PR #7630 完整报告

verl-project/verl

[rollout] fix: DeepSeek continuous token builder cannot render tool appends

合并时间：2026-08-31 15:11

原文链接：http://prhub.com.cn/verl-project/verl/pull/7630

执行摘要

- 一句话：修复 DeepSeek 连续 token 构建器工具追加 TypeError
- 推荐动作：值得精读，尤其是维护自定义 tokenizer 模板或连续 token 增量路径的开发者。本 PR 展示了模板拼接契约与合成占位消息之间的类型冲突如何用最小覆盖解决，并提供了完整的真实 tokenizer 验证矩阵方法论。源码部分很小，重点看 continuous_token.py 的家族子类划分，以及测试文件中边界 tokenizer 的设计方式。

功能与动机

PR body 指出 DeepSeek R1、V3.1 和 V3.2 的 chat 模板会通过字符串拼接把 tool['function']['arguments'] 写入 prompt，而基础 ContinuousTokenBuilder._tokenize_tool_group() 会在合成 assistant 消息后渲染工具组，该合成消息的 tool calls 携带 "arguments": {} 的 dict，因此每次工具追加都会在拼接处抛 TypeError。DeepSeek-V3 是家族中唯一幸免的，因为它的模板只在 content is None 时渲染 tool_calls，而合成消息从不满足该条件。该问题是在核对 PR #7628 的 DeepSeek 家族兼容性时发现的。

实现拆解

实现按以下步骤展开：

1. 在 verl/utils/tokenizer/continuous_token.py 的 DeepSeekContinuousTokenBuilder 中新增 _synthetic_assistant_for_tools() 覆盖。该方法先调用父类生成与基础构建器一致的合成助手消息，再把每条 tool_calls 的 function.arguments 从 dict 替换为 JSON 字符串 "{}"。由于合成消息只用于与真实历史对比后 diff 掉，token 输出不变，而真实 DeepSeek 模板做字符串拼接时不再报错。
2. 在 tests/utils/test_continuous_token_on_cpu.py 中新增 _DeepSeekBoundaryTokenizer 模拟类。它以字符串拼接模拟 R1 / V3.1 / V3.2 模板行为：function["arguments"] 为 dict 时抛 TypeError，且工具消息之后不追加生成 prompt，复现真实模板的两个关键边界条件。
3. 新增 test_deepseek_builder_renders_tool_appends_through_string_concatenating_template：先用基础 ContinuousTokenBuilder 断言同样轨迹会抛 TypeError，再验证 DeepSeekContinuousTokenBuilder 走通工具追加并只输出 answer 的 token。新增 test_deepseek_builder_synthetic_tool_call_arguments_are_a_json_string 直接断言合成消息保留 tool_call_id 且所有 arguments 都是 "{}"。

4. 验证矩阵（PR body 中完成）：用真实 tokenizer 在 tool_agent_loop 轨迹上做 24 组案例（system prompt 开关 × 0 / 10 / 50 次历史工具轮次 × 1 / 2 个并行调用 × enable_thinking 默认 / False）。DeepSeek-R1 / V3.1 / V3.2 全部从 TypeError 变为可渲染，DeepSeek-V3与main输出一致；Qwen2/2.5/3/3.5、MiniMax-M2、GLM-4.7、Gemma 4、gpt-oss 的基础构建器 24/24 与 main 一致，确认基础路径无回归。

关键文件：

- verl/utils/tokenizer/continuous_token.py（模块 分词器；类别 source；类型 core-logic；符号 _synthetic_assistant_for_tools）：核心修复文件：DeepSeekContinuousTokenBuilder 新增 _synthetic_assistant_for_tools 覆盖，将合成工具调用的 arguments 从 dict 替换为 JSON 字符串 "{}"，解决 DeepSeek 家族模板字符串拼接抛 TypeError 的问题。
- tests/utils/test_continuous_token_on_cpu.py（模块 分词器；类别 test；类型 test-coverage；符号 _DeepSeekBoundaryTokenizer, init, convert_tokens_to_ids, apply_chat_template）：新增 _DeepSeekBoundaryTokenizer 字符串拼接模拟器与两个测试，复现真实 DeepSeek 模板的崩溃行为，验证修复后的增量 token 正确性，并直接校验合成消息参数为 JSON 字符串。

