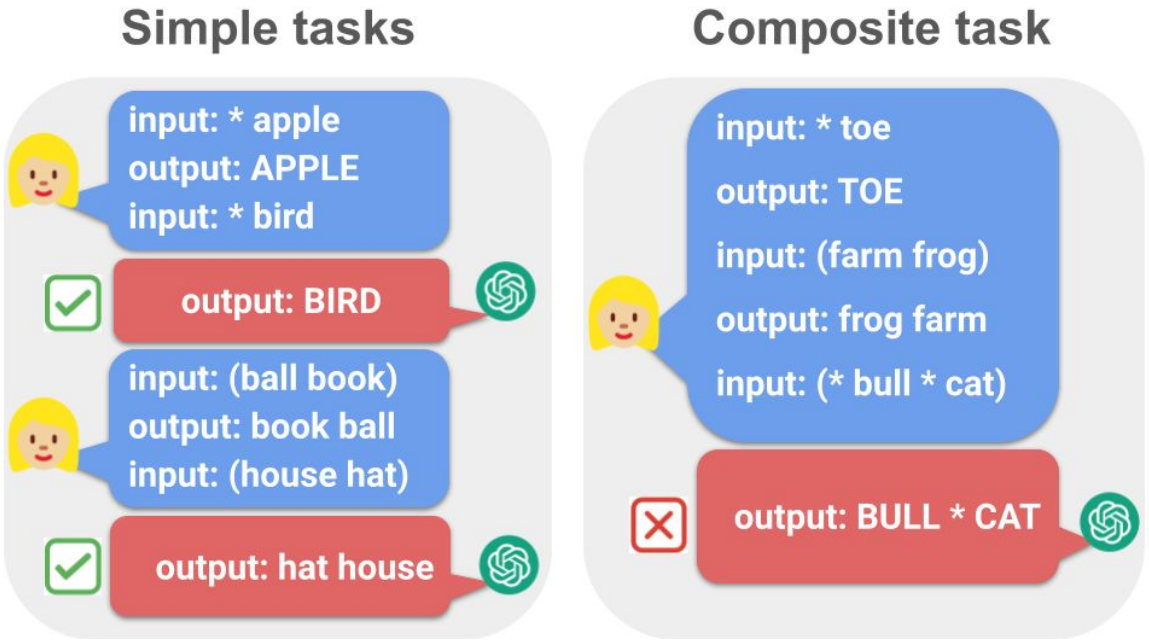
Large Language Models in Compositional Reasoning

Compositional Ability



Inconsistent performance in GPT-4. 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: CAT BULL.*** However, GPT-4 fails to solve composite tasks (right).}

