

# MEM1: Learning to Synergize Memory and Reasoning for Efficient Long-Horizon Agents

开源代码: [github.com/MIT-MI/MEM1](https://github.com/MIT-MI/MEM1)

## MEM1: 针对TIR(Tool-Integrated Reasoning)场景的Memory机制

Zijian Zhou<sup>\*12</sup> Ao Qu<sup>\*13</sup> Zhaoxuan Wu<sup>1</sup> Sunghwan Kim<sup>4</sup>  
Alok Prakash<sup>1</sup> Daniela Rus<sup>13</sup> Jinhua Zhao<sup>13</sup> Bryan Kian Hsiang Low<sup>13</sup>  
Paul Pu Liang<sup>3</sup>

<sup>1</sup>Singapore-MIT Alliance for Research and Technology Centre

<sup>2</sup>National University of Singapore

<sup>3</sup>MIT

<sup>4</sup>Yonsei University

### 简介

本文提出MEM1，一种面向TIR(Tool-Integrated Reasoning)多轮推理场景的记忆(memory)框架，核心创新在于将“记忆(Memory)更新”融入“思考(Thinking)过程”。简单来说，目前TIR做法是将之前所有轮(step/turn)的<think>、<tool calling>和<tool output>拼接起来放到prompt中，而MEM1最多用前两轮的<think>、<tool calling>和<tool output>拼接起来放到prompt中，因为<think>...</think>的内容不仅包含对接下来的思考还包含了对历史轮信息的摘要(看作memory)。通过这种方式，MEM1实现了常量长度的上下文记忆管理，避免了prompt膨胀问题。整体上用的是RLVR训练方式，通过prompt template约束<think>...</think>还要包含历史信息的摘要，此外还提出二维attention mask来约束每个token只能attend 到其生成时可见的真实上下文，避免因上下文错配导致的梯度污染。

### 背景

在TIR(Tool-Integrated Reasoning, 工具增强推理) inference阶段，目前的做法是把之前所有step的<think>、<tool calling>和<tool output>都拼接到prompt中，然后进行下一个step的<think> → <tool calling> → <tool output>直到生成<answer>。如果query比较复杂，TIR推理需要更多的step/turn，本文作者认为这会导致prompt length越来越大甚至过大，为此，作者提出了专门针对TIR场景做动态记忆(memory)压缩的MEM1方案。

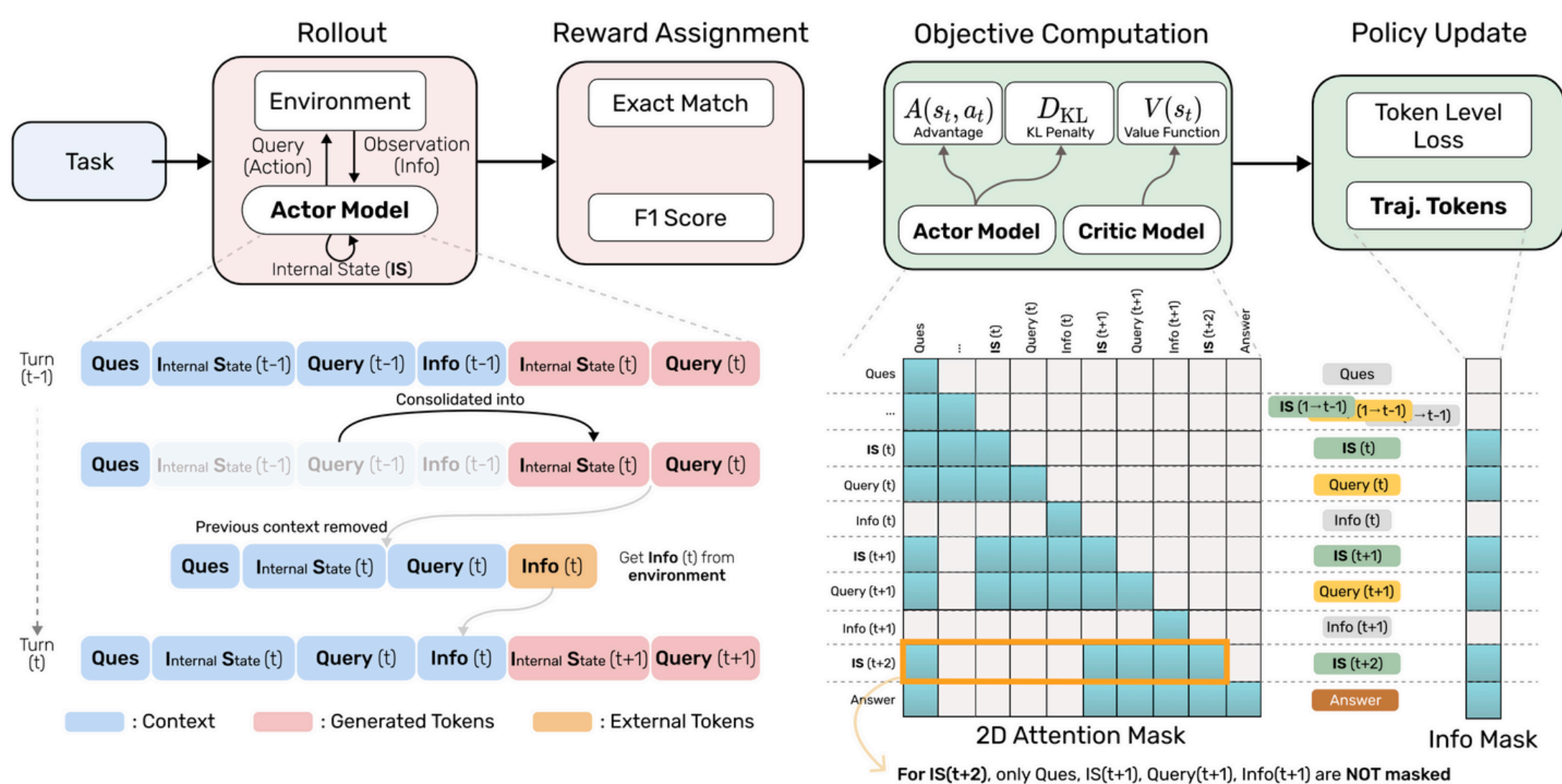
注意：本文的memory机制和之前读过的mem0、MemoryOS完全不同，它们关注的是在通用对话或交互系统中保存长期用户记忆。

### 实验设置

- 训练框架：verl
- 实验对象：Qwen2.5-7B Base；RL算法：PPO
- reward function：根据任务不同选择exact match或F1，没有用format reward
- 右图很好的解释了本文memory机制的使用场景，用于压缩/提炼TIR的reasoning trajectory



### MEM1



### 思考

说实话，在读这篇论文的时候我一直在问自己，这写的是啥？这哪里是memory，和mem0、MemoryOS完全不一样啊，这不就是TIR？最后只能去看代码，总算搞清楚了，本文的memory是针对TIR场景的，看懂之后觉得作者思考的这个问题还挺有意思，此时自动脑补彪哥语录：学吧 学无止境 太深了。

本文的研究领域可以部分归结为memory agent，我觉得还可以和之前看的AutoRefine等工作联系起来，即对TIR的reasoning trajectory做改动，当前目的都是为了得到更准确的answer。