # MTP 层走同样的处理, 只是前缀不同
        mtp_layer_pattern = r"mtp\.layers\.(\d+)\.(\.+)"
        match = re.search(mtp_layer_pattern, megatron_name)
    if not match:
        return converted_named_params

```

```

        layer_idx, rest = match.groups()
        rest = rest.replace("transformer_layer.", "").replace("mtp_model_layer.", "")
    else:
        layer_idx, rest = match.groups()

# --first-last-layers-bf16: 首尾若干层整体跳过 NVFP4, 保持 BF16
if getattr(args, "first_last_layers_bf16", False):
    num_layers = int(args.num_layers)
    num_layers_at_start_in_bf16 = int(getattr(args, "num_layers_at_start_in_bf16", 0))
    num_layers_at_end_in_bf16 = int(getattr(args, "num_layers_at_end_in_bf16", 0))
    head_end_idx = num_layers_at_start_in_bf16
    tail_start_idx = num_layers - num_layers_at_end_in_bf16
    if int(layer_idx) < head_end_idx or int(layer_idx) >= tail_start_idx:
        return converted_named_params

# routed experts: 只有 mlp.experts.linear_fc1 / linear_fc2 进入 NVFP4,
# 且由 _quantize_moe_params 保证 gate/up 成对量化、共享同一个 global amax
expert_pattern = r"mlp.experts\.(.+)\.weight(\d+)"
match = re.match(expert_pattern, rest)
if match:
    rest, _ = match.groups()
    if rest in [
        "linear_fc1",
        "linear_fc2",
    ]:
        return _quantize_moe_params(converted_named_params, ignore_rules)

# base 版本在此处还有一段 mlp.shared_experts.linear_fc1/2 的分支, 本 PR 删除:
# 共享专家从此走原样返回路径保留源精度, 与训练 matcher、离线转换保持一致
return converted_named_params

```

tests/e2e/megatron/test_glm5_2_744b_a40b_5layer_nvfp4.py

覆盖 NVFP4 RL 全链路的 8 卡 B200 E2E。删除冗余的 shared expert matcher 与 `--extra-high-precision-layers-*` 参数, 并新增 `--ci-disable-kl-checker`; 后者是本 PR 中最需要审慎对待的 CI 语义变化。

tests/e2e/megatron/test_glm5_2_744b_a40b_5layer_nvfp4.py (head 版本节选)

训练侧精度配置: 只显式声明路由专家, 共享专家的 bf16 matcher 已被删除,

交给 default_bf16 兜底, 等价于「NVFP4 默认不覆盖共享专家」。

TE_PRECISION_CONFIG = """

configs:

nvfp4:

transformer_engine_config_type: "TEQuantizationParams"

training_recipe:

fp4_quantization_recipe: "nvfp4"

bf16:

transformer_engine_config_type: "TEQuantizationParams"

training_recipe: {}

matchers:

```

routed_experts_fc1_nvfp4:
  type: "glob"
  enabled: true
  pattern: "*.mlp.experts.linear_fc1"
  config: "nvfp4"
routed_experts_fc2_nvfp4:
  type: "glob"
  enabled: true
  pattern: "*.mlp.experts.linear_fc2"
  config: "nvfp4"
default_bf16:
  type: "glob"
  enabled: true
  pattern: "*"
  config: "bf16"
""".strip()

# ci_args 新增 --ci-disable-kl-checker:
# 依据是 signed train/ppo_kl < 1e-8 断言在未改动的基线提交上同样失败,
# 因此该豁免不改变 NVFP4 实现本身
ci_args = (
    "--ci-test " "--ci-disable-kl-checker " "--ci-disable-logprobs-checker " "--ci-disable-weight-update-
    checker "
)

