PR #7513 完整报告

verl-project/verl

[trainer, ckpt, cfg] feat: add config-driven checkpoint callback hook

合并时间：2026-08-25 11:41

原文链接：http://prhub.com.cn/verl-project/verl/pull/7513

执行摘要

- 一句话：新增配置驱动的 checkpoint 保存回调钩子
- 推荐动作：值得精读。设计上有三个可借鉴点：null-object 模式省去调用点空值守卫、fail-fast 语义保证 checkpoint 副作用不静默失败、**kwargs 与 no-op 基类为未来扩展事件预留兼容性。评审中“不碰 deprecated trainer、删冗余 UT”的收敛也值得学习。计划在 verl 上做 checkpoint 上传 / 注册 / 评估接入的团队可直接照此实现。

功能与动机

PR body 明确指出，用户需要在不 fork 或子类化 trainer 的前提下获得 checkpoint 副作用的官方扩展点（上传对象存储、注册模型 registry、触发评估、保留策略）。方案刻意对齐 HuggingFace transformers 的 TrainerCallback.on_save 语义。PR 作者也通过搜索确认没有现存 PR/issue 提供 trainer 级 checkpoint callback，并与 rollout.checkpoint_manager_class（权重同步引擎而非磁盘 checkpoint）、v1 trainer 的 on_* 方法（子类覆写点而非配置可插拔）做了区分。

实现拆解

1. 新增回调基类与工厂：verl/trainer/ppo/checkpoint_callback.py 定义 CheckpointCallback，唯一事件为 on_save(trainer, global_step, checkpoint_dir, async_save=False, **kwargs)，**kwargs 为向前兼容保留；build_checkpoint_callback(config) 在配置未设置时返回 no-op 实例（null-object 模式），否则通过 load_class_from_fqn 加载用户 FQN 并传入完整配置实例化。
2. 接入 v1 trainer：verl/trainer/ppo/v1/trainer_base.py 中 PPOTrainer.__init__ 增加 self.checkpoint_callback = build_checkpoint_callback(config)，使坏的 FQN 在 Ray 资源分配前的 driver 侧立即失败；_save_checkpoint 末尾在两个路径调用 on_save：async_save=True 早退路径携带 async_save=True，正常路径在写入 latest_checkpointed_iteration.txt 后携带 async_save=False。钩子显式放在 worker save 成功之后而非 try/finally，保证保存失败时不触发回调。
3. 配置与生成文件同步：verl/trainer/config/ppo_trainer.yaml 新增带注释的 trainer.checkpoint_callback_class: null，四个 _generated_ppo_*_trainer.yaml 通过 scripts/generate_trainer_config.sh 重新生成，保持配置契约一致。
4. 测试配套：新增 tests/trainer/ppo/v1/test_checkpoint_callback_on_cpu.py，用 _StubTrainer 与 _RecordingCallback 覆盖 4 条路径：worker save 后触发 on_save、

async_save=True 标志传递、worker 保存抛异常时抑制回调、回调异常向上传播中止训练。

5. 文档更新：docs/advance/checkpoint.rst 新增 “Checkpoint Callback” 章节说明钩子契约与异步保存时序，docs/examples/config.rst 补充键参考，docs/extend_guide.rst 加入扩展指南条目。评审后移除对 legacy RayPPOTrainer 及对应 UT 的改动，verl/experimental/fully_async_policy/fully_async_trainer.py 不在覆盖范围。

关键文件：

- verl/trainer/ppo/checkpoint_callback.py（模块 回调机制；类别 source；类型 core-logic；符号 CheckpointCallback, init, on_save, build_checkpoint_callback）：本 PR 核心：新增 CheckpointCallback 基类与 build_checkpoint_callback 工厂，null-object + load_class_from_fqn 模式。
- tests/trainer/ppo/v1/test_checkpoint_callback_on_cpu.py（模块 单元测试；类别 test；类型 test-coverage；符号 _StubTrainer, _RecordingCallback, _make_trainer, test_save_fires_on_save_after_worker_save）：覆盖 v1 训练器钩子接线的 4 条关键路径，验证事件顺序与异常语义。
- verl/trainer/ppo/v1/trainer_base.py（模块 训练器；类别 source；类型 dependency-wiring；符号 PPOTrainer.init, PPOTrainer._save_checkpoint）：钩子接线点：__init__实例化回调，_save_checkpoint 两处调用 on_save，是变更影响的核心训练路径。
- verl/trainer/config/ppo_trainer.yaml（模块 训练配置；类别 config；类型 configuration）：新增配置契约键 checkpoint_callback_class，文档化注释说明 FQN 与语义。
- verl/trainer/config/_generated_ppo_megatron_trainer.yaml（模块 生成配置；类别 config；类型 regenerated-config）：由脚本重新生成的训练配置，携带新配置键，保证 Megatron 训练入口契约一致。
- verl/trainer/config/_generated_ppo_torchtitan_trainer.yaml（模块 生成配置；类别 config；类型 regenerated-config）：由脚本重新生成的训练配置，携带新配置键，保证 TorchTitan 训练入口契约一致。
- verl/trainer/config/_generated_ppo_trainer.yaml（模块 生成配置；类别 config；类型 regenerated-config）：由脚本重新生成的训练配置，携带新配置键，保证默认 PPO 训练入口契约一致。
- verl/trainer/config/_generated_ppo_veomni_trainer.yaml（模块 生成配置；类别 config；类型 regenerated-config）：由脚本重新生成的训练配置，携带新配置键，保证 VeOmni 训练入口契约一致。
- docs/advance/checkpoint.rst（模块 用户文档；类别 docs；类型 documentation；符号 MyCheckpointCallback, on_save）：新增 Checkpoint Callback 章节，完整定义 on_save 钩子契约、异步保存时序与异常语义。
- docs/extend_guide.rst（模块 用户文档；类别 docs；类型 documentation）：扩展指南新增条目，引导用户通过 checkpoint callback 扩展训练器行为。
- docs/examples/config.rst（模块 用户文档；类别 docs；类型 documentation）：配置参考文档同步新增 checkpoint_callback_class 键说明。

