目录

K2 Thinking

智能体 (Agentic) 训练范式

智能体训练概述

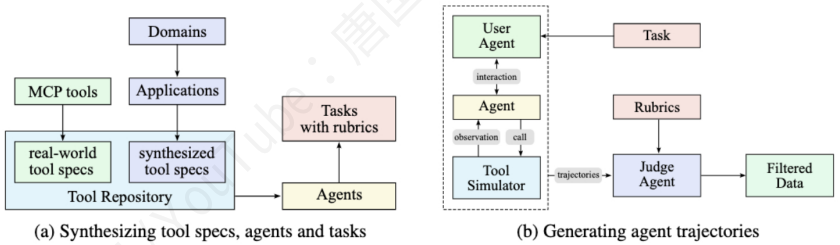
K2 Base 是一个强大的基础模型，而 K2 Thinking 是通过一个精密的后训练 (Post-Training) 过程，将其进化为"智能体"。

K2 Thinking 核心理念：端到端训练，使"函数调用 (API/工具)"成为模型推理流程中的一个**原生动作**。

- **智能体范式**

这种设计使模型超越静态数据的限制，通过自身的探索和工具使用来获取新能力，实现**"思考 → 行动 → 再思考"**的智能体范式。

阶段一：SFT (监督微调)



(a) Synthesizing tool specs, agents and tasks (b) Generating agent trajectories

Figure 8: Data synthesis pipeline for tool use. (a) Tool specs are from both real-world tools and LLMs; agents and tasks are the generated from the tool repo. (b) Multi-agent pipeline to generate and filter trajectories with tool calling.

- 目标

教会模型如何使用工具

- 挑战

真实世界的工具使用数据难以大规模获取

K2 的方案：大规模智能体数据合成流水线 —— 一个三阶段系统，用以模拟真实世界的工具使用场景。

1. 工具规范生成 (Tool Spec Generation)

构建一个庞大的工具库：

3,000+

真实世界工具

(如 GitHub 的 MCP)

20,000+

合成工具

(模拟各种专业场景)

2. 智能体与任务生成 (Agent and Task Generation)

. 基于工具库，生成数千个具有不同能力、专业领域和行为模式的智能体

. 为它们设计从简单到复杂的任务

. 配备明确的**成功标准 (rubric)**

3. 轨迹生成 (Trajectory Generation)

这是一个复杂的多组件系统，用于模拟智能体完成任务：

用户模拟器

由 LLM 扮演不同沟通风格的用户

工具执行环境

一个复杂的工具模拟器（功能上等同于世界模型），负责执行工具调用并提供真实反馈

质量评估与过滤

由一个基于 LLM 的"**评估员**"智能体，根据预设标准评估每个交互轨迹，**只保留成功的轨迹**用于训练

混合方法 (Hybrid Approach)：为弥补模拟环境的真实性不足， 团队在编码和软件工程等关键领域，将模拟与**真实执行沙盒**相结合，确保模型能从真实世界的反馈中学习。

阶段二：RL (强化学习)

- **目标**
  在 SFT 基础上，进一步提升模型的词元效率和泛化能力，特别是在主观偏好任务和复杂推理任务上

K2 的方案：通用强化学习框架，包含两大类奖励机制

1. 可验证奖励的 "Gym" (RLVR)

适用场景：有明确对错标准的任务

数学/STEM

提供具有中等难度的高质量问答对

编码与软件工程

从 GitHub 收集问题，并利用**可执行的单元测试**来验证代码的正确性

忠实性 (Faithfulness)

训练一个句子级别的忠实性判断模型，作为奖励模型来提升模型回答的事实准确性

2. 自评奖励机制 (Self-Critique Rubric Reward)

适用场景：没有唯一正确答案的主观任务 (如创意写作)

机制：K2 模型同时扮演"**行动者 (Actor)**"（生成回答）和 "**评论家 (Critic)**"（评估回答）两个角色

工作流程

"评论家"K2 会根据一套预定义的规则 (Rubrics)（如清晰性、客观性、禁止奉承等）， 对"行动者"K2 生成的多个回答进行比较和排序，从而产生偏好信号

闭环优化

"评论家"模型会利用来自"可验证奖励 Gym"的客观信号进行持续更新和校准。 这确保了其主观判断力是建立在可验证的数据基础之上，实现了可靠对齐

RL 算法改进

预算控制

`Budget Control`

对超长回答进行惩罚，鼓励简洁高效

PTX 损失

`PTX Loss (Pre-Training eXtension)`

加入辅助的预训练损失，防止"灾难性遗忘"

在 RL 训练过程中，模型可能过度优化奖励信号而遗忘预训练阶段学到的通用知识。PTX 损失通过在总损失中加入预训练数据的损失项，确保模型在优化特定任务的同时保持原有的语言理解和生成能力。

温度衰减

`Temperature Decay`

从探索转向利用，确保收敛

最终成果：通过 SFT 和 RL 两个阶段的训练， K2 Thinking 成功地将基础模型进化为一个能够自主思考、规划和使用工具的智能体， 能够在复杂动态环境中完成需要 200-300 步连续工具调用的任务。