PR #32401 完整报告

sgl-project/sglang

[Model] Support standalone text-only Qwen3.5 checkpoints

合并时间：2026-07-28 14:08

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/32401

执行摘要

- 一句话：支持独立文本 Qwen3.5 模型部署
- 推荐动作：值得精读，特别是如何通过包装现有 body 快速适配新 checkpoint 格式；建议在类似模型支持中采用此模式。

功能与动机

独立文本 Qwen3.5 检查点使用 Qwen3_5ForCausalLM 或 Qwen3_5MoeForCausalLM 架构，但现有 qwen3_5.py 中的类是供多模态包装器使用的 transformer body，返回 hidden states，缺少顶层 LM head 和 logits processor。此 PR 完成 #27899 中讨论的运行时路径，支持直接部署文本检查点。

实现拆解

1. 新增顶层模型类：在 qwen3_5_text.py 中定义 Qwen3_5ForCausalLM（和 Qwen3_5MoeForCausalLM）类，包装已有的 qwen3_5.Qwen3_5ForCausalLM body，添加 lm_head（Pipeline Parallel 感知）和 LogitsProcessor。
2. 注册新 Config 类型：在 configs/__init__.py 中导出 Qwen3_5TextConfig 和 Qwen3_5MoeTextConfig，并在 model_config.py 中将新架构添加到 draft 模型路由列表，使得它们能正确映射到 Qwen3_5ForCausalLMMTP。
3. 注册 Config Registry：在 utils/hf_transformers/common.py 中将 Qwen3_5TextConfig 和 Qwen3_5MoeTextConfig 加入 _CONFIG_REGISTRY，确保 AutoConfig 能加载对应的 model_type。
4. 权重加载适配：模型加载时从 model.* 键加载 body 权重，可选的未绑定 lm_head 单独加载，同时保持已有 MTP 加载器不变。

关键文件：

- python/sglang/srt/models/qwen3_5_text.py（模块 模型层；类别 source；类型 core-logic；符号 Qwen3_5ForCausalLM, init, start_layer, end_layer）：核心变更文件，包含新顶层模型类 Qwen3_5ForCausalLM 和 Qwen3_5MoeForCausalLM，复用 body 并添加 lm_head 和 logits processor。
- python/sglang/srt/configs/__init__.py（模块 配置；类别 source；类型 dependency-wiring）：导出新 config 类 Qwen3_5TextConfig 和 Qwen3_5MoeTextConfig，使其可通过包路径导入。

- python/sglang/srt/configs/model_config.py（模块 配置；类别 source；类型 data-contract）：在 _config_draft_model 中添加 Qwen3_5ForCausalLM 和 Qwen3_5MoeForCausalLM 架构到 draft 模型路由，确保这些文本架构能正确映射到 MTP 入口。
- python/sglang/srt/utils/hf_transformers/common.py（模块 工具模块；类别 source；类型 core-logic）：将 Qwen3_5TextConfig 和 Qwen3_5MoeTextConfig 注册到全局 _CONFIG_REGISTRY，确保 AutoConfig 能识别对应的 model_type。

关键符号：Qwen3_5ForCausalLM.init, Qwen3_5ForCausalLM.forward, Qwen3_5ForCausalLM.start_layer, Qwen3_5ForCausalLM.end_layer, Qwen3_5ForCausalLM.get_embed_and_head, Qwen3_5ForCausalLM.set_embed_and_head

