

PR #43636 完整报告

vllm-project/vllm

[Misc] Support interleaved custom image benchmark datasets

合并时间: 2026-05-26 18:37

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/43636>

执行摘要

- 一句话: 自定义图像基准数据集支持多图像与交错内容
- 推荐动作: 建议阅读该 PR, 特别是 CustomImageDataset 的 `load_data` 和 `sample` 方法的设计, 以及请求构建的适配模式。对于需要定制 benchmark 数据集的团队具有参考价值。整体清理且安全, 可正常合并。

功能与动机

现有 CustomImageDataset 在 benchmark 时仅使用 `image_files` 中的第一个图像, 且无法表示文本与图像的任意交错顺序, 难以模拟真实多模态负载。Issue #43269 提出了支持多图像和交错内容的需求。此 PR 通过引入 'content' 字段解决了上述限制。

实现拆解

1. 数据集加载与验证(`vllm/benchmarks/datasets/datasets.py`): 重写了 `load_data` 方法, 严格校验每行 JSONL 必须包含旧字段 (`prompt + image_files`) 或新字段 (`content`)。新增 `_validate_content_parts` 静态方法确保 `content` 为合法列表; `_process_content_part` 类方法将每个部件转换为规范格式 (`text/image/image_url`); `_process_interleaved_content` 组装最终内容列表并统计文本部分长度。
2. 样本生成调整(`vllm/benchmarks/datasets/datasets.py`): `sample` 方法针对 `content` 字段使用辅助函数构建多模态内容列表, 并保留顺序。旧 `image_files` 格式改为使用所有图像 (而非仅第一个), 生成列表。同时适配 `enable_multimodal_chat` 选项, 当启用时将内容包装为聊天消息。
3. 请求构建适配(`vllm/benchmarks/lib/endpoint_request_func.py`): `RequestFuncInput.prompt` 类型扩展为 `str | list[str] | list[dict]`。新增 `_is_chat_messages` 函数识别预构建的多轮消息, `_get_chat_messages` 函数在必要时将内容包装为用户消息。`async_request_openai_chat_completions` 和 `async_request_openai_embeddings_chat` 改用 `_get_chat_messages` 构造 payload。
4. 测试覆盖(`tests/benchmarks/test_custom_image_dataset.py`): 新增测试文件, 包含三个端到端测试: 多图像使用所有文件、交错内容顺序保持、以及通过 CLI 路径的采样。
5. 文档更新(`docs/benchmarking/cli.md`): 添加新 JSONL 格式示例, 说明 `content` 字段的用法和 `image_url`、`image` 简写。

关键文件:

- `vllm/benchmarks/datasets/datasets.py` (模块 基准数据集; 类别 source; 类型 core-logic; 符号 `load_data`, `_validate_content_parts`, `_process_content_part`, `_process_interleaved_content`) : 核心变更文件, 扩展 `CustomImageDataset` 支持多种图像输入格式 (旧格式所有图像使用和新 content 交错格式), 新增多个辅助方法。
- `tests/benchmarks/test_custom_image_dataset.py` (模块 测试; 类别 test; 类型 test-coverage; 符号 `_TokenizedPrompt`, `init`, `_Tokenizer`, `call`) : 新增全面的单元测试, 覆盖多图像、交错内容、端到端请求构建等。
- `vllm/benchmarks/lib/endpoint_request_func.py` (模块 请求构建; 类别 source; 类型 core-logic; 符号 `_is_chat_messages`, `_get_chat_messages`) : 调整请求构建逻辑以支持新的 prompt 类型 (`list[dict]`), 新增 `_is_chat_messages` 和 `_get_chat_messages`。
- `docs/benchmarking/cli.md` (模块 文档; 类别 docs; 类型 documentation) : 更新文档以说明新的 JSONL 格式。

关键符号: `load_data`, `_validate_content_parts`, `_process_content_part`, `_process_interleaved_content`, `_get_text_from_content`, `_process_image_files`, `sample`, `_is_chat_messages`, `_get_chat_messages`

