Google Cloud

Learn-by-interact: A Data-Centric Framework For Self-Adaptive Agents in Realistic Environments

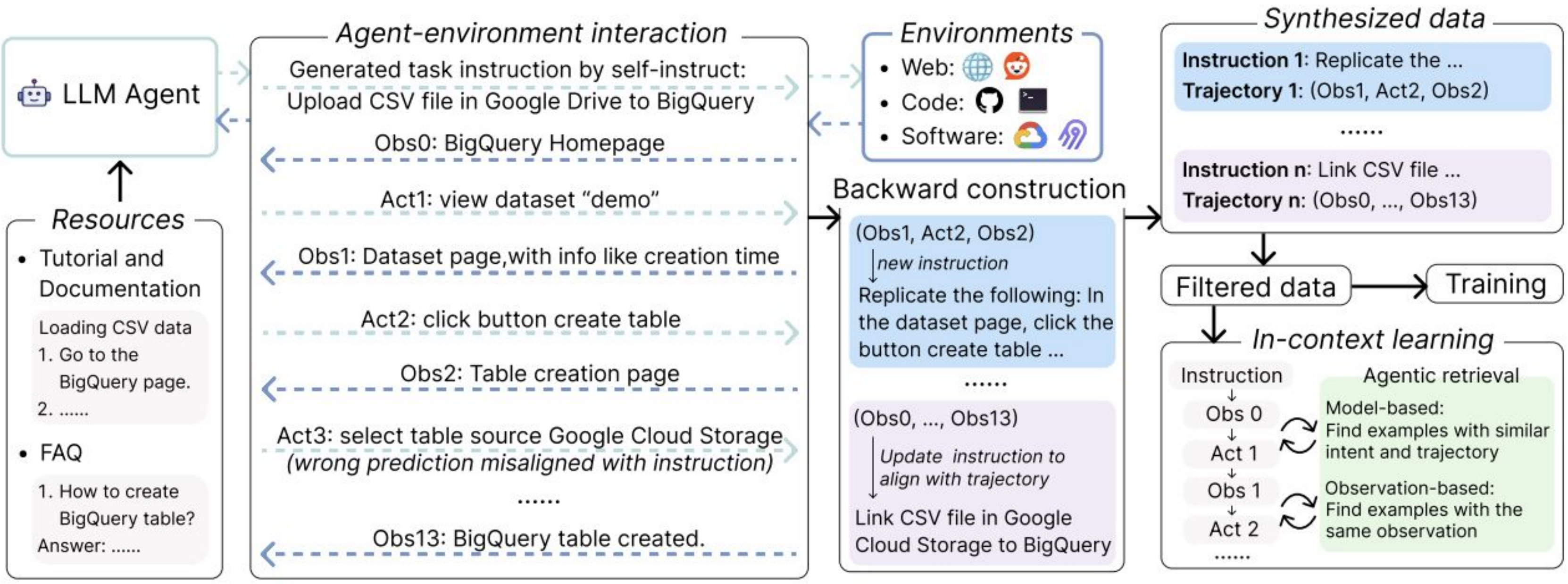
Hongjin Su[1,2], Ruoxi Sun[1], Jinsung Yoon[1], Pengcheng Yin[1], Tao Yu[2], Sercan Arik[1]

[1]Google, [2]The University of Hong Kong

Motivation

- LLMs have great potential for assisting humans with various tasks in digital settings, such as editing images, performing data analysis, resolving software engineering issues, and navigating commercial platforms.
- Low performance in realistic scenarios, e.g., 38% in os, 16% in professional software
- Insufficient data and difficulties for humans to annotate long trajectories in complex environments

Data synthesis pipeline



Algorithm 1: Agent data synthesis

```
Input: LLM: Large Language Model; E: environment; Doc: standard resources like documentation; N: the number of instructions to generate per document; F: data filter.
Initialization: D = []: synthesized data.
for d in Doc do
    // self-instruct to generate N task instructions
    Instructions = LLM(d, N)
    for I in Instructions do
        E.reset()
        T = []    // initialize interaction trajectory
        while not E.finished() do
            o = E.get_observation()
            a = LLM(I, T, o)
            T += [o, a]
        end while
        T.append(E.get_observation())
        // backward construction
        for i in range(0, len(T) − 1, 2) do
            for j in range(i + 2, len(T), 2) do
                T′ = T[i : j]
                I′ = LLM(T′)
                D.append([I′, T′])
            end for
        end for
    end for
end for
D = F(D) // Filter low-quality data
Return: D
```

Filtering

- Remove duplicate states
- LLM committee check: coherent, natural, reasonable, aligned with instructions

Data statistics

	SWE-bench	WebArena	OSWorld	Spider2-V
Documents	6,464	3,578	7,362	11,231
Raw trajectories	4,568	3,967	1,125	1,226
Examples	41,237	32,319	19,688	21,525
Filtered examples	10,232	10,456	11,782	10,169

Algorithm 2: Agentic retrieval

Input: LLM: Large Language Model; E: environment; D: synthesized data; OR: observation retrieval model; RM: dense retriever; I: task instruction; $m1$: maximum number of examples from observation-based retrieval; $m2$: maximum number of examples from model-based retrieval.