Composite tasks: swap + capitalization

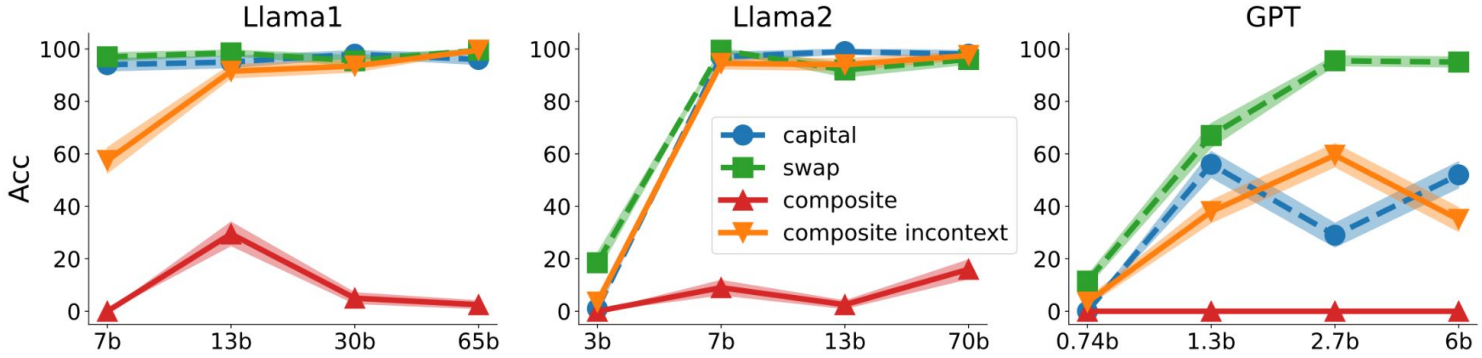


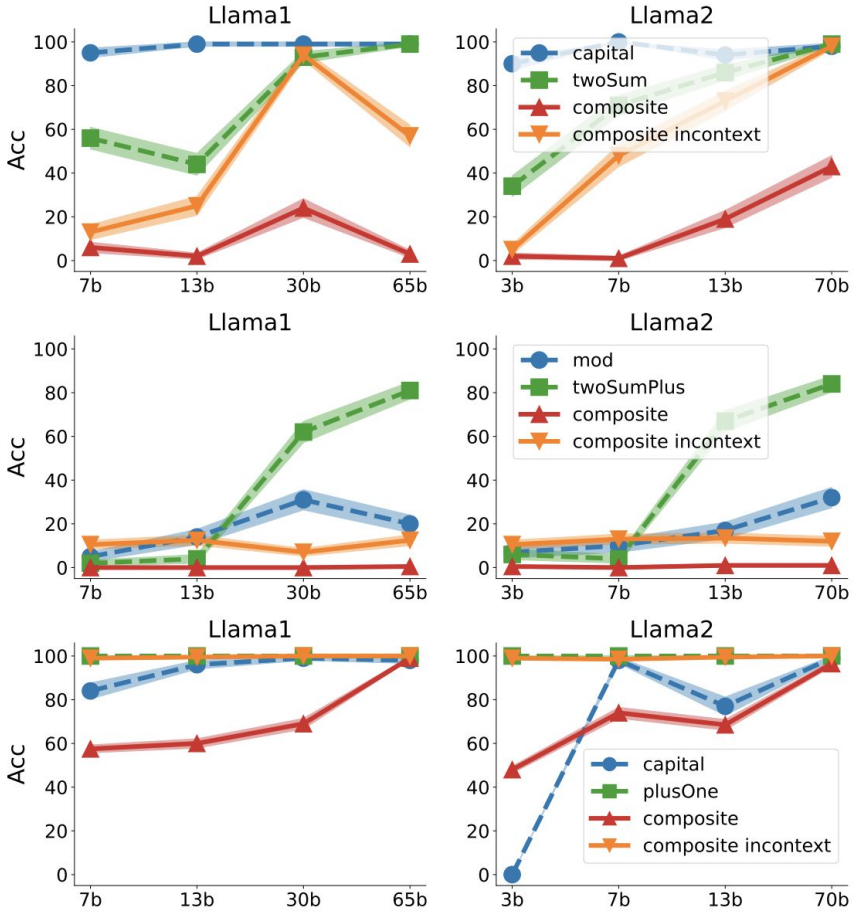
Figure 2. The exact match accuracy (y-axis) vs the model scale (x-axis, "b" stands for billion) for **(T1)** Capitalization & Swap tasks (example in Figure 1). Line *capital*: performance on the simple task of capitalization; *swap*: on the simple task of swap; *composite*: in-context examples are from 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).

	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: (* model * math)* *output: MATH MODEL* *input: (* bicycle * add)*
Truth	*output: FORD BELL*	*output: ADD BICYCLE*

Logic reasoning composite tasks

Tasks	Simple Task	Simple Task	Composite
(T1)	input: * apple output: APPLE	input: (farm frog) output: frog farm	input: * (bell ford) output: FORD BELL
(T2)	input: * *(five)* output: FIVE	input: *twenty* @ *eleven* . output: thirty-one	input: * *(thirty-seven* @ *sixteen)* . output: FIFTY-THREE
(T3)	input: 15 @ 6 output: 3	input: 12 # 5 output: 18	input: 8 # 9 @ 7 Ouput: 4
(T4)	input: 435 output: 436	input: cow output: COW	input: 684 cat output: 685 CAT

(T1) Capitalization & Swap. (T2) Capitalization & Two Sum. (T3) Modular & Two Sum Plus.(T4) Capitalization & Plus One.



