Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Compositional Generalization in Transformers

Masterthesis
at Research Group Knowledge Technology, WTM
Prof. Dr. Stefan Wermter

Department of Informatics
MIN-Faculty
Universität Hamburg

submitted by
Imran Ibrahimli
Course of study: Intelligent Adaptive Systems
Matrikelnr.: 7486484
on
24.10.2024

Examiners: Prof. Dr. Stefan Wermter
Dr. Jae Hee Lee

Abstract

Your English abstract here (mandatory if written in English and recommended otherwise).

Zusammenfassung

Hier die deutsche Zusammenfassung einfügen (notwendig).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Chapter 2

Background

2.1 Transformer

TODO: *lay o how these all a connected*
TODO: *make diagram for tran former architectu*
TODO: *ma a diagram f how transform is trained a inference*
TODO: *go ov with chatgpt, cla ify dimensions math,*

The Transformer architecture, introduced by Vaswani et al. (2017), has revolutionized natural language processing by enabling efficient modeling of sequential data without relying on recurrence or convolution. The core innovation lies in the self-attention mechanism, which allows the model to weigh the importance of different elements in the input sequence when generating representations.

Encoder-Decoder The original Transformer is an *encoder-decoder* architecture (Vaswani et al. 2017). The encoder maps an input sequence $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ into a sequence of continuous representations $\mathbf{z} = (z_1, z_2, \ldots, z_n)$. The decoder then generates an output sequence $\mathbf{y} = (y_1, y_2, \ldots, y_m)$ by predicting the next token y_t based on the previous tokens and the encoder's output.

Mathematically, each encoder layer consists of a multi-head self-attention mechanism followed by a position-wise feed-forward network:

$$\text{SelfAttention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{2.1}$$

$$\text{where } Q = XW_Q, \quad K = XW_K, \quad V = XW_V. \tag{2.2}$$

Here, X represents the input embeddings, and W_Q, W_K, W_V are learned projection matrices.

Encoder-only Encoder-only models utilize only the encoder part to produce contextualized embeddings of the input sequence. These models are effective for understanding and classification tasks where the entire input sequence is available at once. BERT (Devlin et al. 2019) is a prominent example of an encoder-only Transformer.

Decoder-only Decoder-only models focus on autoregressive generation by predicting the next token based solely on previous tokens. GPT-3 (Brown et al. 2020)

is a notable example, where the model generates text by sequentially extending the input prompt. This work focuses on decoder-only Transformers, since this modification is the most commonly used, and is general enough to be applied to many different types of tasks.

2.1.1 Positional Encoding Schemes

Transformers lack inherent positional information due to their permutation-invariant self-attention mechanism. To address this, positional encodings are added to the input embeddings.

Sinusoidal Positional Encoding The original Transformer uses sinusoidal positional encodings (Vaswani et al. 2017):

$$\mathrm{PE}_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right), \tag{2.3}$$

$$\mathrm{PE}_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right), \tag{2.4}$$

where pos is the position index and i is the dimension.

Relative Position Representations Shaw, Uszkoreit, and Vaswani (2018) introduced relative position representations to allow the self-attention mechanism to consider the relative distances between sequence elements. The attention weights are modified to include a term based on the relative position k:

$$e_{ij} = \frac{(x_i W_Q)(x_j W_K + a_{ij}^K)^\top}{\sqrt{d_k}}, \tag{2.5}$$

where a_{ij}^K is a learned embedding for the relative position between positions i and j.

Rotary Position Embedding (RoPE) Su et al. (2024) proposed RoPE, which encodes absolute positions using rotation matrices. This method introduces relative position dependency by rotating the query and key vectors:

$$\mathrm{RoPE}(x, p) = x \cdot \cos(\theta_p) + (\mathrm{rotate}(x)) \cdot \sin(\theta_p), \tag{2.6}$$

where θ_p is a function of the position p, and $\mathrm{rotate}(x)$ rotates the vector x.

