PR #6511 完整报告

verl-project/verl

[veomni, fsdp] feat: enable fused top-K distillation kernel for OPD

合并时间：2026-05-29 13:52

原文链接：http://prhub.com.cn/verl-project/verl/pull/6511

执行摘要

- 一句话：为 VeOmni 引擎集成 fused top-K 蒸馏核，避免材料化完整 logits
- 推荐动作：值得精读，尤其是学习如何在引擎层面注入额外张量以支持 fused kernel 的蒸馏功能。输入注入和输出提取的对称设计（eager vs fused 路径输出格式一致）是一个好实践。但需注意该功能目前仅支持 VeOmni backend，且依赖于 veomni 的 bug 修复。

功能与动机

原先 fused 路径只产生 per-token log_probs/entropy，而 top-K 蒸馏需要完整 logits 计算加权交叉熵，因此 OPD 批次必须设置 use_fused_kernels=False，无法享受 fused kernel 的显存节省。VeOmni 0.1.11 的 chunk_topk_distill_function 在分块 fused-linear 前向中内联计算蒸馏损失，避免了材料化完整 [B, L, V] logits。本 PR 将其接入 verl 的 actor 流水线，使得 fused 路径也能支持 top-K 蒸馏。

实现拆解

1. 依赖升级：将 CI 中的 veomni 从 0.1.10 升级到 0.1.11（涉及 5 个 workflow 文件），获取所需的 chunk_topk_distill_function 和 FusedLinearAuxOutput 中的蒸馏字段。
2. VeOmni 引擎输入注入：在 VeOmniEngineWithLMHead.prepare_model_inputs 中，当 use_fused_kernels=True、distillation_use_topk=True 且 micro_batch 包含 teacher_ids 时，从 micro_batch 读取 teacher_ids 和 teacher_logprobs（均从嵌套张量中提取 values() 并 unsqueeze(0) 得到 (1, total_nnz, K) 形状），若启用 Ulysses SP 则通过 slice_input_tensor 沿 seqlen 维度切片，然后分别以 teacher_topk_ids 和 teacher_topk_log_probs 键放入 model_inputs，使其经模型前向传递到 ForCausalLMLoss，进而路由到 chunk_topk_distill_function。同时增加了 paired validation，防止缺少 teacher_logprobs 时静默失败。
3. FSDP 引擎输出提取：在 FSDPEngineWithLMHead.prepare_model_outputs 的 fused 分支中，当 distillation_use_topk=True 且 output.fused_linear_aux 存在时，从 aux_outputs 中逐个提取 distillation_losses、student_mass、teacher_mass，进行 squeeze(0) 和必要的 Ulysses SP 去填充（通过 gather_outputs_and_unpad），最后封装为嵌套张量（nested_tensor_from_jagged），存入 model_output，与 eager 路径的格式一致，使得下游损失函数无需区分路径。
4. 配套示例脚本：新增两个 OPD 示例脚本（run_qwen3_8b_mopd_veomni.sh 多教师 VL 和 run_qwen3_0.6b_opd_veomni.sh 单教师文本），展示如何通过设置

