

PR #2224 完整报告

radixark/miles

fix: derive --critic-save from --save so PPO critic checkpoints are not silently skipped

合并时间: 2026-08-07 06:29

原文链接: <http://prhub.com.cn/radixark/miles/pull/2224>

执行摘要

- 一句话: --critic-save 默认从 --save 推导, 修复 PPO critic 检查点静默丢失
- 推荐动作: 值得快速精读。改动集中在 miles/utils/arguments.py 的 miles_validate_args, 属于 " 参数默认值推导 " 的经典案例: 与既有 --critic-load / --critic-lr 回退保持一致的风格, 以兄弟目录隔离 actor 与 critic 的状态文件, 测试覆盖推导、尾斜杠、显式优先、无 --save 四种边界。对维护 CLI 参数校验的开发者有直接参考价值。

功能与动机

PR body 明确指出: 当使用 --advantage-estimator ppo 训练时, 训练循环总是尝试保存 critic 模型 (train.py 调用 save_training_model(critic_model)), critic 角色用 args.critic_save 作为保存路径; 但 --critic-save 默认为 None 且从未从 --save 推导, 所以只传 --save 的 PPO 运行会静默跳过所有 critic 检查点。actor 检查点看起来正常, 但恢复运行会让 critic 从零开始, 破坏 PPO 的断点恢复。且 --critic-load 回退 --load、--critic-lr 回退 --lr 的机制早已存在, 本 PR 补齐缺失的 --critic-save 回退。

实现拆解

1. 更新参数帮助文本: 在 miles/utils/arguments.py 的 add_algo_arguments 中, 将 --critic-save 的 help 从 "The checkpoint for critic model." 改为说明默认行为: 未设置时按 --save 加 _critic 后缀推导, 并给出 --save /ckpts/run1 对应 /ckpts/run1_critic 的具体示例。
2. 新增默认推导逻辑: 在 miles_validate_args 中紧跟 critic_load / critic_lr 回退之后新增: 条件是 args.critic_save is None 且 args.save 非 None; 推导值为 args.save.rstrip("/") + "_critic"。选择兄弟目录而非 --save 本身, 原因是 actor 与 critic 各自维护 latest_checkpointed_iteration.txt 一类迭代跟踪文件, 共享目录会互相覆盖。
3. 补齐单元测试: tests/fast/utils/test_arguments.py 新增 TestCriticSaveDerivation 类, 复用真实参数注册与校验入口 (get_miles_extra_args_provider + miles_validate_args), 覆盖四类场景: 从 --save 推导出 _critic、尾斜杠归一化 (/ckpts/run1/ → /ckpts/run1_critic)、显式 --critic-save 优先不被覆盖、未设置 --save 时保持 None。
4. 配套情况: 无配置文件、部署或文档站点改动, 两个提交分别完成实现与帮助文本完善, 属于纯参数校验层的小范围修复。

关键文件:

- miles/utils/arguments.py (模块 参数校验; 类别 source; 类型 core-logic; 符号 miles_validate_args, add_algo_arguments) : 核心修复文件: 在 miles_validate_args 中新增 critic_save 从 save 的默认推导逻辑, 并更新 --critic-save 帮助文本, 决定 PPO critic 检查点默认落盘路径。
- tests/fast/utils/test_arguments.py (模块 参数校验; 类别 test; 类型 test-coverage; 符号 TestCriticSaveDerivation, _validate, test_derives_sibling_dir_from_save, test_trailing_slash_is_stripped) : 新增 TestCriticSaveDerivation 测试类, 覆盖推导、尾斜杠归一化、显式优先、无 --save 保持 None 四种场景, 锁定新默认行为防止回归。

关键符号: miles_validate_args, add_algo_arguments

关键源码片段

miles/utils/arguments.py

核心修复文件: 在 miles_validate_args 中新增 critic_save 从 save 的默认推导逻辑, 并更新 --critic-save 帮助文本, 决定 PPO critic 检查点默认落盘路径。

```
# miles_validate_args 中的 critic 参数回退片段
# 只有 PPO 才启用 critic, Shared Actor/Critic 模式仅支持 Megatron 后端
args.use_critic = args.advantage_estimator == "ppo"
if args.use_critic:
    if args.train_backend != "megatron":
        raise ValueError("Shared Actor/Critic PPO requires the Megatron backend")
    assert args.megatron_to_hf_mode != "bridge", (
        "Critic models are not supported with --megatron-to-hf-mode bridge"
    )
    assert not enable_experimental_ft_trainer(), (
        "Shared Actor/Critic PPO is not supported with MILES_EXPERIMENTAL_FT_TRAINER=1"
    )
    assert args.kl_coef == 0, (
        "Shared Actor/Critic PPO does not support reward-level KL (--kl-coef)"
    )
    args.critic_num_gpus_per_node = args.actor_num_gpus_per_node
    args.critic_num_nodes = args.actor_num_nodes

# --critic-load 与 --critic-lr 早已支持回退, 本 PR 补齐 --critic-save 的缺失回退,
# 否则只传 --save 的 PPO 运行会静默跳过所有 critic 检查点, 恢复时 critic 从头开始
if args.critic_load is None:
    args.critic_load = args.load
if args.critic_lr is None:
    args.critic_lr = args.lr
if args.critic_save is None and args.save is not None:
    # 不能用 --save 本身, actor 与 critic 各自维护 latest_checkpointed_iteration.txt
    # 迭代跟踪器, 共享目录会互相覆盖; 用兄弟目录 <save>_critic 隔离两者
    args.critic_save = args.save.rstrip("/") + "_critic"
```

