



# MINT: Evaluating LLMs in Multi-turn Interaction with Tools and Language Feedback

Xingyao Wang\*, Zihan Wang\*, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, Heng Ji

{xingyao6, zihanw, jiateng5, yangyic3, lifan4, haopeng, hengji}@illinois.edu



## Motivation

- People have been using LLMs in **multi-turn interactions** (e.g., conversations, LLM agents with tools)
- Such multi-turn interactions typically involves **natural language feedback** from human users
- Existing LLM evaluations predominantly focus on single-turn input-output pairs, often overlook user-provided natural language feedback.

MINT benchmark measures LLMs' ability to solve tasks with multi-turn interactions by (1) using tools and (2) leveraging natural language feedback.

## Interaction Framework

LLM can (1) optionally express its reasoning process ("Thought"); (2) then either interact with tools by generating and executing Python code through a Python interpreter ("Execute"), or proposing a solution to the user ("Propose Solution").

- **Baseline:** LLM interacts with a lazy user (w/o language feedback) that only provides minimal feedback on task outcome for up-to  $k$  interaction turns
- **Informative:** LLM interacts with a user (w/ language feedback) for up-to  $k$  interaction turns

**Interaction Turn #1**

**User:** Task: Together Lily, David, and Bodhi collected 43 insects. Lily found 7 more than David. David found half of what Bodhi found. How many insects did Lily find? Solution output format: an integer.

**LLM:** Thought: Now that we know that Bodhi found 18 insects [...] Lily found 7 more than David, so Lily found 18 + 7 insects.  
Execute: `lily_insects = solution[0] + 7`  
`lily_insects`

**Interaction Turn #2**

**Python Interpreter:** Observation: Out[1]: 18.0 - 0.25\*b  
You have 4 steps left and 2 chances to propose solution left.

**User (optional, LLM-simulated):** This is NOT GOOD. [...] You should have used the information that David found half of what Bodhi found [...]. Then, you can express the total insects in terms of Bodhi's insects and solve for it [...].

**LLM:** Thought: [Apologize ...] Bodhi found 18 insects [...] David found half of what Bodhi found, so David found (1/2) \* 18 insects. Lily found 7 more than David, so Lily found (1/2) \* 18 + 7 insects.  
Execute: `lily_insects = (1/2) * solution[0]`  
`lily_insects = david_insects + 7`  
`lily_insects`

**Interaction Turn #3**

**Python Interpreter:** Observation: Out[2]: 18.000000000000000  
You have 3 steps left and 2 chances to propose solution left.