关键符号：CheckpointCallback, CheckpointCallback.on_save, build_checkpoint_callback, PPOTrainer.init, PPOTrainer._save_checkpoint

关键源码片段

verl/trainer/ppo/v1/trainer_base.py

钩子接线点：__init__实例化回调，_save_checkpoint 两处调用 on_save，是变更影响的核心训练路径。

```
actor_ckpt_cfg = self.config.actor_rollout_ref.actor.get("checkpoint", {})
if actor_ckpt_cfg.get("async_save", False):
    # Megatron 异步保存：worker 写盘可能仍在途，跳过迭代 tracker 写入
    logger.info("skip write latest_checkpointed_iteration.txt when async_save is True")
    # 显式传 async_save=True，让回调知道此时不能假设 durable
    self.checkpoint_callback.on_save(
        trainer=self,
        global_step=self.global_steps,
        checkpoint_dir=local_global_step_folder,
        async_save=True,
    )
    return

local_latest_checkpointed_iteration = os.path.join(
    self.config.trainer.default_local_dir, "latest_checkpointed_iteration.txt"
)
with open(local_latest_checkpointed_iteration, "w") as f:
    f.write(str(self.global_steps))

# 同步路径：迭代 tracker 已落盘，表示保存流程完整走完，此时才触发 on_save
self.checkpoint_callback.on_save(
    trainer=self,
    global_step=self.global_steps,
    checkpoint_dir=local_global_step_folder,
    async_save=False,
)
```

verl/trainer/config/ppo_trainer.yaml

新增配置契约键 checkpoint_callback_class，文档化注释说明 FQN 与语义。

```
trainer:
  ...
  # 用户自定义 checkpoint 回调类的全限定名，例如 my_pkg.callbacks.MyCheckpointCallback。
  # driver 侧用完整 trainer config 实例化，每次 checkpoint 保存后调用其 on_save 钩子，
  # 语义与 transformers TrainerCallback.on_save 对齐；null 表示不启用任何回调。
  checkpoint_callback_class: null
```

评论区精华

评审共 4 条评论、2 个主题。wuxibin89 在 verl/trainer/ppo/ray_trainer.py 上留言 “Legacy trainer is deprecated, don't modify it.”，作者回复 “Done — reverted all changes to ray_trainer.py; the callback is now wired into the v1 trainer only”，并同步更新测试与文档；在 tests/trainer/ppo/test_checkpoint_callback_on_cpu.py 上留言 “This UT is not needed.”，作者删除该文件、保留 v1 测试。最终 wuxibin89 给出 APPROVED。

- Legacy RayPPOTrainer 是否应接入 checkpoint callback (design): 移除 legacy trainer 改动，特性明确为 v1-only。
- legacy trainer 的 UT 是否必要 (testing): 删除 legacy 测试文件，保留 v1 测试。

风险与影响

- 风险：
  1. _save_checkpoint 位于 v1 训练核心保存路径，任何回调异常都会中止训练：这是有意的 fail-fast 设计（文档与 docstring 已说明），但第三方回调的 bug 会直接影响训练可用性；
  2. async_save=True（Megatron 异步 checkpoint）时 on_save 在 worker 写入可能仍在途、latest_checkpointed_iteration.txt 未写时触发，用户若将其当作“已持久化”信号可能导致副作用于不可靠数据；
  3. 仅 v1 trainer 接入，legacy RayPPOTrainer 与 fully_async trainer 不触发钩子，用户跨后端迁移时行为不一致；
  4. 新增配置键需要与 4 个生成配置保持同步，后续手工编辑生成文件可能失配。 - 影响：对用户：新增官方扩展点，默认 null 无行为变化，存量配置完全兼容；对系统：每次 checkpoint 保存后 driver 侧多一次回调调用，性能开销可忽略，但回调的 I/O 会叠加在保存路径上；对团队：确立了 trainer 级 checkpoint 副作用的插入规范，未来可在不破坏子类的前提下扩展 load 等事件；对文档：config.rst、checkpoint.rst、extend_guide.rst 三处同步，配置契约清晰。 - 风险标记：核心保存路径变更，异常中断训练，异步保存时序，仅 v1 覆盖

关联脉络

- PR #7363 [ckpt, sft, megatron] fix: finalize Megatron async checkpoint queue on every SFT rank: PR body 列为最接近的 checkpoint bug 修复之一，同属 checkpoint 生命周期完善；其异步保存完成语义与回调的 async_save 时序直接相关。
- PR #7396 [ckpt, fsdp] fix: enforce max-one SFT retention after restart: PR body 列为最接近的 checkpoint bug 修复之一，同属 checkpoint 生命周期与保留策略完善。
- PR #7402 [ckpt, trainer] fix: resume SFT after epoch checkpoints: PR body 列为最接近的 checkpoint bug 修复之一，同属 checkpoint 生命周期完善。
- PR #6537 [RFC] Generic RemoteBackend abstraction for out-of-process RL backends: PR body 论证本特性与 RemoteBackend RFC 不重复，两者同属 verl 可扩展性设计方向，一个在训练器内部扩展点，一个在后端抽象。
- PR #7536 [cfg] fix: drop unused ref router replay config: 同为 trainer 配置变更并同步重新生成 generated_ppo*_trainer.yaml，走同一配置生成链路。