Paper Poster

It Ain't That Bad:

Understanding the Mysterious Performance Drop in OOD Generalization for Generative Transformer Models

Xingcheng Xu, Zihao Pan

Haipeng Zhang, Yanqing Yang

Shanghai AI Lab, ShanghaiTech U, Fudan U

IJCAI-24, Jeju

August 7, 2024

OUTLINE

01

Background and Motivation

Background and Motivation

- Large language models (LLMs) have exhibited remarkable advancements across diverse domains. These models have demonstrated exceptional versatility, tackling tasks ranging from natural language to code and mathematical reasoning.
- However, despite these accomplishments, the generalization ability of LLMs is not fully understood.
- The black-box nature of these models has led researchers to *explore basic mathematical tasks* as a means to gain insights into their generalization behaviors.
  - For example, use n-digit addition (123+456 for $n = 3$) or multiplication to train a model, and test on inputs with length no more than n as *in-distribution (ID) test* such as 378+12 or 12+78, test on inputs with length greater than n as *out-of-distribution (OOD) test* such as 9123+8456.

Performance of LLMs on Arithmetic Taks

Performance of arithmetic tasks among different prominent large language models (LLMs) including GPT-4, ChatGPT, GPT-3.5, Galactica, LLaMA, OPT, BLOOM, and GLM.

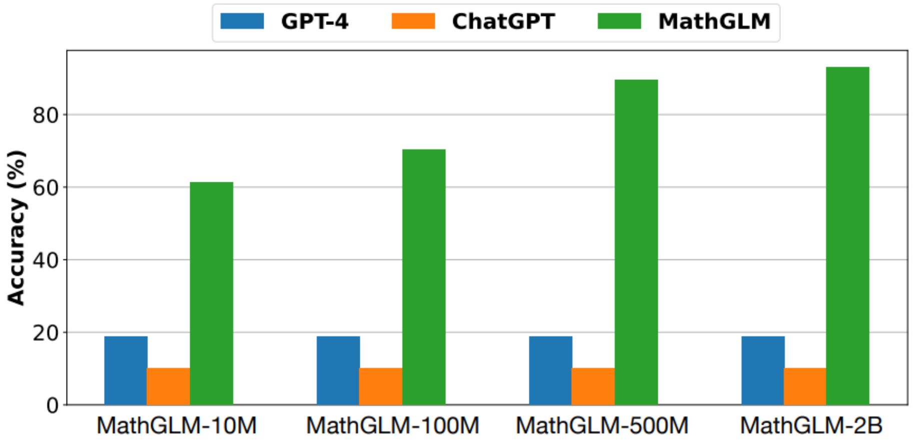


Figure 1: Accuracy scores across various LLMs like GPT-4 and ChatGPT, as well as a series of MathGLM models on the generated test dataset for the arithmetic tasks. Among the different model scales, MathGLM consistently achieves superior performance.

Model	ACC	RE
GPT-4	22.22%	-
ChatGPT	13.25%	-
text-davinci-003	9.79%	-
text-davinci-002	4.08%	-
Galactica-120b	7.97%	-
Galactica-30b	7.02%	-
LLaMA-65b	5.02%	-
OPT-175B	3.83%	-
BLOOM-176B	3.96%	-
GLM-130B	3.06%	-
MathGLM-10M	64.29%	97.96%
MathGLM-100M	73.47%	98.23%
MathGLM-500M	89.80%	98.82%
MathGLM-2B	94.90%	98.98%

Overall performance comparison on various LLMs in term of Accuracy.

Source: Yang, Z., Ding, M., Lv, Q., Jiang, Z., He, Z., Guo, Y., ... & Tang, J. (2023). GPT can solve mathematical problems without a calculator. arXiv preprint arXiv:2309.03241.

Our Study

- Observations have revealed a distinct difference between in-distribution (ID) generalization, where models perform well on familiar inputs, and out-of-distribution (OOD) generalization, where they struggle with longer, unseen cases.
- The paper explores this generalization problem in more depth, focusing on the performance drop observed when models are tested on longer, unseen inputs (OOD generalization).
- Explore the generalization gap by investigating the mechanistic perspectives behind these behaviors, using smaller models to uncover insights that could apply to LLMs.