(T2) Capitalization & Two Sum. (T3) Modular & Two Sum Plus.(T4) Capitalization & Plus One.

Composite Linguistic Translation

Input: The princess teleported a cookie to the goose .

Output: TELEPORT (PRINCESS , COOKIE , GOOSE)

Input: A cake was forwarded to Levi by Charlotte .

Output: FORWARD (CHARLOTTE , CAKE , LEVI)

Composite Linguistic Translation

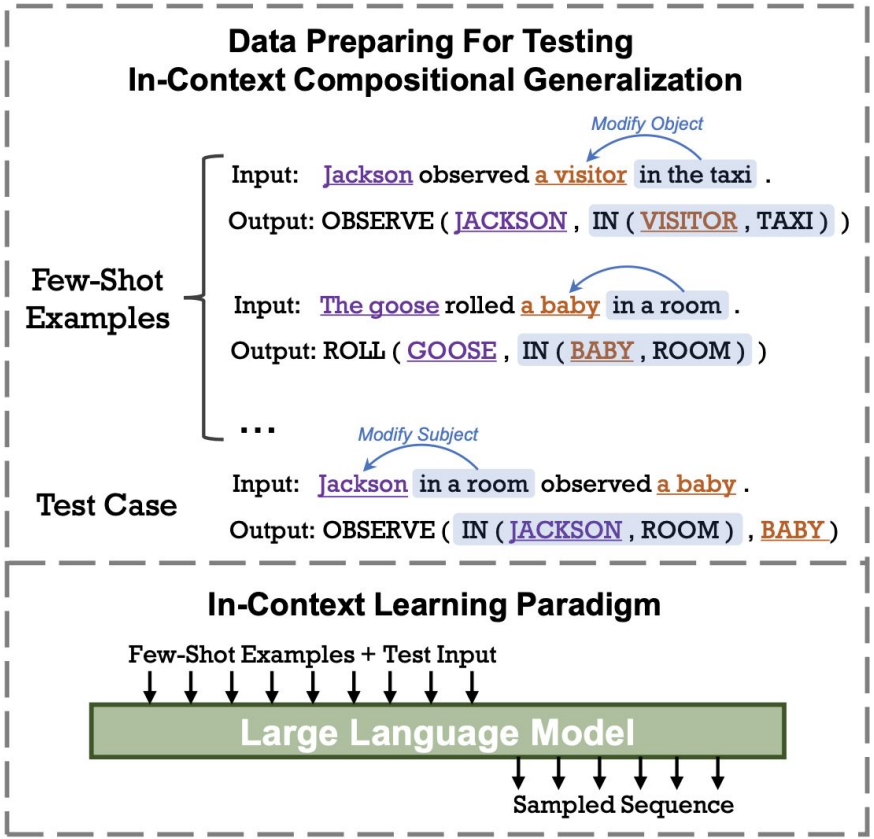


Figure from: An et al. How Do In-Context Examples Affect Compositional Generalization? 2023.

