PR #6323 完整报告

verl-project/verl

[veomni] feat: add veomni qwen3-30b and fix ep

合并时间：2026-05-19 16:29

原文链接：http://prhub.com.cn/verl-project/verl/pull/6323

执行摘要

- 一句话：新增 Qwen3-30B VeOmni 训练脚本并修复 EP 参数获取 bug
- 推荐动作：该 PR 值得单读以了解 VeOmni 后端训练脚本的标准结构和 EP 参数修复方式。特别注意 extra_parallel_sizes 初始化的一致性问题，建议在其相关 PR (#6346 等) 中确认修复。

功能与动机

PR body 指出需要为 VeOmni 后端添加 Qwen3-30B 和 Qwen3-1.7B 的 NPU 训练脚本，并修复 EP 参数访问 bug，确保专家并行在 VeOmni 引擎下正常工作。同时响应 Review 意见，将脚本从 ascend_extras 移至公共目录，合并 GPU/NPU 配置。

实现拆解

1. 创建初始脚本：在 examples/ascend_extras/grpo_trainer/ 下为 NPU 创建 run_qwen3-30b_veomni_npu.sh 和 run_qwen3_1_7b_npu.sh。
2. 根据 Review 重构：删除小模型脚本（避免为每个模型保留），将 30B 脚本移至 examples/grpo_trainer/ 并重命名为 run_qwen3_30b_veomni.sh。
3. 脚本完善：添加 DEVICE 自动检测逻辑（若可导入 torch_npu 则为 npu，否则为 gpu），在 NPU 分支中设置 HCCL、VLLM 等环境变量，并在头部注释说明算法、硬件、推理后端。
4. 修复 EP 参数：在 verl/workers/engine/veomni/transformer_impl.py 的 get_per_tensor_param 方法中，将 Expert Parallel 的 gather 维度从 ps.ep_size 改为 ps.extra_parallel_sizes['ep']，以匹配新的并行状态字典结构。
5. 配套调整：未新增测试，因功能依赖特定硬件环境。

关键文件：

- examples/grpo_trainer/run_qwen3_30b_veomni.sh（模块 训练脚本；类别 other；类型 core-logic）：新增的 VeOmni 训练脚本，是 PR 核心产物，支持 GPU/NPU 自动切换，结构规范。
- verl/workers/engine/veomni/transformer_impl.py（模块 引擎；类别 source；类型 core-logic；符号 get_per_tensor_param）：核心 bug 修复：修正专家并行参数访问方式，影响所有使用 VeOmni 引擎的 MoE 模型训练。

关键符号：get_per_tensor_param

关键源码片段

examples/grpo_trainer/run_qwen3_30b_veomni.sh

新增的 VeOmni 训练脚本，是 PR 核心产物，支持 GPU/NPU 自动切换，结构规范。

```
#!/usr/bin/env bash
# GRPO | Qwen3-30B-A3B (MoE) | VeOmni training | NVIDIA GPUs or Ascend NPU
# Knobs:
# INFER_BACKEND controls rollout backend: vllm

set -x
ENGINE=${1:-vllm}
# 检测硬件类型：如果能导入 torch_npu 则识别为 npu，否则为 gpu
DEVICE=${DEVICE:-$(python3 -c 'import torch_npu' 2>/dev/null && echo npu || echo gpu)}

TRAIN_FILE=dapo-math-17k.parquet
TEST_FILE=aime-2024.parquet
max_prompt_length=$((1024 * 2))
max_response_length=$((1024 * 8))
rollout_max_num_seqs=$((128))
n_devices_per_node=$((8))
actor_ppo_max_token_len=$(((max_prompt_length + max_response_length) * 1))
infer_ppo_max_token_len=$(((max_prompt_length + max_response_length) * 3))

case "${DEVICE}" in
  gpu)
    # GPU 环境无需额外配置
    ;;
  npu)
    # Ascend NPU 专用环境变量：启用任务队列、VLLM 后端、关闭 NZ 格式等
    export TASK_QUEUE_ENABLE=1
    export HCCL_OP_EXPANSION_MODE="AIV"
    export VLLM_USE_V1=1
    export VLLM_VERSION=0.13.0
    export VLLM_ASCEND_ENABLE_NZ=0
    export HCCL_BUFFSIZE=610
    export CKPT_DIR="./ckpt30b"
    export PYTORCH_NPU_ALLOC_CONF=max_split_size_mb:1024
    export CUDA_DEVICE_MAX_CONNECTIONS=1
    n_devices_per_node=16
    ;;
  *)
    echo "Unsupported DEVICE=${DEVICE}. Expected 'gpu' or 'npu'." >&2
    exit 1
    ;;
esac

# 后续参数省略 ...
```