# mixed_precision_args 不再拼接 --extra-high-precision-layers-hf / -megatron:
# 共享专家的排除已经内建在量化默认范围里, 原先的参数是冗余 workaround
mixed_precision_args = (
    "--transformer-impl transformer_engine "
    "--bf16 "
    "--fp4-format e2m1 "
    "--fp4-recipe nvfp4 "
    "--first-last-layers-bf16 "
    f"--num-layers-at-start-in-bf16 {NUM_LAYERS_AT_START_IN_BF16} "
    f"--num-layers-at-end-in-bf16 {NUM_LAYERS_AT_END_IN_BF16} "
    f"--te-precision-config-file {te_precision_config_path} "
)

```

评论区精华

本 PR 的正式评审几乎没有留下技术交锋: [claude\[bot\]](#) 因 PR 来自 fork 而直接关闭自动评审 (“This pull request is from a fork — automated review is disabled”), 人类评审只有 [yueming-yuan](#) 的两次 APPROVED, 正文为空。真正的论证集中在 PR body 的 Validation / Limitations 章节, 作者主动披露了三处争议点:

“Neither exact 8-GPU GLM-5.2 RL E2E passed. Both exercised the affected FlashInfer routed-MoE path through both rollout generations, then failed the same rollout-2 signed-KL assertion.”

作者没有把失败归因于本次改动，而是重跑 before-PR 基线证明该断言在改动前就失败（基线 `train/ppo_kl=0.00005550903006223962`，head `train/ppo_kl=0.00015277016791515052`），并给出 step-0/step-1 的 `ppo_kl`、`train_rollout_logprob_abs_diff`、`train_rollout_kl` 对照表，随后用 `--ci-disable-kl-checker` 绕过断言、保留指标计算与日志。

“The one-sided `train/ppo_kl < 1e-8` gate accepts negative values regardless of magnitude; exit status therefore depends on sign and is not a robust magnitude check.”

这段判断指出仓库 CI 的 KL 门禁本身存在缺陷：两版 step-0 的负 KL 只是「符号正确」才通过，量级反而更大，说明门禁不具备幅值分辨能力。

第三处权衡是测试资产的取舍：`tests/fast/test_nvfp4_scope.py` 在 CUDA 上 13 项全绿（含 `test_nvfp4_hf_conversion_preserves_shared_expert_storage`），但最终被 follow-up commit `da280c4f` Removal 掉，未随 PR 合入，导致「共享专家必须保持 BF16」这条新契约缺少单测守卫。此外 E2E 依赖手动覆盖 FlashInfer 到 0.6.18 以规避 Issue #4486 的 NaN 回归（`pip check` 因此报出预期的不一致），这一环境依赖也未随 PR 固化。

- signed KL 门禁失效与 `--ci-disable-kl-checker` 的取舍 (testing): 以「基线同样失败」为依据关闭该断言，保留 KL/log-prob 指标的计算与日志；门禁本身的判据缺陷未修复。
- `tests/fast/test_nvfp4_scope.py` 在合入前被移除，新契约缺少单测守卫 (testing): 合入版本没有为「共享专家必须保持 BF16」提供自动化回归保护，只能依赖 8 卡 GLM-5.2 E2E。
- NVFP4 shared-expert 默认范围与训练 / 导出路径对齐 (design): 三条路径统一为 routed experts only，共享专家默认保留源精度；`--extra-high-precision-layers-*` 仍保留给首尾层、MLA 投影等例外。
- E2E 依赖 FlashInfer 0.6.18 手动覆盖以规避 NVFP4 MoE NaN 回归 (correctness): 验证结果依赖手工环境覆盖，仓库镜像与依赖声明未随 PR 更新，同路径在 CI 上仍可能复现 NaN。
- 评审覆盖情况 (other): 技术论证完全依赖作者在 PR body 中自证，未经过独立的代码级 review 交锋。

风险与影响

- 风险：兼容性与数据一致性（高）：`tools/convert_hf_to_nvfp4.py` 与 `quantizer_nvfp4.py` 的默认量化半径都变了，此前用旧版工具产出的 NVFP4 checkpoint 中共享专家是 NVFP4 张量，而新的训练 / 导出路径按 BF16 处理，直接复用旧产物会造成精度语义不一致，需要重新执行转换与导出；同时既有 NVFP4 基线的指标（KL、reward）不具备直接可比性。