Randomized Positional Encodings To improve length generalization, Ruoss et al. (2023) proposed randomized positional encodings. By simulating positions of longer sequences during training, the model can better handle unseen sequence lengths.

Functional Interpolation for Relative Positions (FIRE) S. Li et al. (2024) introduced FIRE, a functional relative position encoding with progressive interpolation. This approach generalizes existing relative position encodings and improves long-context Transformer performance.

2.1.2 Training and Inference

TODO: *clean math + notation*

Training Training a Transformer involves minimizing a loss function $\mathcal{L}$ over a dataset $\mathcal{D}$:

$$\min_{\theta} \sum_{(x,y)\in\mathcal{D}} \mathcal{L}(f_\theta(x), y), \tag{2.7}$$

where f_θ is the Transformer model with parameters θ. The loss function is typically the cross-entropy loss for sequence prediction tasks.

$$\mathcal{L}(y, \hat{y}) = -\sum_{t=1}^{m} y_t \log \hat{y}_t, \tag{2.8}$$

where y is the ground-truth sequence and $\hat{y}$ is the predicted sequence.

Inference During inference, the model generates outputs by autoregressively predicting the next token. For a decoder-only model, the probability of the next token is conditioned on all previous tokens:

$$P(y_t|y_{<t}, x) = \text{softmax}(f_\theta(y_{<t}, x)). \tag{2.9}$$

Beam search or other decoding strategies are used to find the most probable output sequence.

2.2 Mechanistic Interpretability

TODO: *expand*

Understanding how Transformers make decisions is crucial for interpretability and debugging. Key concepts include:

Residual Stream The residual stream in a Transformer accumulates information across layers through residual connections. Analyzing this stream helps in understanding how information is propagated and transformed.

Activation Patching Activation patching involves replacing activations in a model with activations from a different context to study causal effects (Olsson et al. 2022). This technique can identify which components contribute to specific model behaviors.

Logit Lens The logit lens method inspects the logits (pre-softmax outputs) at different layers to interpret intermediate representations. It provides insights into how predictions evolve across layers.

Olsson et al. (2022) investigated "induction heads," attention heads that enable in-context learning by recognizing patterns like repeated sequences.

2.3 Realizability

2.4 Deep Double Descent

ODO: *maybe name, fill, ASP/RASP-L nd Ahuja and lansouri paper*

The phenomenon of *deep double descent* refers to the observation that increasing model capacity or training epochs can initially degrade performance before improving it (Nakkiran et al. 2021). This challenges the traditional bias–variance trade-off, where increasing complexity always leads to overfitting.

Belkin et al. (2019) explored this phenomenon, showing that modern machine learning models can benefit from overparameterization, achieving better generalization even when they perfectly fit the training data.

Understanding deep double descent is important for training Transformers effectively, as it informs decisions about model size and training duration to optimize performance on tasks like integer addition.

ODO: *explain hy I care about is*

Chapter 3

Related Work

In this chapter, a literature review is presented, focusing on the topics of transformers, learning arithmetic tasks, reasoning capabilities, compositional learning, and generalization to longer sequence lengths.

TODO: *rewri summary*

3.1 Learning Arithmetics with Transformers

The ability of transformer models (Vaswani et al. 2017) to learn arithmetic operations such as integer addition has been a subject of significant research interest. Evaluations demonstrated that large pre-trained language models such as GPT-3 (Brown et al. 2020) can exhibit emergent capabilities across general-purpose tasks, including basic few-digit arithmetic, despite these tasks not being explicitly encoded by the unsupervised next-token prediction objective. However, even largest state-of-the-art models like GPT-4 (Achiam et al. 2023) struggle to robustly solve multi-digit addition and multiplication, especially when a larger number of digits are involved.

