PR #52948 完整报告

vllm-project/vllm

[Model] Support bidirectional (encoder-only) attention for DeepSeek e···

合并时间：2026-08-20 00:36

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/52948

执行摘要

- 一句话：DeepSeek 骨干双向注意力嵌入模型支持，启用非 MLA 路径
- 推荐动作：该 PR 值得关注，尤其适合要新增 encoder-only 变体的模型维护者阅读：它展示了「HF 配置字段驱动注意力类型 + 全局 MLA 开关短路 + registry 复用已有因果模型实现」的组合套路，且与 Qwen2/Qwen3 双向 embedding 的处理方式一脉相承。不过运行时数值验证尚未自动化，建议在后续补一条针对双向 DeepSeek embedding 的回归测试。

功能与动机

DeepSeek-V2/V3 是 decoder-only（causal）架构，但部分 embedding 模型复用了它的骨干并改成双向（encoder-only）注意力。由于 DeepSeek 使用 MLA，而 MLA 的融合 kernel 只支持 causal attention，vLLM 当前没有任何代码路径能对 DeepSeek 骨干应用双向注意力，导致这类模型无法运行。PR body 明确说明目标是在非 MLA 注意力路径上响应 HF config 的 is_causal=False，并保证对现有 DeepSeek 模型字节级不改变行为。

实现拆解

实现按 4 步完成：

1. 配置开关 vllm/config/model.py：在 ModelConfig.use_mla 的 vLLM 模型实现分支中新增 is_causal=False 短路返回 False。这样做让双向 DeepSeek 变体自动绕过 causal-only 的 MLA 融合 kernel，走 materialized（非 MLA）注意力路径，用户无需设置 VLLM_MLA_DISABLE=1。
2. 注意力层改造 vllm/model_executor/models/deepseek_v2.py：在 DeepseekV2Attention.__init__ 中读取 config.is_causal（默认 True），为 False 时选用 EncoderOnlyAttention 并把 AttentionType.ENCODER_ONLY 传给 self.attn，否则保持原 Attention 与 AttentionType.DECODER；同时补入 EncoderOnlyAttention 与 AttentionType 的 import。这一步是该 PR 让双向注意力真正生效的核心路径。
3. 模型注册 vllm/model_executor/models/registry.py：在 _EMBEDDING_MODELS 中注册 DeepseekV3BidirectionalModel，映射到 deepseek_v2 模块的 DeepseekV3ForCausalLM，使其可作为 embedding 模型被 pooling runner 识别。
4. 测试配套 tests/models/registry.py：在 _EMBEDDING_EXAMPLE_MODELS 增加 ai-sage/Giga-Embeddings-instruct-10B-A1.8B-0826 示例条目，带 trust_remote_code=True 和 hf_overrides（把 model_type 改写为 deepseek_v3、

auto_map 置空），保证 registry import 与覆盖率测试通过。数值 parity 在 PR body 中报告，但未作为自动化测试合入 CI。

关键文件：

- vllm/model_executor/models/deepseek_v2.py（模块 模型实现；类别 source；类型 core-logic；符号 DeepseekV2Attention）：核心实现文件：DeepseekV2Attention 根据 config.is_causal 选择 EncoderOnlyAttention/AttentionType.ENCODER_ONLY，让双向注意力在非 MLA 路径上真正生效。
- vllm/config/model.py（模块 模型配置；类别 source；类型 configuration；符号 ModelConfig.use_mla）：ModelConfig.use_mla 的 is_causal=False 短路分支是双向 DeepSeek 自动选择非 MLA 路径的关键配置开关，review 中对该分支位置有过讨论。
- vllm/model_executor/models/registry.py（模块 模型注册；类别 source；类型 data-contract；符号 _EMBEDDING_MODELS）：注册 DeepseekV3BidirectionalModel 到 _EMBEDDING_MODELS，复用 DeepseekV3ForCausalLM 实现，是模型可被 pooling runner 识别的前提。
- tests/models/registry.py（模块 注册测试；类别 test；类型 test-coverage；符号 _EMBEDDING_EXAMPLE_MODELS）：测试注册表补入 Giga-Embeddings 示例及 hf_overrides，保证 registry import 与覆盖率测试覆盖新架构。

