

PR #1923 完整报告

THUDM/slime

[examples] add coding_agent_rl: agent-in-sandbox RL minimal demo

合并时间: 2026-05-26 23:34

原文链接: <http://prhub.com.cn/THUDM/slime/pull/1923>

执行摘要

- 一句话: 新增编码代理 RL 示例, E2B 沙盒 + Claude Code 集成
- 推荐动作: 值得精读, 尤其关注 `middleware.py` 的设计模式: 如何通过 Chain/Session 实现多对话链跟踪、raw token 拼接保证 TITO、以及 `generate.py` 的容错设计。对于希望实现类似代理 RL 管线的工程师有很强参考价值。

功能与动机

提供一个最小且可读的编码代理 RL 示例, 展示如何将真实编码工具 (Claude Code) 与沙盒执行和基于测试的奖励集成到 slime 训练循环中。PR body 描述了完整的端到端流程, 强调无需事后重新标记化。

实现拆解

1. Sandbox 层 (`sandbox.py`): 封装 E2B API, 提供 `exec`、`upload`、`write_text`、`read_text` 等 5 个原语, 并实现 `_is_transient_rpc_error` 与 `_rpc_retry` 以应对网络波动。
2. LLM 中间件 (`middleware.py`): 实现 Anthropic Messages API ↔ SGLang /generate 协议转换; 核心数据类 Chain 和 Session 维护对话链与会话状态; `_select_chain` 决定延续主链还是子代理链; `_build_prompt` 通过 chat template 加 raw token 拼接避免重新分词; `_generate` 调用 SGLang 并验证 TITO; `_build_reply` 将 `output_ids` 翻译回 Anthropic SSE 块。
3. 样本生成 (`generate.py`): 实现 slime custom-generate 入口 `generate()`; 每个样本: 启动 Agent 沙盒 → 安装 Node 与 Claude Code → 执行代理 → 捕获 git diff → 启动评估沙盒 → 应用 diff 并运行测试 → 从中间件拉取 token 段 → 折叠或扇出为 Sample 对象; 包含墙钟超时保护、失败软降级等容错设计。
4. aiohttp 线程工具 (`aiohttp_threaded.py`): 在后台 daemon 线程中运行 aiohttp 应用, 用于中间件的 HTTP 服务。
5. 启动脚本与文档: `run_qwen36_35b_a3b_swe_8nodes.sh` 展示完整的模型并行与 rollout 配置; `README.md` 说明环境设置、数据集格式和运行指南。

关键文件:

- `examples/coding_agent_rl/middleware.py` (模块 中间件; 类别 source; 类型 dependency-wiring; 符号 `_strip_cache_control`, `_hash`, Chain, Session): 核心中间件, 实现 Anthropic ↔ SGLang 协议转换和 token 捕获, 是整个 RL 数据链路的关键组件。

- `examples/coding_agent_rl/generate.py` (模块 样本生成; 类别 `source`; 类型 `dependency-wiring`; 符号 `_State`, `init`, `_boot_agent_sandbox`, `_write_segment_to_sample`): 自定义 `generate` 入口, 编排整个端到端 RL 样本生成流程, 包括沙盒管理、中间件交互、容错逻辑。
- `examples/coding_agent_rl/sandbox.py` (模块 沙盒; 类别 `source`; 类型 `dependency-wiring`; 符号 `_parse_sandbox_metadata`, `_is_transient_rpc_error`, `_rpc_retry`, `E2BSandbox`): 沙盒封装层, 提供 E2B 原语操作和容错重试机制。
- `examples/coding_agent_rl/aiohttp_threaded.py` (模块 网络服务; 类别 `source`; 类型 `dependency-wiring`; 符号 `AppHandle`, `url`, `stop`, `_shutdown`): 提供在后台线程中运行 `aiohttp app` 的工具, 用于启动中间件 HTTP 服务。
- `examples/coding_agent_rl/run_qwen36_35b_a3b_swe_8nodes.sh` (模块 启动脚本; 类别 `infra`; 类型 `dependency-wiring`): 8 节点启动脚本, 展示了完整的模型并行、rollout 配置和环境变量设置。
- `examples/coding_agent_rl/README.md` (模块 文档; 类别 `docs`; 类型 `documentation`): 文档说明, 帮助用户理解示例的结构、环境设置和运行方法。