Main Contributions

- **Showcasing the power of mechanistic empirical evaluation for LLM generalization:** We train small generative language models (e.g., NanoGPT, MinGPT) on arithmetic tasks to directly investigate ID vs. OOD generalization.
- **Discovering learned structure for OOD generalization:** The discernible algebraic structure and the equivalence generalization would hopefully guide robust essential solutions for strong OOD generalization.
- **Understanding the role of representations in generalization:** We show that representation learning enables strong ID performance, while unanticipated extension of representations to OOD inputs leads to systematic errors.

02

Experiments and Results

Model Details

We employ the model framework of GPT, a Transformer with a decoder-only architecture comprising multiple layers and multi-head attentions. We train several small-scale models, namely NanoGPT and MinGPT (Karpathy, mingpt), from random initialization using character-level tokenization and the conventional next-token prediction objective.

Hyperparameter	**Addition**	**Multiplication**
num layer	3	6
num head	3	6
dim embd	48	192
vocab size	10	10
context window	15	19
dropout prob	0.1	0.1
optimizer	AdamW	AdamW
learning rate	0.0005	0.0005
betas	(0.9, 0.95)	(0.9, 0.95)
weight decay	0.1	0.1
grad norm clip	1.0	1.0

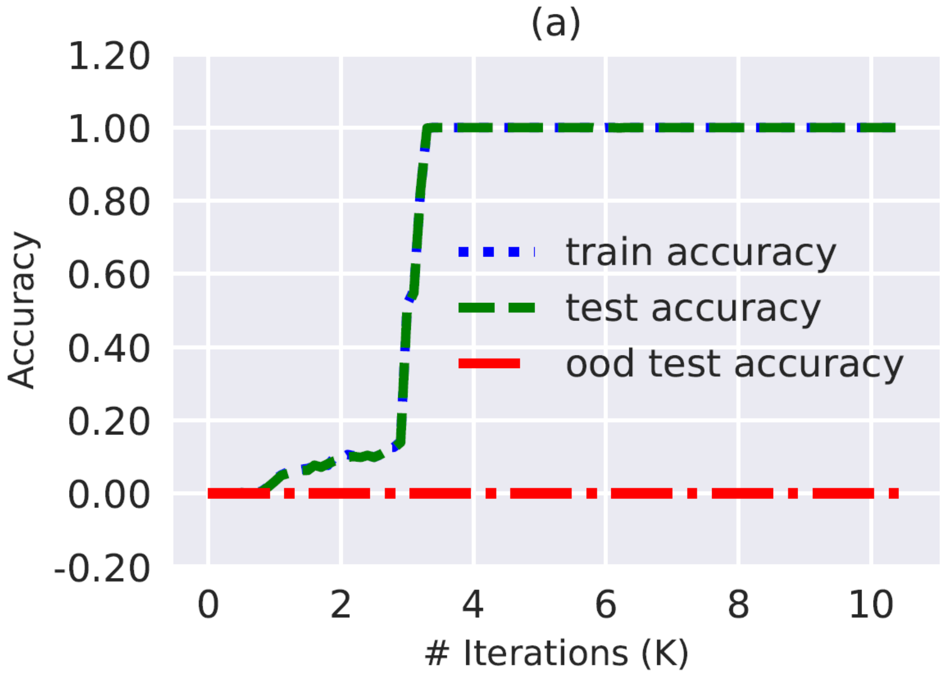
Data Details

The dataset is structured as a concatenation of operand pairs in a natural order, with the reversed order of the operation results and padding before a, b, c.

