Extended Abstract Track

How Capable Can a Transformer Become? A Study on Synthetic, Interpretable Tasks

Abstract

Transformers trained on huge text corpora exhibit a remarkable set of capabilities. Given the inherent compositional nature of language, one can expect the model to learn to compose these capabilities, potentially yielding a *combinatorial explosion* of what operations it can perform on an input. Motivated by the above, we aim to assess in this paper "how capable can a transformer become?". In this work, we train Transformer models on a data-generating process that involves compositions of a set of well-defined monolithic capabilities and show that: (1) Transformers generalize to exponentially or even combinatorially many functions not seen in the training data; (2) Transformers that generate the intermediate outputs of the composition are more effective at generalizing to unseen compositions; (3) The training data has a significant impact on the model's ability to compose functions (4) Attention layers in the latter half of the model seem critical to compositionality.

Keywords: Transformers, Capabilities, Mechanistic interpretability, Synthetic task

1. Introduction

Large Language Models (LLMs) showcase an impressive spectrum of capabilities (Radford et al., 2018, 2019; Brown et al., 2020; Wei et al., 2022a, 2021; Thoppilan et al., 2022; Touvron et al., 2023; Hoffmann et al., 2022), such as generating coherent and contextually relevant text over extended passages or answering complex questions based on provided context. Acquiring such a diverse set of capabilities is crucial to developing artificial intelligence systems that are both general and flexible.

However, true generality is not simply a by-product of accumulation, but emerges when these capabilities are combined in novel ways. Learning a set of monolithic capabilities, and composing them together could potentially lead to a *combinatorial explosion* of what operations can be performed on an input. Although LLMs have shown initial "sparks" (Bubeck et al., 2023) of compositional behavior, the underlying mechanisms and limits of this behavior remains elusive.

Studying compositionality directly in language models is difficult because it is (i) hard to precisely control what set of capabilities are learnable via the training data and (ii) difficult to train a language model once, let alone multiple times. This motivates us to consider a minimal synthetic setup that allows us to study compositionality and characterize phenomena relating to it. The goal of this work is similar in spirit to recent works that study transformers on synthetic datasets (Garg et al., 2022; Liu et al., 2022; Allen-Zhu and Li, 2023a; Li et al., 2023b).

We introduce the synthetic task of applying a sequence of mathematical operations, such as bijections and permutations, to an input to obtain accurate outputs. Our contributions with this framework are: **(1) A Synthetic setup.** We propose a minimal synthetic setup to study compositions of capabilities in transformers. We define two different types of compositions and find that step-by-step compositions seem to generalize compositionally in more scenarios compared to direct compositions. **(2) Demonstration of exponential explosion of capabilities.** We find that transformers compositionally generalize to exponentially or even combinatorially many functions – which would be considered "out-of-distribution". **(3) Mechanistic study of attention layers.** Attention layers between layers 6-10 show a large increase in probe accuracy, suggesting their importance in compositional generalization.

2. Preliminaries

To make progress on the question of how capable a Transformer can become, we aim to evaluate its ability to compose skills acquired via pretraining. To this end, we first present a definition of compositionality and categorize different types of compositionality.

2.1. Definitions

The term *capability* in our setup refers to the ability of a model to accurately implement a function $f : X^k \mapsto X^k$ that maps a sequence of tokens $x \in X^k$ to another sequence in the same domain, i.e., an automorphism. This is motivated by the fact that the input and output of language models is the same. We would like to understand the set of capabilities—or the set of functions—that a Transformer can implement by composing together such capabilities. The notions of compositionality most relevant to our work are systematicity and productivity (Fodor, 1975), i.e., a model is compositional if it can accurately compute a set of functions $\mathcal{F}$ and also compose them together (in a mathematical sense). We formalize this as follows.

Definition 1 ***(Compositional model)*** *We define a model $M : \mathcal{F}^L \times X^k \mapsto X^k$ to be compositional if for any sequence of functions $(f_i)_{i=1}^L \in F^L$, the model M satisfies*

$$M((f_1, f_2, \cdots f_L), x) = f_L \circ f_{L-1} \circ \cdots \circ f_1(x).$$

Consider a family of functions $\mathcal{F}$ containing $N \times L$ different functions. Each function in this set is uniquely indexed by $f_n^{[l]}$ where $n \in \{1, \ldots N\}$ and level $l \in \{1 \ldots, L\}$. We would like to select L functions from this family to compose together.

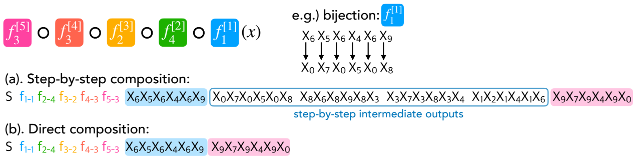
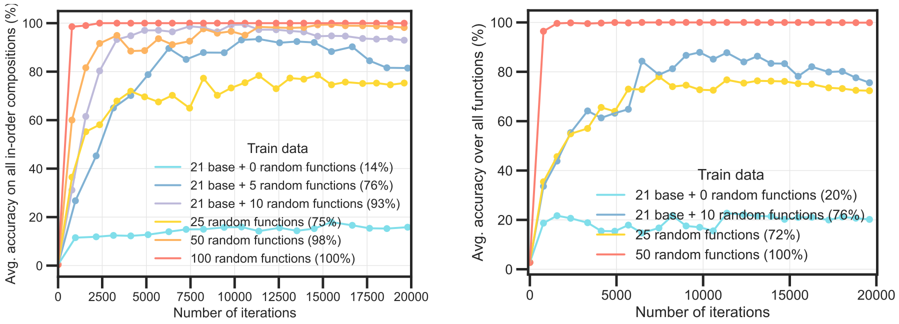

$f_3^{[5]} \circ f_3^{[4]} \circ f_2^{[3]} \circ f_4^{[2]} \circ f_1^{[1]}(x)$
e.g.) bijection: $f_1^{[1]}$
X₆ X₅ X₆ X₄ X₆ X₉
X₀ X₇ X₀ X₅ X₀ X₈
(a). Step-by-step composition:
S f1-1 f2-4 f3-2 f4-3 f5-3 X6X5X6X4X6X9 X0X7X0X5X0X8 X8X6X8X9X8X3 X3X7X3X8X3X4 X1X2X1X4X1X6 X9X7X9X4X9X0
step-by-step intermediate outputs
(b). Direct composition:
S f1-1 f2-4 f3-2 f4-3 f5-3 X6X5X6X4X6X9 X9X7X9X4X9X0

Figure 1: **Step-by-step composition v.s. Direct composition. (a)** Transformers can compose functions while generating intermediate outputs. **(b)** Alternately, the model can compose the functions without the intermediate outputs.

