

PR #7858 完整报告

verl-project/verl

[trainer] fix: pad teacher fields in minimal padding and support fsdp2 in forward topk loss

合并时间: 2026-09-14 22:03

原文链接: <http://prhub.com.cn/verl-project/verl/pull/7858>

执行摘要

- 一句话: 补齐蒸馏 teacher 字段填充并支持 fsdp2 策略
- 推荐动作: 值得精读, 但只需读源码不必读讨论——PR 本身讨论为零。重点关注三个设计决策: 一是 padding 模板选择 "按模板序列长度重建" 而非 "裁剪 / 丢弃教师字段", 意味着合成样本仍需保持可被蒸馏 loss 消费的完整张量结构; 二是用 `.new_full/.new_zeros` 保留 topk 维与 dtype/device, 写法克制, 便于后续扩展更多 teacher 字段; 三是把 fsdp2 直接并入 fsdp/veomni 分支而不是新建实现。建议阅读者在本地补一条 CPU 单测覆盖 teacher_ids/teacher_logprobs 存在与不存在两种情况, 并确认跨 tokenizer 场景下 eos_token_id 作为 teacher topk id 的合法性。

功能与动机

PR body 明确写道: "While adding OPD support to verl-omni, we encountered several issues that required changes to verl.", 作者随后列出两个希望回流到 verl 主仓的修复: 一是 "Add fsdp2 support in compute_topk_loss", 二是 "Add teacher_ids/teacher_logprobs padding to prevent shape mismatch in construct_minimal_padding_template", 并给出上游来源 verl-omni PR#375。也就是说, 动机来自外部仓库集成 OPD 蒸馏时踩到的真实报错: 合成 padding 样本与真实样本的 teacher 张量长度不一致导致拼接失败, 以及 FSDP2 后端在蒸馏 topk loss 分派时直接抛 NotImplementedError。

实现拆解

本次改动入口是 OPD 蒸馏数据流上的两个点: "批次最小填充模板" 与 "topk 蒸馏损失分派", 两者通过 `compute_topk_loss` 末尾的形状断言耦合在一起。

1. 教师字段随模板序列重建 (verl/trainer/ppo/padding_utils.py, 函数 `construct_minimal_padding_template`)。模板原本会先把源样本的每个字段 clone 进 `template_sample`, 因此在蒸馏场景下 teacher_ids / teacher_logprobs 会带着原始序列长度 (如 8K) 进入模板, 而模板随后把序列重写成 prompts (`SYNTHETIC_PADDING_SEQ_LEN - 1`) + responses (1) 的固定最小长度, 第一维立即错位。新增逻辑先取 `sequence_length = input_ids.size(0)`, 再对存在的 teacher_ids 用 `teacher_ids.new_full((sequence_length, *teacher_ids.shape[1:]), eos_token_id)`、对 teacher_logprobs 用 `new_zeros` 同构重建, 保留 topk 维、dtype 与 device。这样 `upsample_batch_to_divisible_size` 在同 batch 内拼接真实样本与合成样本时形状一致。

2. 策略分派补齐 fsdp2 (verl/trainer/distillation/losses.py, 函数 compute_topk_loss) 。
match config.strategy 的分支由 case "fsdp" | "veomni" 扩展为 case "fsdp" | "veomni" | "fsdp2", 让 FSDP2 后端复用 verl.trainer.distillation.fsdp.losses.compute_forward_kl_topk, 而不是落到 case _ 抛 NotImplementedError。这与原注释 "VeOmni uses FSDP2 internally, so its loss computation is identical to FSDP" 的判断一致, 属于把已有等价性结论显式落到分派表。
3. 两处改动共同服务于第 158-160 行的形状契约: compute_topk_loss 会断言每个输出 shape == student_logits.shape[:2] (bsz, seqlen/cp_size) 。若 padding 模板未补齐教师字段, 错误会更晚、更难定位地出现在 forward_kl_topk 内部; 补齐后该断言成为最早的失败点。
4. 配套缺口: 本次没有新增或修改任何测试文件, 也没有 schema、配置项、启动脚本或文档变更; PR 检查表中 "Add unit or end-to-end test(s)" 一项未勾选, Test 章节为空白, 未提供训练曲线或数值验证。

关键文件:

- verl/trainer/ppo/padding_utils.py (模块 批次填充; 类别 source; 类型 core-logic; 符号 construct_minimal_padding_template) : 核心修复点:
construct_minimal_padding_template 在克隆源样本后按最小序列长度重建 teacher_ids / teacher_logprobs, 解决 OPD 蒸馏下 batch upsample 的 shape mismatch。
- verl/trainer/distillation/losses.py (模块 蒸馏损失; 类别 source; 类型 core-logic; 符号 compute_topk_loss) : 策略分派修复: compute_topk_loss 的 match 分支加入 fsdp2, 使 FSDP2 后端复用 FSDP 的 compute_forward_kl_topk, 避免 NotImplementedError。

关键符号: construct_minimal_padding_template, compute_topk_loss,
compute_forward_kl_topk, upsample_batch_to_divisible_size

关键源码片段

verl/trainer/ppo/padding_utils.py

核心修复点: construct_minimal_padding_template 在克隆源样本后按最小序列长度重建 teacher_ids / teacher_logprobs, 解决 OPD 蒸馏下 batch upsample 的 shape mismatch。

```
# verl/trainer/ppo/padding_utils.py
# 构造用于把 batch upsample 到可整除大小的最小文本填充模板。
def construct_minimal_padding_template(
    source_td: dict,
    source_tag: dict,
    eos_token_id: int,
) -> tuple[dict, dict]:
    """Construct a text-only padding template. 返回 (template_sample, template_tag). """

    # 1) 深拷贝源样本作为模板: teacher_ids / teacher_logprobs 也会被一并带过来,
    # 此时它们的序列维仍是原始样本长度 (例如 8K), 与下面构造的最小序列不一致。
    template_sample = {}
    for key in source_td.keys():
        value = source_td[key]
```

```
template_sample[key] = value.clone() if isinstance(value, torch.Tensor) else copy.  
deepcopy(value)
```

```
template_tag = copy.deepcopy(source_tag)
```

```
# 2) 构造最小序列: prompt 段全为 eos, response 段只有 1 个 token,
```

```
# 真实长度固定为 SYNTHETIC_PADDING_SEQ_LEN。
```

```
responses = torch.full((1,), eos_token_id, dtype=torch.int64)
```

```
prompts = torch.full((SYNTHETIC_PADDING_SEQ_LEN - 1,), eos_token_id, dtype=torch.int64)
```

```
input_ids = torch.cat((prompts, responses))
```

```
attention_mask = torch.ones_like(input_ids, dtype=torch.int64)
```

```
response_mask = torch.zeros_like(responses)
```

```
position_ids = build_padding_position_ids(template_sample.get("position_ids"), attention_mask)
```

```
routed_experts = build_padding_routed_experts(template_sample.get("routed_experts"), input_  
ids.size(0))
```

```
# 3) 本次新增: 把教师侧字段按新序列长度重建, 否则 upsample 拼接真实样本时
```

```
# teacher_ids / teacher_logprobs 第一维不匹配, 直接 shape mismatch。
```

```
# * 用 new_full / new_zeros 保留 topk 维 (*shape[1:]) 以及 dtype、device;
```

```
# * response_mask 全 0, 这些位置不参与蒸馏 loss, 只需形状合法。
```

```
sequence_length = input_ids.size(0)
```

```
teacher_ids = template_sample.get("teacher_ids")
```

```
if teacher_ids is not None:
```

```
    template_sample["teacher_ids"] = teacher_ids.new_full((sequence_length, *teacher_ids.  
    shape[1:]), eos_token_id)
```

```
teacher_logprobs = template_sample.get("teacher_logprobs")
```

```
if teacher_logprobs is not None:
```

```
    template_sample["teacher_logprobs"] = teacher_logprobs.new_zeros((sequence_length,  
    *teacher_logprobs.shape[1:]))
```

```
# 4) 用最小序列覆盖所有序列相关字段, 并清理不再适用的内容。
```

```
template_sample.update(  
    prompts=prompts,  
    responses=responses,  
    input_ids=input_ids,  
    attention_mask=attention_mask,  
    position_ids=position_ids,  
    num_turns=0,  
    response_mask=response_mask,  
    loss_mask=response_mask,  
    rm_scores=torch.zeros_like(response_mask, dtype=torch.float32),  
    rollout_log_probs=torch.zeros_like(response_mask, dtype=torch.float32),  
)
```

```
if "multi_modal_inputs" in template_sample:
```

```
    template_sample["multi_modal_inputs"] = {}
```

```
if routed_experts is not None:
```

```
    template_sample["routed_experts"] = routed_experts
```

```

else:
    template_sample.pop("routed_experts", None)

# 5) is_padding 标记用于隔离指标统计 (response length、score、reward) ,
# 保证合成样本不会污染训练指标。
template_tag.update(
    is_padding=True,
    prompt_len=SYNTHETIC_PADDING_SEQ_LEN - 1,
    response_len=1,
    seq_len=SYNTHETIC_PADDING_SEQ_LEN,
)
return template_sample, template_tag