N. Lee et al. 2023 investigate how even small transformers, trained from random initialization, can efficiently learn arithmetic operations such as addition and multiplication. They show that training on chain-of-thought style data that includes intermediate step results significantly improves accuracy, sample complexity, and convergence speed, even in the absence of pretraining. This approach aligns with the experiments presented in this thesis on chain-of-thought training, where models are trained to output the steps involved in solving addition problems. One limitation of their work is that it limits each task to a fixed number of digits (e.g. 7 and 7 digit operands), using padding to ensure uniform input length. In contrast, this thesis extends the task to variable-length addition problems which is more difficult due to the need for models to learn to position-wise align digits as discussed in Chapter 4.

Understanding how transformers learn arithmetic tasks is further explored by Quirke and Barez 2023, who present an in-depth mechanistic analysis of a one-layer transformer model trained for integer addition. They reveal that the model processes the problem in a non-intuitive way: it divides the task into parallel,

digit-specific streams and employs distinct algorithms for different digit positions, merging the results in the MLP layer. This work also restricts the operands to a fixed length of 5 digits and employs padding, apart from restricting the model to a single layer.

Length generalization is a critical challenge in training transformers for arithmetic tasks that comes up in numerous research works. Jelassi et al. 2023 examine how transformers cope with learning basic integer arithmetic and generalizing to longer sequences than seen during training. They find that relative position embeddings enable length generalization for simple tasks such as addition, allowing models trained on 5-digit numbers to perform 15-digit sums. However, this method fails for multiplication, leading them to propose "train set priming" by adding a few (10 to 50) longer sequences to the training set. Despite showing interesting capabilites of learning from few examples, this defeats the purpose of *zero-shot* generalization to unseen lengths.

Similarly, Duan and Shi 2023 investigate the capabilities of transformers in learning arithmetic algorithms and introduce Attention Bias Calibration (ABC), a calibration stage that enables the model to automatically learn proper attention biases linked to relative position encoding mechanisms. Using ABC, they achieve robust length generalization on certain arithmetic tasks. Despite promising results, this work is limited due to the attention bias intervention being task-specific and the need for a modified training with 2 stages (first training to perfect interpolation accuracy to learn the attention biases, then training another model with extracted attention biases). Conversely, the main research interest in this domain lies in architectural modifications that apart from boosting algorithmic capabilities, also preserve or improve performance on other language tasks.

Recent work by McLeish et al. 2024 addresses the poor performance of transformers on arithmetic tasks by adding an embedding to each digit that encodes its position relative to the start of the number. This modification, along with architectural changes like input injection and recurrent layers, significantly improves performance, achieving up to 99% accuracy on 100-digit addition problems. As of now, this work represents the state-of-the-art in length generalization on integer addition, with test sequence lengths up to 6 times longer training sequence lengths (compared to previous SOTA of 2.5 times by Y. Zhou et al. 2024).

Investigations into the symbolic capabilities of large language models by Dave et al. 2024 and the arithmetic properties in the space of language model prompts by Krubiński 2023 further contribute to understanding how transformers process arithmetic operations and the challenges involved in symbolic reasoning tasks.

ODO: *expand on ese 2 papers*

In summary, the literature demonstrates that transformers can learn arithmetic tasks like integer addition, but challenges remain in achieving robust length generalization and understanding the underlying mechanisms by which these models perform arithmetic operations. These findings are directly relevant to the focus of this thesis, i. e. length generalization on integer addition and failure modes of transformers, since works either simplify the task to be modular addition ((**nanda˙modular˙fourier**), (**pizza˙clock**)), fixed digit length ((N. Lee et al. 2023), (Quirke and Barez 2023)), or propose task-specific architectural modifications to improve performance ((McLeish

ODO: *add zza clock, and el nanda fourier ldition*

et al. 2024)) whose performance on other language tasks is not explored.