关键符号：DeepSeekContinuousTokenBuilder._synthetic_assistant_for_tools

关键源码片段

verl/utils/tokenizer/continuous_token.py

核心修复文件：DeepSeekContinuousTokenBuilder 新增 _synthetic_assistant_for_tools 覆盖，将合成工具调用的 arguments 从 dict 替换为 JSON 字符串 "{}"，解决 DeepSeek 家族模板字符串拼接抛 TypeError 的问题。

```
def _synthetic_assistant_for_tools(self, tool_messages: list[dict[str, Any]]) -> dict[str, Any]:
    # R1 / V3.1 / V3.2 的模板会把 tool['function']['arguments'] 直接拼接到 prompt,
    # 如果传 dict 会在拼接时抛 TypeError；这个合成助手消息只用来生成增量 token,
    # 随后会被 diff 掉，所以把 arguments 换成 JSON 字符串 "{}" 即可，不影响最终输出。
    synthetic_assistant = super()._synthetic_assistant_for_tools(tool_messages)
    for tool_call in synthetic_assistant["tool_calls"]:
        tool_call["function"]["arguments"] = "{}"
    return synthetic_assistant
```

tests/utils/test_continuous_token_on_cpu.py

新增 _DeepSeekBoundaryTokenizer 字符串拼接模拟器与两个测试，复现真实 DeepSeek 模板的崩溃行为，验证修复后的增量 token 正确性，并直接校验合成消息参数为 JSON 字符串。

```
class _DeepSeekBoundaryTokenizer(_TemplateTokenizer):
    """最小 DeepSeek 风格渲染器，模拟 R1 / V3.1 / V3.2 模板的两个关键行为：
    字符串拼接 arguments，以及工具消息之后不追加生成 prompt。"""

    name_or_path = "deepseek-ai/DeepSeek-V3.1"
    unk_token_id = 0

    def __init__(self):
```

```
        self.eos_id = 1

    def convert_tokens_to_ids(self, token):
        if token == "<eos>":
            return self.eos_id
        return self.unk_token_id

    def apply_chat_template(self, messages, tokenize=True, add_generation_prompt=True,
                            tools=None, return_dict=False, **kwargs):
        rendered = ""
        last_was_tool = False
        for message in messages:
            role = message.get("role")
            if role == "tool":
                rendered += f"<tool_output_begin>{message.get('content', '')}<tool_output_end>"
                last_was_tool = True
                continue
            last_was_tool = False
            if role == "assistant" and message.get("tool_calls"):
                calls = ""
                for tool_call in message["tool_calls"]:
                    function = tool_call["function"]
                    # 复刻真实模板的拼接写法：arguments 是 dict 时这里会抛 TypeError
                    calls += "<tool_call_begin>" + function["name"] + "<tool_sep>" +
                    function["arguments"]
                    calls += "<tool_call_end>"
                rendered += "<assistant>" + message.get("content", "") + calls + "<eos>"
            else:
                rendered += f"<{role}>{message.get('content', '')}\n"
        # 工具消息之后不加生成 prompt，保证最后一段增量只包含 tool output
        if add_generation_prompt and not last_was_tool:
            rendered += "<assistant>"
        if tokenize:
            return self.encode(rendered, add_special_tokens=False)
        return rendered
```

评论区精华

本 PR 没有公开的 review 评论线程，wuxibin89 直接 APPROVED。值得留意的是 PR body 中作者给出的关键设计取舍：#7628 的融合路径不再对全量历史做重渲染，因此本修复只改合成消息的字段类型就足以覆盖所有 DeepSeek 工具追加场景；同时基础构建器刻意保持不动，因为 Qwen 风格模板把 arguments 交给 tojson 渲染，期望一个 mapping。这一决策把模板契约差异隔离在模型家族子类内部，避免了影响面扩散。

- 暂无高价值评论线程

风险与影响

- 风险：

1. 模板契约依赖：修复依赖 DeepSeek 当前模板的字符串拼接行为，若模板未来改为对 arguments 做 JSON 序列化或接受 dict，该覆盖可能不再匹配，需随模板升级同步核查。
2. 测试覆盖局限：单元测试使用人工构造的 _DeepSeekBoundaryTokenizer，只覆盖拼接加无生成 prompt 两个关键行为；真实 tokenizer 验证在作者本地实验完成，未纳入 CI，模板更新后可能漏检。PR body 也明确 V3 / R1 的 conversation 级 is_output_first 标志无法被有界渲染看到，属已存在且未改动的行为。
3. 影响面隔离：变更只作用于 DeepSeekContinuousTokenBuilder 子类，基础 ContinuousTokenBuilder 与其他家族无 behavior 变化，回归风险较低。 - 影响：对用户：DeepSeek R1 / V3.1 / V3.2 在 agent loop 工具调用场景从每次 append 必崩变为正常可用；DeepSeek-V3 与其余模型家族用户输出完全一致，无 API 变化。对系统：连续 token 增量路径在 DeepSeek 家族恢复稳定，不影响训练、序列化等模块。对团队：变更面极小（13 行源码 + 91 行测试），维护成本低，同时确立了“按模型家族覆盖合成消息字段类型以适配模板拼接”的先例，后续可在 continuous_token.py 的家族子类中复用。 - 风险标记：模型模板拼接契约依赖，测试仅模拟 tokenizer，未纳入真实模板 CI，仅 DeepSeek 子类生效，基础路径零改动

关联脉络

- PR #7628 [rollout] fix: continuous token fuse generation prompt with the final append group: 本 PR 是在验证 #7628 对 DeepSeek 家族兼容性时发现问题后提出的；#7628 的融合路径使本修复仅改合成消息字段类型即可覆盖全部工具追加场景，两个 PR 共同构成连续 token 增量路径的完整演进。