Phrase Recombination + Longer Chain

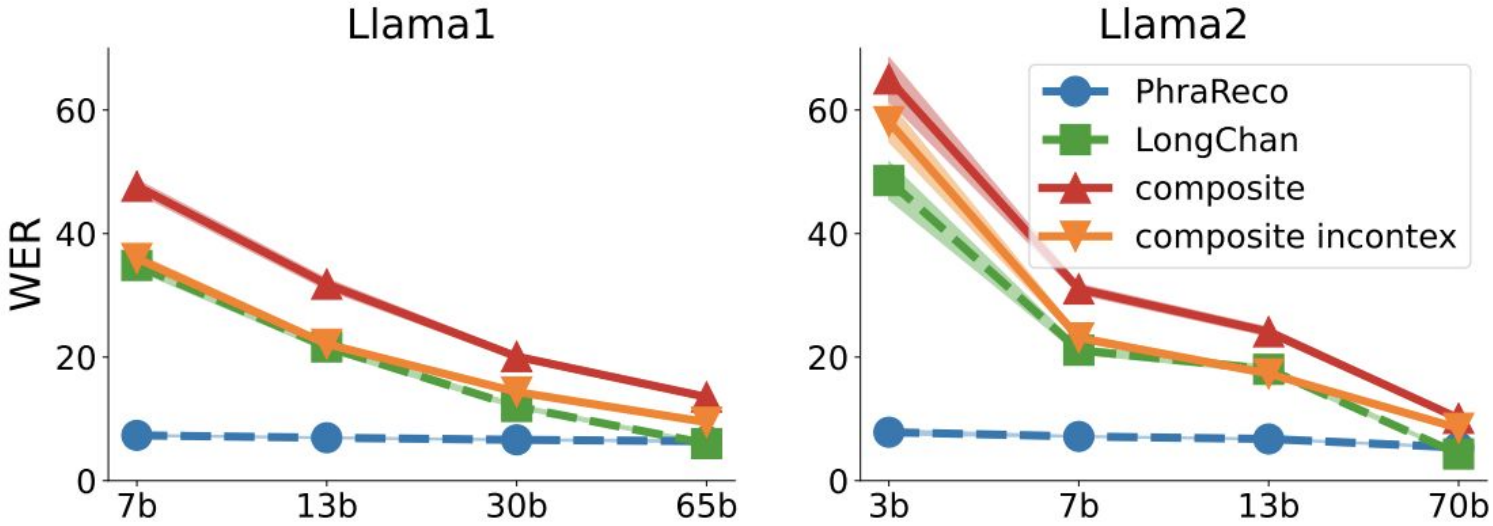
Phrase Recombination	input: Logan mailed Stella **the cake in the pile** . output: MAIL (LOGAN , IN (CAKE , PILE) , STELLA) input: The goose rolled **a baby in a room** . output: ROLL (GOOSE , IN (BABY , ROOM) , NONE)	input: **A visitor in the pile** rolled a resident . output: ROLL (IN (VISITOR , PILE) , RESIDENT , NONE)
Longer Chain	input: The boy admired that Noah confessed that \ Emma was given a cookie . output: ADMIRE (BOY , NONE , NONE) \ **CCOMP** CONFESS (NOAH , NONE , NONE) \ **CCOMP** GIVE (NONE , COOKIE , EMMA)	input: The girl wished that a crocodile declared that \ the boy admired that Emma liked that \ Evelyn was passed a drink . output: WISH (GIRL , NONE , NONE) \ **CCOMP** DECLARE (CROCODILE , NONE , NONE) \ **CCOMP** ADMIRE (BOY , NONE , NONE) \ **CCOMP** LIKE (EMMA , NONE , NONE) \ **CCOMP** PASS (NONE , DRINK , EVELYN)

Figure from: An et al. How Do In-Context Examples Affect Compositional Generalization? 2023.