In appendix A, we define two different ways to compose L functions together from the set of $N \times L$ functions: in-order and out-of-order (fig. 4). We use this characterization to study how the types of compositions in the training data influence the nature of compositional generalization.

2.2. Data Generating Process

We consider two types of functions $f : X^k \mapsto X^k$ that map from a sequence of tokens to another of tokens: bijections and permutations. The rationale for this choice is that both



(a) (b)

Figure 2: **Transformers can generalize to an exponential (a) or combinatorial (b) number of new functions.** We plot the accuracy averaged over all compositions of 5 bijections, where each level of composition has 4+1 choices – with one of them being the identity function. Each curve corresponds to training data generated by a different subset of functions and the Transformer is trained using the step-by-step prompt format. **(a)** The choice of 5 functions are different at different levels of composition – there are 21 different functions (1 identity) which can be composed (in-order) in 3125 different ways. **(b)** The choice of 5 functions are identical across all 4 levels of compositions which means there are 3125 different ways to compose them; only 1365 of them are unique. Both figures are evidence that one can train on a small number of compositions of functions (around 31-100) and generalize to exponentially (a) and combinatorially (b) many functions that would be considered "out-of-distribution".

these functions form groups with composition as the group operator. The co-domain and range of functions are guaranteed to match and the composition of functions will always yield a member of the group.

The vocabulary consists of two types of tokens, task tokens denoted by X_F and the regular tokens denoted by the set X. The task tokens specify the composition of functions to apply on the input $x \in X^k$. We consider two types of prompts to test compositionality in transformers (see fig. 1). The first is the step-by-step composition prompt that consists of a sequence of task tokens, the input which is an element from X^k, followed by the output of every intermediate step of the function compositions. The second is the direct composition prompt which consists of a sequence of task tokens, the input which is an element from X^k, followed by the final output of the composition of all the functions–without the intermediate steps.

3. Results

We systematically investigate the capabilities of a transformer trained on synthetic tasks with compositional structure. We find that: (1) Transformers generalize to an exponential set of compositions (fig. 2(a)) and partially to a combinatorial set of compositions (fig. 2(b)) not present in the training data; (2) Compositional generalization often fails in the direct prompt format compared to the step-by-step format (figs. 3 and 7); (3) The compositions in the training data influence the extent of in-order (exponential) and out-of-order (combinatorial) generalization (fig. 4); (4) The compositions in the training data determine if composition of many functions are learnt before compositions of few of them (fig. 8); (5) A linear probe reveals that attention layers 6-10 seem critical to compositional generalization (fig. 9). We present results (1) and (2) below with the rest presented in appendix E.

Combinatorial explosion and exponential growth in capabilities Do transformers only generalize to functions present in the training data or do they reflect compositional structure present in data? In fig. 2 we train on data consisting of a small subset of (in-order) compositions of functions, in the step-by-step prompt format.

We consider two scenarios which both consider the composition of 5 functions (or 5 levels) in figs. 2(a) and 2(b). The composition of functions at each level can be one of 4 choices, with the 4 choices at each level being different in fig. 2(a) and the same in fig. 2(b). In addition, any of the levels can also assume the identity function.

The training data for fig. 2 has two major variations. The set of functions **random**, considers a random set of compositions of functions from the set of all possible in-order compositions. The set of functions **21 base**, considers each of the 4 functions at each of the 5 levels and the identity function, totalling to 21 functions. The set **21 base** excludes all compositions of 2 or more functions.

We find that Transformers capture the compositional structure in data and generalize to an exponential and combinatorial set of functions in figs. 2(a) and 2(b), despite being trained on a small subset of function compositions. A transformer trained on just 30-100 compositions of bijections generalizes to 3125 unseen compositions of these bijections almost perfectly. This could explain why language models show signatures of compositionality.

Direct vs. Step-by-step composition In fig. 3, we consider the setup identical to fig. 2(a) and train on a different number of **random** functions. Transformers fail to generalize to new in-order compositions with direct compositions when we consider compositions of bijections. We observe this failure even if we train of 2000 of the 3125 possible in-order compositions of functions (see appendix E.2). In fig. 7(b), we see that step-by-step compositions show compositional generalization in the same scenario.

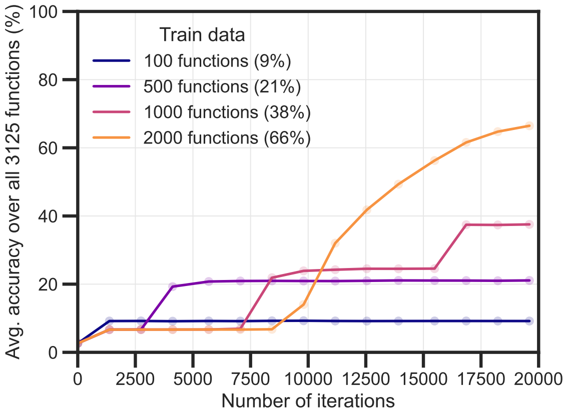


Figure 3: **Compositional generalization is less frequently seen in the direct prompt format compared to the step-by-step format.** We train a Transformer on 20+1 bijections with 5 levels of compositions with 4 choices at each level. The Transformer fails to generalize to all 3125 compositions even if it trained on 2000 such functions.

References

Abubakar Abid, Maheen Farooqi, and James Zou. Persistent anti-muslim bias in large language models. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 298–306, 2021.

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 1, context-free grammar. *arXiv preprint arXiv:2305.13673*, 2023a.

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of Language Models: Part 1, Context-Free Grammar, 2023b.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint. arXiv:2108.07258*, 2021.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Alan Chan, Rebecca Salganik, Alva Markelius, Chris Pang, Nitarshan Rajkumar, Dmitrii Krasheninnikov, Lauro Langosco, Zhonghao He, Yawen Duan, Micah Carroll, et al. Harms from increasingly agentic algorithmic systems. *arXiv preprint arXiv:2302.10329*, 2023.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*, 2023.

Meta Fundamental AI Research Diplomacy Team (FAIR)†, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022.

Jerry A Fodor. *The language of thought*, volume 5. Harvard university press, 1975.

Steven M Frankland and Joshua D Greene. Concepts and compositionality: in search of the brain's language of thought. *Annual review of psychology*, 71:273–303, 2020.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

Ismael Garrido-Muñoz, Arturo Montejo-Ráez, Fernando Martínez-Santiago, and L Alfonso Ureña-López. A survey on bias in deep nlp. *Applied Sciences*, 11(7):3184, 2021.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*, 2020.

Noah D Goodman, Joshua B Tenenbaum, Thomas L Griffiths, and Jacob Feldman. Compositionality in rational analysis: Grammar-based induction for concept learning. *The probabilistic mind: Prospects for Bayesian cognitive science*, 2008.

