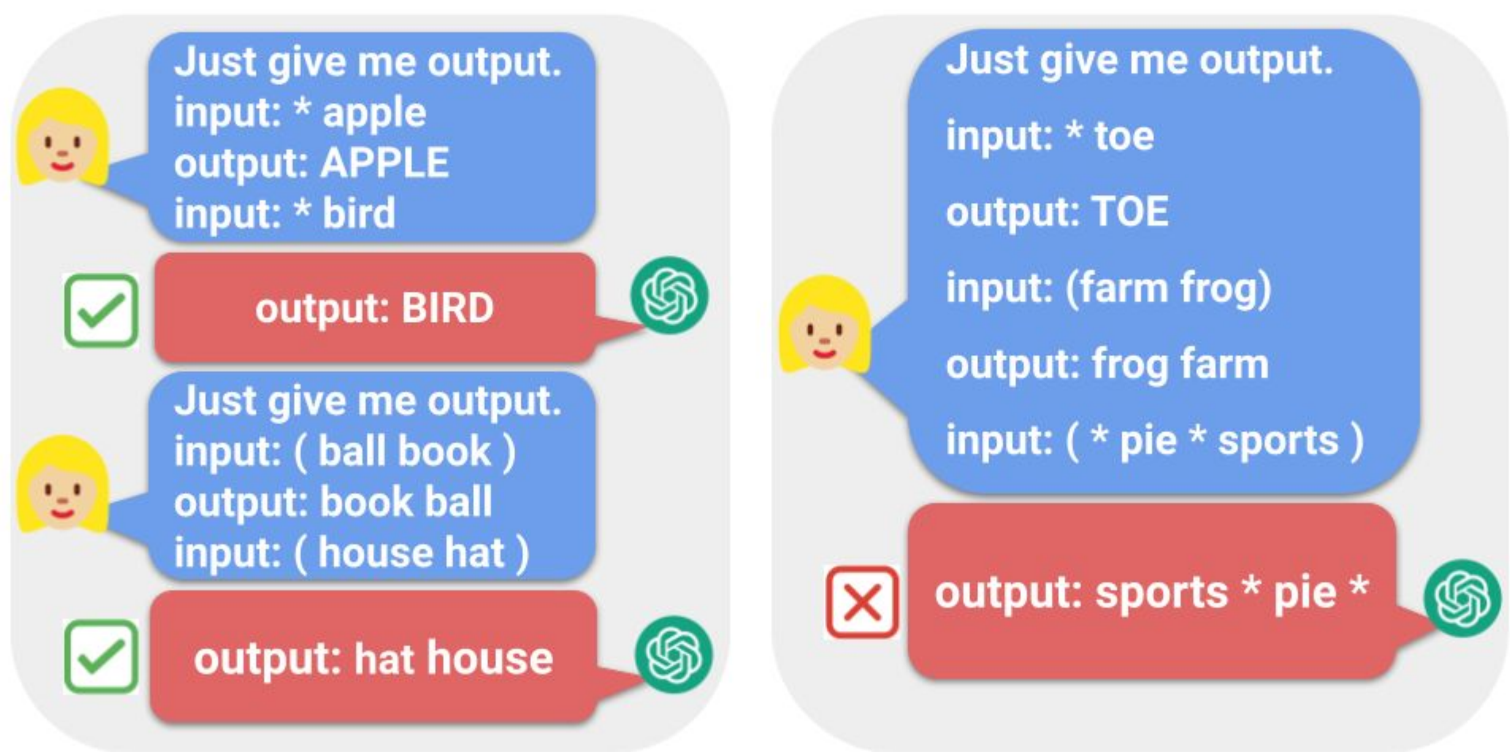
Large Language Models Have Compositional Ability? An Investigation into Limitations and Scalability

