# PR #36040 完整报告

sgl-project/sglang

[diffusion] feat: support mixed w4a4 and int8 checkpoints

合并时间：2026-08-24 20:35

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/36040

## 执行摘要

- 一句话：支持混合 W4A4 与 INT8 序列化检查点加载
- 推荐动作：值得精读 kitchen_w4a4_config.py 中的委托模式，它展示了如何在保持既有量化方法不变的情况下，通过配置聚合支持混合格式。虽然代码量不大，但 get_quant_method 和 supports_input_partition 的双路委托是复用已有能力的好例子。建议阅读 quantization_utils.py 中的格式组合分支，理解自动检测的边界。

## 功能与动机

真实世界的 Comfy 检查点（如 Abiray/Minimax-H3-nvfp4-INT4-INT8-Convrot）将 W4A4 与 INT8 层混合在同一个 safetensors 中。此前 SGLang 仅支持单一格式，遇到混合检查点会报错，用户需要手动拆分或重新导出。PR 的目标是直接消费这类检查点，同时保持 CLI 无新增标志，并保留每种格式的 TP 组边界准入规则。PR body 也明确说明：'admit Comfy checkpoints that mix convrot_w4a4 and serialized int8_tensorwise layers - compose the existing W4A4 and INT8 methods instead of adding another kernel or global quantization mode'。

## 实现拆解

1. 格式识别入口：在 quantization_utils.py 的 resolve_comfy_checkpoint_quantization 中新增 ["convrot_w4a4", "int8_tensorwise"] 组合分支，将其解析到 KitchenW4A4Config，从而让混合检查点自动进入 W4A4 配置作为分发中枢。
2. 配置内委托：在 kitchen_w4a4_config.py 的 KitchenW4A4Config.__init__ 中，先过滤所有 int8_tensorwise 标记并实例化内部 KitchenInt8Config；随后在 get_quant_method 与 supports_input_partition 中按标记格式委托给内部 INT8 配置处理，同时保持 W4A4 原有路径与 TP 组边界校验不变。
3. 测试覆盖：在 test_transformer_quant.py 中新增 test_mixed_w4a4_int8_dispatches_each_serialized_layer，用两个标记构造混合配置，验证 W4A4 层得到 packed 权重（shape (3, 128)）、INT8 层得到非 packed 权重（shape (3, 256)），并断言 config.selected 记录了所有被分发的层。
4. 文档更新：quantization.mdx 更新 comfy-w4a4-convrot 表格说明，明确支持与 int8_tensorwise 混合；MiniMax-H3.mdx 补充混合导出的使用示例，说明 SGLang 自动按层分发而非应用单一全局方法。

关键文件：

- python/sglang/multimodal_gen/runtime/layers/quantization/configs/kitchen_w4a4_config.py（模块 量化配置；类别 source；类型 core-logic；符号 KitchenW4A4Config.init, KitchenW4A4Config.get_quant_method, KitchenW4A4Config.supports_input_partition）：混合分发逻辑的核心：在 W4A4 配置中引入内部 INT8 配置并实现按标记委托。
- python/sglang/multimodal_gen/runtime/utils/quantization_utils.py（模块 量化解析；类别 source；类型 core-logic；符号 resolve_comfy_checkpoint_quantization）：检查点量化格式的自动检测入口，新增混合格式组合，是启用混合加载的前提。
- python/sglang/multimodal_gen/test/unit/test_transformer_quant.py（模块 单元测试；类别 test；类型 test-coverage；符号 test_mixed_w4a4_int8_dispatches_each_serialized_layer）：新增混合分发测试，验证 W4A4 与 INT8 层在同配置下正确路由，并记录 selected 集合。
- docs/docs/sglang-diffusion/quantization.mdx（模块 文档；类别 other；类型 documentation）：更新 comfy-w4a4-convrot 格式说明，补充混合 int8_tensorwise 层的支持描述。
- docs/cookbook/diffusion/MiniMax/MiniMax-H3.mdx（模块 文档；类别 other；类型 documentation）：补充混合检查点的使用示例，说明无需指定 --quantization。

关键符号：KitchenW4A4Config.init, KitchenW4A4Config.get_quant_method, KitchenW4A4Config.supports_input_partition, resolve_comfy_checkpoint_quantization, test_mixed_w4a4_int8_dispatches_each_serialized_layer

## 评论区精华

本 PR 的 review 没有产生实质技术讨论。作者通过 /tag-and-rerun-ci 触发了 CI 重跑，Mintlify 机器人自动生成了文档预览链接；PR 在设计上的核心取舍（复用已有方法而非新增内核）已在 body 中明确交代，未引发争议。

- 暂无高价值评论线程

## 风险与影响

- 风险：数据一致性风险：get_quant_method 与 supports_input_partition 依赖 _int8_config 非空，若未来出现仅含 W4A4 标记但被错误路由到混合分支的调用，会触发断言。当前入口 resolve_comfy_checkpoint_quantization 的分支逻辑保证了正确性，但属于隐式契约。 TP 边界校验分工：混合模式下 W4A4 与 INT8 各自有组边界要求，代码通过委托给内部配置分头校验，但若 TP 分区大小仅满足其中一种格式，会在运行时暴露，需要用户确保分区兼容。 组合覆盖有限：quantization_utils.py 仅新增了 convrot_w4a4 + int8_tensorwise 的组合；其他混合格式（如与 asym_w4a8_int8）仍会抛出 NotImplementedError。 测试覆盖单一：单元测试仅覆盖模拟的两个标记，未覆盖真实检查点的完整标记集；验证依赖对公共检查点的手工检查。
- 影响：对用户来说，可直接加载混合量化检查点，无需额外参数或手动处理，降低 Comfy 生态模型部署成本；对系统来说，仅增加一个格式组合的分发逻辑，没有引入新内核或全局量化模式，风险较低；对团队来说，文档和测试补齐了功能契约，便于后续维护。

- 风险标记：依赖标记数据一致性，组合覆盖有限，断言依赖内部配置，测试覆盖单一

## 关联脉络

- PR #36039 [Diffusion] Load serialized ConvRot W4A4 checkpoints: 本 PR 直接叠加于 #36039 之上，在其基础上扩展混合 INT8 支持；共享 kitchen_w4a4_config.py 与 quantization_utils.py 等文件。