Michael Hahn and Navin Goyal. A theory of emergent in-context learning as implicit structure induction. *arXiv preprint arXiv:2303.07971*, 2023.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Arian Hosseini, Ankit Vani, Dzmitry Bahdanau, Alessandro Sordoni, and Aaron Courville. On the compositional generalization gap of in-context learning. *arXiv preprint arXiv:2211.08473*, 2022.

Po-Sen Huang, Huan Zhang, Ray Jiang, Robert Stanforth, Johannes Welbl, Jack Rae, Vishal Maini, Dani Yogatama, and Pushmeet Kohli. Reducing sentiment bias in language models via counterfactual evaluation. *arXiv preprint arXiv:1911.03064*, 2019.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795, 2020.

Liwei Jiang, Jena D Hwang, Chandra Bhagavatula, Ronan Le Bras, Jenny Liang, Jesse Dodge, Keisuke Sakaguchi, Maxwell Forbes, Jon Borchardt, Saadia Gabriel, et al. Can machines learn morality? the delphi experiment. *arXiv e-prints*, pages arXiv–2110, 2021.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017.

Michael A Lepori, Thomas Serre, and Ellie Pavlick. Break it down: Evidence for structural compositionality in neural networks. *arXiv preprint arXiv:2301.10884*, 2023.

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent World Representations: Exploring a Sequence Model Trained on a Synthetic Task, 2023a. Comment: ICLR 2023 oral (notable-top-5%): https://openreview.net/forum?id=DeG07_TcZvT ; code: https://github.com/likenneth/othello_world.

Yingcong Li, Kartik Sreenivasan, Angeliki Giannou, Dimitris Papailiopoulos, and Samet Oymak. Dissecting chain-of-thought: A study on compositional in-context learning of mlps. *arXiv preprint arXiv:2305.18869*, 2023b.

Tom Lieberum, Matthew Rahtz, János Kramár, Geoffrey Irving, Rohin Shah, and Vladimir Mikulik. Does circuit analysis interpretability scale? evidence from multiple choice capabilities in chinchilla. *arXiv preprint arXiv:2307.09458*, 2023.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

Bingbin Liu, Jordan T Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. *arXiv preprint arXiv:2210.10749*, 2022.

Bingbin Liu, Jordan T Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Exposing attention glitches with flip-flop language modeling. *arXiv preprint arXiv:2306.00946*, 2023.

Kris McGuffie and Alex Newhouse. The radicalization risks of gpt-3 and advanced neural language models. *arXiv preprint arXiv:2009.06807*, 2020.

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. *arXiv preprint arXiv:2309.00941*, 2023.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*, 2021.

Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Samuel R Bowman. Bbq: A hand-built bias benchmark for question answering. *arXiv preprint arXiv:2110.08193*, 2021.

Steven Phillips and William H Wilson. Categorial compositionality: A category theory explanation for the systematicity of human cognition. *PLoS computational biology*, 6(7): e1000858, 2010.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. The woman worked as a babysitter: On biases in language generation. *arXiv preprint arXiv:1909.01326*, 2019.

Alex Tamkin, Miles Brundage, Jack Clark, and Deep Ganguli. Understanding the capabilities, limitations, and societal impact of large language models. *arXiv preprint arXiv:2102.02503*, 2021.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Josef Valvoda, Naomi Saphra, Jonathan Rawski, Adina Williams, and Ryan Cotterell. Benchmarking compositionality with formal languages. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6007–6018, 2022.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022a.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022b.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*, 2021.

Zhengxuan Wu, Atticus Geiger, Christopher Potts, and Noah D Goodman. Interpretability at scale: Identifying causal mechanisms in alpaca. *arXiv preprint arXiv:2305.08809*, 2023.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

Albert Xu, Eshaan Pathak, Eric Wallace, Suchin Gururangan, Maarten Sap, and Dan Klein. Detoxifying language models risks marginalizing minority voices. *arXiv preprint arXiv:2104.06390*, 2021.

Jing Xu, Da Ju, Margaret Li, Y-Lan Boureau, Jason Weston, and Emily Dinan. Recipes for safety in open-domain chatbots. *arXiv preprint arXiv:2010.07079*, 2020.

Haoyu Zhao, Abhishek Panigrahi, Rong Ge, and Sanjeev Arora. Do transformers parse while predicting the masked word? *arXiv preprint arXiv:2303.08117*, 2023.

Appendix A. Preliminaries: In-order and Out-of-order Compositions

An in-order composition is a sequence of L functions denoted by $(f_k^{[1]}, \dots, f_j^{[L-1]}, f_i^{[L]})$ such that the l^{th} element in the tuple is selected from one of the N different choices available to it from the set $\{f_n^{[l]}\}_{n=1}^{N}$. An out-of-order composition is a sequence of L functions, where each element in the sequence is allowed to be any element from $\mathcal{F}$.

