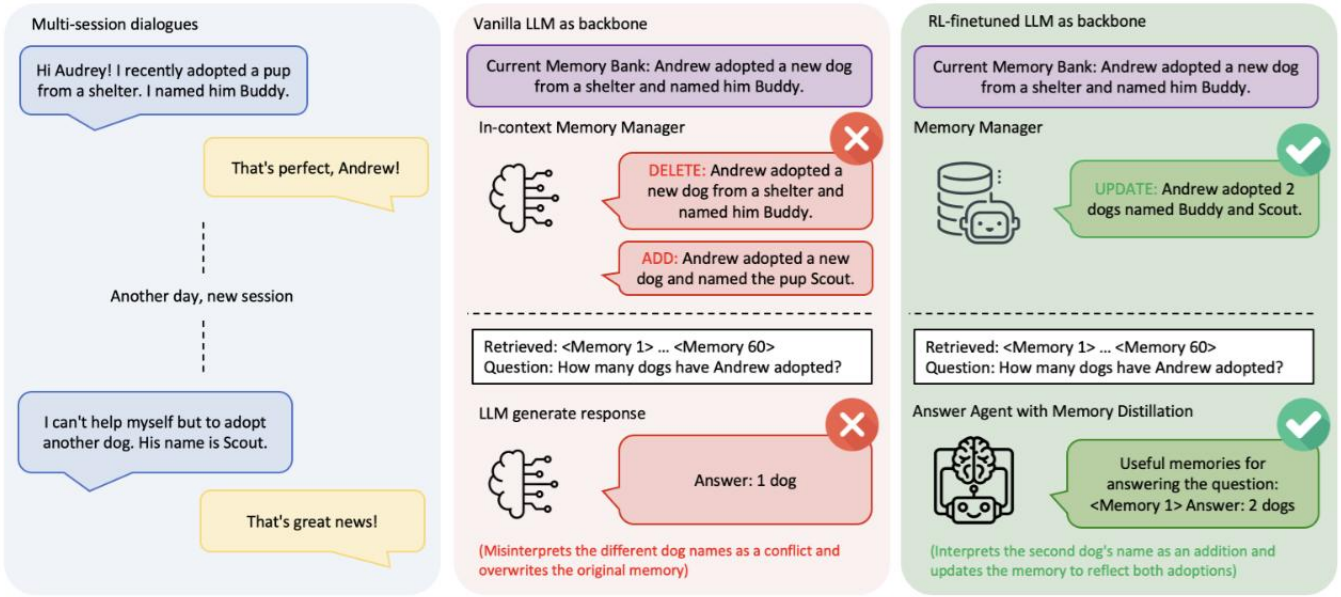
背景

北京大学 PEKING UNIVERSITY

- 当前大语言模型（LLM）虽能力强，但 无状态，受限于上下文窗口，难以长期记忆和跨会话推理。
- 现有解决方案的缺陷：
  - 外部记忆库（如RAG）：
    - 检索过多：无关信息会淹没关键信息
    - 检索过少：可能会遗漏关键信息
  - 启发式记忆管理： 大多依赖固定的规则，缺乏自适应学习能力来决定何时存储、更新或遗忘信息。
- 核心问题：如何让LLM像人一样，学会主动、智能地管理和利用自己的记忆?
- MEM-R1

