PR #23808 完整报告

sgl-project/sglang

[Feature] Xiaomi MiMo-V2.5-Pro day0 support

合并时间：2026-04-28 11:43

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23808

执行摘要

- 一句话：添加 Xiaomi MiMo-V2.5-Pro 模型 day0 支持
- 推荐动作：值得精读：PR 展示了如何在不破坏向后兼容的前提下对已有模型架构进行重命名和扩展，特别是保留旧类名作为别名、处理 fused QKV 检查点加载等设计模式值得参考。

功能与动机

MiMo-V2.5-Pro 是小米发布的 Mixture-of-Experts（MoE）语言模型，拥有 1.02T 总参数、42B 激活参数，支持长上下文（1M tokens）和多 token 预测（MTP），需要在 SGLang 中提供运行支持。

实现拆解

1. 文件搬迁与类重命名：将 mimo_v2_flash.py 和 mimo_v2_flash_nextn.py 分别重命名为 mimo_v2.py 和 mimo_v2_nextn.py，并将内部类 MiMoV2FlashForCausalLM 重命名为 MiMoV2ForCausalLM，MiMoV2MTP 的基类也随之更新；同时在原文件中保留 MiMoV2FlashForCausalLM 作为空继承类，确保旧配置仍可加载。
2. 配置键兼容：在 MiMoV2DecoderLayer 和 MiMoV2MTPLayer 的 __init__ 中，优先从 config.context_len 读取最大位置编码长度，若不存在则回退到 max_position_embeddings，以适配不同模型配置风格。
3. Fused QKV 加载支持：在 MiMoV2ForCausalLM.load_weights 和 MiMoV2MTP.load_weights 中新增对 fused QKV projection 检查点的处理：按 attention_tp_size 对权重行维度分片取当前 rank 部分，解决 FP8 量化情况下的特殊布局兼容问题。
4. 配置与调度适配：在 model_config.py 中，将 MiMoV2ForCausalLM 加入 draft model 映射（MTP）、hybrid SWA compress 列表、attention sinks 检测、hybrid layer ID 获取等全量判断逻辑。在 server_args.py 中，将新架构名加入 multi-layer EAGLE 自动启用、hierarchical cache 设置、自动推测参数选择等分支。在 common.py 中将新架构名加入 is_fa3_default_architecture 列表，确保 FlashAttention 3 被正确选择。
5. 测试与部署: PR 未新增独立测试文件，但已有 CI 测试 test_mimo_models.py 通过运行验证；PR body 提供了详细的 prefill 和 decode 性能基准数据。

关键文件：

- python/sglang/srt/models/mimo_v2.py（模块 模型定义；类别 source；类型 rename-or-move；符号 MiMoV2FlashForCausalLM, MiMoV2ForCausalLM）：核心模型文件，完成类重命名、fused QKV 加载支持、config 兼容性调整，并保留了旧类名作为入口。
- python/sglang/srt/models/mimo_v2_nextn.py（模块 MTP draft；类别 source；类型 rename-or-move；符号 MiMoV2MTP）：MTP 模型文件，同步重命名基类和导入路径，添加 fused QKV 支持，确保 draft 模型可加载。
- python/sglang/srt/configs/model_config.py（模块 模型配置；类别 source；类型 data-contract；符号 _config_draft_model, _derive_hybrid_model, _detect_attention_sinks, is_hybrid_swa_model）：集中控制所有架构检查，确保新架构名在 MTP 映射、hybrid SWA、attention sink 和 layer pattern 中正确参与判断。
- python/sglang/srt/server_args.py（模块 服务器参数；类别 source；类型 core-logic；符号 _handle_model_specific_adjustments, auto_choose_speculative_params）：服务器参数配置中，新增架构名匹配分支以启用多 layer EAGLE、hierarchical cache 调整、推测参数选择等。
- python/sglang/srt/utils/common.py（模块 工具函数；类别 source；类型 core-logic；符号 is_fa3_default_architecture）：将 MiMoV2ForCausalLM 添加到 is_fa3_default_architecture 默认架构列表，确保默认使用 FlashAttention 3 后端。

关键符号：MiMoV2ForCausalLM.init, MiMoV2ForCausalLM.load_weights, MiMoV2MTP.init, MiMoV2MTP.load_weights, MiMoV2FlashForCausalLM, ModelConfig._config_draft_model, ModelConfig._derive_hybrid_model, ModelConfig._detect_attention_sinks, is_hybrid_swa_model, get_hybrid_layer_ids, ServerArgs._handle_model_specific_adjustments, auto_choose_speculative_params, is_fa3_default_architecture

