PR #2377 完整报告

radixark/miles

fix(kimi): align YaRN parameters with checkpoints

合并时间：2026-08-12 05:03

原文链接：http://prhub.com.cn/radixark/miles/pull/2377

执行摘要

- 一句话：修复 Kimi 系列 YaRN 参数与 checkpoint 配置不一致
- 推荐动作：值得精读。该 PR 展示了如何在不修改 Megatron 全局默认值的前提下，通过 per-model 包装器解决配置边界不一致问题，并配套了从 CLI 参数到最终 TransformerConfig 的端到端回归测试。关注点：load_sibling_model_args 的透传机制、K2.5 包装器的设计，以及测试中如何隔离 TE 依赖。

功能与动机

原始 K2 和 K2-Thinking 训练使用了 beta_fast=32，而其 HF rollout 配置要求 beta_fast=1；K2.5 正确要求 32。根因是 kimi-k2*.model_args 遗漏了 checkpoint 特定的 YaRN 字段，导致 MLATransformerConfig.beta_fast 回落到 Megatron 全局默认值 32。该问题由社区成员 zyzshishui 和 Robert Li 提出，需要修复模型配置边界以匹配已发布 checkpoint。

实现拆解

1. 补充 YaRN 参数元组：在 scripts/models/kimi-k2.py 和 scripts/models/kimi-k2-thinking.py 的 model_args 中显式声明 --rope-type yarn、--original-max-position-embeddings 4096、--beta-fast 和 --beta-slow 1。其中 kimi-k2-thinking 增加 beta_fast 参数（默认 1），使 K2 与 K2-Thinking 训练时得到正确的 beta_fast=1。
2. 新增 K2.5 包装器：新增 scripts/models/kimi-k25.py，复用 kimi-k2-thinking 的完整配置但覆盖 beta_fast=32，避免修改 Megatron 全局默认值；同时将 scripts/models/kimi-k25_2layer.py 的引用从 kimi-k2-thinking 改为 kimi-k25，保证 K2.5 全量和 2 层变体都获得 beta_fast=32。
3. 调整启动脚本：scripts/run_kimi_k25.py 和 scripts/run-kimi-k25.sh 中的 megatron_model_type 从 kimi-k2-thinking 改为 kimi-k25，使 K2.5 训练走专用配置；examples/lora/run-kimi-k25-megatron-lora.sh 同步更新。
4. 新增回归测试：在 tests/fast/test_megatron_cli_flags.py 中新增参数化测试 test_kimi_yarn_flags_propagate_to_megatron，通过 expand_model_args 展开四种模型配置，经 parse_args 和 core_transformer_config_from_args 验证最终 MLATransformerConfig 的 rope_type、original_max_position_embeddings、beta_fast、beta_slow 是否符合期望。第二个提交专门在测试中禁用 moe_permute_fusion，以兼容未安装 Transformer Engine 的 CPU CI。

5. 同步快照与文档：更新 tests/snapshots/model_args/ 下 K2、K2-Thinking、K2.5、K2.5-2layer 四个参数快照，以及多个 launch script 快照（.sh 和 .py）；docs/models/kimi/kimi-k2.5.md 同步说明。

关键文件：

- scripts/models/kimi-k25.py（模块 模型配置；类别 source；类型 data-contract；符号 model_args）：新增的 K2.5 专用包装器，复用 K2-Thinking 配置并覆盖 beta_fast=32，是修复的核心入口。
- scripts/models/kimi-k2-thinking.py（模块 模型配置；类别 source；类型 data-contract；符号 model_args）：K2 与 K2-Thinking 共用的基础配置，补全 YaRN 字段并新增 beta_fast 参数，是修复的关键数据契约变更。
- scripts/models/kimi-k2.py（模块 模型配置；类别 source；类型 data-contract）：K2 基础配置，补充 YaRN 字段使 beta_fast 固定为 1。
- scripts/models/kimi-k25_2layer.py（模块 模型配置；类别 source；类型 data-contract）：K2.5 2 层变体，引用从 kimi-k2-thinking 改为 kimi-k25，确保继承 beta_fast=32。
- scripts/run_kimi_k25.py（模块 启动脚本；类别 source；类型 core-logic；符号 ScriptArgs.post_init）：K2.5 训练入口脚本，megatron_model_type 从 kimi-k2-thinking 改为专用 kimi-k25，使 K2.5 训练使用正确的 YaRN 配置。
- tests/fast/test_megatron_cli_flags.py（模块 测试；类别 test；类型 test-coverage；符号 test_kimi_yarn_flags_propagate_to_megatron）：新增参数化回归测试，验证四种 Kimi 变体的 YaRN 配置最终正确传播到 MLATransformerConfig，是本次修复的质量保障。
- tests/snapshots/model_args/kimi-k25.txt（模块 快照；类别 docs；类型 documentation）：新增 K2.5 模型参数快照，固定了预期的完整参数序列，供手动快照测试校验。
- tests/snapshots/model_args/kimi-k2.txt（模块 快照；类别 docs；类型 documentation）：K2 参数快照同步新增 YaRN 字段，确保快照与源码一致。