Initialization: $H = []$: interaction history; R: retrieved examples.

while not E.finished() **do**
    $o = E$.get_observation()
    // observation-based retrieval
    $R = OR(o, D, m1)$
    // model-based retrieval
    $q = LLM(I, H, o)$
    $R \mathrel{+}= RM(q, D, m2, R)$
    $a = LLM(I, H, o, R)$
    $H+ = [o, a]$
end while

Evaluation datasets

- SWE-bench (Jimenez et al., 2023) is an evaluation benchmark on realistic software engineering problems from realistic Github issues. We use the verified version by default throughout the experiments.
- Webarena (Zhou et al., 2023b) evaluates agent capabilities to perform tasks in the web environments such as e-commerce, social forum discussion, and beyond.
- OSWorld (Xie et al., 2024) is an integrated environment for assessing open-ended computer tasks, which involve diverse applications like Terminal, Chrome, etc.
- Spider2-V (Cao et al., 2024) is a multimodal agent benchmark focusing on professional data science and engineering workflows, which includes BigQuery, Airbyte and more.

Baselines

- Baseline: The vanilla prediction pipeline in each benchmark that includes the task instruction, interaction history and the state observation in the prompt. See more implementation details in Appendix A.
- RAG: The conventional RAG pipeline that first retrieves from the resources like documentation based on the instruction, and augments LLMs with the retrieved content.
- Data distill: We follow the same pipeline to synthesize data in Algorithm 1 except backward construction (replace lines 15-22 with $D.append(I, T)$), and follow Algorithm 2 during the evaluation.
- Reflexion (Shinn et al., 2024): A general framework to reinforce language agents through linguistic feedback from both executors and LLMs.
- Language Agent Tree Search (LATS) (Zhou et al., 2023a): It integrates the combinatorial tree search into expanding ReAct (Yao et al., 2022b) and combine agent online reasoning, acting and planning throughout the trajectory.

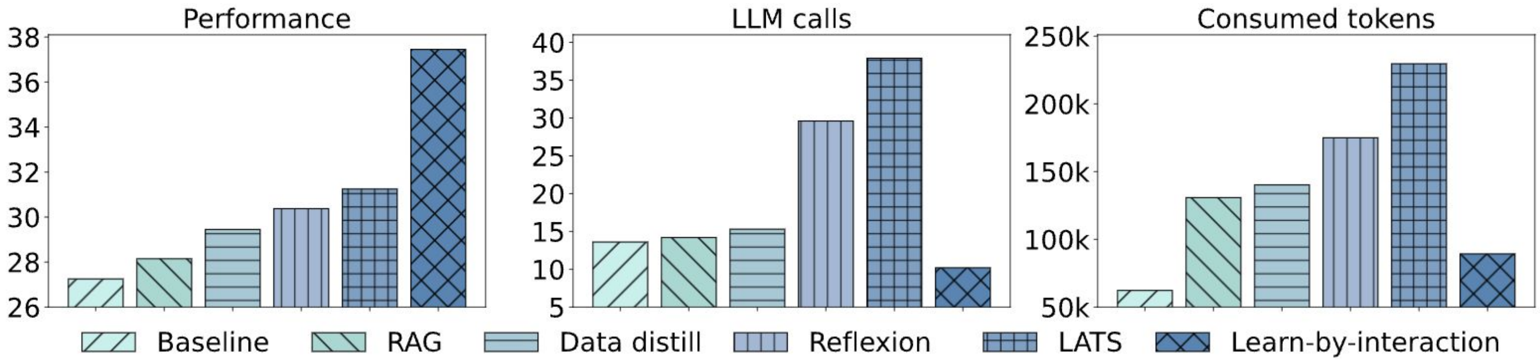
Training-free evaluation

Benchmark →	SWE	Web	OS	Spider2-V	SWE	Web	OS	Spider2-V
Approach ↓	Gemini-1.5-pro				Claude-3.5-sonnet			
	Existing approaches							
Baseline	13.3	17.9	4.9	8.3	51.2	35.8	12.4	8.4
RAG	13.7	19.5	5.1	9.1	51.8	36.9	12.8	9.2
Data distill	14.0	19.8	5.7	9.1	54.0	39.2	12.9	9.7
Reflexion	14.3	20.2	5.7	9.3	54.4	40.4	15.6	10.5
LATS	15.3	21.0	6.5	11.3	55.2	41.3	16.8	11.2
	Ours							
Learn-by-interact	**18.7**	**25.6**	**10.3**	**16.4**	**60.0**	**48.0**	**22.5**	**16.6**
Δ over baseline	+5.4	+7.7	+5.4	+8.1	+8.8	+12.2	+10.1	+8.2