Phrase Recombination + Longer Chain

Task		Example
Phrase Recombination	Input	The baby on a tray in the house screamed .
	Output	SCREAM (ON (BABY , IN (TRAY , HOUSE)) , NONE , NONE)
Longer Chain	Input	A girl valued that Samuel admired that a monkey liked that Luna liked that Oliver respected that Savannah hoped that a penguin noticed that Emma noticed that the lawyer noticed that a cake grew .
	Output	VALUE (GIRL , NONE , NONE) \ CCOMP ADMIRE (SAMUEL , NONE , NONE) \ CCOMP LIKE (MONKEY , NONE , NONE) \ CCOMP LIKE (LUNA , NONE , NONE) \ CCOMP RESPECT (OLIVER , NONE , NONE) \ CCOMP HOPE (SAVANNAH , NONE , NONE) \ CCOMP NOTICE (PENGUIN , NONE , NONE) \ CCOMP NOTICE (EMMA , NONE , NONE) \ CCOMP NOTICE (LAWYER , NONE , NONE) \ CCOMP GROW (NONE , CAKE , NONE)
Composite Task	Input	The baby on a tray in the house valued that Samuel admired that a monkey liked that Luna liked that Oliver respected that Savannah hoped that a penguin noticed that Emma noticed that the lawyer noticed that a cake grew .
	Output	VALUE (ON (BABY , IN (TRAY , HOUSE) , NONE , NONE) \ CCOMP ADMIRE (SAMUEL , NONE , NONE) \ CCOMP LIKE (MONKEY , NONE , NONE) \ CCOMP LIKE (LUNA , NONE , NONE) \ CCOMP RESPECT (OLIVER , NONE , NONE) \ CCOMP HOPE (SAVANNAH , NONE , NONE) \ CCOMP NOTICE (PENGUIN , NONE , NONE) \ CCOMP NOTICE (EMMA , NONE , NONE) \ CCOMP NOTICE (LAWYER , NONE , NONE) \ CCOMP GROW (NONE , CAKE , NONE)

Phrase Recombination + Longer Chain



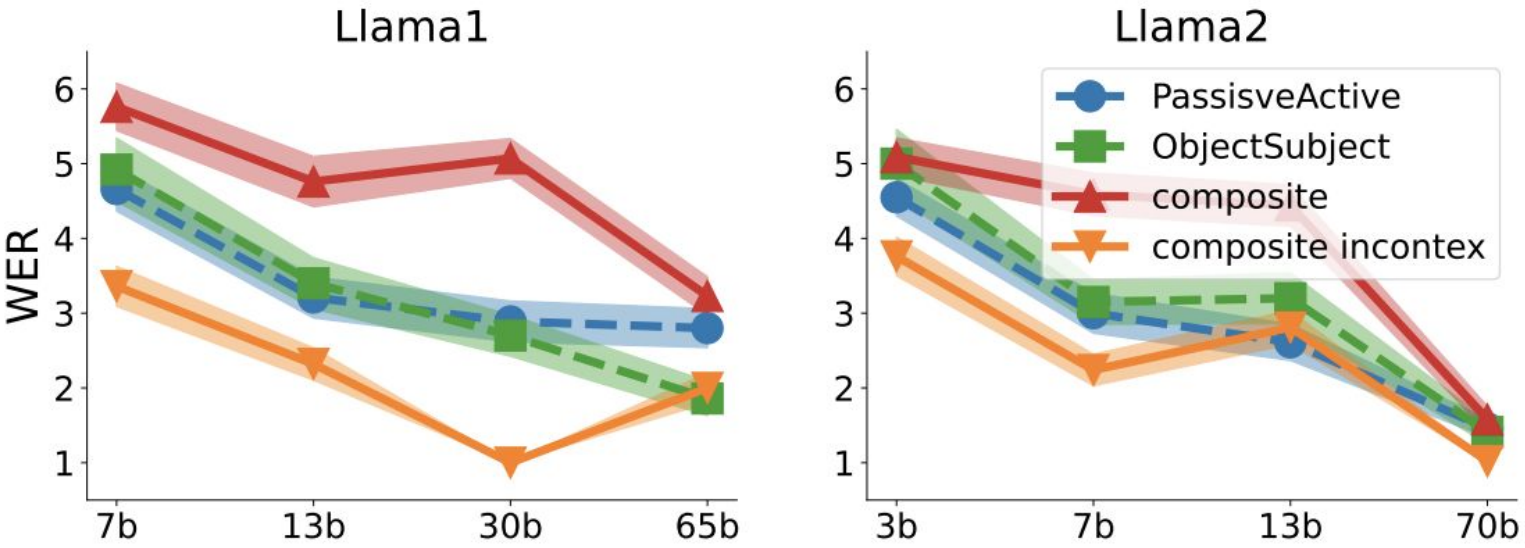
The word error rate (WER) vs the model scale on composite linguistic translation tasks. Dashed lines: simple tasks. Solid lines: composite tasks