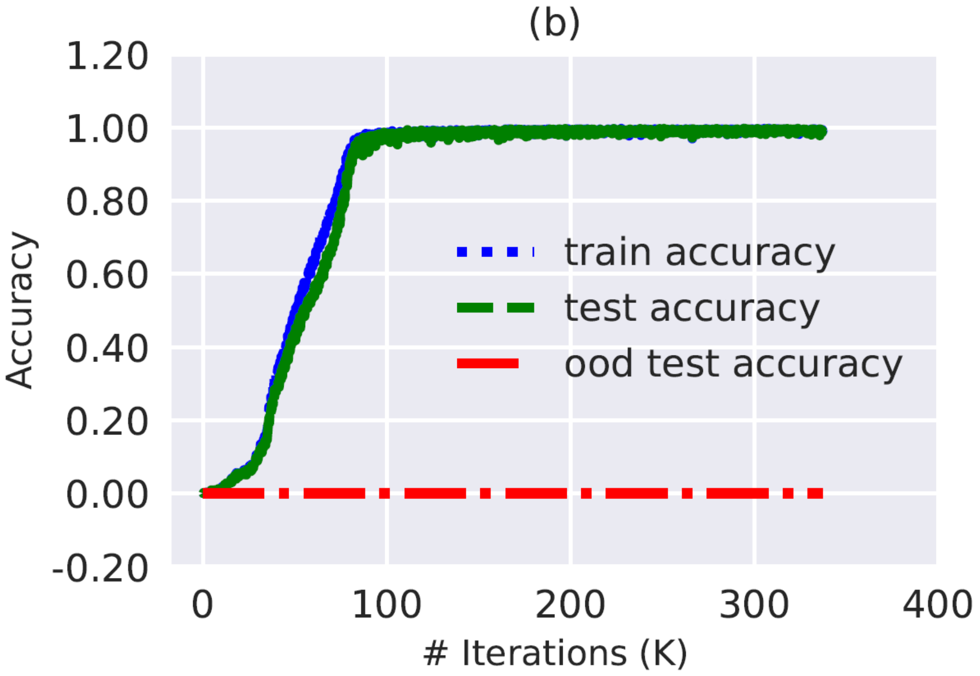
$$a_2a_1a_0 + b_2b_1b_0 = c_0c_1c_2c_3$$

Generalization: Phenomenon

When training on n-digit operations (e.g., addition and multiplication) in which both input operands are n-digit in length, models generalize successfully on unseen n-digit inputs (*in-distribution (ID) generalization*), but fail miserably and mysteriously on longer, unseen cases (*out-of-distribution (OOD) generalization*).



addition



multiplication

Generalization: Phenomenon

Operands	Output Result	Correct Result
$349 + 705$	1,054	1,054
$1{,}349 + 2{,}705$	1,054	4,054
128×256	32,768	32,768
$3{,}128 \times 4{,}256$	32,768	13,312,768

Table 2: Examples on models' outputs for addition and multiplication.