3.2 Reasoning in Transformers

Transformers have shown remarkable abilities in reasoning tasks, particularly when employing techniques such as including a chain-of-thought (CoT) in the sequence instead of directly outputting an answer. Z. Li et al. 2024 provide a theoretical understanding of the power of chain-of-thought for decoder-only transformers, demonstrating that CoT empowers the model with the ability to perform inherently serial computation, which is otherwise lacking in transformers, especially for fewer layers. Intuitively, this is due to the fact that a CoT part in generated sequence allows the model to write out intermediate results and perform more computation before arriving at the final answer. This theoretical perspective also supports the experiments described in Chapter 5 involving chain-of-thought training to enhance the reasoning capabilities of models on arithmetic tasks.

Wang and D. Zhou 2024 explore the idea that chain-of-thought reasoning paths can be elicited from pre-trained language models by simply altering the decoding process, rather than relying on specific prompting techniques. Instead of decoding by taking the tokens with most activation (greedy decoding), they propose evaluating multiple possible first tokens, and then continue with greedy decoding for each of them. This results in multiple decoded sequences instead of a single answer sequence. They find that paths including a chain-of-thought part already frequently exist among these alternative sequences and that the presence of a CoT in the decoding path correlates with higher confidence in the model's decoded answer. This work strengthens the argument that chain-of-thought reasoning is a powerful mechanism for transformers to improve reasoning capabilities.

Another research direction is training the models to output a chain-of-thought sequence using bootstrapping from existing data and pre-trained models, without the need for curated CoT data. Zelikman, Wu, et al. 2022 propose the Self-Taught Reasoner (STaR) method to bootstrap reasoning capabilities using existing models and answer-only datasets. In STaR, a pre-trained LLM is encouraged to generate intermediate steps before answer using few-shot prompting, and it is assumed that if the resulting answer is correct, the generated CoT steps are also correct. Then, the model is fine-tuned on the generated CoT data. This method is shown to improve reasoning capabilities on various tasks, including arithmetic. Further extending the concept of self-generated reasoning, Zelikman, Harik, et al. 2024 introduce Quiet-STaR, where language models learn to generate "rationales" at each token to explain future text using reinforcement learning, thereby improving their predictions. This approach generalizes previous work on self-taught reasoning by Zelikman, Wu, et al. 2022, training the LLM to implicitly reason without explicit generation of a CoT trace, and can leverage unsupervised text datasets since the objective is not task-specific.

Goyal et al. 2024 propose training language models with "pause tokens", allowing the model to perform more computation before outputting the next tokens.

This method improves performance on reasoning tasks, including arithmetic. However, while it is tempting to think that the pause tokens implicitly perform a form of chain-of-thought reasoning, the authors do not explicitly analyze the internal reasoning processes of the model. Moreover, the pause tokens are not directly interpretable as distinct, structured intermediate steps, which is a key feature of chain-of-thought reasoning.

The limitations of transformers in performing counting tasks are highlighted by Yehudai et al. 2024, who focus on simple counting tasks involving counting the number of times a token appears in a string. They show that transformers can solve this task under certain conditions, such as when the dimension of the transformer state is linear in the context length. But in general this ability does not scale beyond this limit, based on the authors' theoretical arguments for the observed limitations. While this work does not directly address arithmetic tasks, it provides insights into the limitations of transformers in handling a related counting task.

Understanding how transformers learn causal structures is investigated by Nichani, Damian, and J. D. Lee 2024, who introduce an in-context learning task requiring the learning of latent causal structures. They prove that gradient descent on a simplified two-layer transformer learns to solve this task by encoding the latent causal graph in the first attention layer, providing insights into the mechanisms by which transformers develop reasoning capabilities. However, this work is highly theoretical and does not directly address arithmetic tasks, nor other algorithmic or language tasks.

Overall, these studies illustrate the importance of internal reasoning processes in transformers and provide various methods to enhance reasoning abilities, which are pertinent to our work on training models to output steps and analyzing how they attempt to solve addition tasks. Despite some theoretical results proving possibility of certain forms of reasoning, the literature is not clear on whether transformers can learn the generalizable algorithm for performing integer addition on an unbounded number of digits, which is the main focus of this thesis. Moreover, even if theoretical possibility was to be established, it would still be of interest to interpret the learned algorithm in a higher-level, human-understandable form akin to **nanda˙fourier**.