use_fused_kernels=True、distillation_use_topk=True 等参数启用 fused 蒸馏路径，并给出默认超参数配置。

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 VeOmniEngineWithLMHead.prepare_model_inputs）：核心改动：在 prepare_model_inputs 中注入 teacher top-K 张量并支持 SP 切片，是 fused 蒸馏路径的入口。
- verl/workers/engine/fsdp/transformer_impl.py（模块 FSDP 引擎；类别 source；类型 core-logic；符号 FSDPEngineWithLMHead.prepare_model_outputs）：在 prepare_model_outputs 的 fused 分支中提取蒸馏输出，转换为嵌套张量，与 eager 路径格式一致。
- examples/on_policy_distillation_trainer/run_qwen3_8b_mopd_veomni.sh（模块 示例脚本；类别 other；类型 configuration）：新增多教师 OPD 示例脚本，展示 fused 蒸馏配置。
- examples/on_policy_distillation_trainer/run_qwen3_0.6b_opd_veomni.sh（模块 示例脚本；类别 other；类型 configuration）：新增单教师 OPD 示例脚本，展示 fused 蒸馏配置。
- .github/workflows/e2e_ppo_trainer_veomni_vllm.yml（模块 CI 配置；类别 infra；类型 infrastructure）：veomni 依赖从 0.1.10 升级到 0.1.11。
- .github/workflows/e2e_ppo_trainer_veomni_vllm_ascend.yml（模块 CI 配置；类别 infra；类型 infrastructure）：veomni 依赖升级，与 NVIDIA 版本同步。
- .github/workflows/e2e_sft_llm.yml（模块 CI 配置；类别 infra；类型 infrastructure）：veomni 依赖升级，但该 workflow 可能不使用 veomni？但变更一致。
- .github/workflows/e2e_sft_llm_ascend.yml（模块 CI 配置；类别 infra；类型 infrastructure）：veomni 依赖升级，与 NVIDIA 版本同步。
- .github/workflows/e2e_sft_vlm.yml（模块 CI 配置；类别 infra；类型 infrastructure）：veomni 依赖升级，但该 workflow 可能不使用 veomni？但变更一致。

关键符号：VeOmniEngineWithLMHead.prepare_model_inputs, FSDPEngineWithLMHead.prepare_model_outputs

关键源码片段

verl/workers/engine/veomni/transformer_impl.py

核心改动：在 prepare_model_inputs 中注入 teacher top-K 张量并支持 SP 切片，是 fused 蒸馏路径的入口。

```
# ... 前面已设置 shift_labels 和 return_log_probs ...
# Pass teacher top-K tensors so ForCausalLMLoss routes to
# chunk_topk_distill_function for fused distillation.
# 从 micro_batch 中获取配置标记
distillation_use_topk = tu.get_non_tensor_data(
    data=micro_batch, key="distillation_use_topk", default=False
)
# 仅当 fused 蒸馏启用且 teacher_ids 存在时才注入
```

```
if distillation_use_topk and "teacher_ids" in micro_batch.keys():
    # 配对校验：如果 teacher_ids 存在但 teacher_logprobs 缺失则报错
    if "teacher_logprobs" not in micro_batch.keys():
        raise ValueError(
            "teacher_ids present without teacher_logprobs; "
            "both must be provided together for fused top-K distillation."
        )
    # 从嵌套张量中提取 values 并添加 batch 维度 → (1, total_nnz, K)
    teacher_topk_ids = micro_batch["teacher_ids"].values().unsqueeze(0)
    teacher_topk_log_probs = micro_batch["teacher_logprobs"].values().unsqueeze(0)
    # 如果启用了 Ulysses 序列并行，沿 seqlen 维度 (dim=1) 进行切片
    if self.use_ulysses_sp:
        from verl.utils.ulysses import slice_input_tensor
        # teacher 张量为 3D，与 input_ids 的 2D 不同，需直接调用 slice_input_tensor
        teacher_topk_ids = slice_input_tensor(teacher_topk_ids, dim=1, padding=True)
        teacher_topk_log_probs = slice_input_tensor(teacher_topk_log_probs, dim=1,
        padding=True)
    # 将处理后的张量放入 model_inputs，供模型前向使用
    model_inputs["teacher_topk_ids"] = teacher_topk_ids
    model_inputs["teacher_topk_log_probs"] = teacher_topk_log_probs
```