**User:** Good job! You have successfully solved the task! [stops interaction]

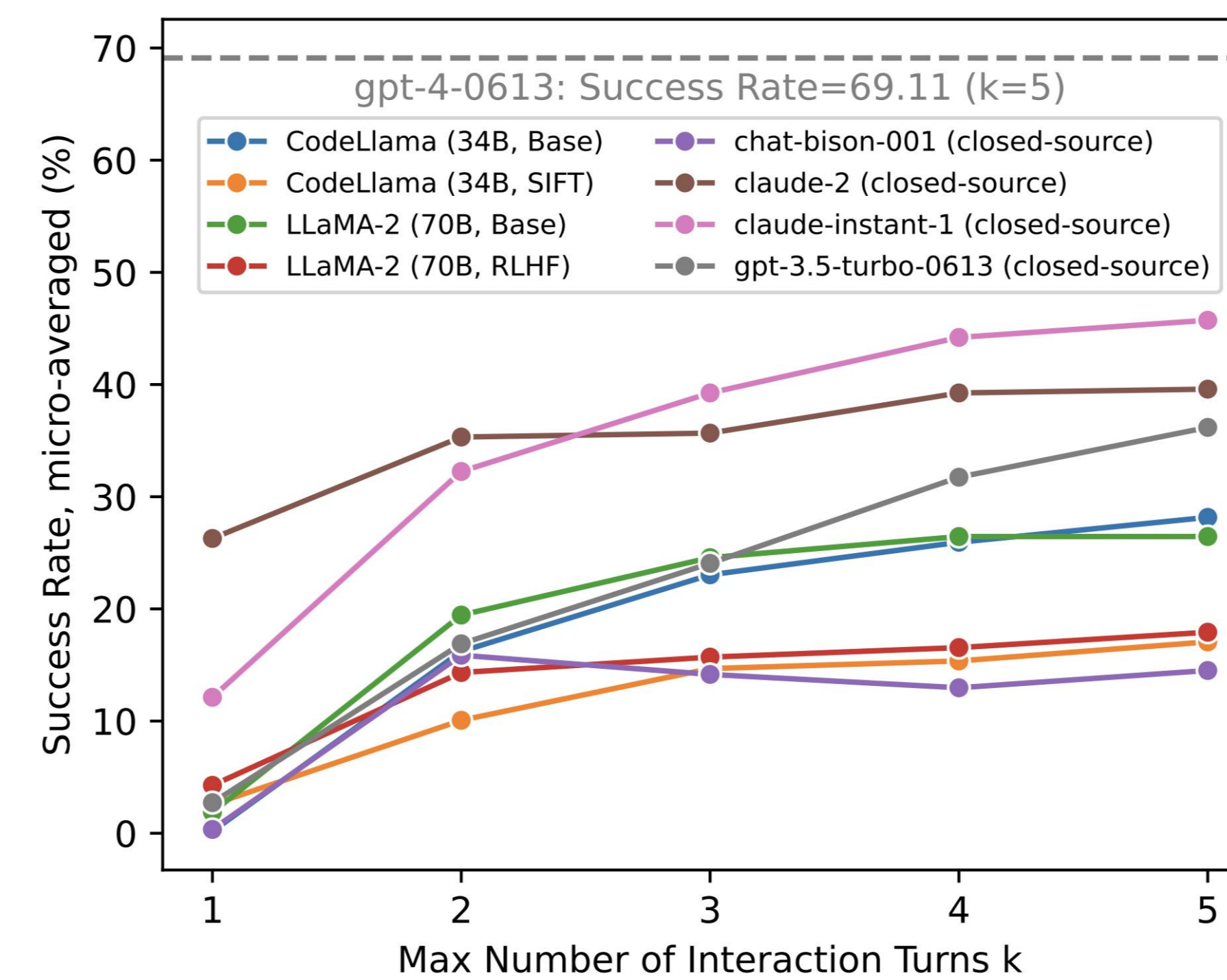
## Evaluation Data

Task Type	Task Name	Original Size	Reduced Size in MINT
Code Generation	HumanEval (Chen et al., 2021)	164	45
	MBPP (Austin et al., 2021)	500	91
Decision Making	ALFWorld (Shridhar et al., 2020)	134	134
Reasoning	GSM8K (Cobbe et al., 2021)	1319	48
	HotpotQA (Yang et al., 2018)	7,405	43
	MATH (Hendrycks et al., 2021)	5,000	100
	MMLU (Hendrycks et al., 2020)	13,985	76
	TheoremQA (Chen et al., 2023a)	800	49
<b>Total</b>		<b>29,307</b>	<b>586</b>

- **Repurposing single-turn tasks into multi-turn:** Reasoning, Code Generation, Decision-making.
- **Keeping instances that require multi-turn interaction:** throw away instances that can be solved by gpt-3.5 within 2 turns.
- **Stratified sub-sampling for efficient evaluation**