符号移除的破坏性（中）：`_is_moe_expert_weight_name` 被重命名为 `_is_routed_expert_weight_name`，常量 `EXPERT_NAME_MARKERS` 被拆分为 `ROUTED_EXPERT_NAME_MARKERS` / `SHARED_EXPERT_NAME_MARKERS`，旧的模块级名字被直接删除。仓库内若有脚本或 notebook import 这两个符号会立刻报 `ImportError`，属于工具层的隐性 breaking change。

测试覆盖不足（中高）：新增的「先否决共享专家、再匹配路由专家」判定顺序是本 PR 的核心逻辑，但唯一针对它的 `tests/fast/test_nvfp4_scope.py` 在合入前被移除；`tests/fast-gpu/test_nvfp4_quantizer.py` 只覆盖 gate/up 成对量化与跨 shard 配对，不覆盖

权重名判定，回归只能依赖 8 卡 GLM-5.2 E2E。

E2E 校验被削弱（中）：`--ci-disable-kl-checker` 叠加原有的 `--ci-disable-logprobs-checker`、`--ci-disable-weight-update-checker` 后，该 E2E 已不再做数值一致性断言，即便变绿也无法证明量化范围收紧没有引入数值副作用（PR 自己也写明“even a pass would not establish numerical parity”）。另外 signed-KL 门禁接受任意负值，属于既有的判据缺陷。

外部依赖风险（中）：E2E 走的是 `--sglang-moe-runner-backend flashinfer_trtllm_routed` 与 per-token 激活量化路径，正是 Issue #4486 中 FlashInfer 0.6.16.post4/0.6.17 在 SM100/SM103 上产生 NaN 的位置；本次靠手动替换 0.6.18 的三个 wheel 才跑通，仓库镜像与依赖声明未同步更新，同路径在 CI 上仍可能复现 NaN 并造成不稳定。

性能与资源（低）：共享专家从 NVFP4 回到 BF16 会略增显存占用与带宽 / 算力开销，但 MoE 中共享专家通常只占少量参数，属于为精度与跨路径一致性付出的可接受代价；dense 层、MLA、MXFP8 与量化数学不受影响。

- 影响：用户侧：使用 NVFP4 RL（GLM-5.2、DeepSeek 系 MoE）的用户升级后，共享专家自动保持源精度，不再需要手工传 `--extra-high-precision-layers-hf / --extra-high-precision-layers-megatron`；但必须重新转换并导出 NVFP4 checkpoint，否则会带着旧的量化共享专家继续训练。文档 docs/advanced/low-precision.md 已同步，降低了误用成本。

系统侧：NVFP4 的作用域在一次 PR 内被三处（离线转换、Megatron 训练 matcher、Megatron→HF 导出 / live weight update）统一成 routed experts only，消除了「转换侧量化、训练侧按 BF16 加载」的隐式不一致；`quantizer_nvfp4.py` 位于权重导出链路上，因此在线权重同步产出的张量布局也随之改变，下游 SGLang 推理侧加载的权重集合发生变化。

团队侧：验证成本高（8×B300、单次 E2E 约 1200–1500 秒、需要 FlashInfer 手动覆盖），短期内对这条 E2E 的信任度会下降，因为 KL 断言被关闭；同时新增的 `--ci-disable-kl-checker` 会与近期仓库里「临时禁用失败测试保 CI 绿灯」的做法叠加，持续累积技术债。

- 风险标记：核心量化契约变更，既有 NVFP4 checkpoint 需重转，KL 断言被禁用，缺少单测守卫，FlashInfer 版本未随 PR 固定

关联脉络

- PR #3148 [CI] temporarily disable failing FT tests: 同类处理方式：为保持 CI 绿灯而临时禁用失败测试。本 PR 的 `--ci-disable-kl-checker` 属于同一类「先豁免、后修复」的妥协，两者的技术债性质一致。
- PR #3160 [CI] disable the two FT determinism tests #3148 missed: 上述豁免的后续补丁，说明这类测试禁用存在扩散风险；GLM-5.2 NVFP4 E2E 关闭 KL 断言后也可能需要类似跟进。
- PR #3147 Bound the trainer's GPU memory during the LoRA weight sync: 同属 Megatron 后端权重同步 / 导出链路（megatron_utils、weight_update）。NVFP4 量化发生在同一导出路径上，本 PR 改变导出张量集合会与该链路的显存与生命周期管理相互影响。

- PR #3162 ci: pin the DSv4 bshd/thd parity test to the miles impl: 同为「量化 / 精度相关 E2E 因默认实现或环境变化而失败, 通过测试侧调整恢复 CI」的处理范式, 可与本 PR 的测试侧收敛对照阅读。