PR #26088 完整报告

sgl-project/sglang

GLM-4.7-Flash: standalone MLA impl and MLA NextN/MTP

合并时间：2026-05-26 13:17

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26088

执行摘要

- 一句话：GLM-4.7-Flash 独立 MLA 实现及 NextN 推测解码
- 推荐动作：建议仔细审查 glm4_moe_lite_nextn.py 的 __init__ 是否按 review 建议修复；若未修复，应及时补充。此 PR 的独立模型设计思路值得参考，尤其 MLA NextN 的 zero_allocator 传递模式。建议合并后补充针对模型加载和 pipeline 的测试。

功能与动机

原 glm4_moe_lite 继承 deepseek_v2.py 包装类，易因 DSV4/DSA/CP 改动而损坏，MoE 门控使用 bf16 导致输出错误，且 MTP 推测解码时 draft 模型映射到 GQA 导致 KV-cache 形状不匹配。详见 PR body。

实现拆解

1. 重写 glm4_moe_lite.py：不再继承 DeepseekV2ForCausalLM 等包装类，改为自包含模型。复用稳定的 DeepseekV2AttentionMLA（组合方式）和 DeepseekV2WeightLoaderMixin。关键改动包括将 MoE 门控投影改为 FP32 精度，引入独立 Glm4MoeLiteSparseMoeBlock 以支持 DeepEP、TBO 和双流处理，并添加 _forward_shared_experts 等前向变体。
2. 新增 glm4_moe_lite_nextn.py：为 MLA 结构创建独立的 NextN 草案模型 Glm4MoeLiteForCausalLMNextN，包含 Glm4MoeLiteModelNextN。该模型在 forward 中构造 BumpAllocator（zero_allocator）传递给 Glm4MoeLiteDecoderLayer，满足 MLA 要求。
3. 更新 model_config.py：在 _config_draft_model 中将 Glm4MoeLiteForCausalLM 独立路由到Glm4MoeLiteForCausalLMNextN（而非共用GQA的Glm4MoeForCausalLMNextN），并在 _derive_model_shapes 中将 Glm4MoeLiteForCausalLMNextN 加入 MLA 检测列表。
4. 扩展 weight_utils.py：在 maybe_add_mtp_safetensors 中增加 Glm4MoeLiteForCausalLM 和 Glm4MoeLiteForCausalLMNextN 架构支持，确保 mtp.safetensors 能被自动加载。

关键文件：

- python/sglang/srt/models/glm4_moe_lite.py（模块 模型层；类别 source；类型 core-logic；符号 Glm4MoeLiteSparseMoeBlock, Glm4MoeLiteDecoderLayer, Glm4MoeLiteForCausalLM, forward）：核心模型重写，从 deepseek_v2 解耦，修复

gate 精度，新增多种前向路径

- python/sglang/srt/models/glm4_moe_lite_nextn.py（模块 推测解码；类别 source；类型 entrypoint；符号 Glm4MoeLiteModelNextN, Glm4MoeLiteForCausalLMNextN, forward, load_weights）：新增 MLA-based NextN 草案模型，实现零分配器传递和正确的 Draft 前向流程
- python/sglang/srt/configs/model_config.py（模块 配置层；类别 source；类型 data-contract）：配置路由和 MLA 检测，确保 draft 正确映射到新模型
- python/sglang/srt/model_loader/weight_utils.py（模块 权重加载；类别 source；类型 data-contract）：扩展 mtp.safetensors 自动加载，支持 Lite 架构

关键符号：Glm4MoeLiteForCausalLM.forward, Glm4MoeLiteModelNextN.forward, Glm4MoeLiteSparseMoeBlock.forward, maybe_add_mtp_safetensors