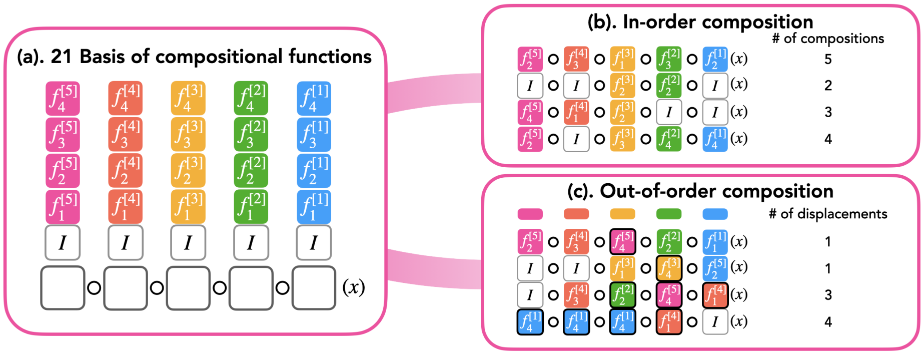

(a). 21 Basis of compositional functions
$f_4^{[5]}$ $f_4^{[4]}$ $f_4^{[3]}$ $f_4^{[2]}$ $f_4^{[1]}$
$f_3^{[5]}$ $f_3^{[4]}$ $f_3^{[3]}$ $f_3^{[2]}$ $f_3^{[1]}$
$f_2^{[5]}$ $f_2^{[4]}$ $f_2^{[3]}$ $f_2^{[2]}$ $f_2^{[1]}$
$f_1^{[5]}$ $f_1^{[4]}$ $f_1^{[3]}$ $f_1^{[2]}$ $f_1^{[1]}$
I I I I I
$\circ$ $\circ$ $\circ$ $\circ$ (x)
(b). In-order composition
# of compositions
$f_2^{[5]} \circ f_3^{[4]} \circ f_1^{[3]} \circ f_3^{[2]} \circ f_2^{[1]} (x)$ 5
$I \circ I \circ f_2^{[3]} \circ f_2^{[2]} \circ I (x)$ 2
$f_4^{[5]} \circ f_1^{[4]} \circ f_2^{[3]} \circ I \circ I (x)$ 3
$f_2^{[5]} \circ I \circ f_3^{[3]} \circ f_4^{[2]} \circ f_4^{[1]} (x)$ 4
(c). Out-of-order composition
# of displacements
$f_2^{[5]} \circ f_3^{[4]} \circ f_4^{[5]} \circ f_2^{[2]} \circ f_1^{[1]} (x)$ 1
$I \circ I \circ f_1^{[3]} \circ f_4^{[3]} \circ f_2^{[5]} (x)$ 1
$I \circ f_3^{[4]} \circ f_2^{[2]} \circ f_4^{[5]} \circ f_1^{[4]} (x)$ 3
$f_4^{[1]} \circ f_4^{[1]} \circ f_4^{[1]} \circ f_1^{[4]} \circ I (x)$ 4

Figure 4: **Data generating process for in-order and out-of-order compositions.** (a) Each of the $L = 5$ levels is associated with $N = 4$ functions $f_i^{[l]}$, in addition to an identity function, resulting in a total of $5 \times 4 + 1 = 21$ basis functions for composition. (b) The in-order compositions select functions within the same level l while (c) out-of-order compositions allow for selecting functions across levels. Each level also include the identity function since it allows us to compute compositions of fewer than $L = 5$ functions. In the examples presented in (c), displaced functions are surrounded by a black line, and we then count the number of displaced functions.

Definition 2 (In-order Generalization) *A model is said to generalize* in-order *if the model generalizes to a test dataset with both the training and test datasets constructed with in-order compositions of functions adhering to the structure of* $f_i^{[L]} \circ f_j^{[L-1]} \circ \dots \circ f_k^{[1]}$*, where* $i, j, \dots, k \in \{1, 2, \dots, N\}$.

Definition 3 (Out-of-order Generalization) *A model is said to generalize* out-of-order *if it generalizes to a test set constructed using out-of-order compositions and a training set with in-order compositions, i.e., the test contains out-of-order compositions of the form* $f_i^{[l]} \circ f_j^{[l']} \circ \dots \circ f_k^{[l'']}$*, where* $l, l', l'' \in \{1, 2, \dots, L\}$.

Note that if the set of functions at every level $l \in \{1, \dots, L\}$ are identical, then in-order and out-of-order compositions correspond to the same set.

Per the definitions above, if a Transformer with N capabilities can perform in-order generalization, its set of capabilities will in fact grow to exponentially many functions—N^L of them to be precise. Further, the ability to compose and generalize out-of-order can increase this set combinatorially, i.e., proportional to $(N \times L)^L$, growing even more quickly compared to the set of in-order compositions. Such an "explosion of capabilities" would imply perfect knowledge of what all tasks a pretrained model can perform is rather difficult to characterize, especially since its pretraining data is generally unknown and hence it is hard to characterize even what monolithic capabilities a model possesses. Our synthetic setup described in Sec.2.2 however endows us the ability to characterize the model's ability to compose. We thus define the following notion of displacement, which serves as a useful tool in our discussion.

Definition 4 (Displacement) *The displacement of an out-of-order composition, represented by an L-tuple $(f_1, \ldots, f_L)$, is defined as the number of elements f_l that do not belong to the set $\{f_n^{[l]}\}_{n=1}^N$.*

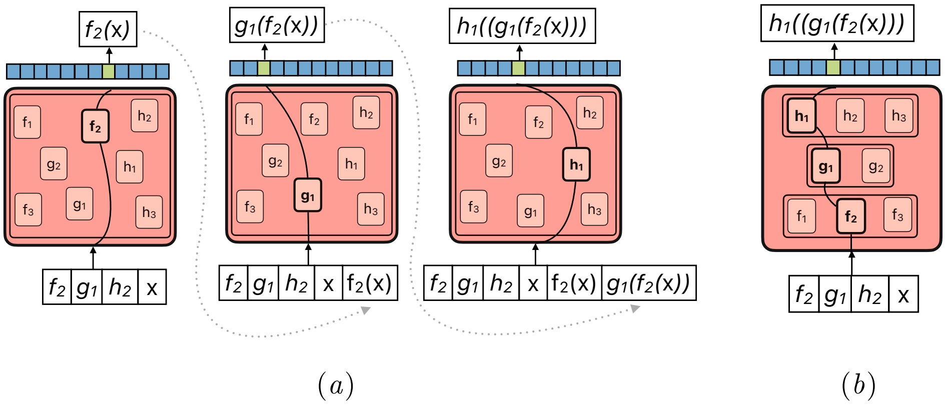

$f_2(x)$
$g_1(f_2(x))$
$h_1((g_1(f_2(x)))$
$h_1((g_1(f_2(x)))$
f_1 f_2 h_2 g_2 h_1 f_3 g_1 h_3
f_2 g_1 h_2 x
f_2 g_1 h_2 x $f_2(x)$
f_2 g_1 h_2 x $f_2(x)$ $g_1(f_2(x))$
h_1 h_2 h_3 g_1 g_2 f_1 f_2 f_3
f_2 g_1 h_2 x
(a)
(b)

Figure 5: **Step-by-step composition v.s. Direct composition. (a)** Transformers can compose functions while generating intermediate outputs. **(b)** Alternately, the model can compose the functions without the intermediate outputs.

Appendix B. Related Work

Capabilities in a Transformer. Prior work evaluating language models pretrained on large-scale, web-crawled text datasets demonstrate very general capabilities, often achieving highly competitive performance on a variety of tasks such as primitive arithmetic, question answering, commonsense knowledge, stylistic transformation of a piece of text, and even multimodal reasoning Radford et al. (2018, 2019); Brown et al. (2020); Bubeck et al. (2023); Wei et al. (2022a, 2021); Rae et al. (2021); Chowdhery et al. (2022); Austin et al. (2021); Chen et al. (2021); (FAIR); Bommasani et al. (2021). However, this generality can come at the cost of a model also learning capabilities that are undesirable Bommasani et al. (2021); Tamkin et al. (2021); Chan et al. (2023), e.g., producing sensitive, biased, or toxic outputs in the pursuit of solving a task Weidinger et al. (2021); McGuffie and Newhouse (2020); Garrido-Muñoz et al. (2021); Lin et al. (2021); Jiang et al. (2021); Abid et al. (2021); Parrish et al. (2021); Xu et al. (2021); Huang et al. (2019); Sheng et al. (2019); Gehman et al. (2020); Xu et al. (2020); Tamkin et al. (2021). We are in fact motivated by this latter position in the current work. For example, if a model possesses the capability to produce biased text, can it compose that capability with the ability to perform reasoning and yield biased reasoning? To get a hold on this question, we argue first an evaluation that demonstrates the extent to which a model can compose its capabilities is necessary—our work is focused on making progress on this front.