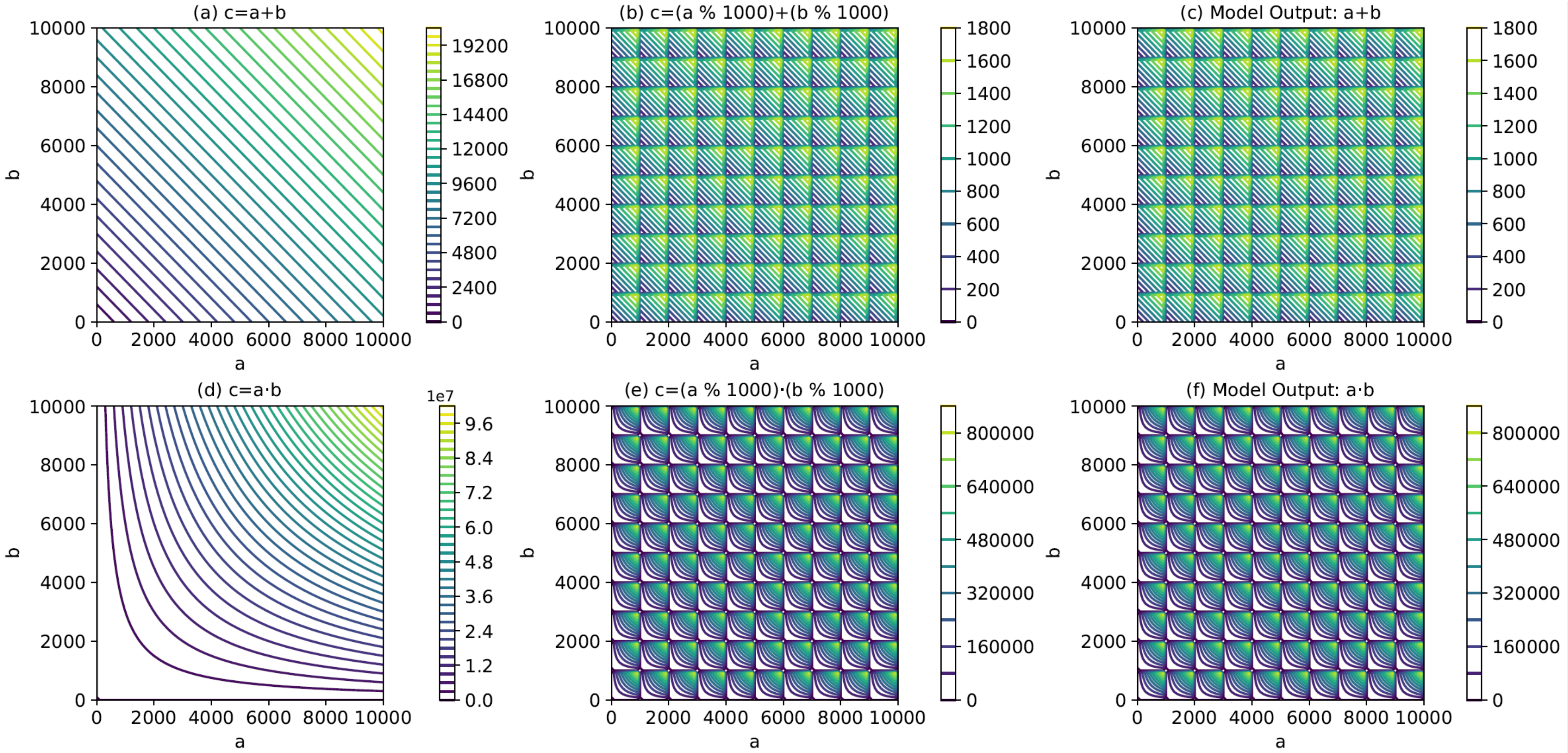
Generalization: Algebraic Structure

The models learn a function $f: \mathscr{D}_1 \cup \mathscr{D}_2 \cup \mathscr{D}_3 \to \mathscr{S}$ (training, ID test, OOD test domains)

The models map unseen OOD inputs to outputs with equivalence relations in the ID domain.

Equivalence Classes: $[(a,b)]_p := \{(x,y) \in \mathbb{N}^2 | \; x \equiv a \,(\mathrm{mod}\, p), \; y \equiv b(\mathrm{mod}\, p)\}$

$$\mathbf{Z}_p^2 = \mathbf{Z}_p \times \mathbf{Z}_p = \{[(a,b)]_p | \; (a,b) \in \mathbb{N}^2\} \quad \mathbf{Z}_p = \mathbb{Z}/p\mathbb{Z}$$



