PR #6265 完整报告

verl-project/verl

[reward, cfg] fix: correctly use RewardModelConfig in reward config

合并时间：2026-05-08 17:47

原文链接：http://prhub.com.cn/verl-project/verl/pull/6265

执行摘要

- 一句话：修复 RewardModelConfig 字段命名、world_size 计算和 YAML 配置缺失项
- 推荐动作：值得精读，尤其是理解 reward model 如何复用 RolloutConfig 以及 world_size 计算公式。设计决策：将 reward 推理配置视为 rollout 配置的复用，并统一字段名 rollout。对于正在使用 reward model 的开发者和用户，建议检查现有 YAML 配置是否使用了 inference 字段，并及时迁移。关注后续是否会有更多同步配置的工作（例如 NPU 专用配置）。

功能与动机

解决 reward model 配置中的四个相关问题：

1) world size 计算未包含 pipeline_model_parallel_size，导致资源分配错误； 2) RewardModelConfig 中嵌套的 rollout 配置被命名为 inference，与 RewardModelManager 中的引用 rollout 不一致； 3) reward.yaml 缺少顶层和子配置段的 target，导致 Hydra 实例化失败，且 RewardManagerConfig 的 target 路径错误（指向 reward_model 而非 reward）； 4) reward.yaml 中缺失多个 RolloutConfig 关键字段（如 pipeline_model_parallel_size、quantization、temperature），导致这些设置无法传递给 reward model。这些 bug 源于之前 #6228 的修复不完整以及 #6258 的回滚。

实现拆解

1. 统一配置字段名：在 verl/workers/config/reward.py 中将 RewardModelConfig 的 inference: RolloutConfig 重命名为 rollout，与 RewardModelManager 中已有的引用一致，同时与其他 rollout 配置风格对齐。相应地更新 verl/experimental/reward_loop/reward_model.py 中所有 self.config.inference 引用为 self.config.rollout。
2. 修正 world_size 计算并增强健壮性：在 RewardModelManager._initialize_llm_servers 中，将 rollout_world_size 从仅取 tensor_model_parallel_size 改为乘以 data_parallel_size 和 pipeline_model_parallel_size，以正确反映 Ray 资源池的分片需求。同时在计算 num_replicas 后添加断言 num_replicas > 0，当 GPU 资源不足时给出明确错误信息而非在空列表中失败。
3. 修复 YAML 配置结构：在 verl/trainer/config/reward/reward.yaml 中：
   - 添加顶层 _target_: verl.workers.config.RewardConfig。

- 修正 reward_manager._target_ 路径从 verl.workers.config.reward_model.RewardManagerConfig 为 verl.workers.config.reward.RewardManagerConfig。
- 为 reward_model 段添加 _target_: verl.workers.config.RewardModelConfig。
- 为 sandbox_fusion 段添加 _target_: verl.workers.config.SandboxFusionConfig。
- 在 reward_model.rollout 下补充缺失的字段：pipeline_model_parallel_size、layered_summon、scheduling_policy、multi_stage_wake_up、ignore_eos、temperature、top_k、top_p、do_sample、n、quantization、quantization_config_file、mtp、disaggregation 等。注意 n 在 YAML 中加引号以免被解析为 boolean。

4. 同步更新自动生成的配置文件：将同样的变更反映到所有 _generated_ 配置文件（_generated_ppo_megatron_trainer.yaml、_generated_ppo_torchtitan_trainer.yaml、_generated_ppo_trainer.yaml、_generated_ppo_veomni_trainer.yaml），确保通过脚本生成的训练配置与 reward.yaml 一致。
5. 修复 SGLang MTP 条件为空的问题：在 verl/workers/rollout/sglang_rollout/async_sglang_server.py 中，将 if self.config.mtp.enable 改为 if self.config.mtp is not None and self.config.mtp.enable，避免当 mtp 配置字段为 None 时访问属性导致 AttributeError。这修复了 reward_model_sglang CI 测试。