Compositionality. While compositionality has been debated since Fodor's hypothesis on its implications in human intelligence (Fodor, 1975), there are several related notions that the term now sees use for both in and outside the purview of machine learning (Valvoda et al., 2022; Johnson et al., 2017; Hosseini et al., 2022; Lepori et al., 2023; Frankland and Greene, 2020; Phillips and Wilson, 2010; Goodman et al., 2008). However, we specifically highlight the work by Hupkes et al. (2020), who ground two important notions that relate to our work: *systematicity* and *productivity*. Systematicity is akin to "out-of-distribution" generalization, whereby structured variations of data should affect a system's outputs predictably; e.g., changing the color of a dog should still lead the system to predict that it is a dog. Meanwhile, productivity is closer to the mathematical notion of chaining two functions, provided their domain and co-domain match.

Understanding transformers via synthetic tasks. Several intriguing works have recently utilized synthetic tasks to assess the limits of Transformers trained on autoregressive, distribution modeling tasks such as learning formal grammars, hidden markov models, and even board games (Allen-Zhu and Li, 2023b; Liu et al., 2022, 2023; Zhao et al., 2023; Hahn and Goyal, 2023; Nanda et al., 2023; Liu et al., 2022; Xie et al., 2021; Valvoda et al., 2022; Liu et al., 2023; Li et al., 2023a). We emphasize that such works, including ours, do not aim to unveil accurate justifications for the success of large-scale models; instead, the goal is to develop mechanistic and behavioral hypotheses that can be useful to develop grounded theories or tools to capture relevant phenomenology seen in large-scale models, hopefully leading to progress on characterizing the models at-scale themselves (Lieberum et al., 2023; Wu et al., 2023; Eldan and Li, 2023).

Appendix C. Discussion

In this work, we explored whether transformers are capable of generalizing to a combinatorial or exponential number of functions not present in the training data by exploiting the compositional structure in data. We conducted a systematic study of compositionality using synthetic data and provide credence to the hypothesis that languages models could generalize to novel compositions of functions not seen in the training data through compositionality.

Understanding transformers Our work raises questions about why transformers exhibit compositionality. While we find preliminary evidence for the importance of attention layers, future directions include further mechanistic interpretability analysis to elucidate the circuit-level mechanisms that drive compositionality. Another unanswered question in this work is a precise understanding of which types of functions can transformers learn to compose, both through direct composition and through step-by-step composition. We find that Transformers with direction compositions are able to compose bijections and permutations but not bijections with other bijections and it would be interesting to understand why this is the case. Apart from studying different functions, another related direction for exploration is to verify if compositionality is seen across different prompt formats, like the one used in Garg et al. (2022).

Caveats to using Synthetic data Synthetic data offers a promising approach to quickly (and cheaply) falsify or verify potential hypotheses. It allows us to work with a precise set of claims since the properties of the data are interpretable and controllable unlike natural language data. However, synthetic setups also present a number of challenges that need to be acknowledged. The primary challenge is to build faithful setups that correctly reproduce the phenomena observed at scale.

Implications for large language models The fact that Transformers can potentially generalize to combinatorial or exponentially many functions implies that language models are capable of doing the same. However, it is hard to characterize compositionality in natural language data which makes it hard to verify that language models exhibit compositional behavior of any kind. Overall, this study establishes a foundational framework for further precise and systematic studies of compositional generalization in Transformer-based autoregressive models.

Appendix D. Experimental Details

Data The inputs contains a sequence of elements from a vocabulary X of size 10. Each input $x \in X^6$ is a sequence of 6 elements. When generating prompts, x is generated by sampling 6 elements from X uniformly at random and with replacement. Similarly, the sequence of task-tokens are draw uniformly at random from the set of functions seen during training.

The training data consists of 100,000 sequences for all training datasets. Each training data point is generated according to the format described in section 2.2.

When evaluating the trained Transformers, we evaluate on 100 different inputs for every single function. Since fig. 6 requires us to evaluate on a combinatorial set of functions, we sampled 1000 functions (or the total number of functions, whichever was lower) from each cell and compute the accuracy averaged of those functions to populate the cell.

Transformer architecture We train nanoGPT using an auto-regressive next-token prediction objective on the entire sequence. The transformer consists of 12 layers and 12 heads with an embedding dimension of 120. The input is tokenized to be a one-hot vector and the context size is at most 64. The model makes use of no dropout and no biases in the Layer norm layers. Finally, we make use of mixed-precision training (bf16 in torch) to speedup training.

Optimizer The transformers are typically trained for 100 epochs with a cosine-annealed scheduled with warmup. We use an initial learning rate of 3e-4 annealed eventually to 6e-5. We use AdamW as the optimizer with weight decay 1e-3 and a batch-size of 512. We also make use of gradient clipping with a magnitude of 1.

Linear probe Due the weight tying, the initial and final layer of the transformer share their weights. We make use of these weights to compute the linear probe accuracy after every attention and MLP layer. Note that the linear probe is applied to the output of the attention or MLP layer summed with the output from the residual connection.

