PR #1897 完整报告

THUDM/slime

Migrate internal feature

合并时间：2026-05-09 15:21

原文链接：http://prhub.com.cn/THUDM/slime/pull/1897

执行摘要

- 一句话：新增训练进度条和 on-policy 优化
- 推荐动作：值得精读，特别是 on-policy 跳过 logprob 的条件设计，展示了如何在保证正确性的前提下进行性能优化。建议添加单元测试覆盖关键条件组合。

功能与动机

PR body 明确说明："training tqdm - don't compute logprob when doing on policy training"。一方面是提升训练过程中的可观测性，让用户能直观看到微批次的处理进度；另一方面是在 on-policy 场景下避免重复计算 logprob，减少不必要的计算开销。

实现拆解

1. 新增进度条支持（model.py）：在 slime/backends/megatron_utils/model.py 中，新增了三个辅助函数：_disable_tqdm_for_non_main_rank() 用于判断当前 rank 是否需要显示进度条（仅 data_parallel_rank=0、tensor_model_parallel_rank=0 且 pipeline_model_parallel_rank 为最后一阶段时显示）；_should_update_microbatch_pbar(model) 负责判断是否应更新进度条（考虑虚拟流水线阶段）；_wrap_forward_step_with_microbatch_pbar(forward_step_func, pbar) 将前向步骤函数包装，在每次执行后更新进度条。在 forward_step 函数中创建 tqdm 实例，将总微批次数量作为 total，并传递给 forward_backward_func；训练完成后关闭进度条。
2. 优化 on-policy logprob 计算（actor.py）：在 slime/backends/megatron_utils/actor.py 的 train_actor 方法中，新增变量 can_reuse_log_probs_in_loss，其值为一系列条件的与运算：只有单个微批次、loss_type 为 policy_loss、KL 系数为 0、不使用 rollout_logprobs、不获取 mismatch 指标、不使用 critic、不 keep_old_actor、不使用 OPD、不使用 routing replay、且 advantage_estimator 不是 gspo。当这些条件全部满足时，跳过额外的 compute_log_prob 调用，因为损失函数可以直接复用前向传播中计算的 logits。
3. 修正 loss 函数中的 logprob 获取逻辑（loss.py）：在 slime/backends/megatron_utils/loss.py 中，修改 compute_advantages_and_returns 中 log_probs 的获取方式，先获取 rollout_log_probs 再按条件决定使用哪个；同时修正了 KL 计算时 xs 的 fallback 链：log_probs or rollout_log_probs or values。policy_loss_function 中也将 old_log_probs 的获取改为 batch.get("log_probs")（带默认值 None），并在 use_rollout_logprobs=False 且 old_log_probs 为 None 时，通过 .

detach() 从当前 logits 复制一份，同时处理 TIS 所需的 train_log_probs_for_tis。

关键文件：

- slime/backends/megatron_utils/model.py（模块 Megatron 后端；类别 source；类型 data-contract；符号 _disable_tqdm_for_non_main_rank, _should_update_microbatch_pbar, _wrap_forward_step_with_microbatch_pbar, wrapped_forward_step）：核心变更文件，新增进度条相关辅助函数，并在 forward_step 中集成 tqdm
- slime/backends/megatron_utils/actor.py（模块 运行器；类别 source；类型 core-logic）：在 train_actor 方法中实现 on-policy logprob 跳过逻辑
- slime/backends/megatron_utils/loss.py（模块 损失函数；类别 source；类型 core-logic）：修改 logprob 获取逻辑以支持可选 log_probs，并添加 fallback 机制

关键符号：_disable_tqdm_for_non_main_rank, _should_update_microbatch_pbar, _wrap_forward_step_with_microbatch_pbar, wrapped_forward_step, compute_advantages_and_returns, policy_loss_function

关键源码片段

slime/backends/megatron_utils/model.py

核心变更文件，新增进度条相关辅助函数，并在 forward_step 中集成 tqdm

```
# slime/backends/megatron_utils/model.py

def _disable_tqdm_for_non_main_rank() -> bool:
    # 只有 data_parallel_rank=0、tensor_parallel_rank=0 且
    # pipeline_parallel_rank 为最后一阶段时才显示进度条
    return not (
        mpu.get_data_parallel_rank(with_context_parallel=True) == 0
        and mpu.get_tensor_model_parallel_rank() == 0
        and mpu.get_pipeline_model_parallel_rank() == mpu.get_pipeline_model_parallel_world_
        size() - 1
    )

def _should_update_microbatch_pbar(model) -> bool:
    if _disable_tqdm_for_non_main_rank():
        return False
    # 解包可能的 DDP 包装
    while hasattr(model, "module"):
        model = model.module
    vp_stage = getattr(model, "vp_stage", None)
    if mpu.get_virtual_pipeline_model_parallel_world_size() is not None and vp_stage is not None:
        return mpu.is_pipeline_last_stage(ignore_virtual=False, vp_stage=vp_stage)
    return mpu.is_pipeline_last_stage(ignore_virtual=True)

def _wrap_forward_step_with_microbatch_pbar(forward_step_func, pbar):
    if pbar is None:
        return forward_step_func
```

```
    def wrapped_forward_step(*args, **kwargs):
        result = forward_step_func(*args, **kwargs)
        # args[1] 即为 model 参数
        model = args[1] if len(args) > 1 else kwargs.get("model")
        if model is not None and _should_update_microbatch_pbar(model):
            pbar.update(1)
        return result
    return wrapped_forward_step
```

评论区精华

该 PR 没有 review 评论，但实现本身包含清晰的设计权衡：进度条只在主 rank 显示，避免分布式环境下输出混乱；logprob 跳过条件非常严格，确保在正确性不受影响的前提下优化性能。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 进度条包装可能引入细微的语义变化：forward_backward_func 现在接收的是包装后的函数，如果内部有异常处理或状态重置，可能受影响。
  2. on-policy 跳过 logprob 的条件链很脆弱：新增任何一个参数或变更现有条件都可能导致优化失效或更严重地错误跳过必要计算。
  3. loss.py 中将 batch["log_probs"] 改为 batch.get("log_probs")，当 log_probs 不存在时会返回 None，后续逻辑依赖于这一行为。 - 影响：影响范围集中在 Megatron 后端的训练流程。进度条功能无性能开销（不在主 rank 时 disable）；on-policy 优化仅影响特定配置（单微批次、纯策略损失等）下的训练流程，可减少一次完整前向传播。 - 风险标记：核心路径变更，缺少测试覆盖，条件链脆弱

关联脉络

- PR #1883 fix(qwen3_next): use torch.get_default_dtype() — get_current_dtype do…: 同样涉及 Megatron 后端的 loss 函数修复
- PR #1866 Rename critic config to megatron config: 重构了 actor-critic 配置，与本 PR 的 actor.py 修改属于同一模块
- PR #1856 refactor/ppo: PPO 训练架构重构，本 PR 的 actor 优化建立在该重构基础上