PR #1979 完整报告

THUDM/slime

[agent] Add openai and anthropic adapters

合并时间：2026-05-29 19:30

原文链接：http://prhub.com.cn/THUDM/slime/pull/1979

执行摘要

- 一句话：新增 OpenAI 和 Anthropic 适配器并重构 agent 模块
- 推荐动作：建议所有参与 agent RL 训练的成员阅读此 PR，尤其是 common.py 中的 AdapterChain 和 call_sglang_generate 设计，它为后续添加新的 adapter 提供了清晰的模式。openai.py 中的 _translate_chat_messages 和 _responses_input_to_messages 体现了多协议消息格式转换的典型做法，值得参考。

功能与动机

为了支持更多 LLM API 协议（OpenAI 和 Anthropic）作为 agent 推理的驱动，使 slime 的 agent RL 训练可以接入更广泛的生态工具，同时将适配器逻辑从示例中解耦出来，提升可维护性和可扩展性。

实现拆解

1. 创建公共基模块slime/agent/adapters/common.py，定义了协议无关的 AdapterChain 数据结构、session 生命周期管理函数（register_session、request_session_id）、token 渲染函数 render_token_ids、SGLang 生成调用 call_sglang_generate 以及采样参数合并函数 _sampling_params。
2. 实现 OpenAI 适配器slime/agent/adapters/openai.py，对外暴露 /v1/chat/completions 和 /v1/responses 端点，负责将 OpenAI 格式的消息体通过 chat template 转为 input_ids，调用 SGLang 并记录 token 级推理轨迹。内部实现了消息扁平化 _flatten_content、工具调用标准化 _normalize_tool_call、角色映射 _translate_chat_messages 等函数。
3. 迁移 Anthropic 适配器将 examples/coding_agent_rl/middleware.py 重构并搬迁至 slime/agent/adapters/anthropic.py，复用公共基类中的 Chain、session 管理、render_token_ids、call_sglang_generate 等，精简了约 200 行重复代码。同时保留了对 Claude Code 多轮对话和子 agent 链的分段支持。
4. 添加测试套件新增 tests/test_agent_adapters.py（单元测试，覆盖消息翻译、session 提取、SSE 解析、流式响应等）和 tests/test_agent_sdk_adapters.py（集成测试，模拟 OpenAI SDK 和 Anthropic SDK 的完整工具循环）。
5. 调整下游代码与 CI更新 examples/coding_agent_rl/generate.py 和 sandbox.py 中的导入路径，将 middleware 引用替换为 slime.agent.adapters.anthropic；扩充 CI 配置文件以包含新的 adapter 测试用例。

关键文件：

- slime/agent/adapters/openai.py（模块 OpenAI 适配器；类别 source；类型 dependency-wiring；符号 Session, _flatten_content, _normalize_tool_call, _translate_chat_messages）：新增的 OpenAI 适配器主文件，实现 chat/completions 和 responses 端点，核心消息转换与工具调用标准化逻辑所在。
- slime/agent/adapters/common.py（模块 公共基类；类别 source；类型 dependency-wiring；符号 AdapterChain, strip_cache_control, stable_hash, json_arguments）：适配器公共基础设施，包含 AdapterChain 数据结构、session 管理、token 渲染、采样参数合并和 SGLang 调用封装。
- slime/agent/adapters/anthropic.py（模块 Anthropic 适配器；类别 source；类型 rename-or-move；符号 _strip_cache_control, _hash, Chain, _render_token_ids）：从 examples/coding_agent_rl/middleware.py 重构并搬迁过来的 Anthropic 适配器，复用公共基类，精简了大量重复代码。
- tests/test_agent_adapters.py（模块 测试套件；类别 test；类型 test-coverage；符号 ToyTokenizer, init, apply_chat_template, decode）：新增单元测试，覆盖 OpenAI 和 Anthropic 适配器的消息翻译、session 提取、SSE 解析、流式响应等关键路径。
- tests/test_agent_sdk_adapters.py（模块 集成测试；类别 test；类型 test-coverage；符号 SDKTokenizer, init, apply_chat_template, decode）：集成测试，通过 OpenAI 和 Anthropic SDK 实际调用适配器，验证工具循环和多轮对话的正确性。
- examples/coding_agent_rl/generate.py（模块 示例脚本；类别 source；类型 dependency-wiring）：更新导入路径，从 middleware 切换到 slime.agent.adapters.anthropic，并调整相关注释和配置。
- .github/workflows/pr-test.yml（模块 CI 配置；类别 infra；类型 infrastructure）：扩展 CI 配置以运行新增的 adapter 测试用例，确保新代码持续集成。

关键符号：slime/agent/adapters/openai::_translate_chat_messages, slime/agent/adapters/openai::_flatten_content, slime/agent/adapters/openai::_normalize_tool_call, slime/agent/adapters/common::_sampling_params, slime/agent/adapters/common::call_sglang_generate, slime/agent/adapters/common::register_session, slime/agent/adapters/anthropic::_build_prompt, slime/agent/adapters/anthropic::_select_chain