关键符号: `generate`, `_boot_agent_sandbox`, `_write_segment_to_sample`, `_fan_out_to_samples`, `_handle_request`, `_select_chain`, `_build_prompt`, `_generate`, `_build_reply`, `_strip_cache_control`, `_hash`, `Chain`, `Session`, `_make_segment`, `_flatten`, `_translate_anthropic`, `_parse_sandbox_metadata`, `_is_transient_rpc_error`, `_rpc_retry`, `E2BSandbox`, `run_app_in_thread`, `AppHandle`

评论区精华

1. 文件重命名: zhuzilin 建议将 `bridge.py` 重命名为 `middleware.py`, 以避免与 `slime` 已有的 `mbridge` 和 `megatron bridge` 混淆。作者采纳。
 2. `asyncio` 锁的必要性: zhuzilin 质疑单线程下是否需要 `session` 锁。作者解释 `_run_turn` 包含 `await _post_generate` (HTTP 调用), 同一 `session` 并发请求会交错修改状态, 导致 TITO 校验失败; 锁是 `per-session` 的。
 3. `session_id` 唯一性: 作者将 `sid` 改为 `f'cagent-{instance_id}-{sample.index}-{sample.group_index}'`, 确保在 rollout 步骤内唯一。
 4. TITO 校验失败处理: 讨论确认会 `mask` 整条 `response` 的 `loss_mask`, 代码为 `target.loss_mask[-n:] = [0] * n`。
 5. 流式请求同步化: 作者解释 `_post_generate` 调用同步 `sglang` 接口获取完整 `output_ids`, 再一次性流式返回, 避免增量流式损坏 TITO。
 6. 客户端取消处理: 通过 `handler_cancellation` 和 `/abort_request` 避免已取消生成与 `release_memory_occupation` 竞争。
- 文件重命名: `bridge` → `middleware (design)`: 作者采纳, 将 `bridge.py` 更名为 `middleware.py`, 并更新所有导入。
 - `asyncio` 锁的必要性 (`correctness`): 作者坚持锁的必要性, 解释合理, 保留锁。
 - `session_id` 唯一性 (`correctness`): 作者修改实现, 确保唯一性。

- TITO 校验与 loss_mask 策略 (correctness): 确认策略, 不匹配时置零当前 response 的 loss_mask。
- 流式请求同步化 (performance): 设计确认: 攒齐后再流式发送。

风险与影响

- 风险:
 - 外部服务依赖: E2B 沙盒和 Claude Code 均为外部系统, 网络波动或版本升级可能导致训练故障。_rpc_retry 和超时提供部分容错, 但仍无离线方案。
 - Token 捕获正确性: 中间件的 TITO 验证关键; 当前设计在失败时静默置零 loss_mask, 可能丢失训练信号。
 - 环境一致性: 运行脚本中硬编码的 Node 和 Claude Code tarball 路径需与集群同步, 否则沙盒环境不匹配。
 - 资源开销: 沙盒启动与销毁涉及 I/O 和网络, 每个样本延迟高; 扇出模式可能使样本数激增, 导致显存压力。
 - 模型适配: 示例针对 Qwen3.5 配置, 切换模型需调整 chat template、tool parser 和 reasoning parser。
- 影响:
 - 用户: 提供了一个开箱即用的编码代理 RL 示例, 只需配置 E2B 和 Claude Code 路径即可实验。
 - 系统: 未修改 slime 核心模块, 仅通过 --custom-generate-function-path 挂钩; 不影响现有训练流程。
 - 团队: 为后续复杂代理 RL workflow 提供了可扩展的参考架构, middleware 的设计模式可直接复用。
 - 维护成本: 作为示例代码, 需与主仓库同步更新; 环境变量接口的变更可能破坏现有脚本。
 - 风险标记: 外部服务依赖, token 捕获正确性, 环境一致性, 资源开销, 模型适配

关联脉络

- PR #1933 [2/N] Support training with variable global batch size: 该示例依赖 PR #1933 的动态 batch size 功能以实现扇出样本的正确损失计算, 提交日志中多次出现 [pr1933 port] 说明直接引用此 PR。
- PR #1926 Move micro-batch scheduling from training side to rollout side: PR #1926 的 micro-batch 调度移动影响了 generate 函数的接口, 该示例的脚本中使用了 --use-dynamic-batch-size 等配置项。