PR #6823 完整报告

verl-project/verl

[BREAKING][trainer, cfg] chore: enable V1 trainer by default

合并时间：2026-06-24 16:51

原文链接：http://prhub.com.cn/verl-project/verl/pull/6823

执行摘要

- 一句话：默认启用 V1 PPO trainer，修正配置键名与奖励计算填充
- 推荐动作：建议所有开发者关注此 PR 的变更细节，特别是配置键名修改和参考策略引用修正。对于自定义 trainer 的项目，需检查 CriticConfig 字段重命名。讨论中的 temperature 冗余问题建议在后续 PR 中统一清理。

功能与动机

V1 PPO trainer 经过多个迭代（如 colocated reward model、fully async 支持等）已趋于成熟，团队决定将其设为默认，以简化用户配置并推动后续功能统一演进。

实现拆解

1. 默认 trainer 切换：在 PPO 配置基类（verl/trainer/config/_generated_ppo_trainer.yaml 等）中将 use_v1 默认值改为 true，同时更新 CI 配置（.github/workflows/model.yml 等）以使用 V1 入口。
2. 配置键名统一：将 CriticConfig 的 model_config 字段重命名为 model，同步修改 verl/trainer/ppo/v1/trainer_base.py、verl/experimental/separation/ray_trainer.py 及对应 YAML 生成文件中引用该字段的位置。
3. 参考策略引用修复：V1 trainer_base 的 _setup 中，当 ref_in_actor=True（LoRA 场景）时，ref_policy_wg 应指向 actor_rollout_wg 而非 all_wg[str(Role.ActorRolloutRef)]，避免分离资源池引用错误。
4. 奖励计算数据填充：verl/experimental/reward_loop/reward_loop.py 的 compute_rm_score 中，当数据条目不能被 worker 数整除时，先通过 pad_dataproto_to_divisor 填充再分片，最后截断至原始长度，防止 Ray 任务分片不均。
5. 其他清理：verl/utils/fsdp_utils.py 中的 collect_lora_params 在 layered_summon 回退路径中移除了多余的 state_dict 构造；verl/trainer/ppo/v1/trainer_separate_async.py 删除了对 need_reward_model 的断言，允许 colocate 奖励模型；_compute_values 与 _update_critic 中显式传递 temperature 字段。

关键文件：

- verl/trainer/ppo/v1/trainer_base.py（模块 训练器；类别 source；类型 core-logic；符号 _setup, _compute_values, _update_critic）：V1 训练器基类，核心变更包括 reference policy 引用修正（支持 LoRA 场景）、计算 value 时传递 temperature、更新器传入

temperature。

- verl/experimental/reward_loop/reward_loop.py（模块 奖励循环；类别 source；类型 dependency-wiring；符号 compute_rm_score）：奖励循环核心计算逻辑，新增数据填充以解决分片不均问题，避免多 worker 时数据分配遗漏。
- verl/workers/config/critic.py（模块 配置；类别 source；类型 core-logic）：配置模型字段名重命名（model_config → model），影响所有 CriticConfig 的调用方，是配置兼容性关键点。

关键符号：_setup, _compute_values, _update_critic, compute_rm_score, collect_lora_params, _create_critic_class

关键源码片段

verl/experimental/reward_loop/reward_loop.py

奖励循环核心计算逻辑，新增数据填充以解决分片不均问题，避免多 worker 时数据分配遗漏。

```
# verl/experimental/reward_loop/reward_loop.py compute_rm_score 方法
from verl.protocol import DataProto, pad_dataproto_to_divisor

def compute_rm_score(self, data: DataProto) -> DataProto:
    if self.reward_model_manager is not None:
        self.reward_model_manager.wake_up()

    num_workers = len(self.reward_loop_workers)
    # 将数据填充至能被 num_workers 整除的长度，确保分片均衡
    padded_data, pad_size = pad_dataproto_to_divisor(data, num_workers)
    chunks = padded_data.chunk(num_workers)

    outputs = ray.get([
        worker.compute_score_batch.remote(chunk)
        for worker, chunk in zip(self.reward_loop_workers, chunks, strict=True)
    ])
    outputs_flat = [item for sublist in outputs for item in sublist]
    if pad_size > 0:
        outputs_flat = outputs_flat[:len(data)] # 移除填充部分

    scores = [item["reward_score"] for item in outputs_flat]
    rm_scores = self.reward_manager_cls.assemble_rm_scores(data, scores)
    # ... 后续计算
```

verl/workers/config/critic.py

配置模型字段名重命名（model_config → model），影响所有 CriticConfig 的调用方，是配置兼容性关键点。

```
# verl/workers/config/critic.py CriticConfig 类
class CriticConfig(BaseConfig):
    # ... 其他字段
    _keys_not_saved = [
```

```
            "ppo_mini_batch_size",
            "ppo_micro_batch_size",
            "engine",
            "model", # 原为 "model_config"，改为 "model"
        ]
        strategy: str = MISSING
```

评论区精华

唯一实质 review 来自维护者 wuxibin89：在 verl/workers/engine_workers.py 中新增 temperature=1.0 被提出异议，认为该值已在 trainer 层显式设置（verl/trainer/ppo/v1/trainer_base.py#L417），不应在 worker 中重复默认赋值，以免导致后续覆盖或混淆。该评论未形成结论，但说明了对参数传递职责边界的关注。

- engine_workers.py 中重复设置 temperature (design): 未明确解决，但建议删除该行。当前 PR 仍保留该行，需后续跟进。

风险与影响

- 风险：
  1. 兼容性风险：默认 trainer 切换为 breaking change，外部用户依赖 V0 的配置和脚本需要手动设置 use_v1=False，CI 中已有部分配置被更新，但用户侧可能遗漏。
  2. 配置键名变更：model_config 改为 model 会影响所有传递 CriticConfig 的代码，虽然已同步修改所有已知引用，但可能存在第三方程 序或自定义脚本未覆盖。
  3. 温度参数冗余：engine_workers.py 中 temperature=1.0 的硬编码可能与 trainer 传递的温度冲突，需确认是否会被覆盖或忽略。
  4. 奖励循环性能：pad_dataproto_to_divisor 的引入会增加少量额外通信与计算开销，但影响应较小。 - 影响：用户：现有使用默认配置的用户将自动切换到 V1 trainer，需验证训练流程是否正常；若使用 V0 特定功能（如不兼容 V1 的奖励配置），需显式降级。团队：此 PR 是 V1 成熟化的里程碑，后续新功能将默认基于 V1 开发，V0 维护负担降低。系统：CI 配置全面转向 V1，覆盖了 Megatron、FSDP、TRTLLM 等多种后端，确保了兼容性。影响范围中等，但涉及配置文件和多个核心模块。 - 风险标记：breaking change，配置键名变更，温度参数冗余，奖励循环填充开销

关联脉络

- PR #6818 [reward] feat: colocated reward model for v1 sync/colocate_async trainer: 为 V1 trainer 添加共置奖励模型支持，是本 PR 启用 V1 的前置依赖。
- PR #6796 [fully_async, trainer] fix: align aggregated metrics logging with current step: 修复 fully async trainer 的指标日志，V1 trainer 也使用了类似逻辑。