Zhuoyan Xu*, Zhenmei Shi*, Yingyu Liang

WISCONSIN UNIVERSITY OF WISCONSIN-MADISON

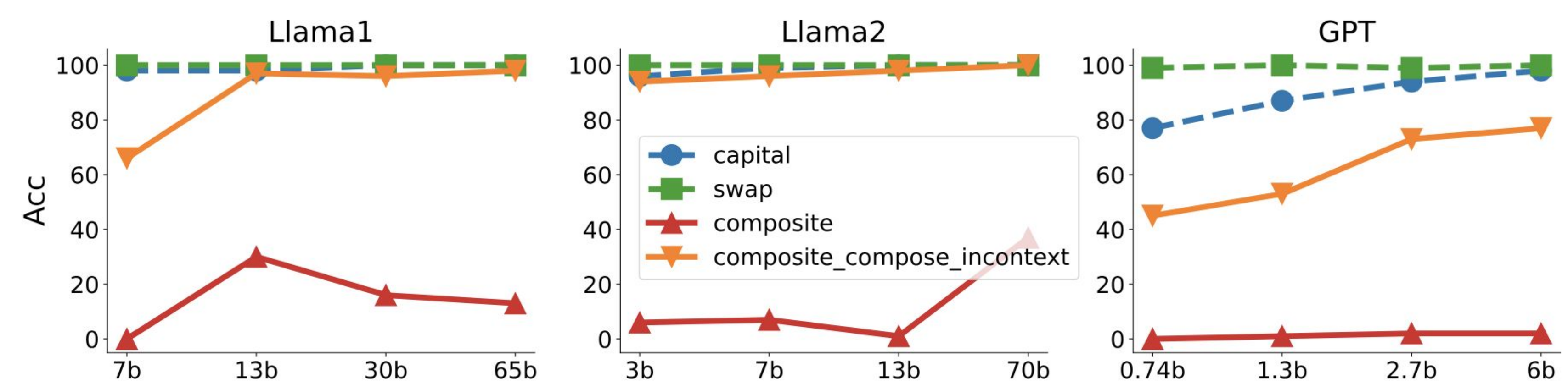
Motivation



Inconsistent performance in GPT-4. Consider 2 simple tasks: If a word is followed by an asterisk (*), capitalize the letter. If two words are surrounded by parenthesis, swap the positions. GPT-4 correctly solves two simple tasks based on demonstrations (left). The composite tasks have test input with both asterisk (*) and parenthesis. The correct answer should be *output: SPORTS PIE*. However, GPT-4 fails to solve composite tasks (right). The same failure was observed in Claude 3.

	Composite	Composite in-context
Prompt	*input: * apple* *output: APPLE* *input: (farm frog)* *output: frog farm* *input: (* bell * ford)*	*input: (* good * zebra)* *output: ZEBRA GOOD* *input: (* bicycle * add)*
Truth	*output: FORD BELL*	*output: ADD BICYCLE*

Table 1: Examples of two settings on composite tasks. Composite: in-context examples are about simple tasks while the test input is about the composite task. Composite in-context: both in-context examples and the test input are about the composite task.



The exact match accuracy (y-axis) vs the model scale (x-axis, "b" stands for billion) for (T1) Capitalization & Swap tasks (example in Figure 1). Line ***capital***: performance on the simple task of capitalization; ***swap***: on the simple task of swap; ***composite***: in-context examples are from simple tasks while test input from the composite task. ***composite incontext***: in-context examples and test input are all from the composite task (example in Table 1).

Take-Home Message

In this study, we delve into the ICL capabilities of LLMs on composite tasks, with only simple tasks as in-context examples. We develop a test suite of composite tasks that include logical challenges and perform empirical studies across different LLM families.

Key Intuition

- For simpler composite tasks that apply distinct mapping mechanisms to ***different input segments***, the models demonstrate decent compositional ability, while scaling up the model enhances this ability.
- For more complex composite tasks that involving ***reasoning multiple steps***, which each step represent one task, models typically underperform, and scaling up does not generally lead to improvements.

Compositional Logical Tasks

Tasks	Simple Task	Simple Task	Composite
(A) + (B)	input: * apple output: APPLE	input: (farm frog) output: frog farm	input: (* bell * ford) output: FORD BELL
(A) + (C)	input: * (*five*) output: FIVE	input: *twenty @ eleven* output: thirty-one	input: * (*thirty-seven @ sixteen*) output: FIFTY-THREE
(G) + (H)	input: 15 @ 6 output: 3	input: 12 # 5 output: 18	input: 8 # 9 @ 7 Ouput: 4
(A) + (F)	input: 435 output: 436	input: cow output: COW	input: 684 cat output: 685 CAT

Examples of the four logical composite tasks. Note that in (G) + (H), the output of the composite task can be either 4 or 11 depending on the order of operations and we denote both as correct.