Passive to Active + Object to Subject

Task	In-context Example	Testing Example
Passive to Active	The book was squeezed . SQUEEZE (NONE , BOOK , NONE)	Sophia squeezed the donut . SQUEEZE (SOPHIA , DONUT , NONE)
Object to Subject	Henry liked a cockroach in a box . LIKE (HENRY , IN (COCKROACH , BOX)	A cockroach inflated a boy . INFLATE (COCKROACH , BOY , NONE)
Composite Task	The book was squeezed . SQUEEZE (NONE , BOOK , NONE) Henry liked a cockroach in a box . LIKE (HENRY , IN (COCKROACH , BOX)	A cockroach squeezed the hedgehog . SQUEEZE (COCKROACH , hedgehog , NONE)

Testing examples of Passive to Active and Object to Subject, red text shows the verbs changing from passive to active voice in simple tasks, and blue text shows the nouns from objective to subjective.

Passive to Active + Object to Subject



The word error rate (WER) vs the model scale on composite linguistic translation tasks. Dashed lines: simple tasks. Solid lines: composite tasks

Conjecture behind the experiments

- If composite tasks contain simple tasks related to different parts or perspectives of the input, the model will tackle the composite tasks well.

- One natural explanation is the model processes the input in some hidden embedding space, and decomposes the embedding of the input into different “regions”:
  - word-level modifications
  - arithmetic calculations
  - linguistic acceptability, etc.

- If the two simple tasks correspond to two different task types where they relates to separate regions of the embedding, the model can effectively manage the composite task by addressing each simple task operation within its corresponding region.

Theoretical Analysis

Data. Assume $x \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Lambda)$, where $\Lambda \in \mathbb{R}^{d \times d}$ is the covariance matrix. Assume $y = Wx$, where $W \in \mathbb{R}^{K \times d}$. Then for any simple task $k \in [K]$, its label is the k-th entry of y, which is $y^{(k)} = \left\langle w^{(k)}, x \right\rangle$, where $w^{(k)}$ is the k-th row of W. We also assume each task weight $w^{(k)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_d)$.

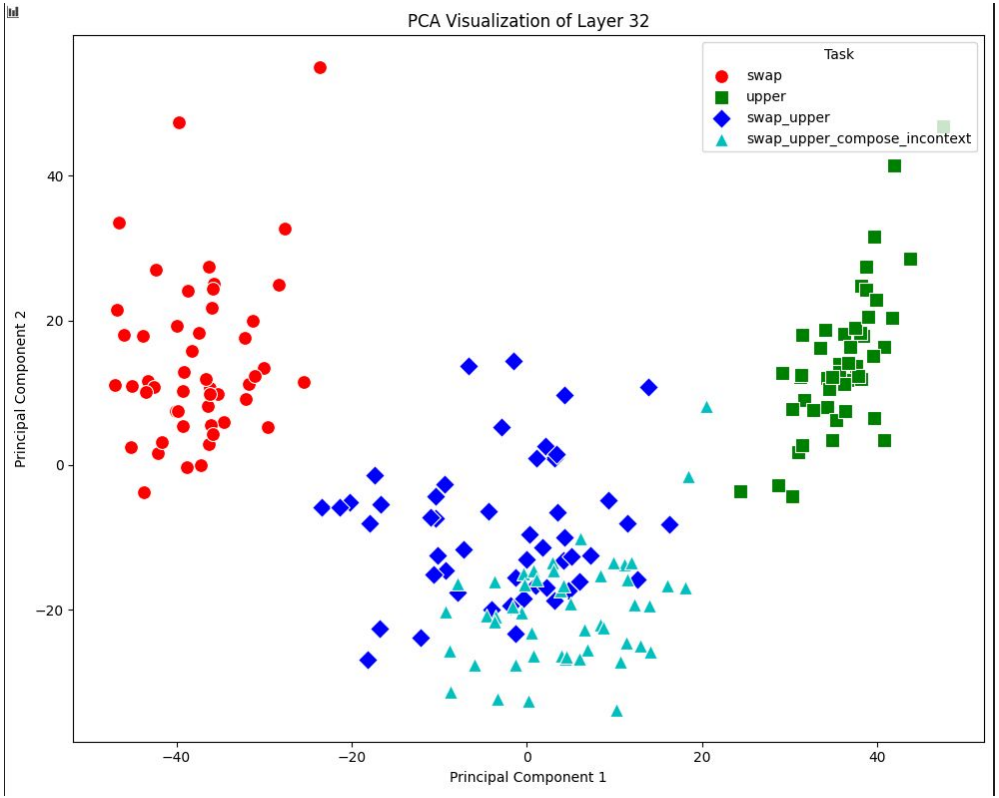
$$f_{\text{LSA},\theta}(E) = E + W^{PV} E \cdot \frac{E^\top W^{KQ} E}{N}.$$