关键源码片段

python/sglang/srt/models/qwen3_5_text.py

核心变更文件，包含新顶层模型类 Qwen3_5ForCausalLM 和 Qwen3_5MoeForCausalLM，复用 body 并添加 lm_head 和 logits processor。

```
# qwen3_5_text.py —— 顶层文本入口
# 复用 qwen3_5.Qwen3_5ForCausalLM body，添加 LM head 和 logits processor

class Qwen3_5ForCausalLM(nn.Module):
    body_cls = qwen3_5.Qwen3_5ForCausalLM # 复用已有多模态 body

    def __init__(self, config, quant_config=None, prefix=""):
        super().__init__()
        self.config = config
        self.quant_config = quant_config
        self.pp_group = get_pp_group()

        # 代理 quant config 的 packed_modules_mapping
        if quant_config is not None and hasattr(quant_config, "packed_modules_mapping"):
            quant_config.packed_modules_mapping = self.packed_modules_mapping

        # 实例化 body
        self.model = self.body_cls(
            config=config,
            quant_config=quant_config,
            prefix=add_prefix("model", prefix),
        )

        # 在 last rank 创建 lm_head
        if self.pp_group.is_last_rank:
            if self.pp_group.world_size == 1 and config.tie_word_embeddings:
                self.lm_head = self.model.embed_tokens
            else:
                self.lm_head = ParallelLMHead(
```

```
                config.vocab_size,
                config.hidden_size,
                quant_config=quant_config,
                org_num_embeddings=config.vocab_size,
                prefix=add_prefix("lm_head", prefix),
                use_attn_tp_group=get_server_args().enable_dp_lm_head,
            )
        else:
            self.lm_head = PPMissingLayer()

        self.logits_processor = LogitsProcessor(config)

        # 文本 checkpoint 保留 mrope_section，但与 1D RoPE 等价
        rope_config = getattr(config, "rope_parameters", None) or getattr(
            config, "rope_scaling", None
        )
        self.is_mrope_enabled = bool(rope_config) and "mrope_section" in rope_config
        self.capture_aux_hidden_states = False

    @property
    def start_layer(self):
        return self.model.start_layer

    @property
    def end_layer(self):
        return self.model.end_layer

    def get_input_embeddings(self):
        return self.model.embed_tokens

    def get_embed_and_head(self):
        return self.model.embed_tokens.weight, self.lm_head.weight

    def set_embed_and_head(self, embed, head):
        del self.model.embed_tokens.weight
        del self.lm_head.weight
        self.model.embed_tokens.weight = embed
        self.lm_head.weight = head
        torch.cuda.empty_cache()
        torch.cuda.synchronize()

    @torch.no_grad()
    def forward(
        self,
        input_ids,
        positions,
        forward_batch,
        input_embeds=None,
        pp_proxy_tensors=None,
```

```
        **kwargs,
    ):
        if self.is_mrope_enabled:
            positions = forward_batch.mrope_positions

        hidden_states = self.model(
            input_ids,
            positions,
            forward_batch,
            input_embeds,
            pp_proxy_tensors=pp_proxy_tensors,
        )
        # LogitsProcessor 生成最终 logits
        # 返回 logits 或 PPProxyTensors（依赖 pp_group 位置）
```

评论区精华

该 PR 没有收到 reviewer 评论或实质性讨论。仓库 CI 状态显示 PR Test 通过，PR Test Extra 失败，但未提供详细分析。Gemini Code Assist 机器人发表了一条关于服务终止的通知，与代码内容无关。

- 暂无高价值评论线程

风险与影响

- 风险：主要风险包括：
  - 权重加载一致性：新的 Qwen3_5ForCausalLM 类从 model.* 前缀加载 body 权重，但若 Hub 上 checkpoint 结构不符预期可能导致加载失败。
  - MTP 路由漏配：在 model_config.py 中新增的架构条件若字母排序有误或遗漏变体，可能导致 draft 模型无法正确初始化。
  - mrope 等效性：文本 checkpoint 保留 mrope_section，但注释说明其等价于 1D RoPE，若实际效果有偏差可能导致位置编码错误。
  - 没有单元测试：PR 未添加单元测试，潜在回归只能通过集成测试覆盖。
  - 影响：对用户：Qwen3.5 文本模型用户可以无需多模态包装器直接部署，减少配置复杂度。对系统：新增约 208 行核心代码，但仅在加载 Qwen3.5 文本模型时生效，无通用性能影响。对团队：模式可复用为其他多模态模型提供文本入口。
  - 风险标记：权重加载路径，MTP 路由变更，缺少测试覆盖

关联脉络

- 暂无明显关联 PR