Simple logical Tasks

Tasks	Task	Input	Output
Words	(A) Capitalization	apple	APPLE
	(B) Swap	bell ford	ford bell
	(C) Two Sum	twenty @ eleven	thirty-one
	(D) Past Tense	pay	paid
	(E) Opposite	Above	Below
Numerical	(F) Plus One	435	436
	(G) Modular	15 @ 6	3
	(H) Two Sum Plus One	12 # 5	18

This table contains a collection of simple logical tasks. The ***Words*** category encompasses tasks that modify words at the character or structural level. In contrast, the ***Numerical*** category is devoted to tasks that involve arithmetic computations performed on numbers.

Composite tasks results

		Mistral		Llama2			Llama1			
	Tasks	7B	8x7B	7B	13B	70B	7B	13B	30B	65B
(A) + (B)	Capitalization	99	98	99	100	100	98	98	100	100
	swap	100	100	100	100	100	100	100	100	100
	Compose	16	42	7	1	37	0	30	16	13
	Com. in-context	95	96	96	98	100	66	97	96	98
(A) + (C)	twoSum	71	100	72	93	99	62	56	98	99
	Capitalization	98	99	100	95	99	97	98	99	99
	Compose	8	19	3	23	44	3	3	31	2
	Com. in-context	31	65	52	77	100	9	22	93	69
(A) + (F)	Capitalization	97	99	98	77	99	84	96	99	98
	PlusOne	100	99	100	100	100	100	100	100	100
	Compose	92	96	74	69	97	57	60	69	99
	Com. in-context	99	98	99	100	100	99	99	100	100
(B) + (D)	Swap	100	100	100	100	100	100	100	100	100
	Past Tense	97	99	97	100	99	97	98	100	100
	Compose	6	12	0	1	62	57	34	46	5
	Com. in-context	92	98	86	95	98	86	95	89	94
(B) + (E)	Swap	100	100	100	100	100	100	100	100	100
	Opposite	61	62	58	68	65	51	58	64	63
	Compose	0	0	0	0	0	0	0	0	0
	Com. in-context	35	32	12	37	37	0	9	7	9
(D) + (F)	Past Tense	100	100	98	100	100	100	100	100	100
	Plus One	100	100	100	100	100	99	100	100	100
	Compose	71	46	32	80	80	40	44	14	74
	Com. in-context	98	100	98	99	100	95	96	98	100
(G) + (H)	Modular	25	22	5	23	43	9	16	29	29
	twoSumPlus	38	42	3	77	90	14	10	40	87
	Compose	4	5	0	1	1	0	0	0	5
	Com. in-context	4	8	13	13	12	11	13	7	12

Results evaluating composite tasks on various models. The accuracy are showed in %.

Theoretical Analysis

In-context Learning

Embedding matrix: $E := \begin{pmatrix} x_1 & x_2 & \dots & x_N & x_q \\ y_1 & y_2 & \dots & y_N & 0 \end{pmatrix}$

Linear self-attention with parameter matrix $\theta = (W^{PV}, W^{KQ})$

$$f_{\text{LSA},\theta}(E) = E + W^{PV} E \cdot \frac{E^\top W^{KQ} E}{N}$$

Data distribution

- input features: $x \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Lambda), \Lambda \in \mathbb{R}^{d\times d}$
- $y = Wx, W \in \mathbb{R}^{K\times d}$
- $W = [w^{(1)}, w^{(1)}, \cdots, w^{(K)}]^\top, w^{(k)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_d)$

Definition1 (Compositional Ability)

Consider a composite tasks combines two simple tasks (A) and (B). Consider each simple tasks contains samples with $\{x_i, y_i\}$. Given a composite test prompt x_q, we say model has compositional ability on composite task (A) + (B) if model has higher accuracy using in-context examples from both (A) and (B) than from either single one.

Theorem (Compositional ability under confined support (Informal))

Consider input embedding $x \in \mathbb{R}^d$ of each simple tasks. Consider each simple has a disjoint subset of indices from $1, 2, \dots, d$. Each simple task only has large values within its corresponding subsets of dimensions of input embeddings. Then with high probability, the model has the compositional ability.