Appendix E. Additional Experiments

E.1. In-order vs. Out-of-order generalization

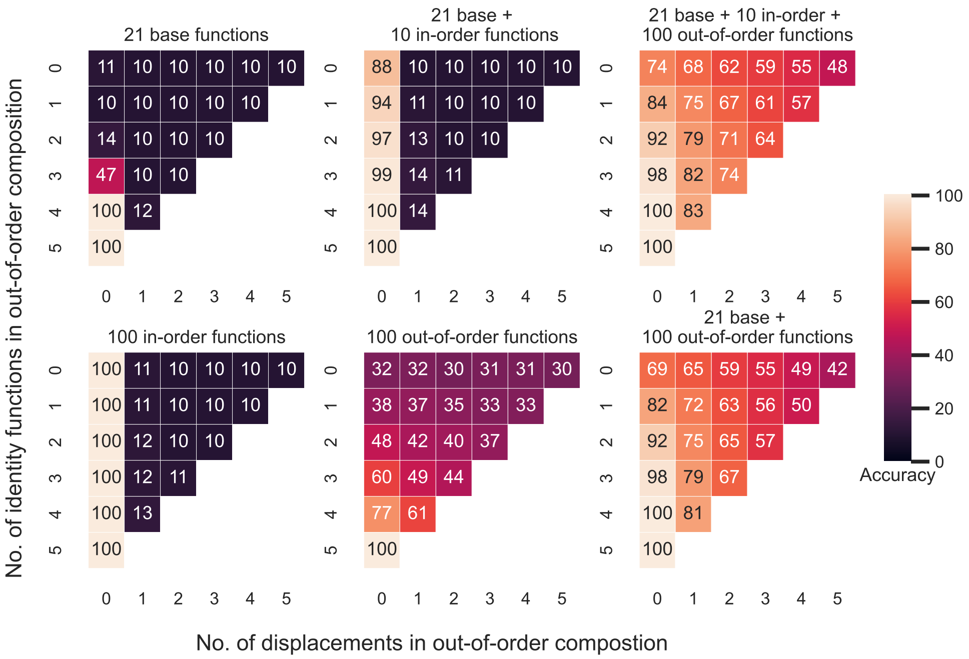


Figure 6: **The training data determines if the Transformer generalizes to an exponential (in-order generalization) or combinatorial (out-of-order generalization) number of functions.** Each sub-plot uses a different subset of bijections to generate the training and evaluate them on combinatorial set of functions generated from 20+1 functions (one of them being identity). The x-axis varies the number of levels of compositions that consider out-of-order functions and the y-axis varies the number of compositions – equivalently the number of functions that aren't identity. We make the following observations: (1) A Transformer trained on just 31 functions (top-middle) generalize to nearly exponentially many or 3125 compositions of functions. (2) All the above configurations do not generalize perfectly to the entire combinatorial set. They however partially generalize to nearly 4 million compositions of functions. The generalization is worse if we decrease the number of identity functions or the number of displacements in the out-of-order composition. (see fig. 4 for pictorial description of displacements)

Are Transformers capable of in-order and out-of-order generalization and how does it depend on the nature of training data? For the functions in fig. 2(a), the number of in-order compositions is $5^5 = 3125$ and the number of out-of-order compositions is a whopping $(21)^5 = 4084101$; most of these functions are different from the ones seen in the training data. Like in section 3, we only consider Transformers trained with the step-by-step prompt format.

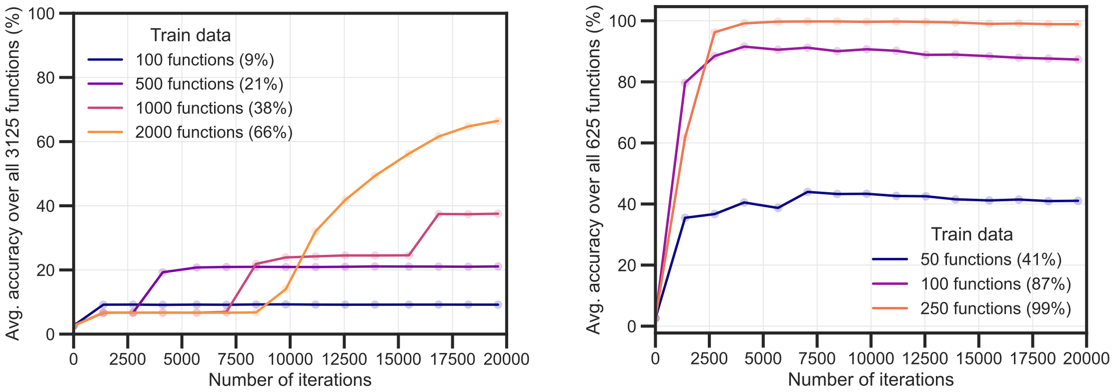
In fig. 6, we consider the training data to have functions from **21 base**, some in-order and out-of-order compositions. We fail to see in-order or out-of-order generalization unless the data also includes in-order or out-of-order compositions respectively. **However, a small number of in-order (10 of them) or out-of-order compositions (100 of them) in the training data is enough for in-order generalization and limited out-of-order generalization.**

Finally, all scenarios in fig. 6 do not fully generalize to out-of-order compositions. This indicates that out-of-order compositions may require a lot more data compared to in-order compositions.

E.2. Different types of compositions

Both section 3 and appendix E.1 use step-by-step compositions but do these results also hold for direct composition? Figures 3 and 11 answer this question in the negative.

In fig. 3, we consider the setup identical to fig. 2(a) and train on a different number of **random** functions. **Transformers fail to generalize to new in-order compositions with direct compositions when we consider compositions of bijections**. We observe this failure even if we train of 2000 of the 3125 possible in-order compositions of functions. In contrast, in fig. 2(a), 100 compositions in the step-by-step format suffices to generalizes to all possible in-order compositions of functions.



(a) (b)

Figure 7: **Compositional generalization is less frequently seen in the direct prompt format compared to the step-by-step prompt format. (a)** We train a Transformer on 20+1 bijections with 5 levels of compositions with 4 choices at each level. The Transformer fails to generalize to all 3125 compositions even if it trained on 2000 such functions. **(b)** We train the Transformer on a composition of two functions, with one function being one of 25 bijections and the other function being one of 25 permutations (totalling to 625) compositions. The Transformer is able to compose previously unseen combinations of functions when trained on 250 of these functions in this scenario.

On the other hand, we see in-order generalization if the Transformer is trained on a composition of a bijection and a permutation. In fig. 7(b), we train on a composition of one of 25 bijections, with one of 25 permutations and find 250 compositions in the training data is enough for the Transformer to generalize to all 625 possible compositions of the bijections and permutations. We note that bijection and permutations operate on orthogonal features of the input: bijections operate on the value of the token while permutations operate on the position of the token.

Direct compositions occur less frequently compared to step-by-step compositions and this could be indicative of why chain-of-thought is a popular prompting strategy (Nye et al., 2021; Wei et al., 2022b). A precise answer when such direct compositions succeed or fail remains unclear though.

Why is out-of-order generalization harder for direct compositions? We believe that direct compositions are unlikely to generalize to the out-of-order compositions or at least require more samples. For example, consider functions f and g and consider a Transformer that computes the function $g \circ f$. Since $g \circ f$ is computed using a single forward pass through the transformer for direct compositions, g must occur in a layer after f (as shown in fig. 5(b)). As a result, the Transformer cannot generalize to $f \circ g$ since f occurs after g in the layers of the transformer. Hence, the Transformer may have to learn copies of f and g at multiple layers of the transformer if it is going to generalize to $f \circ g$ and $g \circ f$.

E.3. Analyzing trained Transformers

Training dynamics In fig. 8, we consider a fine-grained version of fig. 2(a) to understand if the Transformer generalizes to composition of fewer functions before it generalizes compositions of many functions. We find that the answer depends on the nature of the training data.

If the training data consists of **21 base** and very few in-order compositions, then the Transformer generalizes to fewer compositions (more identities) first before generalizing to compositions of multiple functions. On the other hand, if the Transformer is trained on 25 **random** in-order compositions, then the Transformer is better at generalizing to more complex compositions of these functions; this trend is lost when we train on 50 **random** in-order compositions.

