PR #6056 完整报告

verl-project/verl

[fully_async, rollout] feat: enable online policy distillation in fully async training

合并时间：2026-05-06 18:55

原文链接：http://prhub.com.cn/verl-project/verl/pull/6056

执行摘要

- 一句话：实现完全异步模式多教师在线策略蒸馏
- 推荐动作：建议关注以下方面：
  - 资源池设计：当前在 rollouter 中分配教师资源池，但未来规划中 rollouter 将不再负责资源分配。关注后续 PR 对资源池分离的重构。
  - 蒸馏配置传递：distillation_config 参数的风险需要验证 worker 类是否兼容。建议在合并后运行全量 CI 确认无 TypeError。
  - 异步兼容性：run_in_executor 的使用是临时方案，需跟踪 MultiTeacherModelManager 的异步化进展。
  - 测试覆盖：如果使用 FSDP 或其他后端，考虑扩展测试矩阵。

功能与动机

在完全异步训练中支持在线策略蒸馏，使得知识蒸馏可以与异步训练流水线无缝结合，提高模型训练效率和效果。PR #6051 已实现多教师 OPD 的基本框架，本 PR 将其扩展到 fully_async 模式，填补异步训练中蒸馏能力的空白。

实现拆解

1. 资源池扩展：在 verl/experimental/separation/utils.py 的 create_resource_pool_manager 中新增 Role.TeacherModel 的处理逻辑，根据配置中的 distillation.n_gpus_per_node 和 distillation.nnodes 分配专用资源池。
2. 入口层调整：修改 verl/experimental/fully_async_policy/fully_async_main.py 的 _create_rollouter 方法，当蒸馏启用时，在创建 rollouter 的资源池管理器中加入 Role.TeacherModel，并提前创建资源池；随后将完整的 resource_pool_manager 传入 FullyAsyncRollouter。
3. Rollouter 集成：在 verl/experimental/fully_async_policy/fully_async_rollouter.py 中新增 _create_teacher_model_manager 异步方法，使用 asyncio.get_running_loop().run_in_executor 解决 MultiTeacherModelManager.__init__ 内部 asyncio.run() 与已有事件循环的冲突；在 init_workers 流程中插入该调用，并在 _init_async_rollout_manager 中将 teacher_client 传递给 FullyAsyncAgentLoopManager。

4. 训练器适配：在 verl/experimental/fully_async_policy/fully_async_trainer.py 的 __init__ 中解析蒸馏配置，存储为 self.distillation_config；在 _create_actor_rollout_classes 中将 distillation_config 传递给 worker 类的初始化参数，供下游 _update_actor 使用。
5. 测试与 CI：新增 tests/special_e2e/run_fully_async_policy_opd.sh 端到端测试脚本，配置学生模型和两个教师模型，使用 Megatron 训练后端。新增 .github/workflows/e2e_fully_async_policy_opd.yml CI 工作流，在触发条件中监控相关路径并运行测试。

关键文件：

- verl/experimental/fully_async_policy/fully_async_rollouter.py（模块 异步 rollout；类别 source；类型 core-logic；符号 _create_teacher_model_manager）：核心文件：新增 _create_teacher_model_manager 方法，管理教师模型资源的初始化和生命周期，并传递 teacher_client 给 AgentLoopManager。
- tests/special_e2e/run_fully_async_policy_opd.sh（模块 端到端测试；类别 test；类型 test-coverage）：新增的端到端测试脚本，覆盖多教师 OPD 在完全异步模式下的完整流程，使用 Megatron 后端。
- verl/experimental/fully_async_policy/fully_async_main.py（模块 异步主入口；类别 source；类型 entrypoint）：入口文件：修改 _create_rollouter 以在蒸馏启用时预分配教师资源池。
- verl/experimental/fully_async_policy/fully_async_trainer.py（模块 异步训练器；类别 source；类型 dependency-wiring）：训练器：解析蒸馏配置并传递给 worker 类，是蒸馏损失计算的数据来源。
- verl/experimental/separation/utils.py（模块 资源池工具；类别 source；类型 core-logic）：资源池工具：新增 TeacherModel 角色资源池分配逻辑。
- .github/workflows/e2e_fully_async_policy_opd.yml（模块 CI 配置；类别 infra；类型 infrastructure）：新增 CI 工作流，触发条件精确监控相关路径，保障 OPD 功能持续集成。

关键符号：_create_teacher_model_manager, _create_rollouter, _create_actor_rollout_classes, create_resource_pool_manager

