PR #6162 完整报告

verl-project/verl

[repo] refactor: move experimental/vla to standalone verl-vla repository

合并时间：2026-04-27 13:31

原文链接：http://prhub.com.cn/verl-project/verl/pull/6162

执行摘要

- 一句话：将 experimental/vla 模块迁移至独立仓库 verl-vla
- 推荐动作：建议阅读以了解仓库拆分、模块解耦的工程实践。VLA 相关开发者应关注新仓库 verl-project/verl-vla。

功能与动机

VLA stack has grown significantly in complexity (including environments, datasets, algorithms, and training pipelines). Keeping it under experimental/ introduces tight coupling with the core verl codebase, slower iteration for VLA-specific features, and increased maintenance burden for unrelated modules. (引自 PR body)

实现拆解

1. 确定迁移范围：识别出需要迁出的 VLA 模块位于 experimental/vla 目录，包含 OpenVLA-OFT 模型、pi0-torch 模型、SAC 算法、机器人环境、训练器等多个子模块。
2. 删除源代码：从主仓库中移除 experimental/vla 下所有文件，共删除 50 个文件、约 12,490 行代码。
3. 迁移至新仓库：将删除的代码完整迁移至新建的独立仓库 verl-project/verl-vla，并为其建立独立的依赖管理、CI 与文档体系。本 PR 不涉及对新仓库的联动修改，若主仓库其他位置存在引用，将在后续 PR 中更新。

关键文件：

- verl/experimental/vla/models/openvla_oft/modeling_prismatic.py（模块 VLA 模型；类别 source；类型 deletion；符号 unpack_tuple, wrapper, _ls_new_forward, ls_apply_patch）：OpenVLA-OFT 模型核心文件，包含 PrismaticVisionBackbone、补丁逻辑等，被删除 2000 行，是 VLA 模块中最关键的文件。
- verl/experimental/vla/models/pi0_torch/model/paligemma_with_expert.py（模块 pi0 模型；类别 source；类型 deletion；符号 get_transformers_siglip_vision_config, GemmaRMSNorm, init, _norm）：pi0 模型的核心组件，包含 PaliGemmaWithExpertModel、SiglipVisionTransformer 等，被删除 722 行。
- verl/experimental/vla/rob_ray_trainer.py（模块 训练器；类别 source；类型 deletion；符号 compute_response_mask, flatten_trajectories, RobRayPPOTrainer, _start_profiling）：机器人 PPO 训练器的主入口，包含 RobRayPPOTrainer 类，被删除 666 行，是 VLA 训

练流程的关键文件。

关键符号：PrismaticVisionBackbone, PI0ForActionPrediction, RobRayPPOTrainer, SACReplayPool, LiberoEnv

关键源码片段

verl/experimental/vla/models/openvla_oft/modeling_prismatic.py

OpenVLA-OFT 模型核心文件，包含 PrismaticVisionBackbone、补丁逻辑等，被删除 2000 行，是 VLA 模块中最关键的文件。

```
# 被删除的 OpenVLA-OFT 模型实用函数与 PrismaticVisionBackbone 定义

'''来自原始文件的实用工具：用于避免 HF Transformers 参数覆盖的猴子补丁函数'''


def unpack_tuple(fn: Callable[[Any], tuple[Any]]) -> Callable[[Any], Any]:
    '''将返回元组的函数包装成返回单一元素'''

    def wrapper(*args, **kwargs):
        result = fn(*args, **kwargs)
        return result[0] if isinstance(result, tuple) else result
    return wrapper


def ls_apply_patch(ls_module: LayerScale):
    '''将 TIMM 的 LayerScale 中名为 `gamma` 的参数重映射为 `scale_factor`，
    避免与 HF Transformers 的参数重写冲突'''
    ls_module.scale_factor = nn.Parameter(ls_module.gamma.clone())
    ls_module.forward = _ls_new_forward.__get__(ls_module, LayerScale)
    del ls_module.gamma


class PrismaticVisionBackbone(nn.Module):
    '''支持单/双融合视觉主干（SigLIP / DINOv2）的特征提取器'''
    def __init__(self, use_fused_vision_backbone, image_sizes, timm_model_ids, timm_override_
    act_layers):
        super().__init__()
        self.use_fused_vision_backbone = use_fused_vision_backbone
        # 创建主要特征器（单主干或双主干中的第一个）
        self.featurizer = self._create_featurizer(
            model_id=timm_model_ids[0], img_size=image_sizes[0], act_layer=timm_override_act_
            layers[0]
        )
        if use_fused_vision_backbone:
            # 双主干时创建第二个特征器，并与第一个融合
            self.fused_featurizer = self._create_featurizer(
                model_id=timm_model_ids[1], img_size=image_sizes[1], act_layer=timm_override_act_
                layers[1]
```

)

评论区精华

- 自动化审查发现潜在 Bug：gemini-code-assist[bot] 指出 sac_actor.py 中梯度累积步数计算错误地包含了 world_size，导致 FSDP 下有效学习率异常。由于该文件已被删除，问题无需在当前 PR 中修复。
- 仓库管理员批准：wuxibin89 批准了该 PR，认可直接移除的决策。
  - SAC actor 梯度累积 Bug (correctness): 代码已被删除，不需要在当前 PR 中修复。
  - PR 审批 (other): 同意合并。

风险与影响

- 风险：
  - 引用断裂风险：若主仓库其他模块（如 verl/trainer、verl/workers）存在对 experimental/vla 的导入，删除后将导致 ImportError，需全面扫描并更新或移除这些引用。
  - 用户中断：已使用 verl 中 VLA 功能的用户需迁移至新仓库，可能导致暂时不可用。
  - 遗留 Bug 封闭：被删除代码中可能包含未修复的 Bug（如自动审查发现的梯度累积问题），这些 Bug 不再在主仓库中修复，需由新仓库维护者处理。
- 影响：
  - 对用户：VLA 相关功能从 verl 中移除，需安装 verl-vla 独立包才能继续使用。
  - 对核心系统：主仓库代码量减少约 12,490 行，experimental 目录结构更清晰，不再受 VLA 模块的复杂性影响。
  - 对团队：VLA 团队获得独立迭代能力，不再受核心仓库 PR 流程约束；核心团队减少维护负担。
  - 风险标记：模块外部依赖可能存在引用，用户功能中断，遗留 Bug 未修复

关联脉络

- 暂无明显关联 PR