3.3 Compositional Learning

ODO: *cite Hupes et al. 2020 and xpand definition*

Compositionality refers to the ability of models to understand and generate new combinations of known components.

The work of Hupkes et al. 2020 provides a theoretical framework for understanding compositional generalization in neural networks. They propose tests to investigate whether models systematically recombine known parts and rules, generalize to longer sequences, and favor rules over exceptions during training. This framework is relevant to our analysis of how transformers learn and generalize in arithmetic tasks.

Dziri et al. 2023 investigate the limits of transformers on compositional tasks, such as multi-digit multiplication and logic grid puzzles. They find that transformers tend to solve compositional tasks by reducing multi-step reasoning into *linearized subgraph matching rather* than developing systematic problem-solving skills, suggesting limitations in compositional generalization. Apart from suggesting an interesting description for how multi-step reasoning is realized on transformers, this work does not explore integer addition problems, instead focusing on ¡5 digit multiplication and logic puzzles.

Press et al. 2023 measure the compositionality gap in language models by evaluating how often models can correctly answer all sub-problems but fail to generate the overall solution. They find that as model size increases, single-hop question answering performance improves faster than multi-hop performance, indicating that while larger models have better factual recall, they do not necessarily improve in their ability to perform compositional reasoning. They propose methods like chain-of-thought and self-ask prompting to narrow the compositionality gap.

H. Zhou et al. 2023 propose the RASP-Generalization Conjecture, suggesting that transformers tend to length generalize on a task if it can be solved by a short program (in a domain-specific language describing the transformer architecture) that works for all input lengths. They use this framework to understand when and how transformers exhibit strong length generalization on algorithmic tasks, which is closely related to compositional learning.

In summary, while transformers have demonstrated some ability to perform compositional tasks, significant challenges remain in achieving systematic compositional generalization. These findings inform this thesis' focus on multi-task learning and understanding how models can be trained to perform compositional operations like digit alignment and modular sum, before correctly combining results from these sub-tasks into a final answer.

3.4 Generalization to Longer Sequence Lengths

Generalization to longer sequence lengths is a critical challenge in training transformers for tasks like integer addition. Positional encoding plays a significant role in length generalization. Kazemnejad et al. 2023 conduct a systematic empirical study comparing different positional encoding approaches, including Absolute Position Embedding (APE) (Vaswani et al. 2017), Relative position encoding (Shaw, Uszkoreit, and Vaswani 2018), ALiBi (**alibi**), Rotary osition encoding (RoPE) (Su et al. 2024), and Transformers without positional encoding (NoPE). Interestingly, they find that explicit position embeddings are not essential for decoder-only transformers to generalize to longer sequences and that models without positional encoding outperform others in length generalization. The results of experiments conducted in this thesis are not consistent with these findings, observing much steeper performance drops on longer sequences when using NoPE, RoPE or APE.

Ruoss et al. 2023 introduce randomized positional encodings to boost length generalization of transformers. They demonstrate that their method allows trans-

formers to generalize to sequences of unseen length by simulating the positions of longer sequences and randomly selecting an ordered subset to fit the sequence's length. Experiments in this thesis also confirm that randomized positional encodings can improve length generalization on integer addition tasks. Moreover, experiments show that even without architectural changes, simply adding empty spaces to the input sequence can boost out-of-distribution accuracy for task lengths near to those encountered during training.

S. Li et al. 2024 propose a novel functional relative position encoding with progressive interpolation (FIRE) to improve transformer generalization to longer contexts. They theoretically show that FIRE can represent popular relative position encodings and empirically show that FIRE models have better generalization to longer contexts on both zero-shot language modeling and long text benchmarks.

The success of length generalization is also linked to data format and position encoding type. Y. Zhou et al. 2024 test the transformer's ability to generalize using the two integer addition task. They show that transformers can achieve limited length generalization, but find that the performance depends on random weight initialization as well.