关键源码片段

verl/experimental/fully_async_policy/fully_async_rollouter.py

核心文件：新增 _create_teacher_model_manager 方法，管理教师模型资源的初始化和生命周期，并传递 teacher_client 给 AgentLoopManager。

```
async def _create_teacher_model_manager(self):
    """Create MultiTeacherModelManager for distillation if enabled.

    Allocates a big resource pool for all teachers and passes it to
    MultiTeacherModelManager, which splits it internally per teacher.

    NOTE: MultiTeacherModelManager.__init__ calls _run_all internally which uses
    asyncio.run(), conflicting with the already-running event loop.
```

```
    Run in a thread executor as a workaround.
    """
    from verl.trainer.distillation.losses import is_distillation_enabled
    from verl.trainer.ppo.utils import Role

    self.teacher_model_manager = None
    if is_distillation_enabled(self.config.get("distillation")):
        from verl.experimental.teacher_loop import MultiTeacherModelManager

        # Fetch the pre-allocated resource pool for teacher models
        teacher_resource_pool = self.resource_pool_manager.get_resource_pool(Role.
        TeacherModel)
        # MultiTeacherModelManager.__init__ uses asyncio.run() internally,
        # which conflicts with the running event loop.
        # Use run_in_executor to offload to a separate thread.
        loop = asyncio.get_running_loop()
        self.teacher_model_manager = await loop.run_in_executor(
            None,
            lambda: MultiTeacherModelManager(config=self.config, resource_pool=teacher_
            resource_pool),
        )
```

评论区精华

核心讨论点：

- 资源池分配策略：wuxibin89 建议分配一个大资源池并在 MultiTeacherModelManager 内部分割，避免 placement group 碎片。作者采纳并修改。
- resource_pool=None 路径：JacobHelwig 质疑 teacher_model.py 中 resource_pool=None 路径的必要性，作者回复已改为传递预分配资源池。
- 蒸馏配置参数传递风险：gemini-code-assist[bot] 指出将 distillation_config 直接传入 RayClassWithInitArgs 可能导致 TypeError，因为目标 worker 类的 __init__ 可能未接受该参数。该问题在最终 PR 中未明确解决，构成潜在风险。
- 后端选择：wuxibin89 建议使用 Megatron 而非 FSDP 作为 E2E 训练后端，作者已添加相应的 CI。
- 导入冗余：JacobHelwig 指出 _create_teacher_model_manager 中重复导入 Role，作者已移除。
- 未来方向：ArronHZG 说明未来规划中 rollouter 将不再负责资源分配，teacher 需使用 standalone 模式，作者表示等待后续变更。
- 教师模型资源池分配策略 (design): 采用大资源池内部分割方案
- distillation_config 参数传递可能引发 TypeError (correctness): 未解决，需后续验证 worker 类兼容性
- E2E 测试后端选择 (design): 采用 Megatron 后端
- teacher_model.py resource_pool=None 路径 (design): 移除 None 路径，统一使用预分配资源池

风险与影响

- 风险：
  1. 蒸馏配置参数兼容性风险：在 fully_async_trainer.py 中，distillation_config 作为关键字参数传递给 RayClassWithInitArgs。若下游 worker 类（如 PPOWorker 或 FullyAsyncAgentLoopWorker）的 __init__ 未接受 distillation_config 参数，会导致实例化时 TypeError，阻断训练初始化。当前代码中未看到对应 worker 类的适配，该风险未被充分解决。
  2. 资源池配置缺失风险：若用户启用蒸馏但未在配置中提供 distillation.n_gpus_per_node，断言会失败。但在 create_resource_pool_manager 中该断言在 TeacherModel in roles 时才执行，如果 roles 不包含 TeacherModel 但蒸馏仍启用（逻辑矛盾），可能绕过检查导致后续错误。
  3. 异步事件循环嵌套风险：_create_teacher_model_manager 使用 run_in_executor 规避 MultiTeacherModelManager.__init__ 内部的 asyncio.run()，这是一种临时 hack。如果 MultiTeacherModelManager 的初始化进一步引入异步逻辑，仍可能引发事件循环冲突。
  4. CI 覆盖有限：E2E 测试仅覆盖 Megatron 后端，未测试 FSDP 或其他后端。同时测试使用 8 块 H100 GPU，对资源要求较高，可能存在环境差异。
- 影响：影响范围：
  - 用户：提供完全异步训练中在线策略蒸馏的能力，用户可通过配置 distillation 字段启用多教师蒸馏，无需手动管理教师资源。
  - 系统：新增 TeacherModel 角色资源池，增加整体 GPU 占用。以测试配置为例，2 个教师各占 1 块 GPU，共 2 块额外 GPU。对多作业环境下的资源调度有影响。
  - 团队：该 PR 是 #6051 的扩展，统一了同步 / 异步模式下的 OPD 接口。但引入的 resource_pool=None 路径虽被修复，相关讨论和遗留代码（如 teacher_model.py 中的条件分支）可能为后续维护增加复杂度。
  - 部署：新增了 CI 工作流，增加 CI 耗时约 30 分钟，但通过路径过滤减少了不必要的触发。
  - 风险标记：蒸馏配置参数兼容性，资源池断言风险，异步事件循环临时方案，测试覆盖有限

关联脉络

- PR #6129 [BREAKING][rollout] refactor: move LLMServerManager out of AgentLoopManager: 本 PR 需要适配该 refactoring，commit 修复了 teacher_client 传递；讨论中标记为 blocker。