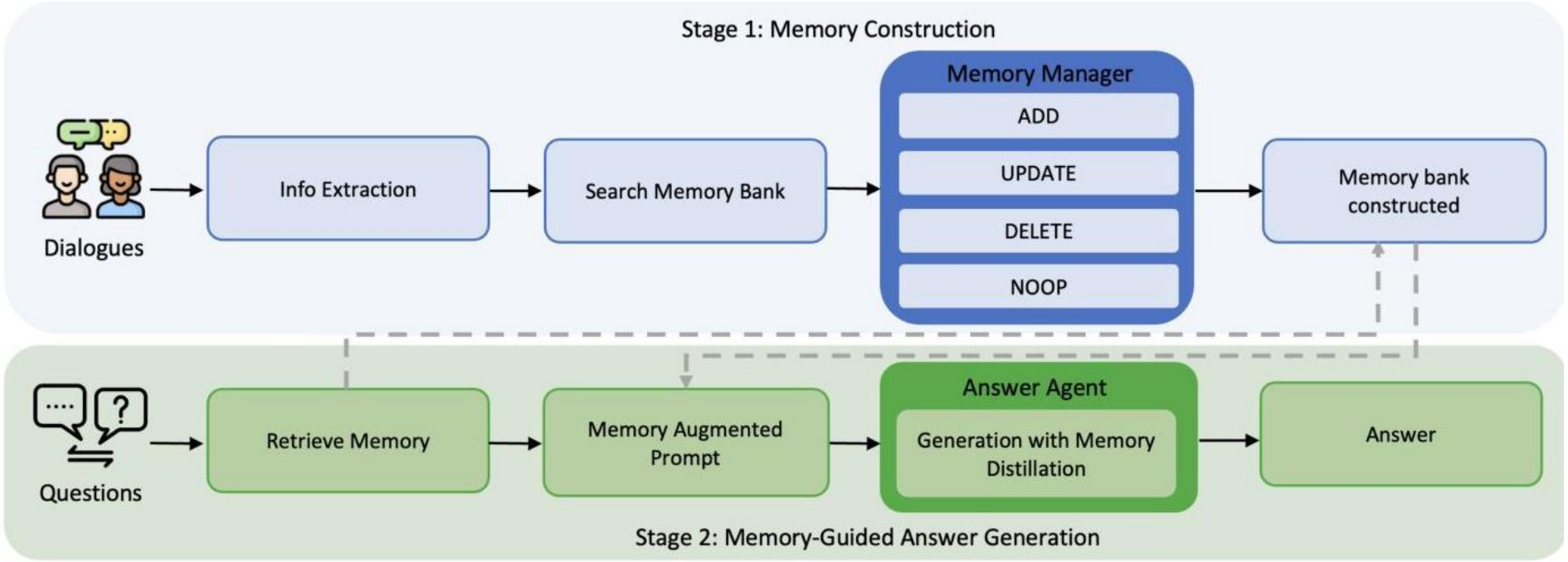
研究目标和主要贡献

- 设计一个框架，使LLM代理能够 主动管理（增删改查）记忆条目，并 合理检索-使用这些记忆，从而增强跨会话、长时程的推理能力。
- 贡献概括：提出 Memory-R1 框架：
  - 两个代理——Memory Manager 与 Answer Agent。
  - 用强化学习（PPO / GRPO）对两个代理进行微调，仅 152 对 QA 样本就能达到显著效果。
  - 在多个基准（如 LoCoMo、MSC、LongMemEval）和多种模型规模（3B–14B）上验证可扩展性与泛化性



具体方法

- 外部记忆库（memory bank）与代理系统交互。
- Memory Manager：负责对记忆条目执行 ADD、UPDATE、DELETE、NOOP 操作。
- Answer Agent：在检索出的多个记忆条目中进行“记忆蒸馏（Memory Distillation）”，筛选出少数相关条目，再结合这些条目＋当前输入生成答案。



具体方法

- Memory Manager
  - 操作集合：{ADD, UPDATE, DELETE, NOOP}
  - 使用强化学习(PPO/GRPO)，使 Manager 根据 downstream 回答质量（奖励）来优化操作选择
  - 根据旧的记忆和输入，策略模型给出Operation以及具体内容。Answer Agent（冻结，不参与训练）据此得到Answser，从而就计算Reward

$$(o, m') \sim \pi\theta(\cdot \mid x, \mathcal{M}_{\text{old}}),$$

- Answer Agent
  - 从检索模块（RAG）得到一批记忆条目，典型多达几十条。然后通过 Memory Distillation 筛选出真正相关的条目（减少噪声）。
  - 经过筛选后，这些条目 + 当前输入一起被用来生成答案。
  - Agent 同样通过 RL 微调（PPO/GRPO），奖励与最终答案准确度相关。

$$y \sim \pi_{\text{ans}}(\cdot \mid q, \mathcal{M}_{\text{ret}}).$$

实验结果

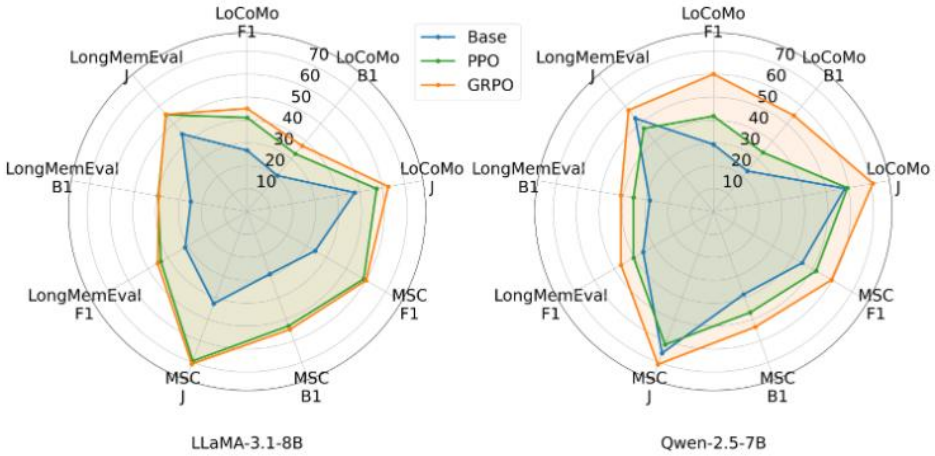
北京大学 PEKING UNIVERSITY

- 数据集：
  - LoCoMo：一个多轮对话数据集，旨在跨对话推理。1：1：8 train/validation/test划分
  - MSC, LongMemEval：未经训练，直接进行测试以验证泛化性
- 模型：Llama-3.1-8B-Instruct, Qwen-2.5-Instruct (3B, 7B, 14B).
- 评测指标：Token级别的F1-score，BLEU-1，LLM-as-Judge
- 训练资源：4张H100(80G)，全量微调，因为训练样本数少，所以时间可接受

