PR #7236 完整报告

verl-project/verl

[data] feat: Add processor hook for multimodal RoPE kwargs

合并时间：2026-08-04 17:23

原文链接：http://prhub.com.cn/verl-project/verl/pull/7236

执行摘要

- 一句话：为多模态 RoPE 计算新增 processor hook，支持音频等额外参数
- 推荐动作：值得快速精读，尤其是 _compute_position_ids 中 hook 的接入方式。这是一个小而清晰的扩展点设计，适合作为 agent_loop 多模态能力演进的参考；对于后续要接入 Qwen3-Omni 的开发者，建议阅读测试文件了解 hook 的约定。

功能与动机

分支名 qwen3omni 以及测试中的 audio_seqlens、feature_attention_mask 表明，这是为 Qwen3-Omni 的音频输入准备的前置能力。此前 _compute_position_ids 只传递图像 / 视频相关的 image_grid_thw、video_grid_thw 和 mm_token_type_ids，无法为需要音频长度信息的 RoPE 计算提供参数。通过可选的 get_rope_index_kwargs hook，将“从多模态输入推导 RoPE 参数”的职责下放给模型专用 processor，避免在 agent_loop 中为每个模型硬编码。

实现拆解

1. 变更入口：verl/experimental/agent_loop/agent_loop.py 中的 _compute_position_ids 是 agent_loop 多模态 trajectory 计算位置 ID 的唯一入口，此前只支持 image_grid_thw、video_grid_thw 与 mm_token_type_ids 三类 RoPE 输入。
2. 核心改动：在该方法中通过 getattr(self.processor, "get_rope_index_kwargs", None) 探测处理器是否提供可选 hook。若处理器实现了 get_rope_index_kwargs(multi_modal_inputs)，则将其返回值（例如 audio_seqlens）合并进 multi_modal_kwargs，再传给已动态绑定到处理器的 get_rope_index。未实现 hook 的处理器行为与之前完全一致，保证向后兼容。
3. 测试配套：新增 tests/experimental/agent_loop/test_multimodal_position_ids_on_cpu.py，构造带 get_rope_index_kwargs 的假 Processor，用 feature_attention_mask 求和生成 audio_seqlens，验证参数正确传递且 position_ids 形状为 (1, 4, 3)。
4. 配置与部署：无配置、schema、CI 或部署相关改动。

关键文件：

- verl/experimental/agent_loop/agent_loop.py（模块 代理循环；类别 source；类型 core-logic；符号 _compute_position_ids）：核心变更文件，在 _compute_position_ids 中新增 processor hook，将模型自定义 RoPE kwargs 合并进 get_rope_index 调用，是本次功能的主要实现点。

- tests/experimental/agent_loop/test_multimodal_position_ids_on_cpu.py（模块 多模态位置；类别 test；类型 test-coverage；符号 test_compute_position_ids_accepts_processor_rope_kwargs_hook, Processor, get_rope_index_kwargs, get_rope_index）：新增测试覆盖 hook 行为，用假 Processor 验证 audio_seqlens 从 feature_attention_mask 正确推导并传给 get_rope_index，以及输出位置 ID 的形状。

关键符号：_compute_position_ids, get_rope_index_kwargs

关键源码片段

verl/experimental/agent_loop/agent_loop.py

核心变更文件，在 _compute_position_ids 中新增 processor hook，将模型自定义 RoPE kwargs 合并进 get_rope_index 调用，是本次功能的主要实现点。

```
def _compute_position_ids(
    self,
    input_ids,
    attention_mask,
    multi_modal_inputs,
    mm_processor_kwargs: Optional[dict[str, Any]] = None,
) -> torch.Tensor:
    """Compute position ids for multi-modal inputs."""
    # 纯文本或非 M-RoPE 多模态（如 Gemma4）走标准 1D 位置 ID
    if self.processor is None or not hasattr(self.processor, "get_rope_index"):
        return compute_position_id_with_mask(attention_mask) # (1, seq_len)

    multi_modal_kwargs = {
        "image_grid_thw": multi_modal_inputs.get("image_grid_thw"),
        "video_grid_thw": multi_modal_inputs.get("video_grid_thw"),
    }
    # transformers >= 5.3.0 时 mm_token_type_ids 仅用于位置 ID 计算
    if multi_modal_inputs.pop("mm_token_type_ids", None) is not None:
        mm_token_type_ids = torch.zeros_like(input_ids)
        image_token_id = get_processor_token_id(self.processor, "image")
        video_token_id = get_processor_token_id(self.processor, "video")
        if image_token_id is not None:
            mm_token_type_ids[0][input_ids[0] == image_token_id] = 1
        if video_token_id is not None:
            mm_token_type_ids[0][input_ids[0] == video_token_id] = 2
        multi_modal_kwargs["mm_token_type_ids"] = mm_token_type_ids

    # 新增 hook：允许模型专用 processor 贡献额外 RoPE 参数（例如 Qwen3-Omni 的 audio_
    seqlens）
    get_rope_index_kwargs = getattr(self.processor, "get_rope_index_kwargs", None)
    if get_rope_index_kwargs is not None:
        multi_modal_kwargs.update(get_rope_index_kwargs(multi_modal_inputs))

    # 模型的 get_rope_index 已动态绑定到 processor 上
    vision_position_ids, _ = self.processor.get_rope_index(
```

```
            input_ids=input_ids,
            attention_mask=attention_mask,
            **multi_modal_kwargs,
        )
        vision_position_ids = vision_position_ids.transpose(0, 1) # (3, 1, seq_len) => (1, 3, seq_len)

        valid_mask = attention_mask[0].bool()
        text_position_ids = torch.ones((1, len(input_ids[0])), dtype=torch.long)
        text_position_ids[0, valid_mask] = torch.arange(valid_mask.sum().item())
        text_position_ids = text_position_ids.unsqueeze(0)
        position_ids = torch.cat((text_position_ids, vision_position_ids), dim=1) # (1, 4, seq_length)
        return position_ids
```

评论区精华

PR 没有收到人工 review 评论或审核记录，唯一评论来自 CLAassistant 确认贡献者已签署 CLA。这意味着变更通过静默合入，设计权衡（例如为什么用 getattr 动态探测而不是注册表或配置项）没有留下公开讨论记录。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 兼容性：hook 缺失时走原路径，getattr 探测保证向后兼容，风险低。
  2. 数据可变性：multi_modal_inputs 以引用方式传给 hook，若 hook 内部修改该字典，可能影响调用方后续逻辑；当前没有防御性拷贝。
  3. 测试覆盖：单测使用玩具处理器，未覆盖真实 Qwen3-Omni 的音频数据流，也没有端到端测试验证音频位置 ID 的数值正确性。
  4. API 约定：get_rope_index_kwargs 是新引入的 processor 非正式接口，尚未写入文档，后续模型接入时可能因约定不一致产生混淆。 - 影响：影响范围集中在 agent_loop 多模态轨迹的位置 ID 计算路径，尤其是使用音频输入的 Qwen3-Omni 场景；对纯文本、视觉多模态处理器无行为变化。该 hook 为未来接入更多需要自定义 RoPE 参数的模型提供了扩展点，团队后续可基于此模式演进。单测在 CPU 上运行，不引入额外 GPU 成本。 - 风险标记：处理器 hook 无防御性拷贝，缺少真实音频端到端测试，processor API 约定未文档化

关联脉络

- PR #7204 [rollout] fix: decode per-turn LLM tokens in traces: 与 PR 7236 同改 verl/experimental/agent_loop/agent_loop.py，说明该文件正在承接多模态 agent 循环的持续性改造。