关键符号：model_args, test_kimi_yarn_flags_propagate_to_megatron, ScriptArgs.post_init, load_sibling_model_args

关键源码片段

scripts/models/kimi-k25.py

新增的 K2.5 专用包装器，复用 K2-Thinking 配置并覆盖 beta_fast=32，是修复的核心入口。

```
from model_args_utils import load_sibling_model_args


def model_args(nlayers: int = 61, first_k_dense_replace: int = 1) -> str:
    # K2.5 与 K2-Thinking 架构相同，但已发布 checkpoint 要求 beta_fast=32
    #  （K2 / K2-Thinking 要求 1），因此复用兄弟配置并覆盖该参数。
    return load_sibling_model_args(
        __file__,
        "kimi-k2-thinking",
        nlayers=nlayers,
        first_k_dense_replace=first_k_dense_replace,
```

```
        beta_fast=32,
    )
```

scripts/models/kimi-k2-thinking.py

K2 与 K2-Thinking 共用的基础配置，补全 YaRN 字段并新增 beta_fast 参数，是修复的关键数据契约变更。

```
from model_args_utils import moe_layer_freq


def model_args(nlayers: int = 61, first_k_dense_replace: int = 1, beta_fast: int = 1) -> str:
    return (
        "--disable-bias-linear "
        f"--num-layers {nlayers} "
        "--hidden-size 7168 "
        "--ffn-hidden-size 18432 "
        "--num-attention-heads 64 "
        "--kv-channels 64 "
        "--normalization RMSNorm "
        "--position-embedding-type rope "
        "--rope-type yarn "
        "--norm-epsilon 1e-5 "
        "--swiglu "
        "--untie-embeddings-and-output-weights "
        "--vocab-size 163840 "
        "--multi-latent-attention "
        "--q-lora-rank 1536 "
        "--kv-lora-rank 512 "
        "--qk-head-dim 128 "
        "--qk-pos-emb-head-dim 64 "
        "--v-head-dim 128 "
        "--qk-layernorm "
        "--rotary-scaling-factor 64.0 "
        "--rotary-base 50000 "
        # 以下三行为本次新增的 YaRN 参数，与 checkpoint 配置对齐
        "--original-max-position-embeddings 4096 "
        f"--beta-fast {beta_fast} "
        "--beta-slow 1 "
        "--mscale 1.0 "
        "--mscale-all-dim 1.0 "
        "--attention-softmax-in-fp32 "
        "--no-rope-fusion "
        # moe
        "--num-experts 384 "
        f"--moe-layer-freq {moe_layer_freq(nlayers=nlayers, first_k_dense_replace=first_k_dense_
        replace)} "
        "--moe-ffn-hidden-size 2048 "
        "--moe-router-topk 8 "
        "--moe-shared-expert-intermediate-size 2048 "
```

```
        "--moe-router-pre-softmax "
        "--moe-router-score-function sigmoid "
        "--moe-router-enable-expert-bias "
        "--moe-router-load-balancing-type seq_aux_loss "
        "--moe-token-dispatcher-type alltoall "
        "--moe-aux-loss-coeff 0 "
        "--moe-router-bias-update-rate 0 "
        "--moe-router-group-topk 1 "
        "--moe-router-num-groups 1 "
        "--moe-grouped-gemm "
        "--moe-router-topk-scaling-factor 2.827 "
        "--moe-router-dtype fp32 "
        "--moe-permute-fusion "
    )
```