关键符号：ModelConfig.use_mla, DeepseekV2Attention.init

关键源码片段

vllm/model_executor/models/deepseek_v2.py

核心实现文件：DeepseekV2Attention 根据 config.is_causal 选择 EncoderOnlyAttention/AttentionType.ENCODER_ONLY，让双向注意力在非 MLA 路径上真正生效。

```
# vllm/model_executor/models/deepseek_v2.py —— DeepseekV2Attention.__init__ 关键分支
if getattr(config, 'is_causal', True):
    attn_type = AttentionType.DECODER
else:
    attn_type = AttentionType.ENCODER_ONLY

# 在非 MLA 路径上按配置选择双向或因果注意力：
# - decoder-only DeepSeek 保持原来的 Attention + DECODER；
# - embedding 变体（is_causal=False）改用 EncoderOnlyAttention + ENCODER_ONLY，
# 使双向注意力能够生效，同时避免使用 causal-only 的 MLA kernel。
attn_cls = EncoderOnlyAttention if attn_type == AttentionType.ENCODER_ONLY else Attention
self.attn = attn_cls(
    self.num_local_heads,
    self.qk_head_dim,
    self.scaling,
    num_kv_heads=self.num_local_heads,
    cache_config=cache_config,
    quant_config=quant_config,
    prefix=f'{prefix}.attn',
```

```
    attn_type=attn_type,
)
```

vllm/config/model.py

ModelConfig.use_mla 的 is_causal=False 短路分支是双向 DeepSeek 自动选择非 MLA 路径的关键配置开关，review 中对该分支位置有过讨论。

```
# vllm/config/model.py —— ModelConfig.use_mla
@property
def use_mla(self) -> bool:
    if envs.VLLM_MLA_DISABLE:
        return False
    if self.using_transformers_backend():
        # kv_lora_rank 表示 Transformers 实现使用 MLA
        return getattr(self.hf_text_config, 'kv_lora_rank', None) is not None

    # 双向 DeepSeek 变体（is_causal=False，embedding 模型使用）必须走非 MLA
    # 注意力路径，因为 MLA kernel 只支持 causal attention。
    if not getattr(self.hf_text_config, 'is_causal', True):
        return False

    # vLLM 模型实现的手工维护模型类型列表
    return self.is_deepseek_mla
```

评论区精华

review 中唯一实质讨论集中在 vllm/config/model.py 的分支放置位置：DarkLight1337 指出 is_causal=False 分支应放在 use_mla property 末尾、紧邻 return self.is_deepseek_mla，而不是插在注释与 return 之间，以保持逻辑可读性。作者回复 fixed，并通过提交 Update model.py change the order of branches（82f3783）调整了顺序，随后 DarkLight1337 approve。过程中还触发了两轮 /ci run（Buildkite CI #84605、#84606）。整体是一致性维护型意见，无设计层面的争议。

- use_mla 中 is_causal 分支的放置位置 (style): 作者通过提交 82f3783 调整分支顺序并回复 fixed，维护者随后 approve。

风险与影响

- 风险：风险点如下：
  1. 全局配置分支：ModelConfig.use_mla 是全局 property，新增的 is_causal=False 短路对所有走 vLLM 模型实现路径的配置生效。当前实测只影响 DeepSeek MLA 类模型，但未来若有其他同时具备 is_deepseek_mla 与双向注意力的架构，也会被静默切到非 MLA 路径，需要留意该分支的通用性。
  2. 运行时正确性未入 CI：PR 只在 registry 测试中覆盖导入与映射，双向注意力的数值 parity 是作者在 RTX 5090 上手工验证的，没有自动化回归测试保护 EncoderOnlyAttention 路径。

3. 性能差异：非 MLA 的 materialized attention 与 MLA 相比内存和计算特征不同，作为 embedding 模型使用时吞吐表现需要单独评估。
4. 依赖 hf-overrides 变通：目标模型原生 model_type 是 deepseek_v3_bidirec，vLLM 不识别，需要 serve 时改写 model_type 并 trust-remote-code，属于外围配置依赖，不是模型内部自包含支持。 - 影响：对用户：新增一类可用的 DeepSeek 骨干 embedding 模型，可通过 --runner pooling --convert embed 提供服务，但需要 --hf-overrides 改写 model_type。对现有 DeepSeek decoder-only 用户无影响，因为 is_causal 缺省为 True 时行为与之前完全一致。对系统：双向 DeepSeek 走非 MLA 路径，相关层的 kernel、显存与调度行为与 MLA 不同，是新的运行组合。对团队：改动集中在模型注册与注意力构造，代码量小、模式清晰，为后续 encoder-only/embedding 变体提供了可参考的落地范式。 - 风险标记：核心注意力路径变更，全局配置分支，运行时数值测试未进 CI，依赖 hf-overrides 变通

关联脉络

- PR #52929 Add NemotronH_Omni_Reasoning_V3 as a supported Nemotron architecture: 同样修改 vllm/model_executor/models/registry.py 与 tests/models/registry.py 注册新架构，展示了新增模型架构的标准流程，与本 PR 的注册表改动属于同一模式。
- PR #52690 [Bugfix] Restore model info caching for package backends: 同仓库近期对 registry.py 与模型注册机制的维护，本 PR 在注册表中新增嵌入模型条目需要与该机制保持一致。