$$E := \begin{pmatrix} x_1 & x_2 & \dots & x_N & x_q \\ y_1 & y_2 & \dots & y_N & 0 \end{pmatrix} \in \mathbb{R}^{d_e \times (N+1)}$$

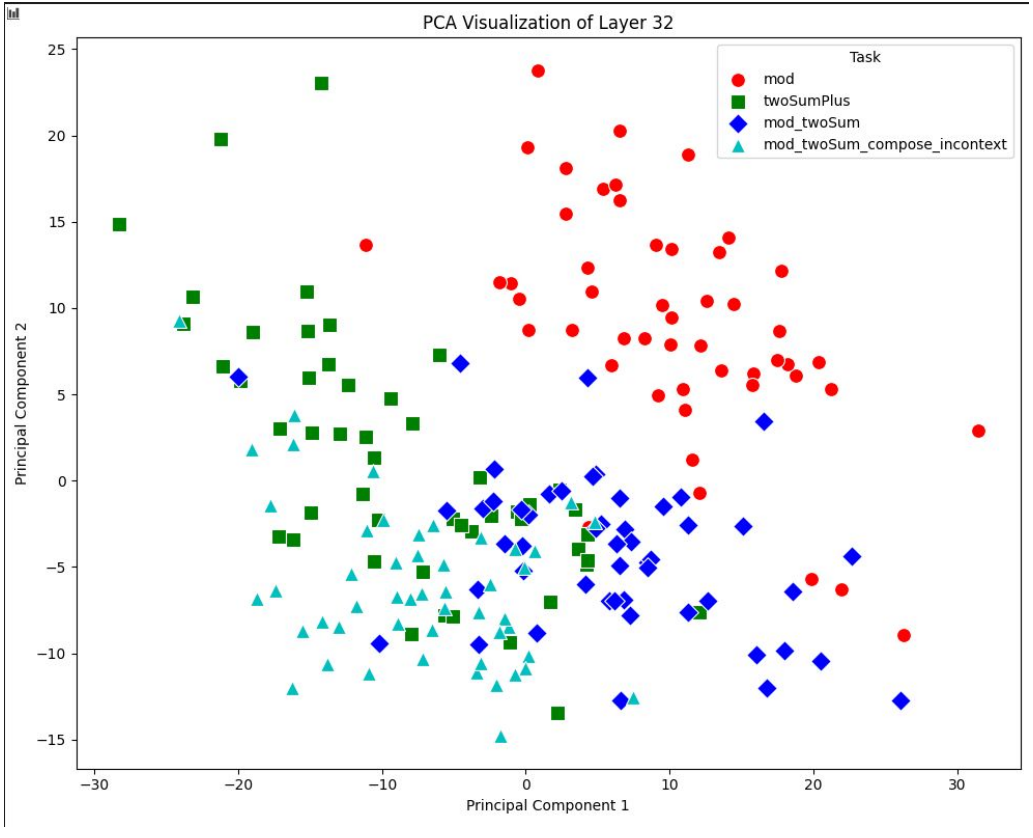
We define compositional ability as:
Given a test input from composite tasks (k+g), if model only given simple examples from simple tasks k or g as in-context demonstrations, the model have average performance. If given examples from both k and g, the mode have better performance.

