这篇技术报告提出了完全通过端到端 agentic reinforcement learning 进行训练的自主智能体 Kimi-Researcher，旨在通过多步骤规划、推理和工具使用来解决复杂问题。

—— **End-to-end agentic RL is promising but challenging**

传统 agent

1. 基于工作流：需要随着模型或环境的变化而频繁手动更新，缺乏可扩展性和灵活性。
2. **使用监督微调 (SFT)进行模仿学习**：在数据标记方面存在困难；特定的工具版本紧密耦合。

Kimi-Researcher：给定一个查询，agent 探索大量可能的策略，获得正确解决方案的奖励 —— 所有技能（规划、感知和工具使用）都是一起学习的，无需手工制作的rule/workflow。

建模

给定状态观察(如系统提示符、工具声明和用户查询)，Kimi-Researcher 会生成 think和action (action 可以是工具调用，也可以是终止轨迹的指示)。

$$\begin{cases} (s_t) \xrightarrow{\text{Kimi-Researcher}} (\text{think}_t, \text{action}_t) & \\ s_{t+1} = \text{context\_manager}(s_t, \text{think}_t, \text{tool\_call\_result}_t) & \text{if action}_t \neq \text{finish} \\ \text{terminate} & \text{if action}_t = \text{finish} \end{cases}$$

Approach

主要利用三个工具：a)并行、实时、内部的 **search tool;** b) 用于交互式 Web 任务的基于文本的 **browser tool**; c) 用于自动执行代码的 **coding tool**.

1. Training data

设计了具有两个互补目标的训练语料库

1. **构建 tool-centric 的任务**： the agent learns when to invoke a tool, and how to orchestrate tool use effectively.
2. **构建 reasoning-intensive 的任务**：数学和代码推理 + Hard Search

开发了完全自动化的管道，以最少的人工干预生成和验证许多问答对。

2. RL training

主要使用 REINFORCE 算法

1. **On-policy Training**：每个轨迹都完全基于模型自身的概率分布生成。
2. **Negative Sample Control**：负样本会导致 token probabilities 降低。
3. **使用 outcome rewards 进行训练**：

Format Reward：惩罚 a)无效工具调用的轨迹 b)上下文/迭代超过最大限制

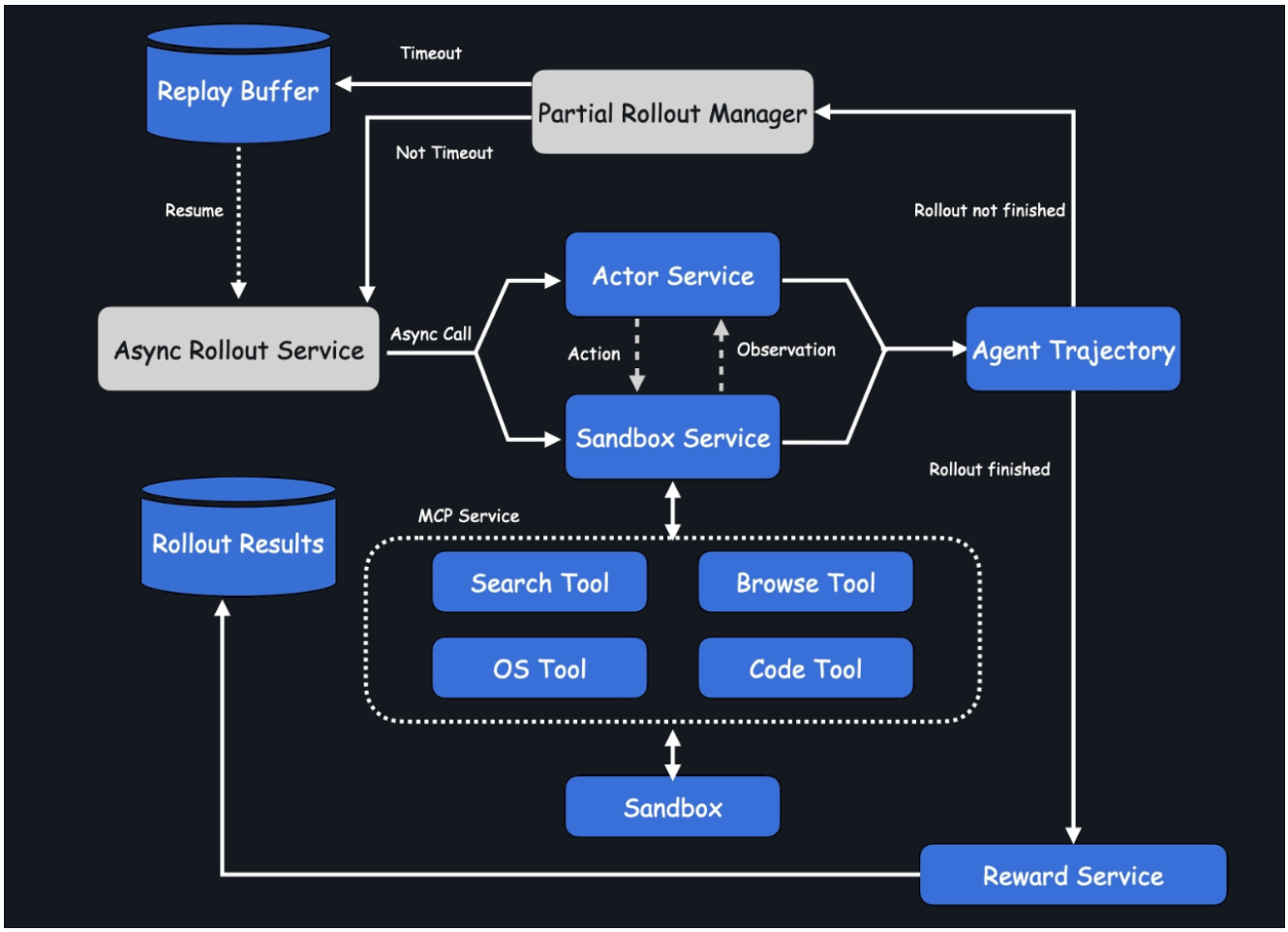
Correctness Reward：对于没有格式错误的轨迹，基于答案与真实值直接的比较进行奖励

应用 gamma-decay factor 来校正轨迹，鼓励更短更高效的探索。

3. Context management

允许模型保留重要信息，同时丢弃不必要的文档。

4. Large-scale agent RL infra



Highlights

1. 解决冲突

当看到来自多个来源的冲突信息时，Kimi-Researcher 通过iterative hypothesis refinement 和 self-correction 来解决不一致。

2. 小心严谨

即使是看似简单的问题，Kimi-Researcher 也会有意在回答之前进行额外的搜索，并交叉验证信息。