PR #50950 完整报告

vllm-project/vllm

[Bugfix] Resolve seq-cls `num_labels` from the top-level config for multimodal checkpoints

合并时间：2026-08-04 19:39

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/50950

执行摘要

- 一句话：修复多模态 seq-cls 分类头 num_labels 解析错误
- 推荐动作：值得精读。_resolve_num_labels 的 fallback 决策体现对 HF 复合配置约定的理解——多模态顶层 config 拥有标签空间而 text_config 只是文本子配置；测试矩阵用 Gemma3Config 构造复合配置，清晰覆盖五种解析分支，可作为同类 config 解析修复的测试范式。建议关注后续是否将逻辑上移到 ModelConfig 层，统一所有 num_labels 读取。

功能与动机

多模态 seq-cls 模型（如 Qwen3-VL）把 id2label / label2id / problem_type 声明在顶层 config.json，而 vLLM 的 as_seq_cls_model._init_pooler 从 get_text_config().num_labels 取标签数。HF 的 num_labels 由 id2label 推导且默认两条，导致 20 标签 checkpoint 的 score.weight 形状 [20, 2560] 与分类头 [2, 2560] 不匹配，权重加载抛出 AssertionError: Tried to load weights of size torch.Size([20, 2560]) to a parameter of size torch.Size([2, 2560])。该问题由 vLLM issue #47956 报告，并在 modelscope/ms-swift#9704 中持续影响 Qwen3-VL 多标签微调后的部署。

实现拆解

1. 变更入口：vllm/model_executor/models/adapters.py，全部改动集中在该文件与对应测试。
2. 新增解析函数：_resolve_num_labels(hf_config, text_config)。策略为：text_config 与 hf_config 同一对象（纯文本模型）时直接返回；否则当顶层 num_labels 不等于 PretrainedConfig().num_labels（说明 checkpoint 在顶层声明了标签空间）时采用顶层值；否则回退到 text_config.num_labels。
3. 调用点替换：ModelForSequenceClassification._init_pooler 中 ReplicatedLinear 的输出维度由 text_config.num_labels 改为 _resolve_num_labels(hf_config, text_config)，这是唯一的运行时行为变更。
4. 测试配套：tests/models/test_adapters.py 新增 _composite_config 构造器（用 Gemma3Config(text_config=...) 模拟多模态组合配置）和 5 个解析矩阵用例，覆盖纯文本、两者均未声明、仅顶层声明、仅 text_config 声明、两者同时声明（顶层优先）。
5. 验证方式：CPU 单测 7 个全部通过，无 GPU 要求；作者在 v0.26.0 + A100-40GB 上手动验证 Qwen3-VL seq-cls checkpoint，POST /classify 返回 20 个独立 sigmoid 概率，符合 multi_label_classification 预期。

关键文件：

- vllm/model_executor/models/adapters.py（模块 模型适配；类别 source；类型 data-contract；符号 _resolve_num_labels）：修复核心：新增 _resolve_num_labels，并在 ModelForSequenceClassification._init_pooler 中用其替换 text_config.num_labels 作为分类头输出维度。
- tests/models/test_adapters.py（模块 模型测试；类别 test；类型 test-coverage；符号 _composite_config, test_resolve_num_labels_text_only_config, test_resolve_num_labels_defaults_when_undeclared, test_resolve_num_labels_declared_on_outer_config）：新增 5 个 CPU 单测覆盖 num_labels 解析决策矩阵，验证 text-only、未声明、仅顶层、仅 text_config、同时声明五种情况。

关键符号：_resolve_num_labels, _init_pooler

关键源码片段

vllm/model_executor/models/adapters.py

修复核心：新增 _resolve_num_labels，并在 ModelForSequenceClassification._init_pooler 中用其替换 text_config.num_labels 作为分类头输出维度。

```
def _resolve_num_labels(hf_config: Any, text_config: Any) -> int:
    """解析序列分类头的标签数量。

    ``PretrainedConfig.num_labels`` 由 ``id2label`` 推导，而 ``id2label``
    始终携带默认的两条记录；多模态组合配置（composite config）按 HF 约定把
    ``id2label`` / ``label2id`` / ``problem_type`` 声明在顶层配置上，所以
    直接从 ``get_text_config()`` 读取会拿到默认值 2，导致分类头尺寸错误。
    这里优先采用顶层配置声明的标签空间，与上方 ``classifier_from_token`` /
    ``method`` 的取值方式保持一致；仅当顶层没有声明时才回退到文本配置。
    """
    # 纯文本模型（text-only）下 text_config 就是 hf_config 本身，直接返回
    if text_config is hf_config:
        return hf_config.num_labels

    from transformers import PretrainedConfig

    # 顶层 num_labels 不等于 HF 默认值，说明 checkpoint 在顶层声明了真实标签空间
    if hf_config.num_labels != PretrainedConfig().num_labels:
        return hf_config.num_labels
    # 顶层未声明时回退到 text_config，兼容 overrides 写在文本配置的路径
    return text_config.num_labels
```

```
    def _init_pooler(
        self,
        vllm_config: VllmConfig,
        prefix: str = '',
    ) -> Pooler:
        hf_config = vllm_config.model_config.hf_config
```

```
        text_config = hf_config.get_text_config()
        model_config = vllm_config.model_config

        # 在线转换路径（由 LM head 推导 score）不量化：
        # 输出维度过小会破坏 FP8 / Marlin tile 对齐
        tokens = getattr(
            hf_config,
            'classifier_from_token',
            getattr(text_config, 'classifier_from_token', None),
        )
        method = getattr(
            hf_config,
            'method',
            getattr(text_config, 'method', None),
        )
        quant_config = (
            None
            if (tokens is not None or method is not None)
            else vllm_config.quant_config
        )

        # 关键修改：分类头输出维度从 text_config.num_labels 改为
        # _resolve_num_labels(hf_config, text_config)，优先采用顶层声明的标签数
        self.score = ReplicatedLinear(
            model_config.get_hidden_size(),
            _resolve_num_labels(hf_config, text_config),
            bias=False,
            params_dtype=model_config.head_dtype,
            quant_config=quant_config,
            return_bias=False,
            prefix=maybe_prefix(prefix, 'score'),
        )
```