关键源码片段

slime/agent/adapters/openai.py

新增的 OpenAI 适配器主文件，实现 chat/completions 和 responses 端点，核心消息转换与工具调用标准化逻辑所在。

```
# 代码片段：_translate_chat_messages 将 OpenAI 消息列表转换为 chat-template 兼容格式
# 处理角色映射、内容扁平化、工具调用标准化等

from slime.agent.adapters.common import json_arguments as _json_arguments
```

```
from slime.agent.adapters.common import stable_hash as _hash

def _translate_chat_messages(messages: list[dict]) -> list[dict]:
    """OpenAI chat messages -> tokenizer chat-template messages."""
    translated: list[dict] = []
    for msg in messages:
        if not isinstance(msg, dict):
            continue
        role = msg.get("role")
        content = msg.get("content")
        # 将 'developer' 角色统一为 'system'
        if role == "developer":
            role = "system"

        # 系统 / 用户消息：扁平化 content 字段
        if role in {"system", "user"}:
            translated.append({"role": role, "content": _flatten_content(content)})
        # 工具返回消息：保留 tool_call_id
        elif role == "tool":
            tool_msg = {"role": "tool", "content": _flatten_content(content)}
            if msg.get("tool_call_id"):
                tool_msg["tool_call_id"] = msg["tool_call_id"]
            translated.append(tool_msg)
        # 助手消息：携带推理内容和工具调用
        elif role == "assistant":
            assistant: dict[str, Any] = {"role": "assistant", "content": _flatten_content(content)}
            if msg.get("reasoning_content"):
                assistant["reasoning_content"] = msg["reasoning_content"]
            tool_calls = msg.get("tool_calls") or []
            if tool_calls:
                # 标准化每个工具调用格式
                assistant["tool_calls"] = [_normalize_tool_call(c) for c in tool_calls if isinstance(c,
                dict)]
            translated.append(assistant)
    return translated
```

slime/agent/adapters/common.py

适配器公共基础设施，包含 AdapterChain 数据结构、session 管理、token 渲染、采样参数合并和 SGLang 调用封装。

```
# 代码片段：_sampling_params 合并 session 默认参数和请求 body 中的采样参数
# 同时支持上下文长度截断保护

def _sampling_params(session: Any, body: dict, *, max_token_keys: tuple[str, ...], stop_keys:
tuple[str, ...]) -> dict:
    # 默认参数：禁止跳过特殊 token、禁止额外空格、不裁剪 stop
    sp: dict[str, Any] = {
        "skip_special_tokens": False,
        "spaces_between_special_tokens": False,
```

```
        "no_stop_trim": True,
        "max_new_tokens": 4096,
        **(session.sampling_defaults or {}),
    }

    # 从 body 中按优先顺序选取 max_new_tokens
    for key in max_token_keys:
        if body.get(key) is not None:
            sp["max_new_tokens"] = min(int(sp.get("max_new_tokens", body[key])), int(body[key]))
            break

    # 覆盖温度、top_p、top_k
    for src_k, dst_k in (("temperature", "temperature"), ("top_p", "top_p"), ("top_k", "top_k")):
        if src_k in body:
            sp[dst_k] = body[src_k]

    # 处理 stop 序列
    for key in stop_keys:
        if body.get(key):
            sp["stop"] = body[key]
            break

    return sp
```

slime/agent/adapters/anthropic.py

从 examples/coding_agent_rl/middleware.py 重构并搬迁过来的 Anthropic 适配器，复用公共基类，精简了大量重复代码。

```
# 代码片段：Session 数据结构和 _build_prompt 函数
# Session 包含主链和可能的子 agent 链，支持分段轨迹

import dataclasses
from slime.agent.adapters.common import AdapterChain as Chain
from slime.agent.adapters.common import render_token_ids
from slime.agent.trajectory import TurnSegment

@dataclasses.dataclass
class Session:
    main: Chain = dataclasses.field(default_factory=Chain)
    active_sub: Chain | None = None # 最多一个活跃子 agent
    pending_dispatch_id: str = ""
    sampling_defaults: dict = dataclasses.field(default_factory=dict)
    max_context_tokens: int = 0
    lock: asyncio.Lock = dataclasses.field(default_factory=asyncio.Lock)
    segments: list[TurnSegment] = dataclasses.field(default_factory=list) # 冻结的输出

def _build_prompt(target: Chain, body: dict, kind: str, tok) -> list[int]:
    """根据 kind 选择替换或扩展消息，然后渲染 token id。"""
    if kind == "append":
```

```
        _extend_chat_messages(target, body)
    else:
        _replace_chat_messages(target, body)
    return render_token_ids(target, tok)
```

评论区精华

该 PR 未产生 review 讨论。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 重构后 examples/coding_agent_rl 的依赖路径发生变化，需确保现有启动脚本和配置兼容。
  2. 新的 OpenAI adapter 处理 responses 格式时对 multi-turn 工具循环的支持尚未在生产环境中验证。
  3. 测试套件依赖于 agents、anthropic、openai SDK，若环境缺少这些依赖会 skip 集成测试，可能掩盖回归。
  4. session_id 提取逻辑同时支持多种 header/body 字段，可能存在优先级冲突。 - 影响：对用户：开发者现在可以通过 OpenAI 或 Anthropic 协议驱动 agent 推理，而不仅限于 Anthropic Claude Code 的专用格式。对系统：适配器代码集中到 slime/agent/adapters，降低了与示例代码的耦合，便于后续扩展新协议（如 Google Gemini）。对团队：需要了解新的模块结构，并保持 common.py 作为协议无关层的通用性。 - 风险标记：核心路径变更，缺少生产验证，重构影响现有流程，依赖外部 SDK 测试

关联脉络

- PR #1956 Add slime/agent/ and move sandbox impl inside: 该 PR 创建了 slime/agent 模块并定义了 sandbox 接口，本次 PR 在此基础上增加了 adapters 子包。
- PR #1960 Extract more util code from coding_agent_rl example: 将 coding_agent_rl 中的通用工具抽取到 slime/agent，为本次适配器迁移提供了 trajectory 和 parsing 基础。
- PR #1963 Fix trajectory merging logic: 修复了轨迹合并逻辑，本次适配器依赖的 merge_turn_segments 等函数正确性由该 PR 保证。
- PR #1965 Don't use sample.index as default rollout_id: 修复了 rollout_id 默认值问题，影响适配器 session 标识的稳定性。