tests/fast/test_megatron_cli_flags.py

新增参数化回归测试，验证四种 Kimi 变体的 YaRN 配置最终正确传播到 MLATransformerConfig，是本次修复的质量保障。

```
@pytest.mark.parametrize(
    ("model_type", "beta_fast"),
    [("kimi-k2", 1), ("kimi-k2-thinking", 1), ("kimi-k25", 32), ("kimi-k25_2layer", 32)],
)
def test_kimi_yarn_flags_propagate_to_megatron(monkeypatch, model_type, beta_fast):
    pytest.importorskip("megatron.training.arguments")

    import torch
    from megatron.core.transformer.transformer_config import MLATransformerConfig
    from megatron.training.arguments import core_transformer_config_from_args

    import miles.backends.megatron_utils.arguments as megatron_arguments
    import miles.utils.arguments as miles_arguments

    monkeypatch.setattr(miles_arguments, "miles_validate_args", lambda args: None)
    monkeypatch.setattr(megatron_arguments, "validate_args", lambda args: None)
    monkeypatch.setattr(
        sys,
        "argv",
        ["pytest", "--train-backend", "megatron", "--rollout-batch-size", "1", *expand_model_
        args(model_type)],
    )

    args = miles_arguments.parse_args()
    if args.bf16:
        args.params_dtype = torch.bfloat16
    elif args.fp16:
        args.params_dtype = torch.float16
    else:
        args.params_dtype = torch.float32
```

```
    # Fused MoE permutation requires Transformer Engine, which is not installed
    # on CPU CI and is unrelated to the YaRN configuration under test.
    args.moe_permute_fusion = False

    config = core_transformer_config_from_args(args)

    assert isinstance(config, MLATransformerConfig)
    assert config.rope_type == "yarn"
    assert config.original_max_position_embeddings == 4096
    assert config.beta_fast == beta_fast
    assert config.beta_slow == 1
```

评论区精华

该 PR 没有 review 评论，reviewer yueming-yuan 直接 APPROVED。Issue 评论中作者 guapisolo 特别感谢了 @zyzshishui 和 Robert Li 提出此问题。因此没有实质性的设计争论，但 PR body 中明确给出了 Review Focus：K2/K2-Thinking 必须保持 yarn/4096/1/1，K2.5 全量和 2 层变体必须保持 yarn/4096/32/1。

- 暂无高价值评论线程

风险与影响

- 风险：主要风险集中在模型配置边界：修改了 scripts/models/kimi-k2*.py 和 scripts/run_kimi_k25.py 的配置来源，可能影响所有使用这些 model_args 的启动脚本（如 run-kimi-k2-Thinking.sh、run-kimi-k2-Instruct.sh 等）。但由于新增的回归测试覆盖四种变体并断言最终 MLATransformerConfig，且快照测试（manual）已验证 135 项模型参数和 18 项启动器检查，回归风险可控。需注意：kimi-k25.py 依赖 load_sibling_model_args 的 beta_fast 透传能力，若该工具函数未同步支持，会引入配置缺失；但测试已验证。此外，测试在 CPU 上运行，禁用了 moe_permute_fusion，这不影响配置验证的正确性。
- 影响：影响范围：所有使用 Kimi K2、K2-Thinking、K2.5（含 2 层变体）的 Megatron 训练 / 微调启动脚本。修复后训练时的 YaRN 旋转位置编码参数与 HF rollout 配置一致，避免因 beta_fast 不一致导致的训练不稳定或结果偏差。对系统而言，无全局默认值改动，仅模型配置层修正，影响面集中在 Kimi 模型家族。对团队而言，新增的回归测试为后续配置变更提供了安全网。
- 风险标记：模型配置边界变更，快照同步更新，测试依赖 CPU 可用性

关联脉络

- PR #1910 Replace the model config shell scripts with python: 引入 model_args_utils 和 Python 模型配置脚本模式，本 PR 的 load_sibling_model_args 和 scripts/models 结构依赖此前的重构。
- PR #1909 Expand the model args in python before building the command: 模型参数改为 Python 预展开，本 PR 的快照和启动脚本依赖这一机制。

- PR #1901 Snapshot the commands and generated configs of every python launch script: 快照测试体系覆盖了本 PR 修改的 kimi 启动脚本，本 PR 同步更新了多个快照文件。