PR #6231 完整报告

verl-project/verl

[trainer] fix: update TorchTitanEngine for latest torchtitan API

合并时间：2026-05-06 17:38

原文链接：http://prhub.com.cn/verl-project/verl/pull/6231

执行摘要

- 一句话：修复 TorchtitanEngine 适配最新 torchtitan API
- 推荐动作：建议合并，这是一个兼容性和 bug 修复 PR，解决已知问题并跟上上游变化。团队可以关注 torchtitan 的稳定性，未来考虑锁定依赖版本。

功能与动机

torchtitan 上游 API 发生了 breaking change，如 expert_tensor_parallel_degree 被删除、model_registry 新增 attn_backend 参数等，导致 TorchtitanEngine 无法正常工作。Issue #6182 报告了 attn_type="flex" 被静默忽略的问题，根本原因是 model_registry 调用时未传递 attn_backend。

实现拆解

1. 调整 parallelism 配置: 删除 expert_tensor_parallel_degree（已在 torchtitan 上游移除）和 etp 参数。
2. 修复 attn_type 传递: 在 model_registry() 调用中传入 attn_backend=self.engine_config.attn_type，替换之前的手动覆写 model_spec.model.layer.attention.attn_backend 的逻辑。
3. 修复 get_attention_masks 调用: 将 attn_type 的读取路径从 self.trainer.model_config.layer.attention.attn_backend 修正为 self.engine_config.attn_type，解决属性不存在的问题。
4. 新增损失函数配置: 导入 CrossEntropyLoss.Config 并添加到 Trainer 配置中，因为 torchtitan 的 Trainer 现在需要显式提供损失函数配置。
5. 重构 flavor 推导逻辑: 在 utils.py 中，使用 model_registry(flavor_name).model 获取模型配置，并支持 n_layers 回退到 len(layers)，以兼容不同配置格式。
6. 修复测试脚本: 在 tests/special_e2e/run_ppo_trainer_torchtitan.sh 中，将硬编码的 TP 大小 8 改为 NUM_GPUS 变量，并修正实验名称。

关键文件：

- verl/workers/engine/torchtitan/transformer_impl.py（模块 引擎；类别 source；类型 dependency-wiring；符号 TorchtitanEngine）：核心引擎文件，修复了 attn_backend 传递、删除已弃用参数、添加 CrossEntropyLoss 配置等关键变更。

- verl/workers/engine/torchtitan/utils.py（模块 工具；类别 source；类型 core-logic；符号 derive_torchtitan_name_and_flavor）：重构了 derive_torchtitan_name_and_flavor 函数，使用 model_registry 并支持 n_layers 回退到 len(layers)。

关键符号：derive_torchtitan_name_and_flavor, TorchtitanEngine.init, TorchtitanEngine.model_forward_step, TorchtitanEngine.prepare_model_inputs, TorchtitanEngine._init_device_mesh

关键源码片段

verl/workers/engine/torchtitan/transformer_impl.py

核心引擎文件，修复了 attn_backend 传递、删除已弃用参数、添加 CrossEntropyLoss 配置等关键变更。

```
    # ... 在 __init__ 方法中
    # 使用 model_registry 并传递 attn_backend 参数
    model_spec = model_module.model_registry(
        torchtitan_flavor,
        attn_backend=self.engine_config.attn_type
    )

    # 注意：不再需要手动覆写 attn_backend，因为 torchtitan 现在负责处理

    # 配置 ParallelismConfig 时删除已弃用的 expert_tensor_parallel_degree
    parallelism = ParallelismConfig(
        data_parallel_replicate_degree=self.engine_config.data_parallel_replicate_size,
        data_parallel_shard_degree=self.engine_config.data_parallel_shard_size,
        fsdp_reshard_ratio=self.engine_config.fsdp_reshard_ratio,
        tensor_parallel_degree=self.engine_config.tensor_parallel_size,
        pipeline_parallel_degree=self.engine_config.pipeline_parallel_size,
        context_parallel_degree=self.engine_config.context_parallel_size,
        expert_parallel_degree=self.engine_config.expert_parallel_size,
        # expert_tensor_parallel_degree 已在 torchtitan 中移除
    )

    # 添加 CrossEntropyLoss.Config，因为 Trainer 需要显式损失配置
    training = TrainingConfig(
        batch_size=self.engine_config.training_batch_size,
        num_steps=self.optimizer_config.total_training_steps,
        compile=CompileConfig(),
        loss=CrossEntropyLoss.Config(), # verl 使用自己的损失函数，此仅为占位
    )
```

verl/workers/engine/torchtitan/utils.py

重构了 derive_torchtitan_name_and_flavor 函数，使用 model_registry 并支持 n_layers 回退到 len(layers)。

```
def derive_torchtitan_name_and_flavor(hf_config) -> tuple[str, str]:
    # ... 获取 model_type 和 name ...
```

```
    model_module = importlib.import_module(f"torchtitan.models.{name}")
    model_registry = model_module.model_registry

    # 通过遍历模块变量找到 flavor 名称列表
    flavor_names = None
    for attr, obj in vars(model_module).items():
        if attr.endswith("_configs") and isinstance(obj, dict):
            flavor_names = list(obj.keys())
            break

    if flavor_names is None:
        raise ValueError(...)

    hidden_size = hf_config.hidden_size
    num_layers = hf_config.num_hidden_layers
    vocab_size = hf_config.vocab_size

    for flavor_name in flavor_names:
        cfg = model_registry(flavor_name).model # 使用 model_registry 构建配置
        n_layers = getattr(cfg, "n_layers", None) or len(getattr(cfg, "layers", [])) # 支持 layers 回退
        if (
            getattr(cfg, "dim", None) == hidden_size
            and n_layers == num_layers
            and getattr(cfg, "vocab_size", None) == vocab_size
        ):
            return name, flavor_name

    raise ValueError(f"No match for {hidden_size}, {num_layers}, {vocab_size}")
```

评论区精华

无 review 评论，PR 由 wuxibin89 直接批准合并。

- 暂无高价值评论线程

风险与影响

- 风险：主要风险在于 torchtitan 上游 API 仍然不稳定，未来可能再次发生 breaking change，需要持续跟踪。同时，本次改动删除了一些参数（如 etp），如果用户配置中依赖这些参数，可能导致配置错误。但该模块仍属于 experimental 阶段，风险可控。
- 影响：影响范围为使用 TorchtitanEngine 的训练任务。修复了 attn_type 被忽略的严重 bug，并确保了与最新 torchtitan 版本的兼容性。用户需要更新 torchtitan 到对应 HEAD 版本才能配合使用。
- 风险标记：上游 API 不稳定，实验性模块，无测试配套

关联脉络

- PR #6182 [trainer] bug: TorchtitanEngine silently ignores attn_type="flex" — no clear BKM for which torchtitan version to use: 该 issue 报告了 attn_type 被忽略的 bug，本 PR 是其修复。