# PR #41277 完整报告

vllm-project/vllm

Fix error in Dynamic NTK scaling

合并时间：2026-05-20 05:27

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/41277

## 执行摘要

- 一句话：修复 Dynamic NTK RoPE 缩放公式为变量而非常量
- 推荐动作：建议合并。核心贡献是修正了一个长期存在的公式错误，并使 Nomic 模型的默认行为与社区对齐。推荐阅读 dynamic_ntk_scaling_rope.py 中的核心公式修正和 config.py 中简化后的配置逻辑。开发者在升级后应检查自家 Nomic 模型是否通过 rope_parameters 自定义了缩放参数。

## 功能与动机

Issue #41236 指出 Dynamic NTK 缩放公式错误：max_len = self.max_position_embeddings * self.scaling_factor 代入后变成常量，失去了长度依赖。PR body 引用社区公式 (α * current length / original length) - (α - 1)，要求根据实际序列长度动态缩放。此外，Nomic 模型需要正确支持从训练长度向 n_positions 的扩展（如 2048→8192）。

## 实现拆解

1. 修正核心公式：修改 dynamic_ntk_scaling_rope.py 中的 _compute_cos_sin_cache 方法，删除错误的最大长度 max_len，直接使用 self.max_position_embeddings 作为分母，用 max_trained_positions 参数替代原来的常量缩放分子，使 base 计算依赖 self.max_position_embeddings / self.max_trained_positions，变量实际来自 self.max_position_embeddings。
2. 扩展参数传递：在 __init__.py 的 get_rope 构建函数中，从 rope_parameters 读取 max_trained_positions（默认值为 max_position），并传递给 DynamicNTKScalingRotaryEmbedding 构造函数，同时添加 max_trained_positions 属性。
3. 简化配置逻辑：在 config.py 的 NomicBertModel 配置类中，删除原有复杂的上下文扩展分支（约 70 行替换为 10 行），不再手动修改 max_model_len 限制到训练长度，而是直接设置 rotary_kwargs 中的 max_position 为 model_config.max_model_len，并将 max_trained_positions 注入 rope_parameters，同时修复 config.rope_parameters 可能为 None 的潜在异常。
4. 更新测试覆盖：调整 test_nomic_max_model_len.py，删除 test_set_max_model_len_illegal 和 test_use_rope_scaling_illegal 测试（因新逻辑不再主动限制长度），修改 test_default 断言 nomic-embed-text-v1 默认 max_model_len 为 8192，并扩展 test_set_max_model_len_legal 支持设置超过 2048 的长度（如 4096）。

关键文件：

- vllm/model_executor/layers/rotary_embedding/dynamic_ntk_scaling_rope.py（模块 旋转嵌入；类别 source；类型 core-logic；符号 DynamicNTKScalingRotaryEmbedding.init, DynamicNTKScalingRotaryEmbedding._compute_cos_sin_cache）：核心修复所在：修改 _compute_cos_sin_cache 方法，修正 base 计算公式，引入 max_trained_positions 参数。
- vllm/model_executor/models/config.py（模块 配置处理；类别 source；类型 data-contract；符号 NomicBertModel.verify_and_update_model_config）：重构 Nomic 模型配置，简化 context extension 逻辑，移除约 70 行条件分支，改为直接传递 max_trained_positions 和 max_model_len。
- vllm/model_executor/layers/rotary_embedding/__init__.py（模块 旋转嵌入；类别 source；类型 core-logic；符号 get_rope）：构建 DynamicNTKScalingRotaryEmbedding 时传递 max_trained_positions 参数，从 rope_parameters 获取。
- tests/models/language/pooling/test_nomic_max_model_len.py（模块 测试；类别 test；类型 test-coverage；符号 test_default, test_set_max_model_len_legal, test_use_rope_scaling_legal）：更新测试以匹配新的默认长度和行为，删除非法长度测试。

关键符号：DynamicNTKScalingRotaryEmbedding.init, DynamicNTKScalingRotaryEmbedding._compute_cos_sin_cache, NomicBertModel.verify_and_update_model_config, get_rope

## 关键源码片段

### vllm/model_executor/layers/rotary_embedding/dynamic_ntk_scaling_rope.py

核心修复所在：修改 _compute_cos_sin_cache 方法，修正 base 计算公式，引入 max_trained_positions 参数。

```
# vllm/model_executor/layers/rotary_embedding/dynamic_ntk_scaling_rope.py

class DynamicNTKScalingRotaryEmbedding(RotaryEmbedding):
    """RotaryEmbedding extended with Dynamic NTK scaling."""

    def __init__(
        self,
        head_size: int,
        rotary_dim: int,
        max_position_embeddings: int,
        max_trained_positions: int, # <-- 新增参数：模型训练的原始长度
        base: float,
        is_neox_style: bool,
        scaling_factor: float,
        dtype: torch.dtype,
    ) -> None:
        self.scaling_factor = scaling_factor
        self.max_trained_positions = max_trained_positions # 保存训练长度
        super().__init__(
```

```
            head_size, rotary_dim, max_position_embeddings, base, is_neox_style, dtype
        )

    def _compute_cos_sin_cache(self) -> torch.Tensor:
        # 正确公式: base' = base * (( α * cur_len / trained_len ) - (α - 1))^(dim/(dim-2))
        # 其中 cur_len = self.max_position_embeddings # 当前推理长度
        # 之前的错误: 使用了 max_len = cur_len * α, 导致公式变成常量
        base = self.base * (
            (
                self.scaling_factor
                * self.max_position_embeddings
                / self.max_trained_positions
            )
            - (self.scaling_factor - 1)
        ) ** (self.rotary_dim / (self.rotary_dim - 2))
        inv_freq = self._compute_inv_freq(base)
        # 注意: t 的范围基于当前 max_position_embeddings, 而非缩放后的长度
        t = torch.arange(self.max_position_embeddings, dtype=torch.float)
        freqs = torch.einsum("i,j -> ij", t, inv_freq)
        cos = freqs.cos()
        sin = freqs.sin()
        cache = torch.cat((cos, sin), dim=-1)
        return cache
```