verl/workers/engine/fsdp/transformer_impl.py

在 prepare_model_outputs 的 fused 分支中提取蒸馏输出，转换为嵌套张量，与 eager 路径格式一致。

```
        # After extracting log_probs and entropy from fused kernel...
        # 如果启用了 fused top-K 蒸馏，从 aux_outputs 中提取逐 token 蒸馏结果
        if distillation_use_topk:
            # fused_linear_aux 由 chunk_topk_distill_function 填充
            aux_outputs = getattr(output, "fused_linear_aux", None)
            if aux_outputs is not None and aux_outputs.distillation_losses is not None:
                cu_seqlens = input_ids.offsets() # 原始嵌套张量的 cu_seqlens
                # 需要提取的三个字段
                for field_name in ("distillation_losses", "student_mass", "teacher_mass"):
                    # 去掉 batch 维度 → (total_nnz,) 或 (total_nnz, K)
                    v = getattr(aux_outputs, field_name).squeeze(0)
                    # 如果使用了 Ulysses SP，需要先 gather 并去填充
                    if self.use_ulysses_sp:
                        pad_size = output_args["pad_size"]
                        v = gather_outputs_and_unpad(
                            v, gather_dim=0, unpad_dim=0, padding_size=pad_size
                        )
                    # 重新包装为嵌套张量，与 eager 路径的输出格式一致
                    model_output[field_name] = torch.nested.nested_tensor_from_jagged(v, cu_
                    seqlens)
```

评论区精华

- gemini-code-assist[bot] 发现 SP 切片缺失：在 Ulysses 序列并行启用时，teacher 的 top-K 张量未进行 padding 和切片，会导致与 student 序列维度不匹配而崩溃。作者在后续提交中修复，先尝试 ulysses_pad_and_slice_inputs，后改用更通用的 slice_input_tensor 处理 3D 张量。
- wuxibin89 询问 FSDP backend 支持：询问 chunk_topk_distill_function 是否也适用于 FSDP backend。Luosuu 明确回答“Not yet”，表明该功能当前仅 VeOmni backend 支持。
  - SP 切片缺失导致形状不匹配 (correctness): 作者在后续提交中通过 slice_input_tensor 对 teacher 张量沿 seqlen 维度进行了 SP 切片。
  - FSDP backend 是否支持 fused 蒸馏 (question): Luosuu 回答 'Not yet'，表明当前仅 VeOmni backend 支持。

风险与影响

- 风险：
  1. 兼容性风险：依赖 veomni >=0.1.11，且存在已知 bug (#804) 阻碍端到端运行，需要 veomni 修复后才能正常使用。单元测试验证了路径正确，但生产使用需等待修复。
  2. 正确性风险：teacher 张量 SP 切片逻辑已修复，但如果未来修改 SP 切片规则，需要同步更新此处的切片调用。耦合在 prepare_model_inputs 中的切片逻辑可能成为维护点。
  3. 性能风险：无负面性能影响，但 fused 路径的蒸馏内核行为与 eager 不同，需确保 veomni 的 kernel 实现正确，目前单元测试验证了有限性和非负性。
  4. 配置风险：新增 distillation_use_topk 等配置项，若在其他引擎中误用可能静默忽略，目前未做严格校验。
- 影响：
  - 用户影响：为使用 VeOmni 引擎进行 OPD 训练的用户提供显存更友好的选项，尤其适合大词表模型。现有 eager 路径不受影响。
  - 系统影响：仅影响 VeOmni 引擎和 FSDP 引擎的 fused 分支（+25/+15 行），其他引擎无改动。示例脚本提供新参考，但端到端运行受阻于 veomni #804。
  - 团队影响：需要维护与 veomni 版本的兼容性；建议在 veomni #804 修复后重新验证端到端示例，并将其纳入 CI。
  - 风险标记：依赖 veomni bug #804, SP 切片对齐需维护，仅 VeOmni backend 支持

关联脉络

- PR #6325 [veomni] feat: add MoE router replay (R2/R3) support: 修改了相同的 VeOmni 引擎文件 verl/workers/engine/veomni/transformer_impl.py，属于同一引擎模块的演进。
- PR #6506 [megatron, trainer] fix: preserve BSHD top-k distillation shape: 同样是蒸馏相关的 bugfix，但涉及不同引擎（Megatron），体现了蒸馏功能在多个引擎上的推进。