PR #6184 完整报告

verl-project/verl

[veomni] feat: use VeOmni's native return_log_probs path to compute log_probs

合并时间：2026-05-06 10:22

原文链接：http://prhub.com.cn/verl-project/verl/pull/6184

执行摘要

- 一句话：使用 VeOmni 原生路径计算 log_probs/entropy，修复 fused kernel crash
- 推荐动作：值得阅读。展示了如何通过 VeOmni 的原生 API 替代手工补丁，以及如何通过 OpsImplementationConfig 统一 kernel 选择。设计清晰，文档丰富。

功能与动机

VeOmniEngine 在 use_fused_kernels=True 时，FSDPEngine.prepare_model_outputs 在 _compute_old_log_prob 处因 CausalLMOutputWithPast 对象缺少 log_probs 属性而崩溃。此 PR 通过 VeOmni 的原生 return_log_probs 路径填充这些字段，从而消除 verl 侧的手动 patch 依赖。

实现拆解

1. 在 verl/workers/engine/veomni/transformer_impl.py 中，从 veomni.arguments 导入 OpsImplementationConfig，在 _build_model_optimizer 中根据引擎配置构造 OpsImplementationConfig 实例并传递给 build_foundation_model，使 VeOmni 的 per-model patches 能根据配置选择 kernel backend。
2. 在 VeOmniEngineWithLMHead.prepare_model_inputs 中，当 use_fused_kernels=True 且 use_remove_padding=True 时，将 labels（packed input_ids_rmpad）、shift_labels（already-rolled labels）和 return_log_probs=True 注入 model_inputs，激活 VeOmni 的 ForCausalLMLoss 内部短路的 chunked log-probs/entropy 路径。
3. 在 verl/workers/config/engine.py 的 VeOmniEngineConfig 中加入 5 个 kernel 实现选择字段（cross_entropy_loss, rms_norm, swiglu_mlp, rotary_pos_emb, load_balancing_loss），默认均为 "eager"，供用户通过 Hydra 配置。
4. 更新 Hydra schema yamls（engine/veomni.yaml, _generated_ppo_veomni_trainer.yaml, ref/veomni_ref.yaml）以包含新字段，确保启动解析。
5. 将 CI workflows 中的 veomni 依赖版本从 0.1.9a1 升级到 0.1.9a4，以包含 return_log_probs 路径所需的 #720 特性。

关键文件：

- verl/workers/engine/veomni/transformer_impl.py（模块 VeOmni 引擎；类别 source；类型 core-logic；符号 VeOmniEngine._build_model_optimizer,

VeOmniEngineWithLMHead.prepare_model_inputs）：核心变更：构造 OpsImplementationConfig、注入 return_log_probs 参数

- verl/workers/config/engine.py（模块 配置定义；类别 source；类型 configuration；符号 VeOmniEngineConfig）：添加 5 个 kernel 实现选择字段到 VeOmniEngineConfig
- verl/trainer/config/engine/veomni.yaml（模块 引擎配置；类别 config；类型 configuration）：Hydra 默认配置增加 5 个 implementation 字段
- verl/trainer/config/_generated_ppo_veomni_trainer.yaml（模块 引擎配置；类别 config；类型 configuration）：生成的训练配置同步添加 implementation 字段
- verl/trainer/config/ref/veomni_ref.yaml（模块 引擎配置；类别 config；类型 configuration）：参考配置更新
- .github/workflows/e2e_ppo_trainer_veomni_vllm.yml（模块 CI 工作流；类别 infra；类型 infrastructure）：CI 依赖版本升级到 0.1.9a4
- .github/workflows/e2e_ppo_trainer_veomni_vllm_ascend.yml（模块 CI 工作流；类别 infra；类型 infrastructure）：CI 依赖版本升级
- .github/workflows/e2e_sft_llm.yml（模块 CI 工作流；类别 infra；类型 infrastructure）：CI 依赖版本升级
- .github/workflows/e2e_sft_llm_ascend.yml（模块 CI 工作流；类别 infra；类型 infrastructure）：CI 依赖版本升级
- .github/workflows/e2e_sft_vlm.yml（模块 CI 工作流；类别 infra；类型 infrastructure）：CI 依赖版本升级

