PR #1974 完整报告

THUDM/slime

[docs] Add finer explanation for re-tokenizationi issue

合并时间：2026-05-29 14:53

原文链接：http://prhub.com.cn/THUDM/slime/pull/1974

执行摘要

该 PR 在 examples/coding_agent_rl/README.md 中新增了一个章节，详细解释了训练中 re-tokenization 带来的正确性问题以及 slime 如何通过“string-in, token-out”设计原则来保证训练轨迹中 token 的完全可溯性。纯文档改进，无代码变更。

功能与动机

此前文档对 re-tokenization 问题描述不够细致，用户可能不清楚为什么训练必须使用模型实际采样的 token 而不是重新编码解码后的字符串。新增章节清晰地解释了：

- 环境交互基于字符串，但训练必须基于 token。
- 多轮对话中，后续 prompt 可能无法与之前采样的 output token 精确匹配，造成 token 级溯源断裂。
- slime 的 merge_turns 通过智能的 loss_mask 分配来避免在这些情况下进行错误的反向传播。

实现拆解

1. 在 README 的 Fan-out Semantics 前插入 ## String-in, Token-out Trajectories 小节。
2. 描述核心契约：每个消息历史使用模型的 chat template 渲染后转为 input_ids，SGLang 调用 return_logprob=True 记录准确的 token ID 和 logprob。
3. 解释 merge_turns 如何拼接多轮 token 流：新添加的工具 / 用户 / 环境 prompt 后缀带 loss_mask=0，模型输出带 loss_mask=1。
4. 重点说明最后一个正确性保护规则：当后续 prompt 不再与之前采样的 output token 完全匹配时，保留匹配前缀，丢弃不匹配后缀，并给对应输出添加 loss_mask=0，避免反向传播到无法追溯的 token。
5. 提及 tests/test_agent_trajectory.py 覆盖了各种边缘情况（匹配前缀、跳过轮次、分割输出漂移、token 计数变化、prompt 基础重启）。

无源码变更。

评论区精华

无 review 讨论。

风险与影响

纯文档更新，无技术风险。影响范围限于文档读者，使训练流程设计更加透明，降低用户误用 re-tokenization 的风险。

关联脉络

本文档关联 PR#1963（修复 trajectory 合并逻辑），`merge_turns` 函数的正确性保证在此文档中做了完整解释。属于 *agent* 功能线文档完善的一部分。