这些变更没有修改任何公共 API，但用户如果直接使用老 YAML（含有 reward_model.inference）需要更新为 rollout。

关键文件：

- verl/experimental/reward_loop/reward_model.py（模块 奖励循环；类别 source；类型 data-contract；符号 RewardModelManager, _initialize_llm_servers）：核心修复文件：修正 world_size 计算、添加断言、更新字段引用。
- verl/workers/config/reward.py（模块 配置定义；类别 source；类型 core-logic；符号 RewardModelConfig）：配置数据类定义，字段名 inference → rollout，是字段统一的关键。
- verl/trainer/config/reward/reward.yaml（模块 奖励配置；类别 config；类型 configuration）：YAML 配置修复，补充 target、路径修正、添加大量缺失字段，是配置完整性的关键。
- verl/workers/rollout/sglang_rollout/async_sglang_server.py（模块 推理引擎；类别 source；类型 bugfix；符号 AsyncSglangServer.launch_server）：修复 MTP 条件避免 None 访问，影响 SGLang rollouts，并修复 CI。
- verl/trainer/config/_generated_ppo_megatron_trainer.yaml（模块 训练配置；类别 config；类型 configuration）：同步更新自动生成的 Megatron 训练配置，确保一致。
- verl/trainer/config/_generated_ppo_torchtitan_trainer.yaml（模块 训练配置；类别 config；类型 configuration）：同步更新自动生成的 TorchTitan 训练配置。
- verl/trainer/config/_generated_ppo_trainer.yaml（模块 训练配置；类别 config；类型 configuration）：同步更新自动生成的 PPO 训练配置。
- verl/trainer/config/_generated_ppo_veomni_trainer.yaml（模块 训练配置；类别 config；类型 configuration）：同步更新自动生成的 VeOmni 训练配置。

关键符号：RewardModelManager._initialize_llm_servers, RewardModelManager.init, AsyncSglangServer.launch_server (MTP 条件), RewardModelConfig.inference -> rollout

关键源码片段

verl/experimental/reward_loop/reward_model.py

核心修复文件：修正 world_size 计算、添加断言、更新字段引用。

```
    def _initialize_llm_servers(self):
        rollout_config = self.config.rollout # 字段已统一为 rollout
        # 正确计算 rollout_world_size：tensor_model_parallel_size * data_parallel_size * pipeline_
        model_parallel_size
        rollout_world_size = (
            rollout_config.tensor_model_parallel_size
            * rollout_config.data_parallel_size
            * rollout_config.pipeline_model_parallel_size
        )
        world_size = (
            self.resource_pool.world_size
            if self.resource_pool # colocate 模式
            else self.config.n_gpus_per_node * self.config.nnodes # standalone 模式
        )
        num_replicas = world_size // rollout_world_size
        # 断言防止资源不足导致空列表无声失败
        assert num_replicas > 0, (
            f"Not enough GPUs to run the reward model. "
            f"world_size ({world_size}) < rollout_world_size ({rollout_world_size}). "
            f"Check your resource pool or standalone config (n_gpus_per_node, nnodes)."
        )

        rollout_replica_class = get_rollout_replica_class(rollout_config.name)
        model_config = HFModelConfig(path=self.config.model_path)
        self.tokenizer = model_config.get_processor()
        self.rollout_replicas = [
            rollout_replica_class(
                replica_rank=replica_rank,
                config=rollout_config,
                model_config=model_config,
                gpus_per_node=self.config.n_gpus_per_node,
                is_reward_model=True,
            )
            for replica_rank in range(num_replicas)
        ]
```