verl/workers/engine/veomni/transformer_impl.py

核心 bug 修复：修正专家并行参数访问方式，影响所有使用 VeOmni 引擎的 MoE 模型训练。

```
# verl/workers/engine/veomni/transformer_impl.py
# 在 get_per_tensor_param 方法的内部闭包 param_generator 中

def param_generator():
    for name, param in params.items():
        unsharded_tensor = param.full_tensor() if isinstance(param, DTensor) else param

        is_expert_layer = "mlp.experts." in name
        is_proj = any(p in name for p in ["down_proj", "gate_proj", "up_proj", "gate_up_proj"])

        if is_expert_layer and is_proj and ps.ep_enabled:
            output_shape = list(unsharded_tensor.shape)
            # 修复：原使用 ps.ep_size，现改为从 extra_parallel_sizes 字典中获取 EP 大小
            # 注意：需确保 extra_parallel_sizes 已被初始化为字典（如 {'ep': N}），
            # 否则访问 ['ep'] 会引发 TypeError
            output_shape[0] *= ps.extra_parallel_sizes["ep"]
            stacked_tensor = torch.empty(output_shape, dtype=unsharded_tensor.dtype, device=
            device)

            # all gather expert tensors [32, H, I] -> [128, H, I]
            torch.distributed.all_gather_into_tensor(stacked_tensor, unsharded_tensor, group=ps.
            ep_group)
            yield from process_func(name, stacked_tensor)
            del stacked_tensor
        else:
            if is_expert_layer:
                yield from process_func(name, unsharded_tensor)
            else:
                yield name, unsharded_tensor
```

评论区精华

1. extra_parallel_sizes 类型风险：gemini-code-assist 指出 ps.extra_parallel_sizes 初始化为元组（(self.engine_config.expert_parallel_size,)），字典访问将导致 TypeError。建议在同一 PR 中修复初始化。该评论未被回复，但 PR 已合并，需注意初始化是否已改为字典。
2. 脚本目录迁移：wuxibin89 建议将 NPU 脚本从 ascend_extras 移至公共 examples/grpo_trainer。作者采纳并删除原有位置的文件。
3. 命名规范和硬件合并：wucong25 要求脚本命名遵循 run_<model>_<train-backend>.sh 格式，并合并 NPU/GPU 配置，通过 DEVICE 变量区分。作者添加了 USE 判断和头部注释。

- extra_parallel_sizes 类型错误风险 (correctness): 作者未公开回复，PR 已合并；需确认初始化已改为字典。
- 脚本目录和命名规范 (design): 作者采纳：删除原始位置文件，在 examples/grpo_trainer/run_qwen3_30b_veomni.sh 中加入 DEVICE 判断。

- 脚本增加硬件注释和参数参考 (documentation): 作者添加头部注释和 DEVICE 逻辑，并调整参数格式。

风险与影响

- 风险:
  1. 类型错误风险：transformer_impl.py 修改依赖 ps.extra_parallel_sizes 被正确初始化为字典。若初始化仍为元组（如行 107），则运行时在 param_generator 中访问 ps.extra_parallel_sizes['ep'] 会触发 TypeError。需确认该初始化已在同系列 PR 中修复。
  2. 配置兼容性：脚本中设置的 NPU 环境变量（如 VLLM_USE_V1=1、VLLM_VERSION=0.13.0）可能随 VLLM 版本变化而失效，需持续维护。
  3. 无测试覆盖：改动依赖特定硬件（NPU），未添加单元测试或 CI 步骤，回归风险仅靠人工验证。 - 影响：对用户：新增可直接使用的 Qwen3-30B VeOmni 训练脚本，支持 GPU 和 Ascend NPU 自动切换，降低了 NPU 用户的使用门槛。删除了小模型脚本，用户若需训练 Qwen3-1.7B 需自行适配。对系统：修改了 VeOmni 引擎中 EP 参数的获取方式，影响所有使用 MoE 模型且开启 EP 的训练流程。对团队：规范了脚本命名和组织结构，便于后续维护。 - 风险标记：潜在类型错误风险，缺少对应测试覆盖

关联脉络

- PR #5275 [veomni] feat: Add GRPO training scripts for Qwen3-VL-30B-MOE (VeOmni Backends): 同属 veomni 后端 GRPO 训练脚本系列，扩展模型支持，结构相似。