

# PR #6644 完整报告

verl-project/verl

[veomni] fix: pass mixed precision config to veomni

合并时间: 2026-06-08 10:08

原文链接: <http://prhub.com.cn/verl-project/verl/pull/6644>

## 执行摘要

- 一句话: 修复 VeOmni 混合精度配置传递
- 推荐动作: 作为一个接口对齐的小型 bugfix, 变更清晰、自包含, 已获 Committer 批准, 可以直接合并, 无需额外 review。

## 功能与动机

VeOmni 的 API 从旧版 `enable_mixed_precision` 布尔值升级为 `MixedPrecisionConfig` 对象, 原有代码仅在做模型并行化时临时转换, 但基础模型加载的 dtype 选择仍直接使用原始布尔值, 导致两者不一致。PR body 明确说明根因在于 'VeOmni API expects `mixed_precision` instead of the old `enable_mixed_precision` boolean'。

## 实现拆解

1. 引入 `MixedPrecisionConfig` 导入: 在文件头导入新增 `from veomni.arguments import MixedPrecisionConfig, OpsImplementationConfig`, 确保可实例化配置对象。
2. 创建统一配置对象: 在 `_build_model_optimizer` 方法中, 将 `self.engine_config.mixed_precision` 布尔值包装为 `veomni_mixed_precision_config = MixedPrecisionConfig(enable=self.engine_config.mixed_precision)`, 作为后续所有混合精度判断的唯一来源。
3. 对齐模型加载 dtype: 将 `build_foundation_model` 的 `torch_dtype` 参数判断从 `self.engine_config.mixed_precision` 改为 `veomni_mixed_precision_config.enable`, 保证逻辑等价且数据源一致。
4. 修正并行化 API 参数: 将 `build_parallelize_model` 调用中的 `enable_mixed_precision=...` 替换为 `mixed_precision=veomni_mixed_precision_config`, 符合 VeOmni 新 API 签名。

关键文件:

- `verl/workers/engine/veomni/transformer_impl.py` (模块引擎; 类别 source; 类型 dependency-wiring): 唯一变更文件, 集中修复 VeOmni 引擎的混合精度配置传递, 涉及导入、配置对象创建和两处参数调用。

关键符号: `_build_model_optimizer`

## 关键源码片段

## verl/workers/engine/veomni/transformer\_impl.py

唯一变更文件，集中修复 VeOmni 引擎的混合精度配置传递，涉及导入、配置对象创建和两处参数调用。

# 文件: verl/workers/engine/veomni/transformer\_impl.py

# 关键变更集中在 \_build\_model\_optimizer 方法内

from veomni.arguments import MixedPrecisionConfig, OpsImplementationConfig # 新增导入

class VeOmniEngine:

def \_build\_model\_optimizer(self):

# 现有 ops\_implementation 构建 ...

# 新增: 将 engine\_config.mixed\_precision 布尔值包装为 VeOmni 期望的  
 MixedPrecisionConfig 对象

# 这样 build\_parallelize\_model 就能收到正确的配置类型

veomni\_mixed\_precision\_config = MixedPrecisionConfig(  
 enable=self.engine\_config.mixed\_precision  
 )

# 修改前: 直接使用 self.engine\_config.mixed\_precision 判断 dtype

# 修改后: 使用 veomni\_mixed\_precision\_config.enable, 保证与并行化参数同源

module = build\_foundation\_model(  
 config\_path=self.\_get\_model\_config\_path(),  
 weights\_path=self.model\_config.local\_path,  
 torch\_dtype="float32" if veomni\_mixed\_precision\_config.enable else "bfloat16",  
 # ... 其他参数不变 ...  
 )

# 修改前: 传递布尔值给 enable\_mixed\_precision 参数

# 修改后: 传递 MixedPrecisionConfig 对象给 mixed\_precision 参数, 对齐 VeOmni 新版 API

module = build\_parallelize\_model(  
 module,  
 init\_device=self.engine\_config.init\_device,  
 weights\_path=self.model\_config.local\_path,  
 enable\_full\_shard=self.engine\_config.enable\_full\_shard,  
 mixed\_precision=veomni\_mixed\_precision\_config, # 原为 enable\_mixed\_precision=self.  
 engine\_config.mixed\_precision  
 # ... 其他参数不变 ...  
 )

## 评论区精华

没有实质性 review 讨论。仅 Gemini Code Assist 机器人发表了无内容的评论（告知其功能将于 2026 年停止）。

- 暂无高价值评论线程

## 风险与影响

- 风险：本次变更仅涉及一个文件内的三处修改，且逻辑等价（布尔值到 MixedPrecisionConfig 的包装），回归风险极低。主要风险在于若 VeOmni 的 MixedPrecisionConfig 构造函数签名未来变动，可能需同步更新。但未引入新的测试覆盖，若 VeOmni 环境安装有滞后，CI 可能无法验证。
- 影响：仅影响 VeOmni 后端引擎的混合精度配置流程，对使用其他引擎（FSDP、Megatron、vLLM）的训练不受影响。用户无需调整配置，无外部 API 变更。
- 风险标记：缺少测试覆盖

## 关联脉络

- PR #6505 [config] add missing config fields: PR body 提及该 PR 添加了缺失的配置字段，但未覆盖本次 VeOmni 的 API 适配，是本 PR 的铺垫关联。
- PR #6612 [veomni] feat: reduce extra memory from  $O(E)$  to  $O(E/ep\_size)$  during weight refit: 同为 VeOmni 引擎变更，涉及同一文件 transformer\_impl.py，展示 VeOmni 模块的活跃开发。