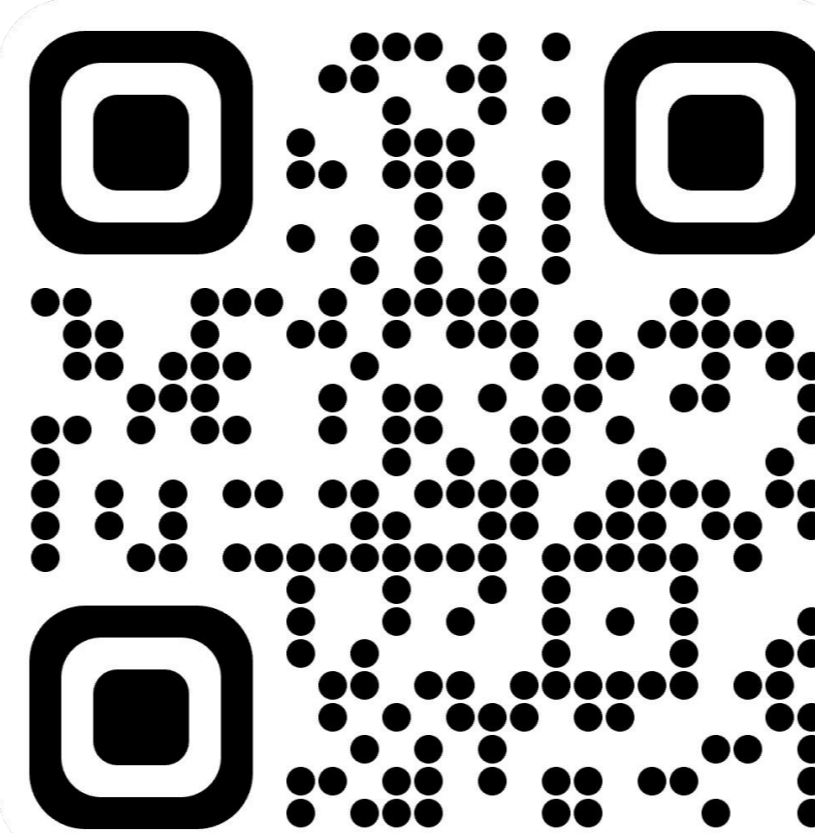
## Evaluating LLM in Multi-Turn Interaction

- **Better single-turn performance does not necessarily entail better multi-turn performance** (claude-2 vs. claude-1)
- **Absolute performance and improvement-per-turn scale with model size** (Llama-2, CodeLLaMA)
- **SIFT on multi-turn data can be helpful for multi-turn interaction** (Vicuna-v1.5, Lemur-v1)
- **RLHF might hurt LLM-tool multi-turn interaction** (LLaMA-2)