## vllm/model_executor/models/config.py

重构 Nomic 模型配置，简化 context extension 逻辑，移除约 70 行条件分支，改为直接传递 max_trained_positions 和 max_model_len。

```
# vllm/model_executor/models/config.py
# NomicBertModel 的 verify_and_update_model_config 新实现（关键片段）

@staticmethod
def verify_and_update_model_config(model_config: "ModelConfig") -> None:
    config = model_config.hf_config
    # ... 前面的断言和属性映射省略 ...

    head_dim = config.hidden_size // config.num_attention_heads
    # 优先取 max_position_embeddings, 兜底 2048
    max_position_embeddings = getattr(config, "max_position_embeddings", 2048)
    max_trained_positions = getattr(
        config, "max_trained_positions", max_position_embeddings
    )

    # 将 max_trained_positions 注入 rope_parameters, 供后续构建使用
    rope_parameters = {
        "max_trained_positions": max_trained_positions,
        **(config.rope_parameters or {}), # 安全解包, 防止 None
    }
```

```
# rotary_kwargs 直接使用 model_config.max_model_len（用户可自由设置）
config.rotary_kwargs = {
    "head_size": head_dim,
    "max_position": model_config.max_model_len,
    "rope_parameters": rope_parameters,
}
# 不再覆盖 max_model_len，删除警告，简化逻辑
```

## 评论区精华

1. 默认值安全性：gemini-code-assist[bot] 提出 __init__.py 中 max_trained_positions 默认取 max_position 会导致缩放比率为 1.0，对未显式提供该参数的模型禁用动态缩放，可能构成回归。作者 maxdebayser 回应这是有意为之，因为如果没有上下文扩展（序列长度不大于训练长度），不需要动态缩放，且该默认不会破坏原有行为。最终未修改默认值。
2. None 解包风险：gemini-code-assist[bot] 指出 config.rope_parameters 可能为 None，导致 ** 解包 TypeError。该建议被采纳，最终代码改为 (config.rope_parameters or {})。
3. 对齐验证：noooop 请求核心维护者 double-check 修改，并建议更新模型代码版本以对齐最新 transformers。同时 ieBoytsov 建议增加 embedding 相似度测试，确认与 sentence-transformers 一致。最终 mgoin 表示认可修改并批准合并。

- 默认 max_trained_positions 为 max_position 的影响 (correctness): 保持默认值不变，无修改。
- rope_parameters 可能为 None 导致 TypeError (correctness): 建议被采纳，最终使用 (config.rope_parameters or {})。
- 请求核心维护者 double-check 公式修改 (design): mgoin 表示希望合入，认为改动合理。
- 增加 embedding 数值比对测试 (testing): 未在本次 PR 中实现，但存在其他测试（如 test_embeddings）可能部分覆盖，未被强制要求。

## 风险与影响

- 风险：
  1. 兼容性风险：未显式设置 max_trained_positions 的模型默认使用 max_position，缩放比率为 1.0，这符合没有上下文扩展的场景，不会引起数值错误。但对于已通过自定义 rope_parameters 依赖原错误的模型，升级后行为可能发生非兼容变化。需要用户确认配置。
  2. 测试覆盖不足：虽然测试已更新，但未增加直接验证 embedding 数值一致性的端到端测试（如与 sentence-transformers 比较），仅验证了 max_model_len 配置。潜在的实际推理输出差异可能未被捕获。
  3. 简化影响：config.py 中删除了大量上下文扩展的警告和条件逻辑，对依赖原有 max_model_len 自动裁剪的用户可能带来意外改变。但新逻辑更简单透明，通过配置明确启用扩展。 - 影响：影响范围：直接修正 Nomic 系列 Embedding 模型（如 nomic-embed-text-v1, v1.5, CodeRankEmbed, Snowflake/snowflake-arctic-embed-m-long）的动态 NTK 缩放行为，使 vLLM 可与 sentence-transformers 对齐。影响程度：中等。对使用这些模型且需要上下文扩展的用

户是 bugfix；其他用户无影响。团队影响：简化了 config.py 中历史遗留逻辑，降低维护成本。 - 风险标记：配置行为变化，核心路径变更，缺少端到端数值验证

## 关联脉络

- 暂无明显关联 PR