Linear probe In fig. 9, we use a linear probe (frozen to the last linear layer of the transformer) to analyze the importance of attention layers and contrast them with the MLP layers. We use a Transformer trained on 100 **random** in-order compositions of 5 functions identical to the model in fig. 2(a). We find that attention in layers 5-10 have high linear prove accuracy and we hypothesize that they are critical to compositionality in Transformers.

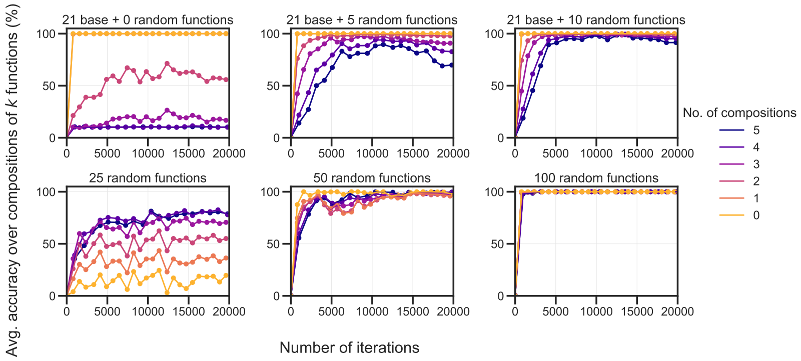
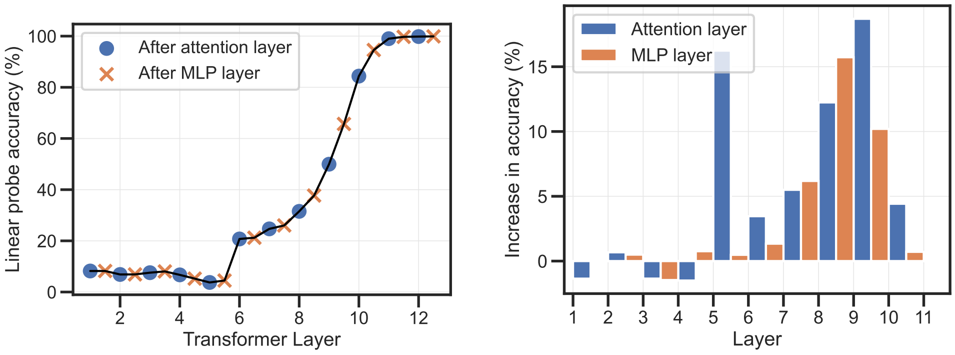


Figure 8: **A Transformer trained on a random subset of functions generalizes first to a composition of more functions before it generalizes to a composition of few of them.** Each line is the average accuracy over all composition of k functions and each subplot is a Transformer trained on a different subset of functions. The **21 base** is trained on the individual functions and these Transformers learn to compose a smaller set of functions (more functions in composition are identity) before learning to compose many of them. The opposite is true when the Transformer is trained on a random subset of 25 compositions of functions.



(a)

(b)

Figure 9: **Attention layers between layers 6-10 see a high increase in probe accuracy, hinting at its importance in compositional generalization.** We compute the linear probe accuracy — averaged over all in-order compositions of functions — after the MLP and attention layers at every layer of the Transformer. **(a)** The accuracy increases sharply at layers 6 and around layers 8-10. **(b)** The increase in accuracy shows that the MLP is important around layer 9-10. The attention layers have a larger contribution to an increase in accuracy. While these plots are not conclusive evidence, they indicate that the attention is important for compositional generalization.

Appendix F. Additional Analysis

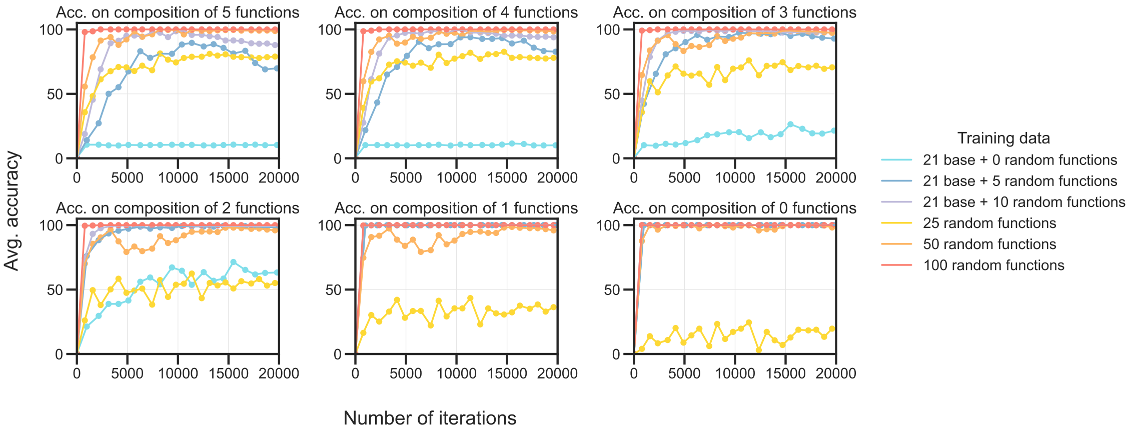


Figure 10: This is a fine-grained version of fig. 2(*a*). Model trained on 50 **random** compositions generalizes poorly compositions of small number of functions while a model trained on the **21 base** generalizes poorly to composition of 4 or 5 functions.

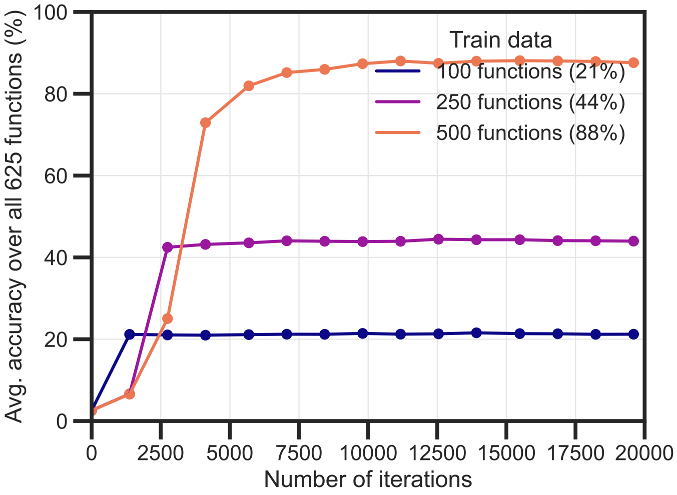


Figure 11: **Another failure of direct compositions.** The curve depicts the accuracy over all 625 in-order compositions of two bijections (25 choices for each bijection) when trained on different subsets of in-order compositions. The model is trained with direct composition. Even if we train on 500 such compositions, the model fails to generalize to the remaining 125 compositions. This is additional evidence that the model is incapable composing bijections through direction composition.