```

评论区精华

本 PR 讨论量极小: review_comments_count = 0, 只有维护者 wuxibin89 给出一次 APPROVED 空评论, 随后合并。因此没有技术交锋可提炼, 能看到的只有两类 " 沉默信号 ": 其一, 作者在 PR body 中直接向维护者发问 "We would like to ask whether the following two fixes could be incorporated into the verl codebase", 说明这是外部集成的回流补丁, 评审者的 approve 即视为接受该结论; 其二, 检查表中测试项留空、正文未附任何实验证据, 评审并未要求补充验证或测试, 属于依赖作者声誉的快速通道。

- 外部集成补丁能否回流主仓 (question): 维护者 wuxibin89 未留下任何文字意见, 直接 APPROVED 并合并, 等价于接受这两处修复进入主线。
- 缺少测试与实验验证 (testing): 未解决: teacher 字段补齐分支 (存在 / 不存在) 与 fsdp2 分派分支都没有自动化覆盖, 回归只能依赖 verl-omni 侧的实际训练发现。

风险与影响

- 风险:
 1. 填充语义风险: teacher_ids 用 eos_token_id 填充 (verl/trainer/ppo/padding_utils.py)。teacher topk id 通常来自 teacher 词表, 若学生 / 教师 tokenizer 词表不同, eos_token_id 可能越界或在后续按 teacher 词表 gather 时语义错误; 虽然合成样本 response_mask 全 0、理论上不参与 loss, 但当前没有任何测试证明下游不会在语义层面消费这些位置。
 2. 零值 logprob 风险: teacher_logprobs 用 new_zeros 填充, 若下游对 teacher 分布做 softmax 归一化或质量统计 (teacher_mass), 非 -inf 的 0 会带来一个 " 均匀分布 " 的假信号; 当前改动只保证形状合法, 未保证数值中性。
 3. 无测试覆盖: 两个文件都是训练主路径 (批次填充 + 蒸馏损失), 却零测试改动。construct_minimal_padding_template 新增的 if teacher_ids is not None / if teacher_logprobs is not None 两个分支都没有 CPU 单测覆盖, CI 里通常也不跑 OPD 端到端, 回归只能靠接入方 (verl-omni) 发现。
 4. fsdp2 分支假设未验证: 把 fsdp2 与 fsdp 共用 compute_forward_kl_topk, 前提是二者 logits 布局与 CP 切分行为完全一致; 注释中的等价性结论只针对 VeOmni, 本次把它外推到 fsdp2 之上, 缺少任何数值对照。

5. 兼容性面相对窄：改动是纯增量（新增字段处理 + 扩充分支），不改变已有 fsdp/megatron/veomni 路径与函数签名，回归风险主要局限在同时使用蒸馏与 batch padding 的组合场景。 - 影响：用户侧：使用 OPD/ 蒸馏 + FSDP2 或需要把 batch upsample 到可整除大小的训练任务，可以直接运行不再报 NotImplementedError 与 shape mismatch，属于解除阻塞而非改变精度语义；不使用蒸馏的普通 GRPO/PPO 训练完全不受影响。系统侧：verl/trainer/ppo/padding_utils.py 与 verl/trainer/distillation/losses.py 都是被多条数据流复用的公共工具，任何 shape 或分派改动都会影响所有走 padding 模板的调用方，因此后续若引入新的 teacher 字段（如 teacher_hidden_states），需要同步在模板中补齐。团队侧：这是 verl-omni 向主仓回流补丁的典型示例，说明外部 OPD 集成的改动路径已经建立，后续可能持续出现同类小补丁。 - 风险标记：蒸馏训练主路径变更，缺少测试覆盖，跨 tokenizer 填充语义未验证，fsdp2 与 fsdp 等价性仅靠注释假设

关联脉络

- PR #7702 [megatron] fix: pad VLM THD input to TP/CP alignment before SP scatter: 同属 " 填充 / 对齐上游形状以适配下游并行或拼接 " 的修复模式，都改动 padding/ 对齐逻辑 (verl/models/mcore/util.py 与 verl/trainer/ppo/padding_utils.py)，可作为同类 shape 契约问题的参照。
- PR #7805 [megatron] fix: make R2 router replay model-scoped and opt-in: 与本 PR 同为窄范围、带形状 / 作用域契约检查的蒸馏或回放相关修复，都通过显式分支与断言把隐式假设显式化，属于同一类小步修正的演进节奏。