Generalization: Algebraic Structure

Robustness: Different model scales and data sizes

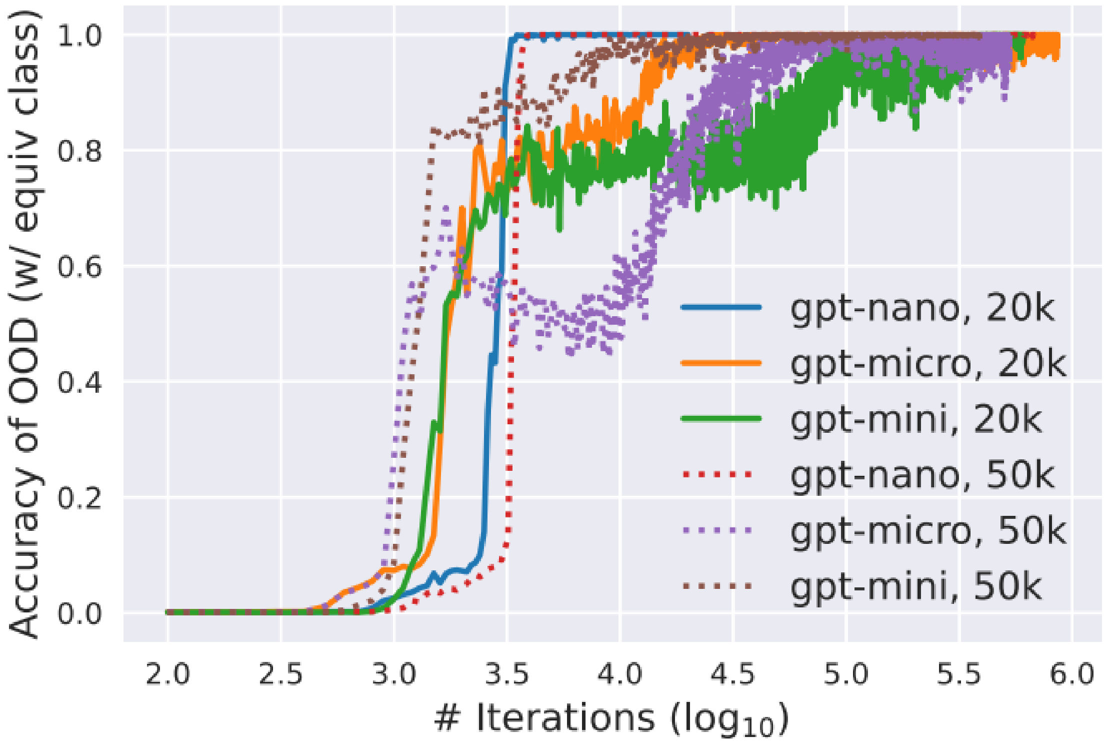


Figure 5: The accuracy of OOD test on equivalence for different model and data scales

Generalization: Algebraic Structure

Robustness

- Encoding method
- Scope of the dataset and training scheme

Versions	ID	OOD
V_1: rightmost digit be 0	100%	0
V_2: tens digit be 0	100%	0
V_3: non-reverse encoding	100%	0
V_4: extended OOD	100%	0

Table 3: The accuracy of ID test and OOD test in different addition variations.

$$[(a,b)]_p := \{(x,y) \in \mathbb{N}^2 \mid x \equiv \lfloor \tfrac{a}{p} \rfloor \cdot p,\ y \equiv \lfloor \tfrac{b}{p} \rfloor \cdot p\}. \tag{1}$$

$$\begin{aligned}[(a,b)]_p := \{(x,y) \in \mathbb{N}^2 \mid & x \equiv \lfloor \tfrac{a}{10p} \rfloor \cdot 10p + a \bmod p, \\ & y \equiv \lfloor \tfrac{b}{10p} \rfloor \cdot 10p + b \bmod p\}.\end{aligned} \tag{2}$$