verl/trainer/config/reward/reward.yaml

YAML 配置修复，补充 target、路径修正、添加大量缺失字段，是配置完整性的关键。

```
# Inference config for reward models (now under rollout)
reward_model:
```

```
_target_: verl.workers.config.RewardModelConfig
enable: False
enable_resource_pool: False
n_gpus_per_node: 8
nnodes: 0
model_path: null
rollout: # 统一字段名
  _target_: verl.workers.config.RolloutConfig
  name: ???
  dtype: bfloat16
  gpu_memory_utilization: 0.5
  enforce_eager: true
  cudagraph_capture_sizes: null
  free_cache_engine: true
  data_parallel_size: 1
  expert_parallel_size: 1
  tensor_model_parallel_size: 2
  pipeline_model_parallel_size: 1 # 新增：支持 PP
  max_num_batched_tokens: 8192
  max_model_len: null
  max_num_seqs: 1024
  load_format: auto
  layered_summon: false # 新增
  scheduling_policy: fcfs # 新增
  multi_stage_wake_up: false # 新增
  engine_kwargs: {}
  limit_images: null
  enable_chunked_prefill: true
  enable_prefix_caching: true
  disable_log_stats: true
  skip_tokenizer_init: false
  ignore_eos: false # 新增
  temperature: 1.0 # 新增
  top_k: -1 # 新增
  top_p: 1 # 新增
  do_sample: true # 新增
  n: 1 # 注意：YAML 中需防解析为 bool，reward.yaml 省略引号但 generated 文件加引号
  prompt_length: 2048
  response_length: 2048
  quantization: null # 新增
  quantization_config_file: null # 新增
  mtp: null # 新增
  disaggregation: # 新增
    enabled: false
    prefill_replicas: 1
    decode_replicas: 1
    decode_tensor_model_parallel_size: null
    transfer_backend: nixl
    bootstrap_port: null
```

ib_device: null

评论区精华

- num_replicas 断言（gemini-code-assist, correctness）：指出 world_size 小于 rollout_world_size 时 num_replicas 可能为 0，导致空列表和后续初始化失败，建议添加明确断言。已采纳，在 _initialize_llm_servers 中添加了 assert num_replicas > 0 并给出详细错误信息。
- YAML 中 n 键引号（gemini-code-assist, correctness）：建议将 n:1 加引号为 'n':1 以避免 YAML 1.1 解析为 boolean false。作者回应遵循了 rollout 配置的相同模式（即未加引号），但最终在自动生成的配置文件中加了引号，而 reward.yaml 中保留无引号形式。该差异未影响实际使用。
- 统一字段名为 rollout（yyDing1, design）：要求统一使用 reward_model.rollout 使命名一致且易于理解。已采纳，作者修改了所有相关引用。
- 修复 CI 测试（yyDing1, testing）：要求修复 reward_model_sglang CI。已修复，通过修改 SGLang MTP 条件解决。
- 添加 num_replicas > 0 断言防止无声失败 (correctness): 作者采纳，在代码中添加了 assert num_replicas > 0 并给出详细错误信息。
- YAML 中 'n' 键引号问题 (correctness): reward.yaml 中未加引号，但 generated 文件中加了引号。差异未影响实际使用。
- 统一字段名 inference→rollout (design): 作者修改了 RewardModelConfig 和所有引用，使用 rollout。
- 修复 CI reward_model_sglang (testing): 修复了 SGLang 条件判断，避免 None 访问错误。

风险与影响

- 风险：
  1. 配置兼容性风险：字段名从 inference 改为 rollout，用户自定义 YAML 或编程引用旧字段会导致实例化失败。但由于该配置区域之前处于错误状态（缺少 _target_ 等），有效使用场景有限，影响较小。
  2. 资源分配变化风险：rollout_world_size 现在正确包含 data_parallel_size 和 pipeline_model_parallel_size，对于使用多副本 reward model 的场景，所需 GPU 数量可能增加（之前被低估）。用户需确保资源池或 standalone 配置有足够 GPU，否则新添加的断言将阻止启动并给出明确错误。
  3. 缺少 GPU 测试覆盖：PR 仅运行了 CPU 配置测试和一次完整训练（vLLM），但未针对不同并行策略（PP、DP 组合）添加专门的 GPU 测试，可能存在 corner case。
  4. 自动生成配置文件同步：虽然已更新四个 _generated_ 文件，但可能存在其他由不同工具链生成的配置未同步，需关注后续 CI。
- 影响：
  - 用户：使用 reward model 的用户现在可以配置 pipeline_model_parallel_size、quantization、temperature 等参数，使 reward 推理更灵活。需注意 YAML 字段名变更。

- 系统：reward model 的 GPU 分配更准确，资源不足时快速报错而非挂起。
- 团队：统一配置风格降低了后续维护困惑。修复与 #6228 和 #6258 相关的反复回滚问题，明确正确实现。
- 风险标记：配置字段不兼容，资源分配变化，缺少 GPU 测试覆盖

关联脉络

- PR #6258 Revert "[reward] fix: compute correct rollout world size": 本 PR 修复了被该回滚撤销的 world_size 计算问题，并进一步修复了配置字段名和缺失项。