PR #7605 完整报告

verl-project/verl

[fsdp] feat: add Qwen3.5-2B on-policy distillation FSDP script

合并时间：2026-08-31 16:50

原文链接：http://prhub.com.cn/verl-project/verl/pull/7605

执行摘要

- 一句话：新增 Qwen3.5-2B 对策略蒸馏 FSDP 双平台脚本
- 推荐动作：值得精读的人群是需要在 GPU/NPU 双平台跑通 on-policy distillation 的用户，可将其作为 OPD 示例模板；关注 DEVICE 自动探测、NPU 环境变量分支和 uv gate 的组合方式。核心代码无改动，评审价值主要体现在示例脚本的可维护性与仓库惯例一致性上。

功能与动机

PR body 明确说明目标是 “The script unifies GPU and NPU into a single entry point with auto-detected DEVICE”，即在 geo3k 数据集上支持从 Qwen3.5-35B-A3B teacher 蒸馏到 Qwen3.5-2B student，并同时覆盖 GPU 与 NPU 平台。作者特意声明这不是 #6638 的重复：该 PR 针对 Qwen3.5-4B student，而本 PR 补充并验证 Qwen3.5-2B 配置，避免后续重复提交。

实现拆解

实现拆解如下：

1. 新增示例入口脚本：examples/on_policy_distillation_trainer/run_qwen3_5_2b_fsdp.sh 将所有实验参数集中到文件头部的 user-adjustable 区域（如 STUDENT_MODEL、TEACHER_MODEL、TRAIN_BATCH_SIZE=128、MAX_PROMPT_LENGTH=1024），再以 DATA、MODEL、ACTOR、REF、ROLLOUT、TRAINER、EXTRA 数组形式拼装成 verl.trainer.main_ppo 命令行。
2. 设备自动探测与环境分支：DEVICE 默认由 python3 -c 'import torch_npu' 探测，case 分支为 npu 注入华为集合通信与 vLLM Ascend 环境变量（HCCL_*、VERL_PLATFORM=huawei、VLLM_ASCEND_ENABLE_NZ=0、PYTORCH_NPU_ALLOC_CONF 等）；gpu 分支不做额外设置，非法值直接报错退出。
3. 蒸馏参数配置：EXTRA 数组开启 distillation.enabled=True，为 teacher 配置 vLLM 推理参数（tensor_model_parallel_size、expert_parallel_size、gpu_memory_utilization），并设置 distillation_loss 的 loss_mode=k1、topk=64、use_policy_gradient=True 及数值上下界保护。
4. 启动逻辑与 uv gate：GPU + vLLM/sglang 组合下使用 uv run --frozen --all-packages --extra ${INFER_BACKEND} --extra fsdp 启动，保证 Ray worker 与 driver 依赖版本一致；NPU 或其他后端回退到系统 python，避免 CANN 环境与 uv 冲突。NPU 分支还会额

外追加 vllm_mm_processor_cache_gb=0 以节省显存。

5. 验证与配套：仅做静态检查（bash -n、check_uv_gpu_only.py、check_example_naming.py），并在 GPU/NPU 上完成 400 步长跑验证蒸馏曲线对齐；提交历史中两次对齐仓库惯例（移除自定义确定性控制、checkpoint 频率改为 200 步默认值）。

关键文件：

- examples/on_policy_distillation_trainer/run_qwen3_5_2b_fsdp.sh（模块 蒸馏示例；类别 other；类型 entrypoint）：本 PR 唯一变更文件，新增 Qwen3.5-2B OPD FSDP 训练脚本，包含 GPU/NPU 自动探测、平台环境变量分支、蒸馏参数配置与 uv 启动逻辑，是全部功能和验证的载体。