As mentioned in Section 3.1, McLeish et al. 2024 address length generalization by adding an embedding to each digit that encodes its position relative to the start of the number. This fix, along with architectural modifications, allows transformers to generalize to larger and more complex arithmetic problems, achieving state-of-the-art performance on addition tasks with longer sequences.

In this thesis' experiments, different positional encodings and CoT training are explored to improve length generalization in integer addition tasks. The literature suggests that carefully designed positional encodings and data formatting can significantly impact the ability of transformers to generalize to longer sequences. Moreover, this work isolates the causes of failure in length generalization on integer addition, and proposes minimally invasive modifications to improve performance on longer sequences.

Chapter 4

Research Questions

Research Questions

1. How do (smaller) transformers learn to compose sub-tasks for integer addition?
2. What architectural features or training strategies enhance compositional learning? (we want them to be applicable to other domains as well, at least not hurt in regular unsupervised text setting)
3. How do transformers generalize to larger sequences than seen during training?

Compositionality gap: difference between performance on subtasks and the full task. In our case full task is integer addition, with subtasks being digit-wise alignment, addition, and carry operations.

Experiments / Hypotheses

- RQ3: Models generalize to in-distribution sequence lengths but not OOD (longer or shorter) lengths.
  - Baseline performance.
  - Smaller 3×3, 7×7 experiments, 1-19 digits and 18/20+ digits
  - Scratchpad can improve it to 18 and 20, not 21+
- RQ1: Sub-task decomposition: models decompose addition into:
  - digit-wise alignment
  - modular digit-wise addition
  - carry operations
  - Probing tasks for sub-task neural networks, saliency, etc. (TODO)
- RQ2: Positional embeddings are important for digit-wise addition and source of errors (e.g., alignment).

  - Random spaces
  - Abacus
  - Relative pos encodings

- RQ1: Compositional learning strategies:
  - Simple to complex tasks (curriculum)
  - Multi-task learning (aux loss) (TODO)
- Error analysis: TODO for robustness.

Chapter 5

Approach

Chapter 6

Conclusion

Appendix A

Nomenclature

Symbol	Description
SOTA	State-of-the-art
MLP	Multi-layer perceptron
RNN	Recurrent neural network
GPT	Generative pre-trained transformer
BERT	Bidirectional encoder representations from transformers
T5	Text-to-text transfer transformer
LLM	Large language model
CoT	Chain-of-thought
OOD	Out-of-distribution

Table A.1: Table of nomenclature

Bibliography

Achiam, OpenAI Josh et al. (Mar. 15, 2023). "GPT-4 Technical Report". In: URL: https://www.semanticscholar.org/paper/GPT-4-Technical-Report-Achiam-Adler/163b4d6a79a5b19af88b8585456363340d9efd04 (visited on 09/22/2024).

Belkin, Mikhail et al. (Aug. 6, 2019). "Reconciling modern machine-learning practice and the classical bias–variance trade-off". In: *Proceedings of the National Academy of Sciences* 116.32. Publisher: Proceedings of the National Academy of Sciences, pp. 15849–15854. DOI: 10.1073/pnas.1903070116. URL: https://www.pnas.org/doi/10.1073/pnas.1903070116 (visited on 09/29/2024).

Brown, Tom et al. (2020). "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., pp. 1877–1901. URL: https://papers.nips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html (visited on 09/22/2024).

Dave, Neisarg et al. (May 2024). "Investigating Symbolic Capabilities of Large Language Models". In: URL: https://www.semanticscholar.org/paper/Investigating-Symbolic-Capabilities-of-Large-Models-Dave-Kifer/cb1addf9cefe4e96763d28437f72a3d3cbfa7225 (visited on 06/15/2024).

Devlin, Jacob et al. (June 2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. NAACL-HLT 2019. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423 (visited on 09/29/2024).