tests/fast/utils/test_arguments.py

新增 TestCriticSaveDerivation 测试类，覆盖推导、尾斜杠归一化、显式优先、无 --save 保持 None 四种场景，锁定新默认行为防止回归。

```
class TestCriticSaveDerivation:
    """覆盖 --critic-save 从 --save 推导的四种场景"""

    def _validate(self, extra):
        # 复用真实参数注册与校验入口，避免测试与实现脱节
        parser = argparse.ArgumentParser()
        get_miles_extra_args_provider()(parser)
        args = parser.parse_args(extra + ["--num-rollout", "1"] + REQUIRED_ARGS)
        miles_validate_args(args)
        return args

    def test_derives_sibling_dir_from_save(self):
        # 未显式传 --critic-save 时，默认推导为 <save>_critic
        args = self._validate(["--advantage-estimator", "ppo", "--save", "/ckpts/run1"])
        assert args.critic_save == "/ckpts/run1_critic"

    def test_trailing_slash_is_stripped(self):
        # 尾斜杠必须先去掉，否则会得到 /ckpts/run1/_critic
        args = self._validate(["--advantage-estimator", "ppo", "--save", "/ckpts/run1/"])
        assert args.critic_save == "/ckpts/run1_critic"

    def test_explicit_critic_save_is_respected(self):
        # 显式传 --critic-save 时保持原值，不被推导覆盖
        args = self._validate(
            ["--advantage-estimator", "ppo", "--save", "/ckpts/run1", "--critic-save", "/elsewhere/critic"]
        )
        assert args.critic_save == "/elsewhere/critic"

    def test_stays_none_without_save(self):
        # --save 未设置时保持 None，避免凭空生成无意义路径
        args = self._validate(["--advantage-estimator", "ppo"])
        assert args.critic_save is None
```

评论区精华

本 PR 没有产生 review 评论与讨论线程，guapisolo 直接批准（APPROVED）。关键设计决策（兄弟目录避免迭代跟踪器覆盖、rstrip("/") 处理尾斜杠）在 PR body 中已充分说明，代码注释也给出了选型理由。

- 暂无高价值评论线程

风险与影响

- 风险：

1. 行为变更（预期内）：此前只传 `--save` 的 PPO 运行不落盘任何 critic 检查点，现在默认写入 `_critic`，磁盘占用增加，运行中断后恢复时需使用新路径。
2. 恢复不对称需留意：`--critic-load` 的回退目标仍是 `--load`（actor 检查点目录），与新的默认保存路径 `_critic` 不一致。恢复 PPO 运行若要加载此前保存的 critic，需要显式传 `--critic-load _critic`，否则 critic 仍会从 actor 检查点或随机初始化开始，与本 PR 的保存修复需要配套使用。
3. 极端输入：`--save "/"` 经 `rstrip("/")` 后会推导出 `_critic`，属于极端边缘情况，正常路径不受影响。
4. 非 PPO 场景：推导逻辑未检查 `args.use_critic`，非 PPO 运行也会设置该字段，但不会创建 critic 角色，字段无人消费，无实际影响。
5. 回归面：改动限于参数校验阶段，且有 4 个单元测试兜底；但缺少 e2e 级别的 "保存后恢复" 测试，恢复路径的验证依赖人工或后续补充。
 - 影响：影响所有使用 PPO + `--save` 训练且未显式设置 `--critic-save` 的用户：修复后 critic 检查点默认落盘，恢复行为更可预期。对团队的意义在于消除 PPO 断点恢复的隐性故障，并补上 `--critic-*` 参数族最后一块缺失的默认回退，使 `--critic-load`、`--critic-lr`、`--critic-save` 三者行为对齐。改动面小，无部署与配置配套。
 - 风险标记：行为变更：新增默认 critic 保存路径，恢复时 `--critic-load` 与默认保存路径不对齐，磁盘占用增加，仅单元测试覆盖

关联脉络

- PR #2226 fix: preserve routing-replay state around MTP spec creation: 同属 PPO critic 训练正确性修复线，一个解决 critic 崩溃，一个解决 critic 检查点静默丢失。
- PR #2203 Revert "Reject `--disable-weights-backuper` for LoRA + colocate + offload-train" (#2077): 同改 `miles/utils/arguments.py` 的参数校验逻辑，属于该文件参数组合与默认值维护线的延续。
- PR #2077 Reject `--disable-weights-backuper` for LoRA + colocate + offload-train: 同样在 `miles/utils/arguments.py` 中做参数组合校验，与本 PR 共享同一维护区域。