PR #25023 完整报告

sgl-project/sglang

[NemotronH] V3 Omni wrapper: WeightsMapper + config round-trip

合并时间：2026-05-27 04:34

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25023

执行摘要

- 一句话：修复 NemotronH V3 Omni NVFP4 权重加载
- 推荐动作：建议检查并合并，属于 bugfix，逻辑清晰，影响范围小。

功能与动机

解决 Nemotron-3-Nano-Omni NVFP4 检查点在加载时由于 quantized_layers 的键前缀 language_model.backbone. 在 sglang 中未正确映射而失败的问题；以及 to_dict() 后恢复配置时丢失 raw_vision_config 的问题。

实现拆解

1. 添加 WeightsMapper：在 NemotronH_Nano_VL_V2 类上定义 hf_to_sglang_mapper，将 language_model.backbone. 前缀映射为 language_model.model.，修复量化层配置加载。
2. 配置往返修复：在 NemotronH_Nano_VL_V2_Config.__init__ 中，当 vision_config 为 None 时，从 kwargs 中弹出 raw_vision_config 赋值给 vision_config，确保 to_dict() / from_dict() 往返不丢失视觉配置。

关键文件：

- python/sglang/srt/models/nano_nemotron_vl.py（模块 模型定义；类别 source；类型 data-contract；符号 NemotronH_Nano_VL_V2）：为主要模型类添加了 WeightsMapper，修复量化层名称映射。
- python/sglang/srt/configs/nano_nemotron_vl.py（模块 配置；类别 source；类型 core-logic；符号 NemotronH_Nano_VL_V2_Config）：在配置类中支持 vision_config 与 raw_vision_config 的别名，修复配置往返。

关键符号：NemotronH_Nano_VL_V2.init, NemotronH_Nano_VL_V2_Config.init

关键源码片段

python/sglang/srt/models/nano_nemotron_vl.py

为主要模型类添加了 WeightsMapper，修复量化层名称映射。

```
# 导入 WeightsMapper
from sglang.srt.models.utils import WeightsMapper
```

```
class NemotronH_Nano_VL_V2(EVS):
    # 外层的 mapper 用于将量化配置中的层名从 language_model.backbone.
    # 转换为 runtime 使用的 language_model.model. 前缀
    hf_to_sglang_mapper = WeightsMapper(
        orig_to_new_prefix={
            "language_model.backbone.": "language_model.model.",
        },
    )
```

python/sglang/srt/configs/nano_nemotron_vl.py

在配置类中支持 vision_config 与 raw_vision_config 的别名，修复配置往返。

```
class NemotronH_Nano_VL_V2_Config(PretrainedConfig):
    def __init__(
        self,
        vision_config=None,
        llm_config=None,
        sound_config=None,
        # ... 其他参数 ...
        **kwargs,
    ):
        # 当 vision_config 为 None 时，尝试从 kwargs 中取出 raw_vision_config（V2 存储名称），
        # 以支持 V3->V2 配置往返：to_dict() 生成 raw_vision_config，
        # 而 from_dict() 通过 vision_config 参数重建。
        if vision_config is None:
            vision_config = kwargs.pop("raw_vision_config", None)

        super().__init__(**kwargs)

        if llm_config is not None:
            self.llm_config = NemotronHConfig(**llm_config)
            assert isinstance(vision_config, dict), "vision_config must be a dictionary"
            self.raw_vision_config = vision_config
        else:
            assert vision_config is None
            self.llm_config = NemotronHConfig()
            self.raw_vision_config = {}
```

评论区精华

PR 没有公开的 review 讨论。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低：变更仅影响 NemotronH V3 Omni 模型加载路径，不影响其他模型；增加了导入依赖 WeightsMapper，但该模块已存在。没有测试覆盖，但改动简单清晰。

- 影响：直接影响使用 NVFP4 量化格式的 Nemotron-3-Nano-Omni 模型加载；不影响现有其他模型或推理路径。
- 风险标记：缺少测试覆盖

关联脉络

- PR #25024 [Companion] (unknown): PR body 中提到该 PR 是 #25024 的 companion，可能处理类似问题。