03

Analysis on Probability and Representation

Generalization: Probability

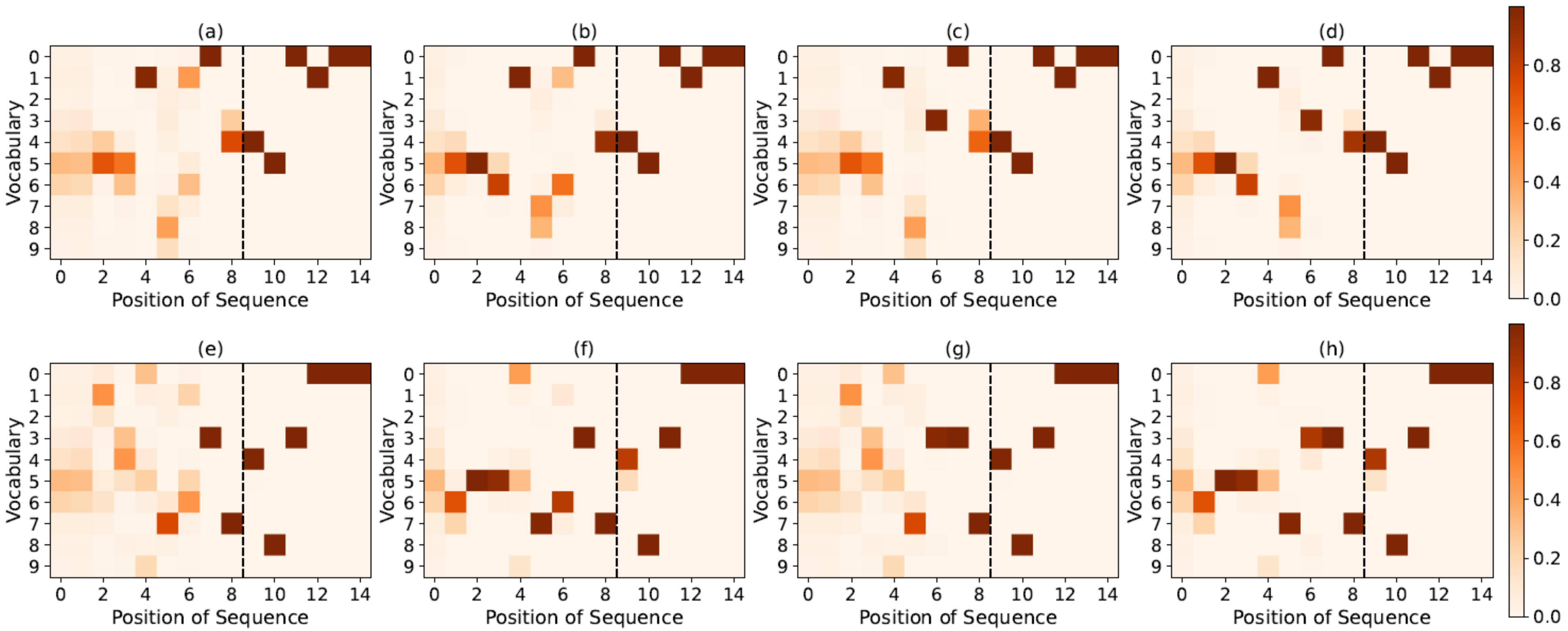
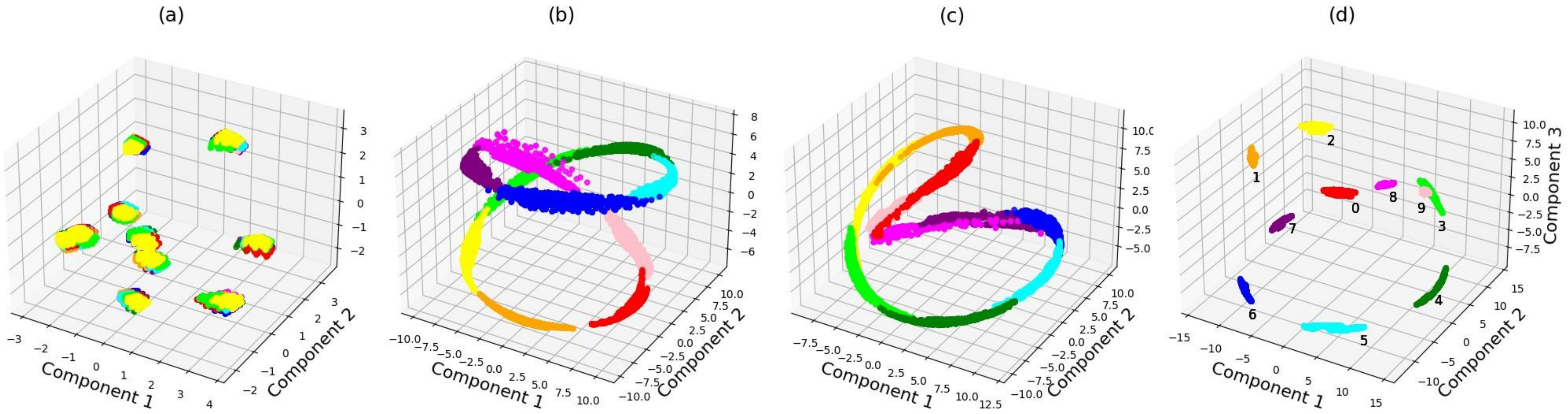


