PR #40720 完整报告

vllm-project/vllm

feat: Enable `prompt_embeds` Content Part Support in vLLM Chat Completions API

合并时间：2026-05-01 10:05

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/40720

执行摘要

- 一句话：Chat Completions API 新增 prompt_embeds 内容部分
- 推荐动作：值得精读。本 PR 展示了在 vLLM 多模态框架中添加新内容类型的完整流程：占位符 token 注册、Chat 消息解析、token 序列展开、嵌入替换、以及嵌入验证。设计模式可复用，对于需要扩展输入模态的开发者有重要参考。讨论中关于 V0/V1 引擎差异也值得关注。

功能与动机

vLLM 已在 Completions API 中支持 prompt_embeds（预计算嵌入），但 Chat Completions API 缺乏相应支持，导致用户需要手动处理聊天模板和 token 化才能利用该特性。RFC #39504 提出了此需求：用户希望在对话中混用 pre-computed embeddings 与文本，并享受 Chat API 的工具调用解析等功能。本 PR 遵循 RFC 设计，将 prompt_embeds 作为新的内容部分类型，API 格式与 Completions API 保持一致，并保持 --enable-prompt-embeds 配置开关。

实现拆解

实现分为五步：

1. 数据结构与占位符定义：在 vllm/entrypoints/chat_utils.py 中新增 ChatCompletionContentPartPromptEmbedsParam TypedDict，定义 PROMPT_EMBEDS_PLACEHOLDER_TOKEN（<prompt_embeds>），并加入 MODALITY_PLACEHOLDERS_MAP。
2. 渲染器占位符处理：在 vllm/renderers/hf.py 中实现 _ensure_prompt_embeds_placeholder_token 注册特殊 token，_build_prompt_embeds_updates 创建 PromptReplacement 对象以在 token 序列中复制占位符，_expand_prompt_embeds_placeholders 根据位置展开，_build_mixed_prompt_embeds 将嵌入向量写入展开后的 token 位置。
3. 嵌入加载与验证：在 vllm/renderers/embed_utils.py 中改进 safe_load_prompt_embeds，增加对 hidden_size 和 dtype 的约束检查，并添加异步版本 safe_load_prompt_embeds_async。
4. 输入传递适配：扩展 vllm/inputs/engine.py 的 EmbedsInput，增加 prompt_token_ids 和 is_token_ids 字段以携带 token 位置信息。

5. 模型运行器集成：修改 vllm/v1/worker/gpu_model_runner.py 和 gpu_input_batch.py 以消费 prompt_token_ids 字段，在生成阶段正确替换嵌入。 测试配套包括：单元测试 tests/renderers/test_chat_utils_prompt_embeds.py（覆盖 token 注册、同步 / 异步解析、错误路径），E2E 测试 test_chat_completion_with_prompt_embeds.py（单 / 多部分、多消息），以及混合模态测试 test_chat_completion_with_mixed_image_embeds.py 和 test_chat_completion_with_mixed_audio_embeds.py。文档更新 docs/features/prompt_embeds.md 和示例脚本。

关键文件：

- vllm/renderers/hf.py（模块 渲染器；类别 source；类型 dependency-wiring；符号 _ensure_prompt_embeds_placeholder_token, _build_prompt_embeds_updates, _expand_prompt_embeds_placeholders, _build_prompt_embeds_positions）：核心渲染器，实现了 prompt_embeds 占位符注册、展开和嵌入替换的核心逻辑，是 PR 最关键的文件。
- vllm/entrypoints/chat_utils.py（模块 入口层；类别 source；类型 dependency-wiring；符号 ChatCompletionContentPartPromptEmbedsParam, model_config, parse_prompt_embeds, _load_prompt_embeds_async）：定义了 prompt_embeds 内容部分的 Pydantic 参数模型、占位符常量、模态占位符映射，以及消息解析逻辑。
- vllm/renderers/embed_utils.py（模块 渲染器；类别 source；类型 dependency-wiring）：提供 prompt_embeds 的安全加载和验证函数，是嵌入数据进入引擎的守门员。
- vllm/inputs/engine.py（模块 输入层；类别 source；类型 core-logic）：扩展 EmbedsInput 以支持 prompt_token_ids 字段，这是将 token 位置信息传递到模型运行器的关键。
- tests/renderers/test_chat_utils_prompt_embeds.py（模块 测试；类别 test；类型 test-coverage；符号 test_prompt_embeds_keys_registered, test_ensure_placeholder_token_is_single_token_and_idempotent, test_parse_chat_messages_openai_format）：单元测试覆盖占位符注册、同步 / 异步解析、错误路径、多种 tokenizer 系列，确保核心逻辑正确。
- tests/entrypoints/openai/chat_completion/test_chat_completion_with_prompt_embeds.py（模块 测试；类别 test；类型 test-coverage；符号 test_single_prompt_embeds_part, test_multiple_prompt_embeds_parts, test_multi_message_conversation）：端到端测试验证实际的 Chat Completions API 请求，确认 prompt_embeds 内容部分可正常生成响应。

关键符号：_ensure_prompt_embeds_placeholder_token, _build_prompt_embeds_updates, _expand_prompt_embeds_placeholders, _build_prompt_embeds_positions, _build_mixed_prompt_embeds, safe_load_prompt_embeds, safe_load_prompt_embeds_async, parse_prompt_embeds