Duan, Shaoxiong and Yining Shi (Oct. 2023). *From Interpolation to Extrapolation: Complete Length Generalization for Arithmetic Transformers*. arXiv:2310.11984 [cs]. DOI: 10.48550/arXiv.2310.11984. URL: http://arxiv.org/abs/2310.11984 (visited on 02/17/2024).

Dziri, Nouha et al. (Nov. 2023). "Faith and Fate: Limits of Transformers on Compositionality". en. In: URL: https://openreview.net/forum?id=Fkckkr3ya8 (visited on 02/17/2024).

Goyal, Sachin et al. (Apr. 2024). *Think before you speak: Training Language Models With Pause Tokens*. arXiv:2310.02226 [cs]. DOI: 10.48550/arXiv.2310.02226. URL: http://arxiv.org/abs/2310.02226 (visited on 07/15/2024).

Hupkes, Dieuwke et al. (Apr. 2020). "Compositionality Decomposed: How do Neural Networks Generalise?" en. In: *Journal of Artificial Intelligence Research* 67, pp. 757–795. ISSN: 1076-9757. DOI: 10.1613/jair.1.11674. URL: https://jair.org/index.php/jair/article/view/11674 (visited on 09/10/2024).

Jelassi, Samy et al. (June 2023). *Length Generalization in Arithmetic Transformers*. arXiv:2306.15400 [cs]. DOI: 10.48550/arXiv.2306.15400. URL: http://arxiv.org/abs/2306.15400 (visited on 02/17/2024).

Kazemnejad, Amirhossein et al. (Nov. 2023). "The Impact of Positional Encoding on Length Generalization in Transformers". en. In: URL: https://openreview.net/forum?id=Drrl2gcjzl (visited on 02/17/2024).

Krubiński, Mateusz (Oct. 2023). "Basic Arithmetic Properties in the Space of Language Model Prompts". en. In: URL: https://openreview.net/forum?id=RCiRtdERCW (visited on 09/09/2024).

Lee, Nayoung et al. (Oct. 2023). "Teaching Arithmetic to Small Transformers". en. In: URL: https://openreview.net/forum?id=YfhuG7xHQ8&referrer=%5Bthe%20profile%20of%20Jason%20D.%20Lee%5D(%2Fprofile%3Fid%3D~Jason_D._Lee1) (visited on 02/17/2024).

Li, Shanda et al. (Mar. 2024). *Functional Interpolation for Relative Positions Improves Long Context Transformers*. en. arXiv:2310.04418 [cs]. URL: http://arxiv.org/abs/2310.04418 (visited on 06/14/2024).

Li, Zhiyuan et al. (May 2024). *Chain of Thought Empowers Transformers to Solve Inherently Serial Problems*. arXiv:2402.12875 [cs, stat]. DOI: 10.48550/arXiv.2402.12875. URL: http://arxiv.org/abs/2402.12875 (visited on 09/16/2024).

McLeish, Sean et al. (May 2024). *Transformers Can Do Arithmetic with the Right Embeddings*. arXiv:2405.17399 [cs] version: 1. DOI: 10.48550/arXiv.2405.17399. URL: http://arxiv.org/abs/2405.17399 (visited on 06/02/2024).

Nakkiran, Preetum et al. (Dec. 2021). "Deep double descent: where bigger models and more data hurt*". In: *Journal of Statistical Mechanics: Theory and Experiment* 2021.12. Publisher: IOP Publishing and SISSA, p. 124003. ISSN: 1742-5468. DOI: 10.1088/1742-5468/ac3a74. URL: https://dx.doi.org/10.1088/1742-5468/ac3a74 (visited on 09/24/2024).

Nichani, Eshaan, Alex Damian, and Jason D. Lee (Feb. 2024). *How Transformers Learn Causal Structure with Gradient Descent*. arXiv:2402.14735 [cs, math, stat]. DOI: 10.48550/arXiv.2402.14735. URL: http://arxiv.org/abs/2402.14735 (visited on 02/26/2024).

