

PR #7461 完整报告

verl-project/verl

[training_utils, env, doc] feat: use Liger fused linear PPO kernel

合并时间: 2026-08-25 11:34

原文链接: <http://prhub.com.cn/verl-project/verl/pull/7461>

执行摘要

- 一句话: 接入 Liger 融合算子, actor 更新提速 13.53%、显存降 5.56%
- 推荐动作: 值得精读。重点看 FusedLinearForPPO.forward 的可选依赖降级模式、2D/3D 形状保真处理, 以及 test_experimental_torch_functional_on_cpu.py 中 FakeLiger 的 dispatch 验证方式。若团队在 GPU 上使用 fused kernel, 建议合入后跑一次 H100 短训练回归; 若维护 NPU/ 其他硬件栈, 需确认 Liger 内核在目标硬件上的可用性。

功能与动机

来自 issue #7424: Liger-Kernel 新增 fused-linear-scaled-cross-entropy 算子, 官方称在 Hopper/Blackwell 上吞吐约 3 倍、内存约 50% 降低, 并提供与其他设备一致的 fallback, 建议作为 verl 融合线性 PPO 的默认实现。本 PR 按该提议接入, 目标是提升 actor 更新速度并降低显存, 同时保持训练行为等价。

实现拆解

1. 可选导入与降级边界: 在 verl/utils/experimental/torch_functional.py 顶部尝试导入 liger_kernel.ops.LigerFusedLinearScaledCrossEntropyFunction, 仅当缺失模块名恰为 liger_kernel 时静默置 None; 其余 ModuleNotFoundError 直接抛出, 避免掩盖损坏的 Liger 安装。
2. 新增 Liger 分支: 在 FusedLinearForPPO.forward 中先校验 hidden_states 是 2D/3D 且 input_ids 形状匹配, 再展平为 2D 并调用 apply(hidden, weight, labels, temperature, -100, 1, True), 将返回的 NLL 取负得到 log_probs, 并 reshape 回原始形状; Liger 缺失时仍走原 FusedLinearForPPOFunction 分块路径, chunk_size 语义不变。
3. 依赖版本提升: setup.py 的 GPU_REQUIRES 与 pyproject.toml 的 fsdp、megatron extras 均由 liger-kernel 提升为 liger-kernel>=0.8.2, uv.lock 同步更新到 0.8.2。
4. 测试配套: 新增 tests/utils/test_experimental_torch_functional_on_cpu.py, 分别覆盖分块 fallback 与原生 torch 前反向一致性 (2D/3D)、Liger dispatch 参数与形状恢复、fallback 保留 chunking。
5. 文档同步: 更新 docs/perf/perf_tuning.rst、docs/examples/config.rst、docs/start/install.rst 与 requirements.txt, 说明新行为、安装要求和回退条件。

关键文件:

- tests/utils/test_experimental_torch_functional_on_cpu.py (模块 算子测试; 类别 test; 类型 test-coverage; 符号 test_fused_linear_for_ppo_chunked_fallback_matches_torch, test_fused_linear_for_ppo_dispatches_to_liger, FakeLigerFusedLinearScaledCrossEntropyFunction, test_fused_linear_for_ppo_fallback_preserves_chunking) : 新增测试文件, 覆盖 fallback 数值一致性、Liger dispatch 参数与形状恢复、fallback chunking 保留, 是本次替换行为的保障。
- verl/utils/experimental/torch_functional.py (模块 算子工具; 类别 source; 类型 core-logic; 符号 FusedLinearForPPO.forward, _LIGER_FUSED_LINEAR_SCALED_CROSS_ENTROPY) : 核心变更文件, FusedLinearForPPO.forward 新增 Liger 分支并保留 fallback。
- setup.py (模块 打包脚本; 类别 source; 类型 configuration) : GPU extras 依赖从 liger-kernel 提升为 >=0.8.2, 保证新路径可用的最低版本。
- pyproject.toml (模块 依赖配置; 类别 config; 类型 configuration) : fsdp 与 megatron extras 同步要求 liger-kernel>=0.8.2, 避免锁文件与依赖声明不一致。
- uv.lock (模块 锁文件; 类别 other; 类型 dependency-lock) : 锁文件同步 liger-kernel 0.8.0 -> 0.8.2, 并更新 requires-dist 版本约束。
- docs/perf/perf_tuning.rst (模块 性能文档; 类别 docs; 类型 documentation) : 说明 Liger 输出头路径、安装要求与 use_liger/use_fused_kernels 的关系。
- docs/examples/config.rst (模块 配置文档; 类别 docs; 类型 documentation) : 配置文档同步提及输出头可选用 Liger scaled cross entropy。
- docs/start/install.rst (模块 安装文档; 类别 docs; 类型 documentation) : 安装文档提示需 liger-kernel>=0.8.2 才能启用新路径。
- requirements.txt (模块 依赖清单; 类别 infra; 类型 configuration) : 依赖清单同步版本约束, 避免手工安装时版本不满足。