关键源码片段

`vllm/benchmarks/datasets/datasets.py`

核心变更文件, 扩展 `CustomImageDataset` 支持多种图像输入格式 (旧格式所有图像使用和新 content 交错格式), 新增多个辅助方法。

```
def load_data(self) -> None:
    # 数据集路径必须给定且为 .jsonl 格式
    if self.dataset_path is None:
        raise ValueError("dataset_path must be provided for loading data.")
    if not self.dataset_path.endswith(".jsonl"):
        raise NotImplementedError("Only JSONL format is supported for CustomImageDataset.")

    self.data = []
    with open(self.dataset_path, encoding="utf-8") as f:
        for line_number, line in enumerate(f, start=1):
            line = line.strip()
            if not line:
                continue
            try:
                item = json.loads(line)
            except json.JSONDecodeError as e:
                raise ValueError(f"Invalid JSON on line {line_number}: {e}") from e
            if not isinstance(item, dict):
                raise ValueError(f"Each line must be a JSON object, got {type(item)} on line {line_number}.")
            # 检查是否包含旧字段 (prompt + image_files) 或新字段 (content)
            has_legacy = "prompt" in item and "image_files" in item
            has_content = "content" in item
            if not has_legacy and not has_content:
```

```

        raise ValueError(f"Line {line_number} must have either 'prompt'+'image_files' or
        'content'.")
    self.data.append(item)
    random.seed(self.random_seed)
    if not getattr(self, "disable_shuffle", False):
        random.shuffle(self.data)

    @staticmethod
    def _validate_content_parts(content: Any) -> list[dict[str, Any]]:
        # 验证 content 必须为非空列表，每项为字典
        if not isinstance(content, list):
            raise ValueError("'content' must be a list of text and image content dictionaries.")
        if not content:
            raise ValueError("'content' must contain at least one item.")
        parts = []
        for part in content:
            if not isinstance(part, dict):
                raise ValueError(f"Each content part must be a dictionary, got {type(part)}.")
            parts.append(part)
        return parts

```

vllm/benchmarks/lib/endpoint_request_func.py

调整请求构建逻辑以支持新的 prompt 类型 (list[dict])，新增 _is_chat_messages 和 _get_chat_messages。

```

def _get_chat_messages(
    request_func_input: RequestFuncInput,
    mm_position: Literal["first", "last"] = "last",
) -> list[dict[str, Any]]:
    prompt = request_func_input.prompt
    # 如果 prompt 已经是一个聊天消息列表，直接返回
    if _is_chat_messages(prompt):
        return prompt
    # 否则创建一个用户消息，content 由 _get_chat_content 构建
    return [
        {
            "role": "user",
            "content": _get_chat_content(
                request_func_input,
                mm_position=mm_position,
            ),
        }
    ]

```

评论区精华

Review 中有两个核心讨论点：

- 文件编码: gemini-code-assist[bot] 建议明确指定 encoding="utf-8" 避免平台编码问题。作者已修复并应用。
- 多模态聊天适配: 当 enable_multimodal_chat 启用时, 交错内容需要包装成 [{"role": "user", "content": content}] 消息格式, 否则模板应用会失败。作者在第二版提交中修复。

最终获得 DarkLight1337 的 approval。

- 指定 UTF-8 编码打开文件 (correctness): 作者在 commit 中已修复。
- enable_multimodal_chat 时内容需包装成消息 (correctness): 作者在 commit 中已修复。

风险与影响

- 风险: 向后兼容风险: 旧格式 image_files 的行为从仅使用第一个图像变为使用所有图像, 可能影响依赖旧行为的用户 (但之前已有 warning 提示)。新的 content 格式与旧格式不冲突。请求构建路径的改动 (_get_chat_messages) 经过测试覆盖, 但可能存在未预料的嵌套消息结构。总体风险较低, 因为变更仅影响 benchmark 工具, 不影响核心推理。
- 影响: 影响范围: 使用 vllm bench serve 进行多模态模型基准测试的用户。他们现在可以创建包含多图像和交错文本的图像数据集, 更真实地模拟生产流量。对内, 新增了测试和文档, 降低了维护成本。影响程度中等。
- 风险标记: 向后兼容改动, 多图像行为变更, 多模态聊天路径依赖

关联脉络

- 暂无明显关联 PR