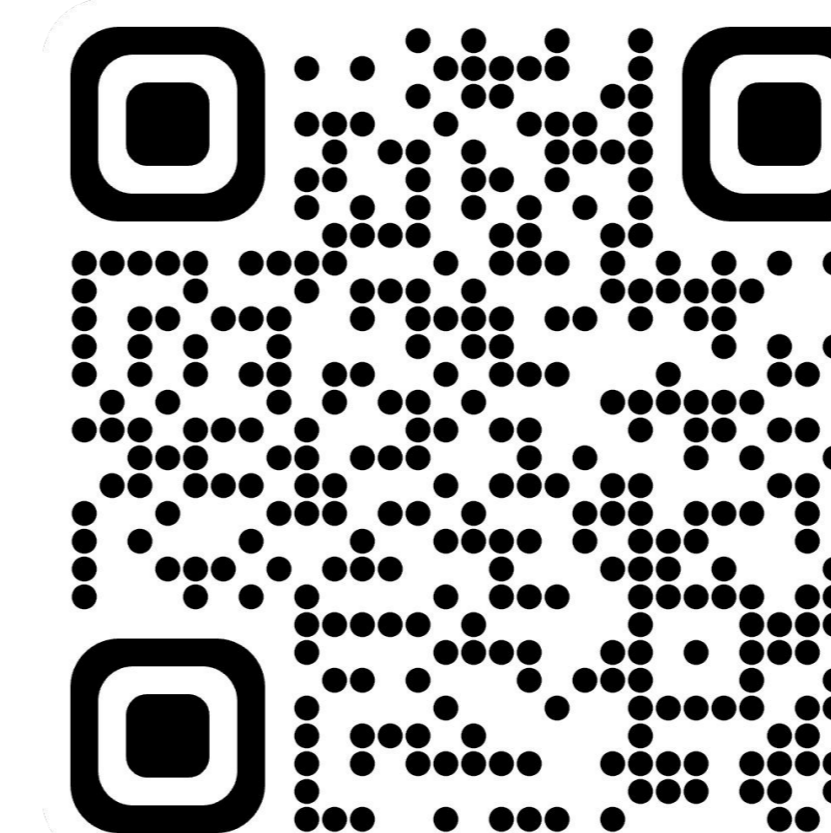


Models	Size	Type	SR (Micro-averaged across tasks)					Improvement Rate	
			k=1	k=2	k=3	k=4	k=5	Slope	R <sup>2</sup>
<b>Open-source LLM</b>									
CodeLLaMA	7B	Base*	0.3	4.1	7.2	7.2	4.3	+1.1	0.38
		SIFT	0.3	7.8	10.2	9.7	8.7	+1.9	0.53
	13B	Base	0.5	13.7	17.9	19.3	18.4	+4.1	0.70
		SIFT*	1.5	12.6	13.1	15.0	14.5	+2.8	0.64
	34B	Base	0.2	16.2	23.0	25.9	28.2	+6.6	0.85
		SIFT*†	2.6	10.1	14.7	15.4	17.1	+3.4	0.86
LLaMA-2	7B	Base	0.2	5.6	7.3	8.9	9.7	+2.2	0.87
		RLHF*	1.0	4.3	6.7	6.5	7.3	+1.5	0.83
	13B	Base	0.2	11.4	15.5	15.2	14.5	+3.2	0.63
		RLHF	4.1	12.5	12.5	13.3	11.9	+1.7	0.47
	70B	Base	1.9	19.4	24.6	26.4	26.4	+5.6	0.73
		RLHF	4.3	14.3	15.7	16.6	17.9	+3.0	0.73
Lemur-v1	70B	Base	1.0	17.9	23.6	25.3	26.3	+5.8	0.77
		SIFT	3.8	27.0	35.7	37.5	37.0	+7.7	0.73
Vicuna-v1.5	7B	SIFT†	0.0	6.7	12.3	15.4	12.6	+3.4	0.77
		SIFT†	0.0	2.2	4.4	6.7	8.4	+2.1	1.00
<b>Closed-source LLM</b>									
chat-bison-001	-	-*	0.3	15.9	14.2	13.0	14.5	+2.5	0.40
claude-2	-	-	26.4	35.5	36.0	39.8	39.9	+3.1	0.81
claude-instant-1	-	-	12.1	32.2	39.2	44.4	45.9	+8.0	0.84
gpt-3.5-turbo-0613	-	-	2.7	16.9	24.1	31.7	36.2	+8.2	0.96
gpt-4-0613	-	-	-	-	-	-	69.5	-	-

\* Evaluated LLM failed to produce parsable output as instructed in some cases. See §3.5 and Tab. A.7 for details.  
† We identified potential undesired artifacts in its training data, which hurt its performance. See §3.5 for details.



