PR #41459 完整报告

vllm-project/vllm

fix(frontend): Add multimodal placeholders to Gemma4 tool message template

合并时间：2026-05-29 05:48

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/41459

执行摘要

- 一句话：修复 Gemma4 工具消息中多模态占位符丢失
- 推荐动作：建议合并。该 PR 修复了用户报告的问题，且与上游 HuggingFace 模板保持同步。测试覆盖充分，风险低。值得关注的是多模态 tool 消息的模板处理方式，可推广到其他支持 tool-calling 的模型。

功能与动机

Issue #41452 报告 Gemma4 无法处理工具消息中的多模态内容。PR body 指出上游 chat_template.jinja 只提取 type == "text" 部分，丢弃图像 / 音频 / 视频条目。通过此 PR 修复了 vLLM 的示例模板，并同时提交了上游 HF PR 修复。

实现拆解

1. 修改 Jinja 模板：在 examples/tool_chat_template_gemma4.jinja 的工具消息文本内容渲染之后，新增循环 tool_body 部分，对每个内容项检查类型，若非文本则输出对应占位符（image/audio/video）。
2. 添加测试用例：在 tests/renderers/test_gemma4_chat_template.py 中新增 test_tool_response_with_multimodal_content 和 test_tool_response_with_all_modalities，分别测试单一模态和所有模态在工具响应中的渲染。
3. 验证：运行通过全部模板测试（16 个）、多模态处理和工具 / 推理解析器测试（77 个），未引入回归。

关键文件：

- tests/renderers/test_gemma4_chat_template.py（模块 模板测试；类别 test；类型 test-coverage；符号 test_tool_response_with_multimodal_content, test_tool_response_with_all_modalities）：新增两个测试用例，验证多模态内容在工具消息中被正确渲染，防止回归。
- examples/tool_chat_template_gemma4.jinja（模块 工具模板；类别 other；类型 core-logic）：核心修复：在工具响应文本后增加非文本内容占位符输出，使多模态内容不再被丢弃。

关键符号：test_tool_response_with_multimodal_content, test_tool_response_with_all_modalities

关键源码片段

tests/renderers/test_gemma4_chat_template.py

新增两个测试用例，验证多模态内容在工具消息中被正确渲染，防止回归。

```
    def test_tool_response_with_multimodal_content(self, gemma4_template):
        """Multimodal placeholders in tool messages are emitted after the
        tool_response block."""
        # 构造包含图像的工具响应消息
        messages = [
            {"role": "user", "content": "Download the image and describe it."},
            {
                "role": "assistant",
                "content": "",
                "tool_calls": [
                    {
                        "id": "call_1",
                        "type": "function",
                        "function": {
                            "name": "download_image",
                            "arguments": '{"url": "https://example.com/x.png"}',
                        },
                    },
                ],
            },
            {
                "role": "tool",
                "tool_call_id": "call_1",
                "content": [
                    {"type": "text", "text": "Image downloaded successfully."},
                    {"type": "image"},
                ],
            },
        ]
        result = _render(gemma4_template, messages, add_generation_prompt=True)
        # 断言工具响应块和图像占位符正确出现
        assert "<|tool_response>" in result
        assert "response:download_image{" in result
        assert "<tool_response|>" in result
        assert "<|image|>" in result

    def test_tool_response_with_all_modalities(self, gemma4_template):
        """All multimodal types (image, audio, video) in a single tool
        response are rendered."""
        # 构造包含多种多模态内容的工具响应
        messages = [
            {"role": "user", "content": "Process media"},
            {
                "role": "assistant",
```

```
            "content": "",
            "tool_calls": [
                {
                    "id": "c1",
                    "type": "function",
                    "function": {
                        "name": "process",
                        "arguments": "{}",
                    },
                },
            ],
        },
        {
            "role": "tool",
            "tool_call_id": "c1",
            "content": [
                {"type": "text", "text": "Results."},
                {"type": "image"},
                {"type": "audio"},
                {"type": "video"},
            ],
        },
    ]
    result = _render(gemma4_template, messages, add_generation_prompt=True)
    # 断言所有占位符都被渲染
    assert "<|image|>" in result
    assert "<|audio|>" in result
    assert "<|video|>" in result
```

评论区精华

Review 中主要讨论了与上游 HuggingFace 模板的同步策略。

- bbrowning：建议等待上游 HF PR 合并后再更新此 PR，保持模板同步。
- lucianommartins：主动与上游协调确保模板唯一性。
  - 最终上游 PR 合并后，此 PR 更新为与上游同步，并解决 DCO 和合并冲突。
  - bbrowning还指出 Chat Completions API 只允许文本在 tool response 中，而 Responses API 允许多模态，需要注意模板适配。
  - 与上游 HuggingFace 模板同步策略 (design): 采用上游模板后合并，保持与 upstream 一致。
  - Chat Completions vs Responses API 对多模态工具消息支持差异 (design): 当前变更支持 Chat Completions 中发送多模态内容，但注意到 API 规范差异，未来可考虑通过 Responses API 路径。

风险与影响

- 风险：低风险。变更仅影响 Gemma4 模型的聊天模板渲染，作用域明确。新增测试覆盖了主要多模态场景，确保模板正确处理。需要注意的是，如果上游模板未来有改动，此示例模板可能再次不同步，但这是维护成本而非直接风险。
- 影响：对使用 Gemma4 模型且启用 tool-calling 并传入多模态 tool 响应的用户有直接影响：之前多模态内容被静默丢弃，现在被正确渲染为占位符。影响程度中等，修复了明确 bug。对其他模型无影响。
- 风险标记：低风险，测试覆盖新增

关联脉络

- 暂无明显关联 PR