Olsson, Catherine et al. (Sept. 2022). *In-context Learning and Induction Heads*. arXiv:2209.11895 [cs]. DOI: 10.48550/arXiv.2209.11895. URL: http://arxiv.org/abs/2209.11895 (visited on 04/07/2024).

Press, Ofir et al. (Dec. 2023). "Measuring and Narrowing the Compositionality Gap in Language Models". In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, pp. 5687–5711. DOI: 10.18653/v1/2023.findings-emnlp.378. URL: https://aclanthology.org/2023.findings-emnlp.378 (visited on 09/09/2024).

Quirke, Philip and Fazl Barez (Oct. 2023). "Understanding Addition in Transformers". en. In: URL: https://openreview.net/forum?id=rIx1YXVWZb (visited on 03/11/2024).

Ruoss, Anian et al. (May 2023). *Randomized Positional Encodings Boost Length Generalization of Transformers*. arXiv:2305.16843 [cs, stat]. DOI: 10.48550/arXiv.2305.16843. URL: http://arxiv.org/abs/2305.16843 (visited on 05/13/2024).

Shaw, Peter, Jakob Uszkoreit, and Ashish Vaswani (June 2018). "Self-Attention with Relative Position Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Ed. by Marilyn Walker, Heng Ji, and Amanda Stent. New Orleans, Louisiana: Association for Computational Linguistics, pp. 464–468. DOI: 10.18653/v1/N18-2074. URL: https://aclanthology.org/N18-2074 (visited on 04/03/2024).

Su, Jianlin et al. (Feb. 2024). "RoFormer: Enhanced transformer with Rotary Position Embedding". In: *Neurocomputing* 568, p. 127063. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2023.127063. URL: https://www.sciencedirect.com/science/article/pii/S0925231223011864 (visited on 06/17/2024).

Vaswani, Ashish et al. (2017). "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc. (Visited on 02/17/2024).

Wang, Xuezhi and Denny Zhou (May 23, 2024). *Chain-of-Thought Reasoning Without Prompting*. DOI: 10.48550/arXiv.2402.10200. arXiv: 2402.10200[cs]. URL: http://arxiv.org/abs/2402.10200 (visited on 09/22/2024).

Yehudai, Gilad et al. (July 2024). *When Can Transformers Count to n?* arXiv:2407.15160 [cs] version: 1. DOI: 10.48550/arXiv.2407.15160. URL: http://arxiv.org/abs/2407.15160 (visited on 09/06/2024).

Zelikman, Eric, Georges Harik, et al. (Mar. 2024). *Quiet-STaR: Language Models Can Teach Themselves to Think Before Speaking*. arXiv:2403.09629 [cs]. DOI: 10.48550/arXiv.2403.09629. URL: http://arxiv.org/abs/2403.09629 (visited on 07/15/2024).

Zelikman, Eric, Yuhuai Wu, et al. (Dec. 2022). "STaR: Bootstrapping Reasoning With Reasoning". en. In: *Advances in Neural Information Processing Systems* 35, pp. 15476–15488. (Visited on 07/31/2024).

Zhou, Hattie et al. (Oct. 2023). *What Algorithms can Transformers Learn? A Study in Length Generalization*. arXiv:2310.16028 [cs, stat]. DOI: 10.48550/arXiv.2310.16028. URL: http://arxiv.org/abs/2310.16028 (visited on 02/17/2024).

Zhou, Yongchao et al. (Feb. 2024). *Transformers Can Achieve Length Generalization But Not Robustly*. arXiv:2402.09371 [cs]. URL: http://arxiv.org/abs/2402.09371 (visited on 02/17/2024).

Erklärung der Urheberschaft

Hiermit versichere ich an Eides statt, dass ich die vorliegende Masterthesis im Studiengang Intelligent Adaptive Systems selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel - insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.

Ort, Datum Unterschrift

Erklärung zur Veröffentlichung

Ich stimme der Einstellung der Masterthesis in die Bibliothek des Fachbereichs Informatik zu.

Ort, Datum Unterschrift