Website

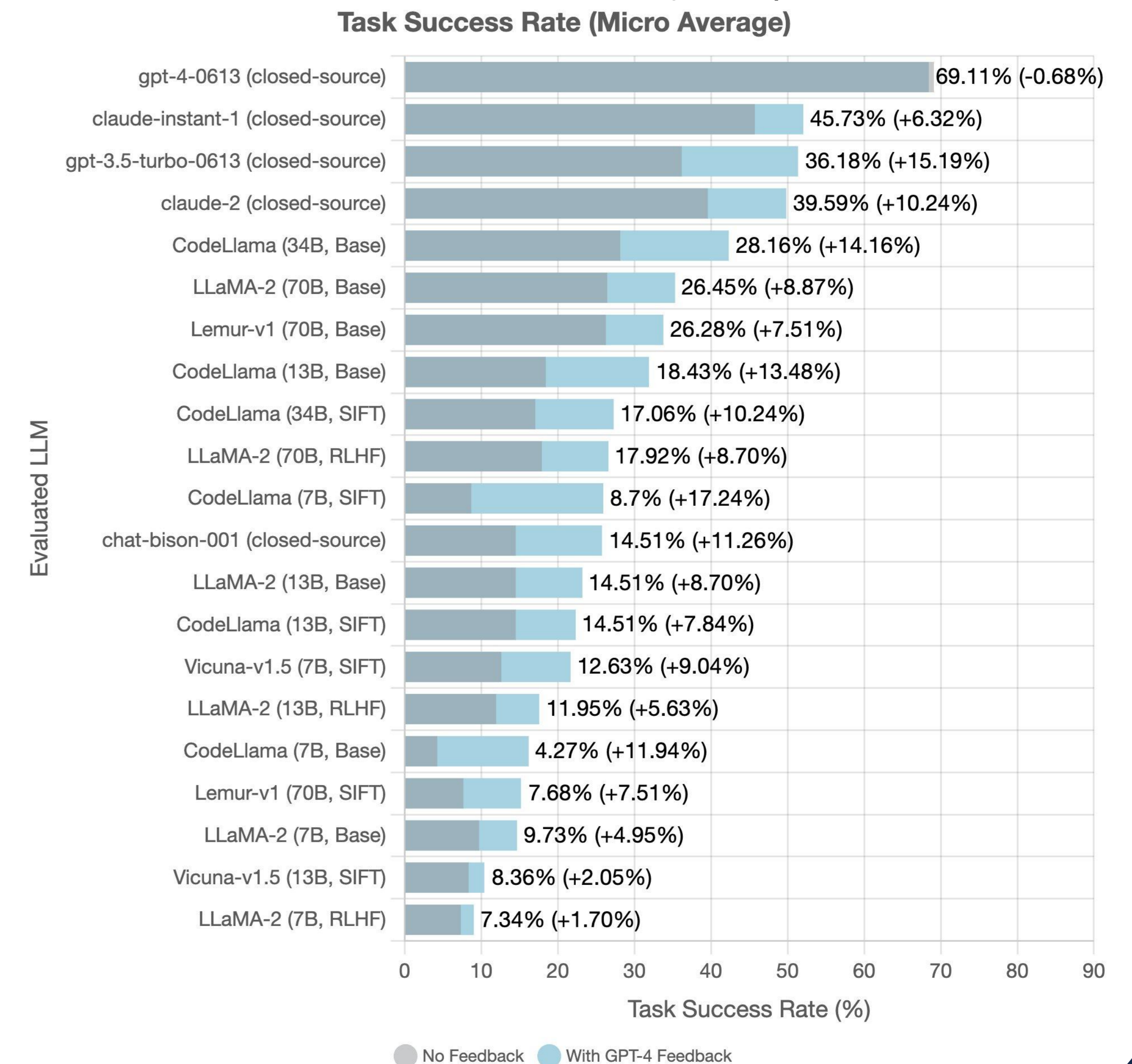


Tweet

## Evaluating LLM with Language Feedback

We use gpt-4-0613 to simulate user feedback:

- **No significant difference between open- and closed-source models in terms of  $\Delta$ feedback**
- **SIFT and RLHF may hurt models' ability to leverage feedback** (CodeLLaMA & LLaMA-2, except Vicuna & Lemur)



## Evaluating LLM as Feedback Provider

Task-solving ability could be orthogonal to feedback-providing ability.

- GPT-3.5 excelled in task-solving but struggled with self-feedback.
- CodeLLaMA-34B-Instruct, despite performing the poorest (-19% difference vs. GPT-3.5), can still provide feedback that improves the stronger GPT-3.5.

Different Feedback Providers' Ability to Improve GPT-3.5's Performance