Training-based evaluation

Benchmark →	Web	OS	Web	OS	Web	OS	Web	OS
Model →	Codegemma-7B		Codestral-22B		Codegemma-7B		Codestral-22B	
Approach ↓	***Before tuning***				***After tuning***			
	Existing approaches							
Baseline	3.3	0.0	4.7	2.2	-	-	-	-
Data distill	4.2	0.0	5.8	2.7	6.2	1.4	10.2	5.4
	Ours							
Learn-by-interact	7.6	3.5	9.9	5.4	14.6	6.5	24.2	11.7
Δ over baseline	+4.3	+3.5	+5.2	+3.2	+11.3	+6.5	+19.5	+9.5

Inference efficiency



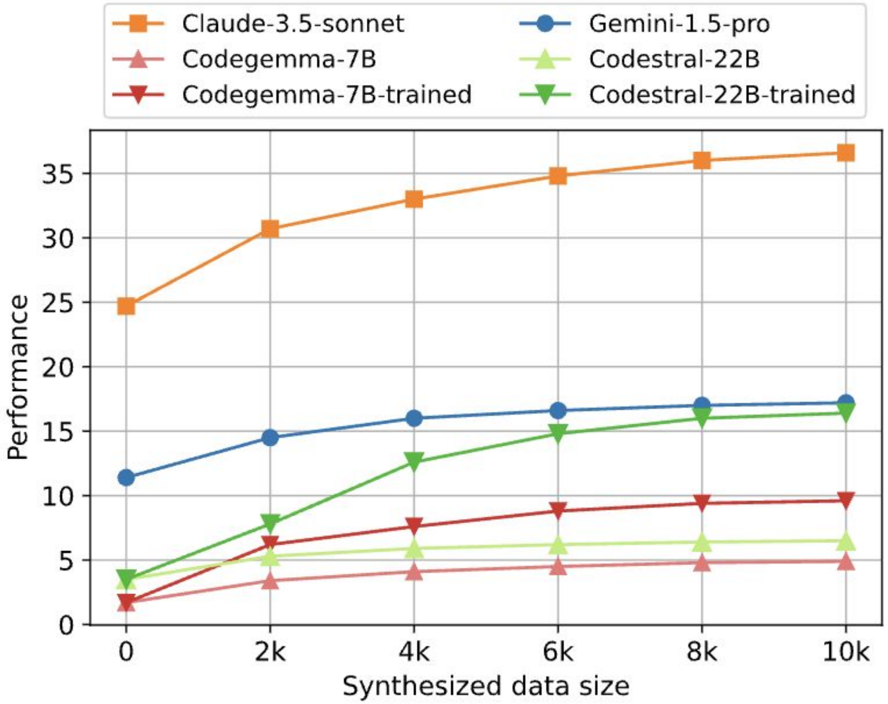
The impact of retrieval

Benchmark →	SWE	Web	OS	Spider2-V	SWE	Web	OS	Spider2-V
Retrieval ↓	Gemini-1.5-pro				Claude-3.5-sonnet			
No retrieval	13.3	17.9	4.9	8.3	51.2	35.8	12.4	8.4
Instruction-based	14.7	21.6	7.0	10.2	52.4	36.6	15.0	9.6
Observation-based	16.3	23.5	8.7	14.6	53.6	42.5	17.2	10.5
Model-based	17.0	24.3	9.5	15.4	57.8	44.8	20.3	13.7
Ours	18.7	25.6	10.3	16.4	60.0	48.0	22.5	16.6

Data granularity

Benchmark →	SWE	Web	OS	Spider2-V	Web	OS
Granularity ↓	Claude-3.5-sonnet				Codestral-22B	
Baseline	51.2	35.8	12.4	8.4	4.6	2.2
Short	54.2	39.4	17.9	10.8	13.5	4.9
Medium	53.6	38.8	16.6	9.7	12.6	4.0
Long	52.2	37.6	15.2	9.2	10.6	3.4
Short+Medium	54.6	41.2	18.8	11.3	14.6	5.7
Short+Long	54.0	40.5	17.8	10.7	14.4	5.3
Medium+Long	53.8	38.6	17.2	10.4	13.2	4.5
Short+Medium+Long	55.0	42.0	19.8	12.3	15.4	6.3

Scaling laws



Thank you!