PR #1928 完整报告

THUDM/slime

fix: avoid applying rollout temperature to critic values

合并时间：2026-05-30 12:09

原文链接：http://prhub.com.cn/THUDM/slime/pull/1928

执行摘要

- 一句话：修复 critic 值头被错误应用 rollout 温度缩放
- 推荐动作：该 PR 是一个轻量的值得注意的 bug 修复，设计清晰（添加可选参数控制温度），适合作为 code review 中“小改解决大问题”的范例。建议合并并关注相关训练结果。

功能与动机

get_responses() 函数被策略 logit 提取和值 head 提取共用。策略 logit 需要温度缩放以重建 rollout logprob，但 critic 的值输出是标量预测，不应除以 rollout 采样温度。原始实现无条件应用温度缩放，导致 critic 值被错误缩放，影响 PPO 等算法的 value loss 计算。

实现拆解

1. 修改 get_responses 函数签名：在 slime/backends/megatron_utils/loss.py 中为 get_responses 新增 apply_temperature: bool = True 参数，保持向后兼容。
2. 添加条件判断：将 if args.rollout_temperature != 1.0 改为 if apply_temperature and args.rollout_temperature != 1.0，确保温度缩放仅在被允许时应用。
3. 更新文档字符串：调整函数和参数说明，反映 rollout_temperature 的可选性。
4. 修改 get_values 调用：在 get_values 内部调用 get_responses 时传递 apply_temperature=False，从而禁用温度缩放。同时更新 get_values 的文档字符串，说明温度缩放已禁用。
5. 新增单元测试：创建 tests/test_value_temperature.py，在零 GPU 环境下使用 monkeypatch 模拟依赖，验证当 rollout_temperature=0.5 时，get_values 输出的值头结果 [2.0, 3.0] 与原始 logits 对应位置一致，证明未应用缩放。

关键文件：

- slime/backends/megatron_utils/loss.py（模块 后端核心；类别 source；类型 core-logic；符号 get_responses, get_values）：核心逻辑变更：get_responses 新增 apply_temperature 参数，get_values 调用时禁用温度缩放。
- tests/test_value_temperature.py（模块 值温度；类别 test；类型 test-coverage；符号 test_get_values_does_not_apply_rollout_temperature）：新增零 GPU 单元测试，验证 get_values 不应用 rollout_temperature。

关键符号：get_responses, get_values

关键源码片段

slime/backends/megatron_utils/loss.py

核心逻辑变更：get_responses 新增 apply_temperature 参数，get_values 调用时禁用温度缩放。

```
# slime/backends/megatron_utils/loss.py
def get_responses(
    logits: torch.Tensor,
    *,
    args: Namespace,
    unconcat_tokens: list[torch.Tensor],
    total_lengths: list[int],
    response_lengths: list[int],
    max_seq_lens: list[int] | None = None,
    apply_temperature: bool = True, # 新增参数，默认开启温度缩放（保持向后兼容）
) -> Iterator[tuple[torch.Tensor, torch.Tensor]]:
    ...
    # 只有 apply_temperature 为 True 且温度不为 1.0 时才缩放
    if apply_temperature and args.rollout_temperature != 1.0:
        logits = logits.div(args.rollout_temperature)
    ...

def get_values(
    logits: torch.Tensor,
    *,
    args: Namespace,
    unconcat_tokens: list[torch.Tensor],
    total_lengths: list[int],
    response_lengths: list[int],
    with_entropy: bool = False,
    non_loss_data: bool = True,
    max_seq_lens: list[int] | None = None,
) -> dict[str, list[torch.Tensor]]:
    ...
    # 调用 get_responses 时显式禁用温度缩放，因为值输出不应被缩放
    for logits_chunk, tokens_chunk in get_responses(
        logits,
        args=args,
        unconcat_tokens=unconcat_tokens,
        total_lengths=total_lengths,
        response_lengths=response_lengths,
        max_seq_lens=max_seq_lens,
        apply_temperature=False,
    ):
        ...
```

tests/test_value_temperature.py

新增零 GPU 单元测试，验证 get_values 不应用 rollout_temperature。

```
# tests/test_value_temperature.py
import sys
import types
from argparse import Namespace
import pytest
import torch

NUM_GPUS = 0 # 零 GPU 测试，无需硬件

def test_get_values_does_not_apply_rollout_temperature(monkeypatch):
    # 模拟 megatron 模块依赖
    previous_loss = sys.modules.pop("slime.backends.megatron_utils.loss", None)
    previous_cp_utils = sys.modules.pop("slime.backends.megatron_utils.cp_utils", None)
    mpu_stub = types.SimpleNamespace(
        get_context_parallel_world_size=lambda: 1,
        get_context_parallel_rank=lambda: 0,
    )
    megatron_mod = types.ModuleType("megatron")
    core_mod = types.ModuleType("megatron.core")
    core_mod.mpu = mpu_stub
    monkeypatch.setitem(sys.modules, "megatron", megatron_mod)
    monkeypatch.setitem(sys.modules, "megatron.core", core_mod)
    try:
        from slime.backends.megatron_utils.loss import get_values
        args = Namespace(qkv_format="thd", rollout_temperature=0.5, allgather_cp=False)
        logits = torch.tensor([[[1.0], [2.0], [3.0], [4.0]]], dtype=torch.float32)
        tokens = [torch.tensor([10, 11, 12, 13], dtype=torch.long)]
        _, result = get_values(
            logits,
            args=args,
            unconcat_tokens=tokens,
            total_lengths=[4],
            response_lengths=[2],
        )
        # 断言值输出未被缩放，即 [2.0, 3.0] 而非 [4.0, 6.0]
        torch.testing.assert_close(result["values"][0], torch.tensor([2.0, 3.0]))
    finally:
        # 恢复模块状态
        if previous_loss is None:
            sys.modules.pop("slime.backends.megatron_utils.loss", None)
        else:
            sys.modules["slime.backends.megatron_utils.loss"] = previous_loss
        if previous_cp_utils is None:
            sys.modules.pop("slime.backends.megatron_utils.cp_utils", None)
        else:
            sys.modules["slime.backends.megatron_utils.cp_utils"] = previous_cp_utils
```

评论区精华

合著者 zhuzilin 评论 "nice catch!"，表明该 bug 的发现和修复受到认可，未产生争议。

- bug 修复确认 (correctness): BUG 修复被认可，无其他讨论。

风险与影响

- 风险：变更较小且语义明确，风险低。但需注意：所有调用 get_responses 的代码路径（包括策略 logprob 计算等）默认保持 apply_temperature=True，行为不变。get_values 是唯一需要传递 False 的显式调用，已在当前 PR 中覆盖。如果未来新增其他调用 get_responses 的路径，需注意温度设置。
- 影响：影响范围：影响 critic value 头输出，从而影响 PPO/GRPO 等算法中的 value loss 和优势函数计算。修复后 value 输出更准确，训练更稳定。零 GPU 测试确保 CI 中不会因缺少 GPU 而跳过验证。
- 风险标记：低风险变更，新增测试覆盖

关联脉络

- PR #1950 fix: drop incorrect critic GPU add to rollout_num_gpus in colocate mode: 同样涉及 critic 相关配置的修复，体现了对 critic/ 值头逻辑的持续完善。