Theorem 1 (Compositional Ability (Informal))
Consider distinct tasks *k* and *g* with corresponding examples *Sk*, *Sg*, and Sk ∪ g = Sk ∪ Sg, If the input embedding x of examples from simple task k and g have support (non-zeros values) on disjoint regions. The model on composite task k and g have the compositional ability.

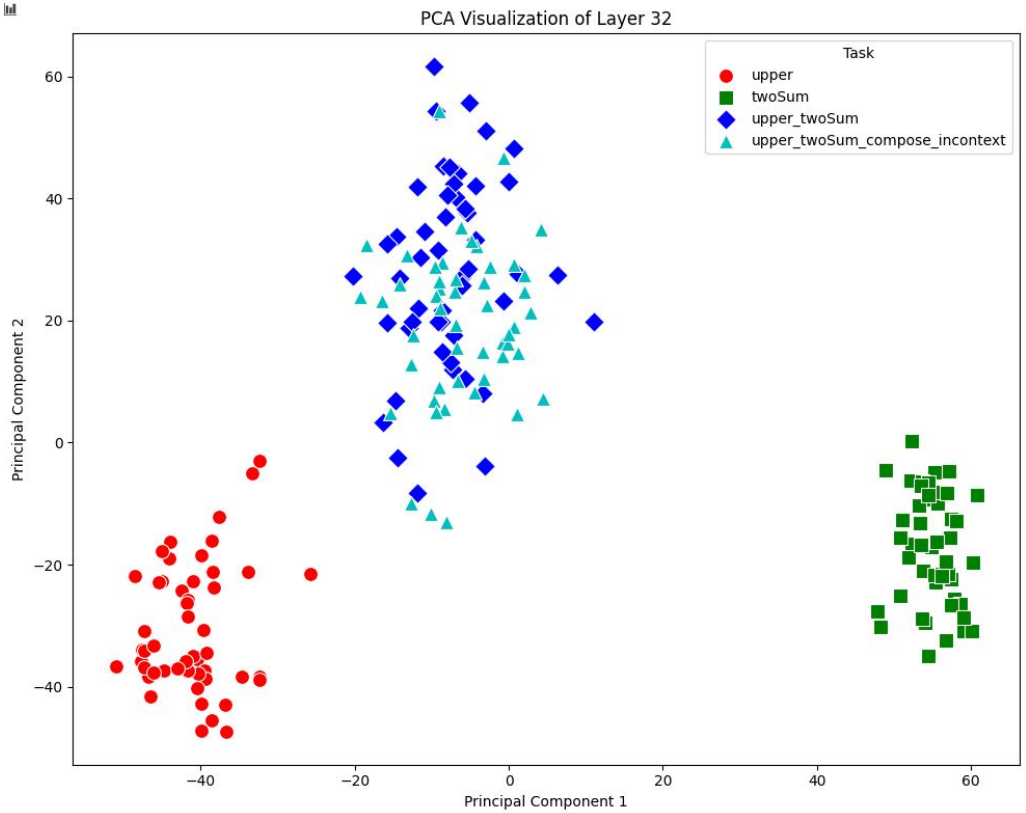
Swap capital



mod_twoSumPlus



upper_twoSum



plusOne_upper