关键符号：VeOmniEngine._build_model_optimizer, VeOmniEngineWithLMHead.prepare_model_inputs

关键源码片段

verl/workers/engine/veomni/transformer_impl.py

核心变更：构造 OpsImplementationConfig、注入 return_log_probs 参数

```
# 文件：verl/workers/engine/veomni/transformer_impl.py

from veomni.arguments import OpsImplementationConfig

class VeOmniEngineWithLMHead(VeOmniEngine):
    # ... 其他代码

    def _build_model_optimizer(self):
        # 构造 OpsImplementationConfig，传递给 build_foundation_model，
        # 使 VeOmni 的 per-model patches 能根据配置选择 kernel backend
        ops_implementation = OpsImplementationConfig(
            attn_implementation=self.engine_config.attn_implementation,
            moe_implementation=self.engine_config.moe_implementation,
            cross_entropy_loss_implementation=self.engine_config.cross_entropy_loss_
            implementation,
            rms_norm_implementation=self.engine_config.rms_norm_implementation,
```

```
            swiglu_mlp_implementation=self.engine_config.swiglu_mlp_implementation,
            rotary_pos_emb_implementation=self.engine_config.rotary_pos_emb_implementation,
            load_balancing_loss_implementation=self.engine_config.load_balancing_loss_
            implementation,
        )
        module = build_foundation_model(
            config_path=self.model_config.local_hf_config_path,
            weights_path=self.model_config.local_path,
            torch_dtype="float32" if self.engine_config.mixed_precision else "bfloat16",
            ops_implementation=ops_implementation, # 传递 ops_implementation
            init_device=self.engine_config.init_device,
        )
        # ...

    def prepare_model_inputs(self, micro_batch: TensorDict):
        # ... 现有逻辑

        # 激活 VeOmni 的 chunk_logprobs 路径：
        # 当 use_fused_kernels=True 且使用 remove padding 时，注入以下参数：
        use_fused_kernels = tu.get_non_tensor_data(data=micro_batch, key="use_fused_kernels",
        default=False)
        if use_fused_kernels and use_remove_padding:
            # labels: packed input_ids_rmpad
            model_inputs["labels"] = input_ids_rmpad
            # shift_labels: 已经 rolled 的 labels，unsqueeze 为 [1, total_nnz]
            shift_labels = output_args["input_ids_rmpad_rolled"].unsqueeze(0)
            model_inputs["shift_labels"] = shift_labels
            # 设置 return_log_probs=True，使 ForCausalLMLoss 短路到 chunk_logprobs_function
            model_inputs["return_log_probs"] = True

        return model_inputs, output_args
```

评论区精华

PR review 讨论较少，wuxibin89 直接批准。gemini-code-assist 自动审查未提出具体问题。无可争议点。

- 代码审查自动反馈 (other): 无需处理

风险与影响

- 风险：
  - VeOmni 版本依赖：必须 ≥0.1.9a4，否则缺少 #720 导致 entropy 缺失。
  - shift_labels 假设：依赖于 VeOmni chunk_logprobs_function 内部跳过 causal shift，若 VeOmni 侧行为变化可能导致对齐失败。
  - 配置字段兼容性：新字段默认值均为 "eager"，向后兼容，但用户自定义时需确保 VeOmni 版本支持相应 backend。
- 影响：

- 用户：VeOmni 用户现在可以启用 use_fused_kernels=True 而不必担心崩溃，并可通过配置精细控制 kernel 实现（如为 DeepSeek-V3 设置 rms_norm_implementation=triton 实现 bitwise 对齐）。
- 系统：减少 verl 侧 monkey-patch 代码，降低维护成本；将 kernel 选择责任转移到 VeOmni。
- 团队：便于支持更多模型，无需为每个模型编写 verl 侧 forward override。
- 风险标记：依赖 VeOmni 版本 0.1.9a4, shift_labels 对齐假设，新配置字段默认值兼容

关联脉络

- 暂无明显关联 PR