Figure 3: The probability distribution of each digit of the sequence in an addition operation $c = a + b$. The left side of the black dashed line represents the input $a + b$, while the right side is the result c. Figure 3(a) and Figure 3(e) represent the $349 + 705$ and $128 + 256$, and the outputs are $1,054$ and 384 (450100 and 483000 in actual sequence output), respectively. In the second column, we perturb the thousands digit of a: Figure 3(b) represents $1,349 + 705$, and Figure 3(f) represents $3,128 + 256$. In the third column, we perturb the thousands digit of b: Figure 3(c) represents $349 + 2,705$, and Figure 3(g) represents $128 + 4,256$. In the fourth column, we simultaneously perturb the thousands digit of a and b: Figure 3(d) represents $1,349 + 2,705$, and Figure 3(h) represents $3,128 + 4,256$.

Generalization: Representation

The representations gradually transition from disorderly to structured throughout the learning process. Initially, the representations appear random with colors mixed together (a). However, as the training progresses, the structure of the learned representations becomes increasingly refined ((b), (c)), ultimately leading to the development of a well-learned representation (d) where each color is separated according to its true label.



the learning process: random initialization → well-trained model

04

Further Research

Discussion

- Our study focused exclusively on arithmetic tasks, such as n-digit addition and multiplication problems. While arithmetic serves as a useful domain for analyzing generalization due to its structured nature, the findings may not directly translate to other types of NLP tasks or domains.
- Defining ID and OOD domains for natural language is challenging. Unlike arithmetic, where the numerical values and operations define the ID domain, NLP involves a wide range of linguistic structures, vocabularies, and contextual dependencies. The boundaries between ID and OOD examples become blurred, making it difficult to precisely delineate the domains.
- Equivalence relations allow the models to map inputs based on shared characteristics or properties and thus plays a key role in the generalization behaviors observed in arithmetic. However, due to the inherent ambiguity and subjectivity of language, much more efforts may be needed to establish clear-cut equivalence relations in NLP tasks.

Extend to Other Mysterious Phenomena

The findings could be used to explain the mysterious phenomena on modular operations learned by Transformer models.

c	PE	Size	Digits 5	10	20	30	35
100	APE	Base	100	98.8	96.2	90.2	88.1
		Large	100	100	100	100	100
	RPE_k	Base	100	100	97.5	85.8	65.2
		Large	100	100	100	100	100
	$\mathrm{RPE}_{k,q}$	Base	100	100	100	100	100
		Large	100	100	100	100	100
1000	APE	Base	80.2	69.8	43.4	26.3	6.4
		Large	28.2	12.2	9.9	8.7	7.7
	RPE_k	Base	100	84.8	4.9	0.2	0
		Large	100	100	100	99.9	26.4
	$\mathrm{RPE}_{k,q}$	Base	100	97.9	82.6	55.1	3.9
		Large	100	84.2	83.0	82.7	20.1
128	APE	Base	14.7	8.4	4.7	4.4	3.8
		Large	9.1	6.9	5.3	4.4	3.9
	RPE_k	Base	19.9	13.3	5.6	3.5	1.2
		Large	11.8	11.5	11.4	11.2	10.0
	$\mathrm{RPE}_{k,q}$	Base	26.9	21.7	14.1	10.3	6.2
		Large	20.4	20.5	19.2	18.4	16.2
101	APE	Base	44.8	2.3	2.4	2.4	2.3
		Large	1.1	1.2	1.2	1.1	1.1
	RPE_k	Base	24.5	2.3	1.9	1.8	1.4
		Large	95.3	2.3	2.2	2.0	2.1
	$\mathrm{RPE}_{k,q}$	Base	99.1	2.5	2.2	2.2	2.1
		Large	9.9	2.4	2.1	1.8	1.8

Table 4: **Modular addition:** Extrapolation results for modulo $c \in \{100, 1000, 128, 101\}$. UTransformer model in their Base and Large format. We report the accuracy reached by the models on 100,000 example test sets.