tests/models/test_adapters.py

新增 5 个 CPU 单测覆盖 num_labels 解析决策矩阵，验证 text-only、未声明、仅顶层、仅 text_config、同时声明五种情况。

```
def _composite_config(outer_labels=None, inner_labels=None):
    """构造多模态风格的组合配置，text_config 是独立对象。"""
    config = Gemma3Config(text_config={'num_hidden_layers': 1})
    if outer_labels is not None:
        config.num_labels = outer_labels # 顶层声明标签空间（HF 约定）
    if inner_labels is not None:
        # 模拟通过 overrides 写入 text_config 的标签数
        config.get_text_config().num_labels = inner_labels
    return config


def test_resolve_num_labels_declared_on_outer_config():
```

```
    """多模态 checkpoint 把 id2label / problem_type 放在顶层配置。"""
    config = _composite_config(outer_labels=20)
    # text_config 里仍是 HF 默认值 2，解析结果应为顶层声明的 20
    assert config.get_text_config().num_labels == PretrainedConfig().num_labels
    assert _resolve_num_labels(config, config.get_text_config()) == 20


def test_resolve_num_labels_outer_wins_when_both_declared():
    """顶层与 text_config 同时声明时以顶层为准。"""
    config = _composite_config(outer_labels=20, inner_labels=5)
    assert _resolve_num_labels(config, config.get_text_config()) == 20
```

评论区精华

PR 无实质性 review 评论：claude[bot] 因 fork PR 自动 review 被禁用，noooop 直接批准（Thanks for your contribution）。作者在 PR body 中主动提出两个待审问题：

1) fallback 逻辑与 vLLM 其他 num_labels 读取点不一致（其他位置都无条件读顶层配置），无条件读取更简单且是 #27338 之前的行为，但会破坏只在 text_config 声明标签的配置，作者因此保留 fallback； 2) 是否将解析函数移到 ModelConfig.get_hidden_size() 旁边，作者判断标签空间属于顶层配置且非架构特定，保留在 adapters.py。两个问题未获回复，按作者取舍合入。

- fallback 逻辑 vs 无条件读顶层 config (design): 无 reviewer 展开讨论；noooop 直接批准。作者保留 fallback 以兼容 text_config 声明标签的路径。
- _resolve_num_labels 的放置位置 (design): 作者决定保留在 adapters.py，未收到反对意见。

风险与影响

- 风险：
  1. 默认值判断：_resolve_num_labels 用 PretrainedConfig().num_labels 判断顶层是否声明过标签，该默认值当前为 2，若 transformers 未来改变默认行为，判断可能失真。
  2. fallback 一致性：仅顶层未声明而 text_config 声明时回退，保持旧行为；两者冲突时以顶层为准。这是 vLLM 中唯一带 fallback 的 num_labels 读取点，存在轻微的语义不一致风险。
  3. 影响面：纯文本模型 get_text_config() 返回自身，第一个分支直接返回，行为不变；在线转换路径（classifier_from_token / method）已在 verify_and_update_config 双写 num_labels，也不受影响。
  4. 测试缺口：CI 仅含 CPU 单测，GPU 端到端验证由作者手动完成，缺少自动化回归防护。 - 影响：用户侧：Qwen3-VL 等多模态 seq-cls checkpoint（num_labels > 2）从无法加载变为可正常部署，/classify 返回正确的标签概率。系统侧：仅影响 pooling runner 启动期的配置解析（adapters.py），不触碰推理、注意力或量化计算路径，对已能加载的模型输出无影响。团队侧：补齐 #31890 之后最后一处未复查的 num_labels 读取位置，为后续统一 config 读取逻辑提供参考。 - 风险标记：配置解析路径变更，依赖 PretrainedConfig 默认值，缺少 GPU 端到端 CI 测试，fallback 一致性权衡

关联脉络

- PR #50890 Touches the same two files in a different function (trivial rebase overlap): PR body 提及：该 PR 触碰相同的 adapters.py 与 tests/models/test_adapters.py，但修改的是 ModelForPooling.load_weights，与本 PR 无语义重叠，仅需平凡 rebase。