

PR #56593 完整报告

vllm-project/vllm

[Bugfix][Rust Frontend] Honor `add\_generation\_prompt` in DeepSeek V3.2/V4/V4.1 renderers

合并时间: 2026-09-15 18:03

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/56593>

执行摘要

本 PR 修复了 `vllm-chat` (Rust 前端) 中 DeepSeek V3.2/V4/V4.1 渲染器的一大类行为不一致问题: 在 `add_generation_prompt=false` 时, 最后一个 user-like 轮次后依然会被错误地追加 `和 thinking/ response` 后缀。修复后, 后缀只在最后一个渲染条目且显式禁用 `generation prompt` 时被省略, 否则照常输出以保证多轮对话语义。核心改动集中在 `deepseek.rs` 的 `next_rendered_entry` 重构与 `deepseek_v32/encoding.rs` 的 `write_user_like_suffix`` 参数化, 并配套新增 9 个测试用例锁定每个分支。改动范围集中、风险可控, 属于对核心 `prompt` 渲染链路的精准修正。

功能与动机

Rust 前端 DeepSeek 渲染器之前无条件地在最后一个 user-like 轮次 (或 V4 的尾部 system、V4.1 的 mid-system 轮次) 后追加生成提示, 任何显式关闭 `add_generation_prompt` 的调用都会收到不符合预期的输入。HF、Kimi K3、Inkling 等渲染器均已正确遵循该标志, 因此 Rust 端与 Python 端行为出现了跨端漂移。PR 作者在描述中声明这不是 #46257 (Python 端 tokenizer 的同类修复) 的重复实现, 而是针对 Rust 渲染器本身的独立修正。

实现拆解

1. 配置提取与传递: `deepseek_v32/encoding.rs` 的 `render_request` 读取 `request.chat_options.add_generation_prompt()`, 并将其作为新增参数贯穿到 `render_message`, 再转发给 `render_developer_message / render_user_message`; `deepseek.rs` 的 V4/V4.1 共用路径则在主循环内直接消费该标志。
2. V3.2 后缀逻辑修正: `write_user_like_suffix` 增加 `add_generation_prompt: bool` 参数, `false` 时直接 `return`; `render_message` 中通过 `let add_generation_prompt = message_index + 1 != messages.len() || add_generation_prompt`; 保证非最后一条消息始终追加, 最后一条消息才受显式配置约束。
3. V4/V4.1 的 `next-rendered` 判定: `deepseek.rs` 将原先返回 `bool` 的 `next_rendered_entry_is_assistant_or_end` 重构为返回 `Option<&ChatMessage>` 的 `next_rendered_entry`, 跳过被合并的尾部 `tool result` 与历史 `developer` 后取下一个真实渲染条目, 再分三种情况决定写入 `transition` (下一条是 `assistant` → 写; 下一条是结尾且 `flag` 关闭 → 不写; 下一条是其他类型 → 不写)。
4. 测试配套: 三个测试文件新增 9 个用例, 覆盖默认追加、关闭 `flag` 不追加、`user` 后接 `assistant` 仍追加、V4 尾部 `system / 合并 tool result / developer`, 以及 V4.1 的 `mid-system` 与 `chat` 模式, 全部使用 `expect!` 精确断言, 锁定了所有关键行为分支。

rust/src/chat/src/renderer/deepseek.rs

V4/V4.1 共用渲染逻辑所在文件，核心重构点：将 next_rendered_entry_is_assistant_or_end 改为返回 Option<&ChatMessage> 的 next_rendered_entry，使 add_generation_prompt 的语义在最后一个渲染条目上生效。

rust/src/chat/src/renderer/deepseek_v32/encoding.rs

V3.2 渲染器入口，write_user_like_suffix 新增 add_generation_prompt 参数并在 false 时跳过后缀，是本 PR 修复的另一半核心逻辑。

rust/src/chat/src/renderer/deepseek_v4/tests.rs

V4 渲染器测试目录，新增多个用例验证 add_generation_prompt=false 时最后一个 user-like 轮次（含合并 tool result、developer、trailing system 场景）不再输出后缀。

关键源码片段

rust/src/chat/src/renderer/deepseek_v4/tests.rs

V4 渲染器测试目录，新增多个用例验证 add_generation_prompt=false 时最后一个 user-like 轮次（含合并 tool result、developer、trailing system 场景）不再输出后缀。

```
#[test]
fn last_merged_tool_result_omits_generation_prompt_when_disabled() {
    let mut request = thinking_request(vec![
        ChatMessage::user("question"),
        // 尾部 tool result 会被合并进当前 user 块，因此不能依赖原始消息索引判断最后一个渲染条目
        ChatMessage::tool_response("result", "call-1"),
    ]);
    request.chat_options.generation_prompt_mode = GenerationPromptMode::
    NoGenerationPrompt;

    let rendered = render_request(&request);

    // 期望输出中没有 后缀，表示 transition 被正确省略
    expect!["用户问题

    <tool_result>result</tool_result>"].assert_eq(&rendered);
    assert!(!rendered.contains("</ | assistant | >"));
}

#[test]
fn last_user_turn_omits_generation_prompt_when_disabled() {
    let mut request = thinking_request(vec![ChatMessage::user("Hello")]);
    request.chat_options.generation_prompt_mode = GenerationPromptMode::
    NoGenerationPrompt;

    let rendered = render_request(&request);

    // 最后一个 user 轮次时，add_generation_prompt=false 不再输出 与 thinking 标签
```

```
assert!(rendered.ends_with("用户 Hello"));
assert!(!rendered.contains("</ | assistant | >"));
assert!(!rendered.contains(" thinking"));
}
```

评论区精华

- BugenZhao 在评审中提出扩展范围：“Would you also cover DSV4/4.1 in this PR? I believe they have the same issue.” 作者随后将 V4/V4.1 纳入修复，并解释了 key 设计选择——用 next-rendered 条目而非 raw index，因为 V4 会把尾部 tool result 合并进当前 user 块，索引判断会失效。
- Codex 自动审查对 commit cbfa5c7 给出无重大问题的结论，未提出异议。
- 讨论中没有涉及性能或安全争议，焦点在于行为一致性、边界条件覆盖与跨端对齐。

风险与影响

- 行为变更影响所有使用 Rust 前端 DeepSeek 系列模型的调用方：关闭 generation prompt 后，prompt 尾部不再出现 `，依赖该分隔符的解析工具（评估脚本、客户端日志等）可能受影响。
- next_rendered_entry 的跳过逻辑由 is_following_user_content 与 is_historical_developer 组合而成，若对这两种判定理解有误，可能把继续对话的 assistant 轮次误判为会话结束，导致 suffix 被错误省略；但新增测试已覆盖这些分支，风险可控。
- 默认行为（add_generation_prompt=true 或不改 flag）完全不变，兼容性良好。

关联脉络

- 46257: Python 端 DeepSeek-V4 tokenizer 的 `add_generation_prompt/continue_final_message` 支持，与本 PR 属同一功能线的不同端实现，PR 正文明确说明不是重复工作。
- 56931: Rust 前端新增 `--hf-overrides` 配置覆盖，体现 vllm-chat 正持续与 Python 端对齐配置与渲染行为，与本 PR 的演进方向一致，共同勾勒出 Rust 前端在 DeepSeek 系列模型上的完整支持面貌。