关键源码片段

vllm/entrypoints/chat_utils.py

定义了 prompt_embeds 内容部分的 Pydantic 参数模型、占位符常量、模态占位符映射，以及消息解析逻辑。

```
# vllm/entrypoints/chat_utils.py ( 部分 )

# 模态占位符映射中新增 prompt_embeds
MODALITY_PLACEHOLDERS_MAP = {
    "image": "<##IMAGE##>",
    "audio": "<##AUDIO##>",
    "video": "<##VIDEO##>",
    "prompt_embeds": "<##PROMPT_EMBEDS##>", # 但最终 renderer 使用 PROMPT_EMBEDS_
    PLACEHOLDER_TOKEN
}

PROMPT_EMBEDS_PLACEHOLDER_TOKEN: Final[str] = "<prompt_embeds>"
"""聊天模板渲染期间每个嵌入位置使用的特殊占位符 token。
当 `--enable-prompt-embeds` 启用时注册为额外的特殊 token。
见 vllm/renderers/hf.py 中的 _ensure_prompt_embeds_placeholder_token。"""

class ChatCompletionContentPartPromptEmbedsParam(TypedDict, total=False):
    """prompt_embeds 内容部分的参数。"""
    data: Required[str]
    """Base64 编码的序列化 torch.Tensor，形状为 (num_tokens, hidden_size)。
   dtype 和 hidden_size 必须与模型嵌入层匹配。"""
    type: Required[Literal["prompt_embeds"]]
    """内容部分的类型。"""
```

评论区精华

1. 占位符一致性：gemini-code-assist 指出 MODALITY_PLACEHOLDERS_MAP['prompt_embeds']（<##PROMPT_EMBEDS##>）与 PROMPT_EMBEDS_PLACEHOLDER_TOKEN（<prompt_embeds>）不一致，导致非 OpenAI 格式失效。作者已修复统一为 <prompt_embeds>。
2. 张量验证缺失：gemini-code-assist 建议在 _build_mixed_prompt_embeds 中显式校验多个张量的 hidden_size 和 dtype；作者解释通过 safe_load_prompt_embeds 已单独校验，但后续仍添加了直接断言。
3. V0 引擎缺失：gemini-code-assist 指出 V0 model_runner 未处理 prompt_token_ids，作者未回应（PR 默认面向 V1）。
4. 测试文件位置：DarkLight1337 建议将单元测试从 tests/entrypoints/ 移至 tests/renderers/，作者已执行。
5. 性能优化：DarkLight1337 建议将 concat-and-index 改为直接切片赋值以避免冗余拷贝，作者已修改。
6. 默认 tokenize 行为：DarkLight1337 指出 render_messages 中 tokenize 默认为 False 时 prompt_embeds 路径会触发警告，作者已在外部调用层将 tokenize 默认覆盖为 True。

- 占位符 token 不一致 (correctness): 作者已修复，统一使用 PROMPT_EMBEDS_PLACEHOLDER_TOKEN。

- V0 引擎未实现 (correctness): 作者未回应此问题；该 PR 主要面向 V1 引擎，V0 缺失可能需后续 PR 跟进。
- 嵌入张量验证 (correctness): 作者说明通过 safe_load_prompt_embeds 已单独校验，但后来仍添加了直接断言。
- concat-and-index 优化 (performance): 作者已采纳，替换为直接循环赋值。
- 默认 tokenize 行为 (correctness): 作者已在调用层拓宽默认值，renderer 内的覆盖保留作为防御。

风险与影响

- 风险：
  1. V0 引擎未支持：当前实现仅修改 V1 模型运行器，V0 引擎未适配，使用 V0 并启用 prompt_embeds 会导致错误或静默失败。
  2. 兼容性：新增内容部分不影响未启用 --enable-prompt-embeds 的旧请求，但若用户端错误发送 prompt_embeds 会返回明确错误信息。
  3. 安全风险：base64 解码加载 torch 张量暴露于 pickle 攻击面；虽通过 safe_load_prompt_embeds 验证 shape 和 dtype，但未验证数据内容，建议后续添加 weights_only=True。
  4. 性能开销：base64 解码和占位符展开增加 CPU 开销，但仅在启用时发生，且 tensor 通常不大。
  5. 回归风险：修改了核心渲染器 hf.py、输入引擎 engine.py 和 GPU 模型运行器，可能影响纯文本聊天路径，测试覆盖了主要场景。 - 影响：对用户：现可直接在 Chat Completions API 中使用 prompt_embeds，无需外部模板处理。支持与文本、image_embeds、audio_embeds 混合，覆盖多模态场景。对系统：需启用 --enable-prompt-embeds 和 / 或 --enable-mm-embeds，扩展了输入数据类型。对团队：核心渲染器和输入管道新增模态扩展点，后续维护需注意占位符 token 注册和展开逻辑的一致性。影响程度：中等（新增可选功能，不破坏现有行为）。 - 风险标记：V0 引擎未支持，base64 解序列化攻击面，核心渲染器修改影响纯文本路径，新内容部分可能影响旧请求（需开启 flag）

关联脉络

- 暂无明显关联 PR