关键源码片段

python/sglang/srt/models/glm4_moe_lite_nextn.py

新增 MLA-based NextN 草案模型，实现零分配器传递和正确的 Draft 前向流程

```
class Glm4MoeLiteModelNextN(nn.Module):
    def __init__(
        self,
        config: PretrainedConfig,
        quant_config: Optional[QuantizationConfig] = None,
        prefix: str = "",
    ) -> None:
        super().__init__()
        # 如果使用 modelopt_fp4 量化，强制覆盖保证兼容
        if quant_config is not None and quant_config.get_name() == "modelopt_fp4":
            logger.warning(
                "Overriding Glm4MoeLiteForCausalLMNextN quant config for modelopt_fp4 "
                "GLM-4.7-Flash model."
            )
            quant_config = None

        self.vocab_size = config.vocab_size

        self.embed_tokens = VocabParallelEmbedding(
            config.vocab_size,
            config.hidden_size,
            use_attn_tp_group=is_dp_attention_enabled(),
            prefix=add_prefix("embed_tokens", prefix),
        )

        self.enorm = RMSNorm(config.hidden_size, eps=config.rms_norm_eps)
        self.hnorm = RMSNorm(config.hidden_size, eps=config.rms_norm_eps)

        self.eh_proj = nn.Linear(2 * config.hidden_size, config.hidden_size, bias=False)
```

```
        self.decoder = Glm4MoeLiteDecoderLayer(
            config,
            0,
            quant_config=quant_config,
            is_nextn=True,
            prefix=add_prefix("decoder", prefix),
        )

        self.shared_head = nn.Module()
        self.shared_head.norm = RMSNorm(config.hidden_size, eps=config.rms_norm_eps)

    def forward(
        self,
        input_ids: torch.Tensor,
        positions: torch.Tensor,
        forward_batch: ForwardBatch,
        input_embeds: torch.Tensor = None,
    ) -> torch.Tensor:
        # 解码层使用 DeepseekV2AttentionMLA 需要 BumpAllocator（zero_allocator），
        # 这与 GQA 路径不同
        zero_allocator = BumpAllocator(
            buffer_size=2,
            dtype=torch.float32,
            device=(
                input_embeds.device if input_embeds is not None else input_ids.device
            ),
        )

        if input_embeds is None:
            hidden_states = self.embed_tokens(input_ids)
        else:
            hidden_states = input_embeds

        if hidden_states.shape[0] > 0:
            # 将编码器 hidden_states 和主模型 hidden_states（来自 spec_info）
            # 拼接后通过 eh_proj 压缩，实现特征融合
            hidden_states = self.eh_proj(
                torch.cat(
                    (
                        self.enorm(hidden_states),
                        self.hnorm(forward_batch.spec_info.hidden_states),
                    ),
                    dim=-1,
                )
            )

        residual = None
        # 禁用 expert distribution recorder，防止 draft 模型干扰主模型记录
        with get_global_expert_distribution_recorder().disable_this_region():
```

```
            hidden_states, residual = self.decoder(
                positions, hidden_states, forward_batch, residual, zero_allocator
            )

        if not forward_batch.forward_mode.is_idle():
            if residual is not None:
                hidden_states, _ = self.shared_head.norm(hidden_states, residual)
            else:
                hidden_states = self.shared_head.norm(hidden_states)

        return hidden_states
```

评论区精华

主要讨论来自 gemini-code-assist[bot] 的 review，指出新类 Glm4MoeLiteForCausalLMNextN 继承 Glm4MoeLiteForCausalLM 但绕过其 __init__ 直接调用 nn.Module.__init__，导致 pp_group 未初始化，会在 load_weights 时出错。建议调用 super().__init__。该评论未被作者采纳，但 PR 仍获批合并（GPU CI 通过），存在潜在风险。

- Glm4MoeLiteForCausalLMNextN 初始化绕过基类导致 pp_group 未初始化 (correctness): 作者未修改，但 PR 仍被批准合并（GPU CI 通过）。

风险与影响

- 风险：主要风险： 1) pp_group 未初始化的潜在问题在 pipeline parallelism 场景下可能暴露权重加载错误； 2) 重写后的模型改动量大（603 行新增），可能引入回归，缺少测试覆盖； 3) 模型配置和权重加载的扩展点到多个文件，耦合性增加。
  - 影响：对用户：修复了 GLM-4.7-Flash 模型的推理正确性，并支持 MLA-based 推测解码，提升性能。 对系统：模型实现更独立，不依赖 deepseek_v2 包装类，减少跨模型耦合影响。 对团队：提升了模型架构的模块化，但需关注 pipeline parallelism 和缺少测试的回归风险。
- 风险标记：核心路径变更，缺少测试覆盖，未解决的 review 问题，pipeline parallelism 潜在风险

关联脉络

- 暂无明显关联 PR