关键符号：未识别

关键源码片段

examples/on_policy_distillation_trainer/run_qwen3_5_2b_fsdp.sh

本 PR 唯一变更文件，新增 Qwen3.5-2B OPD FSDP 训练脚本，包含 GPU/NPU 自动探测、平台环境变量分支、蒸馏参数配置与 uv 启动逻辑，是全部功能和验证的载体。

```
# DEVICE 自动探测：优先尝试导入 torch_npu，成功则判定为 npu，否则为 gpu。
# 用户也可通过环境变量直接覆盖，便于特殊环境（如容器内同时存在 CUDA 与 CANN）。
DEVICE=${DEVICE:-$(python3 -c 'import torch_npu' 2>/dev/null && echo npu || echo gpu)}

# 平台相关环境变量通过 case 分支隔离，NPU 侧集中配置华为集合通信与 vLLM Ascend 参数。
case "${DEVICE}" in
  gpu)
    # GPU 无需额外设置，依赖默认 NCCL/CUDA 环境。
    ;;
  npu)
    export HCCL_CONNECT_TIMEOUT=1500 # 大模型初始化集合通信耗时较长，放宽超时
    export HCCL_HOST_SOCKET_PORT_RANGE=60000-60050
    export HCCL_NPU_SOCKET_PORT_RANGE=61000-61050
    export RAY_EXPERIMENTAL_NOSET_ASCEND_RT_VISIBLE_DEVICES=1
    export VERL_PLATFORM=huawei # 触发 verl 侧的华为平台插件
    export TASK_QUEUE_ENABLE=1
    export HCCL_OP_EXPANSION_MODE="AIV"
    export VLLM_ASCEND_ENABLE_NZ=0 # 关闭 NZ 格式以节省 NPU 带宽
    export HCCL_BUFFSIZE=610
    export PYTORCH_NPU_ALLOC_CONF=max_split_size_mb:1024
    export CUDA_DEVICE_MAX_CONNECTIONS=1
    ;;
  *)
    echo "Unsupported DEVICE=${DEVICE}. Expected 'gpu' or 'npu'." >&2
    exit 1
    ;;
esac
```

```
# NPU 上多模态 processor cache 会占用显存，显式关闭；
# GPU 侧保留默认缓存以加速多模态数据预处理。
if [ "${DEVICE}" = "npu" ]; then
  ROLLOUT+=(actor_rollout_ref.rollout.engine_kwargs.vllm_mm_processor_cache_gb=0)
fi

# 启动方式：GPU + vLLM/sglang 时用 uv run 锁定依赖版本；
# 其他组合（含 NPU）回退到系统 python，避免 CANN/ 昇腾与 uv 环境冲突。
LAUNCH=(python3)
RAY=(ray_kwargs.ray_init.runtime_env.py_executable=null)
if [ "${VERL_USE_UV:-1}" != 0 ] && [ "${DEVICE:-gpu}" = gpu ] && { [ "${INFER_BACKEND}" =
vllm ] || [ "${INFER_BACKEND}" = sglang ]; }; then
  LAUNCH=(uv run --frozen --all-packages --extra "${INFER_BACKEND}" --extra fsdp python3)
  RAY=(ray_kwargs.ray_init.runtime_env.py_executable="uv -v run --frozen --all-packages --extra
  ${INFER_BACKEND} --extra fsdp")
fi

# 蒸馏核心参数：开启 on-policy distillation，teacher 为 Qwen3.5-35B-A3B。
EXTRA=(
  distillation.enabled=True
  distillation.n_gpus_per_node=${TEACHER_WORLD_SIZE}
  distillation.nnodes=${NNODES}
  distillation.teacher_models.teacher_model.model_path="${TEACHER_MODEL}"
  # teacher 使用独立 TP/EP，避免与 student 的 rollout 争抢显存
  distillation.teacher_models.teacher_model.inference.tensor_model_parallel_size=${teacher_tp}
  distillation.teacher_models.teacher_model.inference.expert_parallel_size=${teacher_ep}
  distillation.teacher_models.teacher_model.inference.name=vllm
  distillation.teacher_models.teacher_model.inference.gpu_memory_utilization=${teacher_gpu_
  mem_util}
  distillation.teacher_models.teacher_model.inference.max_model_len=${max_num_tokens}
  # k1 蒸馏损失 + topk 采样，并保留 policy gradient 项
  distillation.distillation_loss.loss_mode=${distillation_loss_mode}
  distillation.distillation_loss.topk=${distillation_topk}
  distillation.distillation_loss.use_task_rewards=False
  distillation.distillation_loss.use_policy_gradient=${use_policy_gradient}
  # 数值保护：限制 loss 与 log_prob 的上下界，防止训练不稳定
  distillation.distillation_loss.loss_max_clamp=10.0
  distillation.distillation_loss.log_prob_min_clamp=-10.0
)
```

评论区精华

本 PR 没有形成实质 review 讨论：审核者 wuxibin89 在无任何评论的情况下直接 APPROVED。从提交历史可看出两次设计收敛：第 2 次提交移除自定义确定性控制，改为仓库标准 GPU uv gate，避免示例脚本引入特例；第 3 次提交将 checkpoint 频率对齐到 200 步默认值。这反映出维护者偏好"示例复用仓库既有惯例、不偏离主流程"的取向。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. NPU 环境变量为特定集群经验值：脚本中 HCCL_HOST_SOCKET_PORT_RANGE、HCCL_BUFFSIZE=610、PYTORCH_NPU_ALLOC_CONF=max_split_size_mb:1024 等配置来自 Ascend 910B 验证环境，迁移到其他 NPU 集群可能需要重新调参。
  2. DEVICE 自动探测可能误判：当容器同时安装 CUDA 与 CANN/torch_npu 时，import torch_npu 会成功导致误判为 npu，从而套用 NPU 环境变量。
  3. 无自动化测试覆盖：脚本仅通过静态检查，未接入 CI 自动执行；仓库近期已有多次清理“过期配置”的 PR，这类示例脚本存在配置漂移风险。
  4. uv *gate* 依赖 uv.lock 中的 *extras*：若锁文件缺少 vllm/sglang/fsdp 对应 *extras*，GPU 启动会失败。
  5. 资源需求高：*teacher* 默认 TEACHER_WORLD_SIZE=8、TEACHER_TP=8、TEACHER_EP=8，需要整机 8 卡级别资源，小规模集群难以直接跑通。
- 影响：
  1. 用户侧：为 on-policy distillation 用户提供开箱即用的 2B student + 35B-A3B *teacher* 训练脚本，统一 GPU/NPU 入口，显著降低了在 Ascend 上运行 OPD 的门槛。
  2. 系统侧：不涉及核心 trainer、rollout、FSDP 实现，对既有训练流程零影响。
  3. 团队侧：脚本的“自动探测 + *case* 分支 + uv *gate*”结构为后续双平台示例提供了可复制的模板，但也意味着示例脚本数量增加，后续维护（配置漂移、环境变量变化）成本会上升。 - 风险标记：示例级变更，无自动化测试，NPU 环境变量为经验配置，DEVICE 自动探测可能误判，依赖 uv.lock *extras*, *teacher* 资源需求高

关联脉络

- PR #6638 Qwen3.5-4B OPD FSDP script: PR body 明确提及该 PR 作为前序实现，二者同属 Qwen3.5 OPD FSDP 示例脚本族；本 PR 是对 4B 方案的补充而非重复，扩展 2B student 配置。
- PR #7577 [veomni] feat: DeepSeek V4 QAT bf16 fake quant training: 该 PR 修改了蒸馏核心逻辑 verl/trainer/ppo/core_algos.py，本 PR 在示例层消费同一蒸馏框架，说明仓库的蒸馏功能线同时在核心算法与示例配置两个层面扩展。