关键符号: FusedLinearForPPO.forward, test_fused_linear_for_ppo_chunked_fallback_matches_torch, test_fused_linear_for_ppo_dispatches_to_liger, test_fused_linear_for_ppo_fallback_preserves_chunking

关键源码片段

tests/utils/test_experimental_torch_functional_on_cpu.py

新增测试文件, 覆盖 fallback 数值一致性、Liger dispatch 参数与形状恢复、fallback chunking 保留, 是本次替换行为的保障。

```
# tests/utils/test_experimental_torch_functional_on_cpu.py 关键片段
def test_fused_linear_for_ppo_dispatches_to_liger(monkeypatch):
    calls = []

    # 用假 Liger 算子记录调用参数, 验证 dispatch 与形状处理是否正确
    class FakeLigerFusedLinearScaledCrossEntropyFunction:
        @staticmethod
        def apply(*args):
```

```

        calls.append(args)
        hidden_states = args[0]
        token_count = hidden_states.shape[0]
        nll = torch.arange(token_count, dtype=torch.float32)
        entropy = torch.arange(token_count, dtype=hidden_states.dtype) + 10
        return nll, entropy

monkeypatch.setattr(
    experimental_F,
    '_LIGER_FUSED_LINEAR_SCALED_CROSS_ENTROPY',
    FakeLigerFusedLinearScaledCrossEntropyFunction,
)
hidden = torch.randn(2, 3, 5)
weight = torch.randn(7, 5)
labels = torch.randint(7, (2, 3), dtype=torch.int32)

log_probs, entropy = experimental_F.FusedLinearForPPO()(hidden, weight, labels,
temperature=0.8)

# 只调用一次, 3D 被展平为 2D, 标签被转换为 int64
assert len(calls) == 1
args = calls[0]
assert args[0].shape == (6, 5)
assert args[1] is weight
assert args[2].shape == (6,)
assert args[2].dtype == torch.int64
assert args[3] == 0.8
assert args[4] == -100
assert args[5] == 1
assert args[6] is True

# 输出形状恢复为 (2, 3), 且 log_probs 是 NLL 取负
torch.testing.assert_close(log_probs, -torch.arange(6, dtype=torch.float32).reshape(2, 3))
torch.testing.assert_close(entropy, (torch.arange(6, dtype=hidden.dtype) + 10).reshape(2, 3))

```

verl/utils/experimental/torch_functional.py

核心变更文件, FusedLinearForPPO.forward 新增 Liger 分支并保留 fallback。

```

# verl/utils/experimental/torch_functional.py 关键片段
import torch

```

```

try:
    from liger_kernel.ops import (
        LigerFusedLinearScaledCrossEntropyFunction as _LIGER_FUSED_LINEAR_SCALED_
        CROSS_ENTROPY,
    )

```

```

except ModuleNotFoundError as exc:

```

```

    # 仅当缺失模块确为 liger_kernel 时降级; 若 Liger 内部依赖缺失则直接抛出,
    # 避免掩盖损坏的安装。

```

```

if exc.name != 'liger_kernel':
    raise
_LIGER_FUSED_LINEAR_SCALED_CROSS_ENTROPY = None

class FusedLinearForPPO(torch.nn.Module):
    def __init__(self, chunk_size: int = 512):
        super().__init__()
        self.chunk_size = chunk_size

    def forward(self, hidden_states, vocab_weights, input_ids, temperature=1.0):
        input_ids = input_ids.to(torch.int64)
        # 未安装 Liger 时保留原有分块 autograd 实现，行为完全不变。
        if _LIGER_FUSED_LINEAR_SCALED_CROSS_ENTROPY is None:
            return FusedLinearForPPOFunction.apply(
                hidden_states, vocab_weights, input_ids, temperature, self.chunk_size
            )

        # Liger 算子只接受 2D 输入，先校验再展平，并记录输出形状以便恢复。
        if hidden_states.ndim not in (2, 3):
            raise ValueError(f'hidden_states must be 2D or 3D, got shape {tuple(hidden_states.shape)}')
        if input_ids.shape != hidden_states.shape[:-1]:
            raise ValueError(
                f'input_ids shape {tuple(input_ids.shape)} must match hidden_states shape '
                f'{tuple(hidden_states.shape[:-1])}'
            )

        output_shape = input_ids.shape
        hidden_states = hidden_states.reshape(-1, hidden_states.shape[-1])
        input_ids = input_ids.reshape(-1)

        # Liger 返回 NLL，取负后即 verl 需要的 log_probs；熵直接透传。
        nll, entropy = _LIGER_FUSED_LINEAR_SCALED_CROSS_ENTROPY.apply(
            hidden_states,
            vocab_weights,
            input_ids,
            temperature,
            -100, # ignore_index
            1, # m_tiles
            True, # return_entropy
        )
        log_probs = -nll
        return log_probs.reshape(output_shape), entropy.reshape(output_shape)

```

评论区精华

本 PR 没有实质性的 review 技术讨论（wuxibin89 直接 APPROVED，仅有 CLA 检查），设计决策主要记录在 PR body 与 issue #7424：

- 是否把 Liger 设为默认：PR body 指出保持现配置不变，仅当 liger-kernel $\geq$ 0.8.2 存在时自动走 Liger 路径，避免破坏现有环境。
- 不兼容安装的处理：PR body 明确选择 'surface broken/incompatible Liger installations instead of silently masking import failures'，即暴露问题而非静默回退。
- 数值与行为保真：通过负 NLL、熵直通、形状恢复、标签 int64 转换保证训练语义一致；四 seed reward 对比 CI 跨零，未发现统计可分辨回归。
- 暂无高价值评论线程

风险与影响

- 风险：
 1. GPU 路径缺少真实 kernel 级测试：新增 CPU 测试只能验证 dispatch 与数值公式，无法覆盖 Liger Triton kernel 在 H100 之外的硬件（如 NPU、AMD）上的行为，安装含 liger-kernel 的 NPU 环境可能触发未验证路径。
 2. 依赖强制升级：所有启用 gpu/fsdp/megatron extras 的环境将被拉升到 liger-kernel $\geq$ 0.8.2，对已有 0.8.0 环境属于 breaking；若环境中存在旧版但模块路径兼容的 Liger，导入成功但 API 不一致时会直接抛错而非回退（这是有意设计，但用户需知晓）。
 3. 数值等价性：Liger 的 scaled CE 与 verl 原 log_softmax+gather 在浮点细节上可能不同，导致 log_probs 微小差异；测试使用 assert_close 容忍一定误差，但长训练下 reward 波动未完全排除（benchmark CI 跨零）。
 4. 3D 展平约束：Liger 路径仅支持 hidden_states 为 2D/3D，且要求 input_ids.shape 与 hidden_states.shape[:-1] 一致；未来若出现更高维输入会直接抛 ValueError，属于行为变更。- 影响：对用户：启用 use_fused_kernels + impl_backend: torch 的训练会无感切换到 Liger 路径（需满足 liger-kernel $\geq$ 0.8.2），获得约 13.53% actor 更新提速与约 5% 显存下降；未安装或版本不足的环境行为与原实现完全一致。对系统：训练吞吐与显存占用均有改善，但引入新的可选依赖和双路径维护成本。对团队：需要保持两条路径的数值一致性，后续 Liger 版本升级需回归验证；文档中明确了两者的关系（use_liger 控制模型内部 kernel，use_fused_kernels 控制输出头）。- 风险标记：依赖强制升级 liger-kernel $\geq$ 0.8.2，GPU 路径缺少端到端验证，旧版 Liger 安装将显式报错而非回退，数值等价性依赖 CPU 单测

关联脉络

- 暂无明显关联 PR