Model	Method	Single Hop			Multi-Hop			Open Domain			Temporal			Overall		
		F1↑	B1↑	J↑	F1↑	B1↑	J↑	F1↑	B1↑	J↑	F1↑	B1↑	J↑	F1↑	B1↑	J↑
LLaMA-3.1-8B Instruct	LoCoMo (RAG)	12.25	9.77	13.81	13.69	10.96	20.48	11.59	8.30	15.96	9.38	8.15	4.65	11.41	8.71	13.62
	Zep	30.15	17.15	52.38	15.04	11.56	33.33	26.67	18.44	45.36	3.49	2.68	27.58	22.60	15.05	42.80
	A-Mem	21.62	16.93	44.76	13.82	11.45	34.93	34.67	29.13	49.38	25.77	22.14	36.43	29.20	24.40	44.76
	Mem0	27.29	18.63	43.93	18.59	13.86	37.35	34.03	24.77	52.27	26.90	21.06	31.40	30.41	22.22	45.68
	Memory-SFT	34.64	23.73	56.90	20.80	16.26	37.35	46.47	37.35	63.27	47.18	34.58	54.65	42.81	32.98	58.76
	Memory-R1-PPO	32.52	24.47	53.56	26.86	23.47	42.17	45.30	39.18	64.10	41.57	26.11	47.67	41.05	32.91	57.54
	Memory-R1-GRPO	**35.73**	**27.70**	**59.83**	**35.65**	**30.77**	**53.01**	**47.42**	**41.24**	**68.78**	**49.86**	**38.27**	**51.55**	**45.02**	**37.51**	**62.74**
Qwen-2.5-7B Instruct	LoCoMo (RAG)	9.57	7.00	15.06	11.84	10.02	19.28	8.67	6.52	12.79	8.35	8.74	5.43	8.97	7.27	12.17
	Zep	31.02	21.39	42.85	20.42	15.76	23.81	25.25	21.34	42.26	8.94	8.42	29.31	23.22	18.78	38.99
	A-Mem	18.96	12.86	40.78	14.73	12.66	31.32	30.58	26.14	46.90	23.67	20.67	28.68	26.08	21.78	40.78
	Mem0	24.96	18.05	61.92	20.31	15.82	48.19	32.74	25.27	65.20	33.16	26.28	38.76	30.61	23.55	53.30
	Memory-SFT	27.81	20.25	57.74	24.62	22.28	46.99	43.33	34.06	66.85	44.41	34.32	**52.71**	39.51	30.84	61.13
	Memory-R1-PPO	**34.22**	23.61	57.74	**32.87**	**29.48**	**53.01**	44.78	38.72	66.99	42.88	30.30	42.25	41.72	33.70	59.53
	Memory-R1-GRPO	33.64	**26.06**	**62.34**	23.55	20.71	40.96	**46.86**	**40.92**	**67.81**	**47.75**	**38.49**	49.61	**43.14**	**36.44**	**61.51**

实验结果

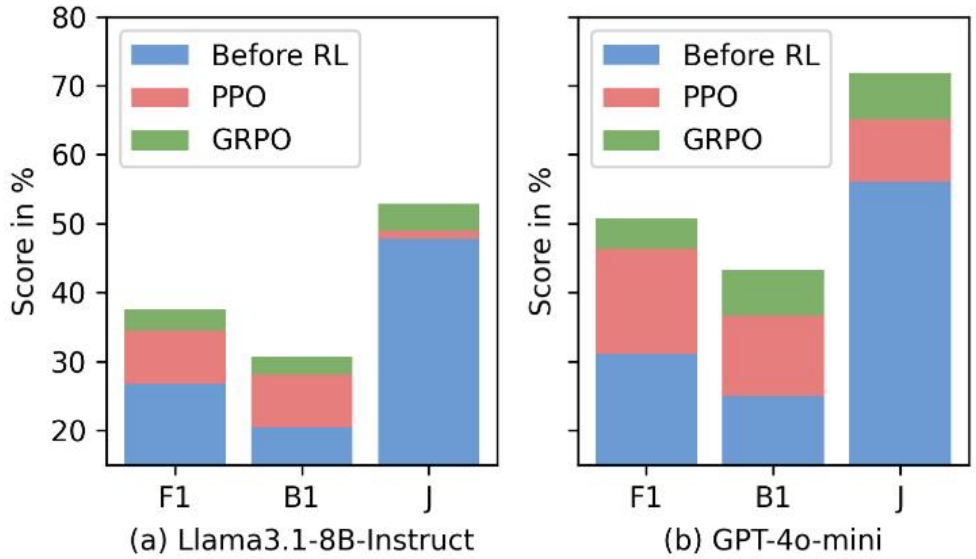
- GRPO的效果略优于PPO，Qwen效果优于Llama



- Exact Answer用以计算Reward效果更好

Method	F1↑	B1↑	J↑
PPO (J-based reward model)	33.69	23.36	**63.58**
PPO (EM-based reward model)	**41.05**	**32.91**	57.54

- 更强大的Memory Manager对下游任务更好



- GRPO强于PPO/SFT的原因可能是样本利用率高

北京大學
PEKING UNIVERSITY

Thanks