关键源码片段

python/sglang/srt/models/mimo_v2.py

核心模型文件，完成类重命名、fused QKV 加载支持、config 兼容性调整，并保留了旧类名作为入口。

```
# 主模型类：支持 MiMo-V2.5-Pro
class MiMoV2ForCausalLM(nn.Module):
    ...
    def load_weights(self, weights: Iterable[Tuple[str, torch.Tensor]]):
        ...
        # 处理 fused qkv_proj 检查点（Pro 格式）
        if "qkv_proj" in name:
            if name in params_dict:
                tp_size = get_attention_tp_size()
                tp_rank = get_attention_tp_rank()
                param = params_dict[name]
                # 按 attention TP 分片取当前 rank 的部分
                loaded_weight = loaded_weight.chunk(tp_size, dim=0)[tp_rank]
```

```
                default_weight_loader(param, loaded_weight)
                continue
        ...

# 保留旧架构名作为别名，确保向前兼容
class MiMoV2FlashForCausalLM(MiMoV2ForCausalLM):
    pass

# 开放两个入口，新配置优先使用 MiMoV2ForCausalLM
EntryClass = [MiMoV2ForCausalLM, MiMoV2FlashForCausalLM]
```

python/sglang/srt/models/mimo_v2_nextn.py

MTP 模型文件，同步重命名基类和导入路径，添加 fused QKV 支持，确保 draft 模型可加载。

```
# 导入路径从 mimo_v2_flash 改为 mimo_v2，类名同步变更
from sglang.srt.models.mimo_v2 import (
    MiMoV2Attention,
    MiMoV2ForCausalLM, # 原来为 MiMoV2FlashForCausalLM
    MiMoV2MLP,
)

MiMoV2Config = None # 原 MiMoV2FlashConfig

class MiMoV2MTPLayer(nn.Module):
    def __init__(
        self,
        config: MiMoV2Config, # 参数类型同步更新
        ...
    ):
        ...
        # 优先读取 context_len 作为 max_position_embeddings，用于长上下文
        max_position_embeddings = getattr(
            config,
            "context_len",
            getattr(config, "max_position_embeddings", 32768),
        )
        ...

class MiMoV2MTP(MiMoV2ForCausalLM): # 基类更新
    ...
    def load_weights(self, weights, ...):
        ...
        # 同样支持 fused qkv_proj
        if "qkv_proj" in name:
            if name in params_dict:
                tp_size = get_attention_tp_size()
                tp_rank = get_attention_tp_rank()
                param = params_dict[name]
                loaded_weight = loaded_weight.chunk(tp_size, dim=0)[tp_rank]
```

```
                default_weight_loader(param, loaded_weight)
                continue
        ...
```

python/sglang/srt/configs/model_config.py

集中控制所有架构检查，确保新架构名在 MTP 映射、hybrid SWA、attention sink 和 layer pattern 中正确参与判断。

```
# 在 draft model 映射中同时匹配新旧架构名
if is_draft_model and self.hf_config.architectures[0] in (
    "MiMoV2ForCausalLM",
    "MiMoV2FlashForCausalLM",
):
    self.hf_config.architectures[0] = "MiMoV2MTP"

# 在 hybrid SWA compress 列表中添加新架构名
self.is_hybrid_swa_compress = self.hf_config.architectures[0] in [
    "MiMoV2ForCausalLM",
    "MiMoV2FlashForCausalLM",
    "MiMoV2MTP",
    ...
]

# 在 attention sinks 检测中添加新架构名
if any(
    a in archs
    for a in (
        "MiMoV2FlashForCausalLM",
        "MiMoV2ForCausalLM",
        "MiMoV2MTP",
    )
):
    ...
```

评论区精华

该 PR 的 review 过程没有产生实质性讨论（review comments 为空）。Issue 评论区有用户提问部署方法，作者已回复提供文档链接。

- 暂无高价值评论线程

风险与影响

- 风险：
  - 向后兼容性：通过保留 MiMoV2FlashForCausalLM 作为别名，旧配置文件的加载不受影响，风险较低。
  - 配置歧义：context_len 的引入可能在某些自定义配置中导致与 max_position_embeddings 不一致，但逻辑中有 fallback，风险可控。

  - Fused QKV 加载: 仅在 checkpoint 中包含 qkv_proj 时触发，若旧模型未使用 fused 格式则不受影响；分片逻辑依赖 get_attention_tp_size，若该函数在非 DP attention 场景返回异常值可能导致加载失败，但已有 CI 验证通过。
  - 测试覆盖: 未增加单元测试，依赖现有注册测试（test_mimo_models.py）和手动性能验证，回归风险存在但较低。
- 影响:
  - 用户影响: 用户现可部署并推理 MiMo-V2.5-Pro 模型，支持长上下文和 MTP 加速。
  - 系统影响: 增加新架构名，需保证后续改动同步更新所有分支检查点。
  - 团队影响: 维护复杂度略有增加，需同时维护 MiMoV2 和 MiMoV2Flash 两个入口；但重命名统一了命名风格。
  - 风险标记：核心路径变更（模型加载类），缺少测试覆盖，配置兼容性（context_len vs max_position_embeddings）

关联脉络

- 暂无明显关联 PR