c	PE	Size	Digits 5	10	20	30	35
100	APE	Base	100	98.8	96.2	90.2	88.1
		Large	100	100	100	100	100
	RPE_k	Base	100	100	97.5	85.8	65.2
		Large	100	100	100	100	100
	$\mathrm{RPE}_{k,q}$	Base	100	100	100	100	100
		Large	100	100	100	100	100
1000	APE	Base	80.2	69.8	43.4	26.3	6.4
		Large	28.2	12.2	9.9	8.7	7.7
	RPE_k	Base	100	84.8	4.9	0.2	0
		Large	100	100	100	99.9	26.4
	$\mathrm{RPE}_{k,q}$	Base	100	97.9	82.6	55.1	3.9
		Large	100	84.2	83.0	82.7	20.1
128	APE	Base	14.7	8.4	4.7	4.4	3.8
		Large	9.1	6.9	5.3	4.4	3.9
	RPE_k	Base	19.9	13.3	5.6	3.5	1.2
		Large	11.8	11.5	11.4	11.2	10.0
	$\mathrm{RPE}_{k,q}$	Base	26.9	21.7	14.1	10.3	6.2
		Large	20.4	20.5	19.2	18.4	16.2
101	APE	Base	44.8	2.3	2.4	2.4	2.3
		Large	1.1	1.2	1.2	1.1	1.1
	RPE_k	Base	24.5	2.3	1.9	1.8	1.4
		Large	95.3	2.3	2.2	2.0	2.1
	$\mathrm{RPE}_{k,q}$	Base	99.1	2.5	2.2	2.2	2.1
		Large	9.9	2.4	2.1	1.8	1.8

Table 5: **Modular multiplication:** Extrapolation results for modulo $c \in \{100, 1000, 128, 101\}$. UTransformer model in their Base and Large format. We report the accuracy reached by the models on 100,000 example test sets.

Source: Jelassi et al., 2023.

Further Research

- Our new work on principled understanding of generalization:

  Xingcheng Xu, Zibo Zhao, Haipeng Zhang, and Yanqing Yang. "*Relating the Seemingly Unrelated: Principled Understanding of Generalization for Generative Models in Arithmetic Reasoning Tasks*." *arXiv preprint arXiv:2407.17963*, 2024.

Further Research

上海人工智能实验室 Shanghai Artificial Intelligence Laboratory

Experiments on Modular Addition

Modulus	Test Accuracy (%) w.r.t. the Ground Truth on the Domain $\widetilde{\mathcal{D}}_i$ 1	2	3	4	5	6	7	8	9	Theory $1/p'$
$p=50$	100	100	100	100	99.3	92.0	93.1	95.2	91.4	100
$p=51$	100	98.5	99.9	99.3	0.3	1.8	1.9	1.9	1.6	1.96
$p=100$	100	100	100	100	100	100	100	100	100	100
$p=101$	100	100	100	100	0.0	1.2	0.9	1.1	1.0	0.99
$p=150$	100	100	100	100	33.2	33.6	32.3	33.0	33.7	33.3
$p=151$	100	99.9	99.9	100	0.0	0.6	0.7	0.7	0.6	0.66
$p=200$	100	100	100	100	99.8	98.9	93.7	94.1	93.5	100
$p=201$	100	100	99.9	99.9	0.0	0.0	0.5	0.4	0.5	0.50

Table 3: Modular Addition: Test Accuracy w.r.t. the Ground Truth $f^p(a,b)=\overline{a+b}^p$ on $\widetilde{\mathcal{D}}_i$

Note: All the Transformer models in above experiments are instances of MiniGPT, which have been trained on a random sample drawn from $\mathcal{D}_4$ (except $p=150$). The accuracy is tested on 10,000 random test samples (when $n>2$), otherwise on the entire dataset. The outputs of models are generated using maximum probability sampling.

Source: Xu, Zhao, Zhang, and Yang. "*Relating the Seemingly Unrelated: Principled Understanding of Generalization for Generative Models in Arithmetic Reasoning Tasks.*" *arXiv preprint arXiv:2407.17963*, 2024.

Links

Paper

Poster

Our New Paper

Thanks for Your Attention!

https://www.shlab.org.cn

Shanghai Artificial Intelligence Laboratory