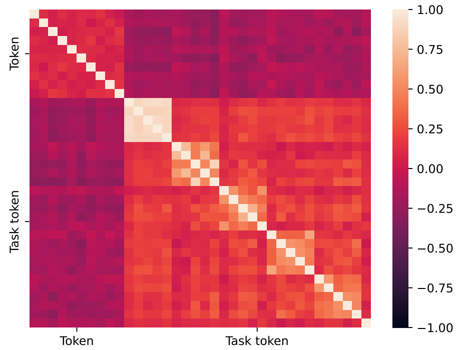


Figure 12: We plot the inner product between all pairs of word embeddings of the tokens. The task tokens are orthogonal to the set of input tokens. Different functions in the same level, i.e. $\{f_i^l\}_{i=1}^N$ for a fixed l, form a block-diagonal in theis matrix.

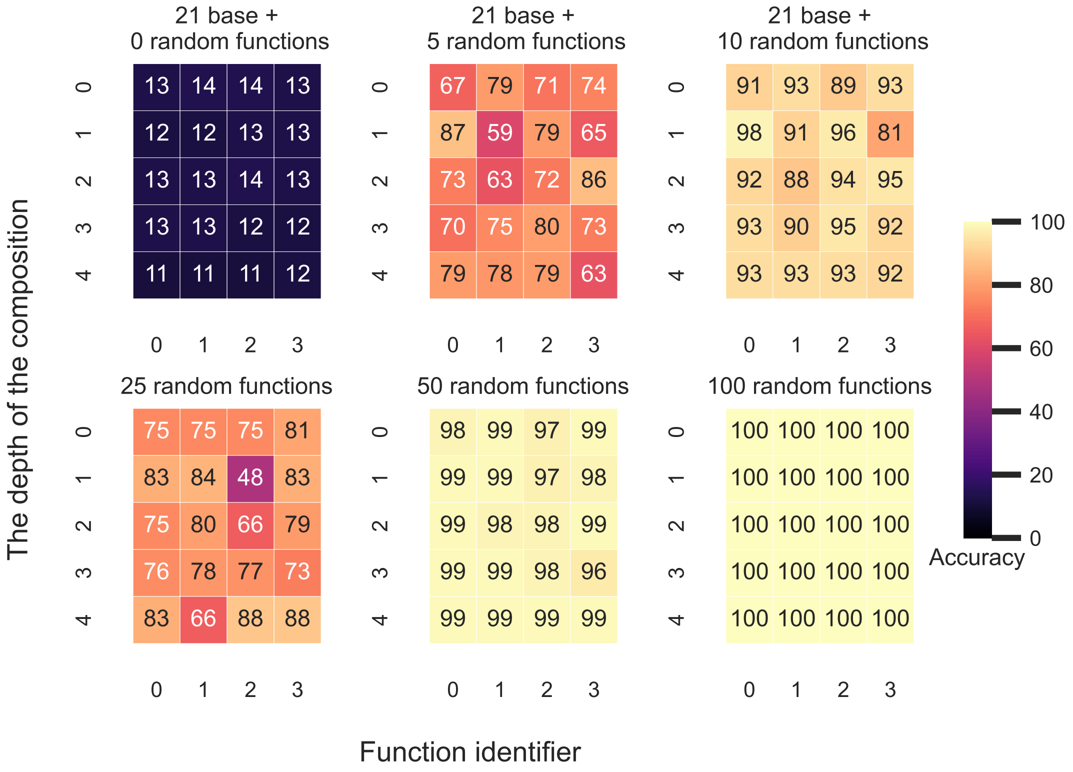


Figure 13: **Systematicity.** We consider trained models from fig. 2(*a*). We analyze the accuracy of each of the 20 functions when averaged all instances in which a particular function was used to compute some composition. Models typically learn all functions equally well.

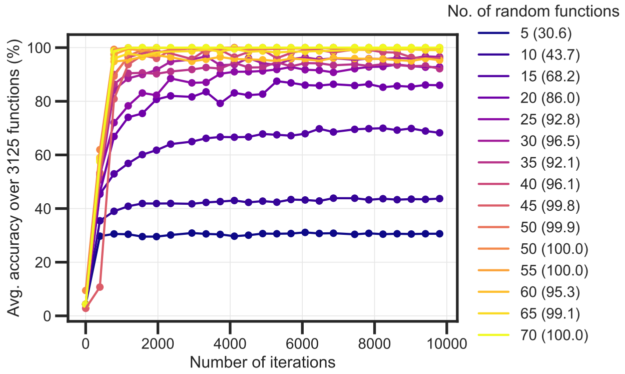


Figure 14: **Training different numbers of random functions.** We train on a different number of random functions ranging from 5-70 in steps of 5. These plots are the accuracies averaged over all in-order compositions of 5 bijections over the course of training.

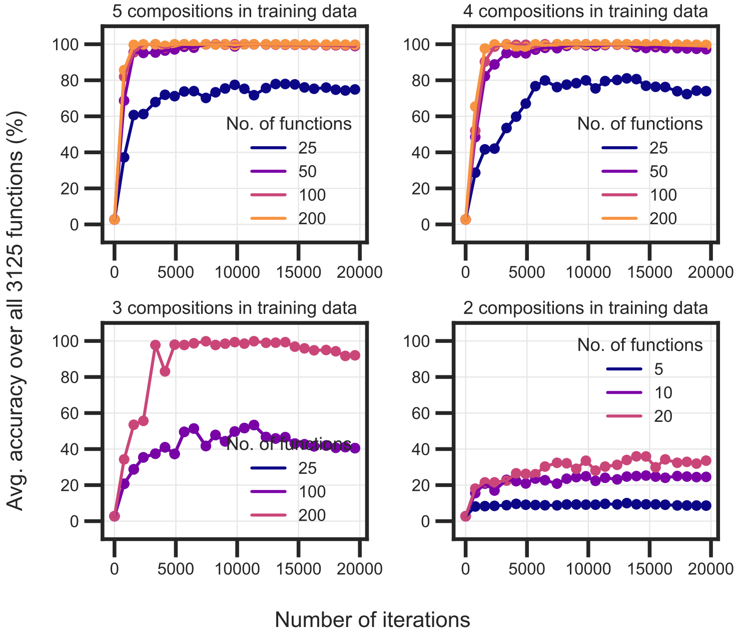


Figure 15: **Number of compositions in the data.** The figure plots the accuracy on all in-order compositions against the number of training iterations. Each sub-plot considers compositions of size exactly 2, 3, 4, 5, respectively in the training data. The model is able to generalize to most in-order compositions only if the training data consists of compositions of size at least 3 (bottom-right).

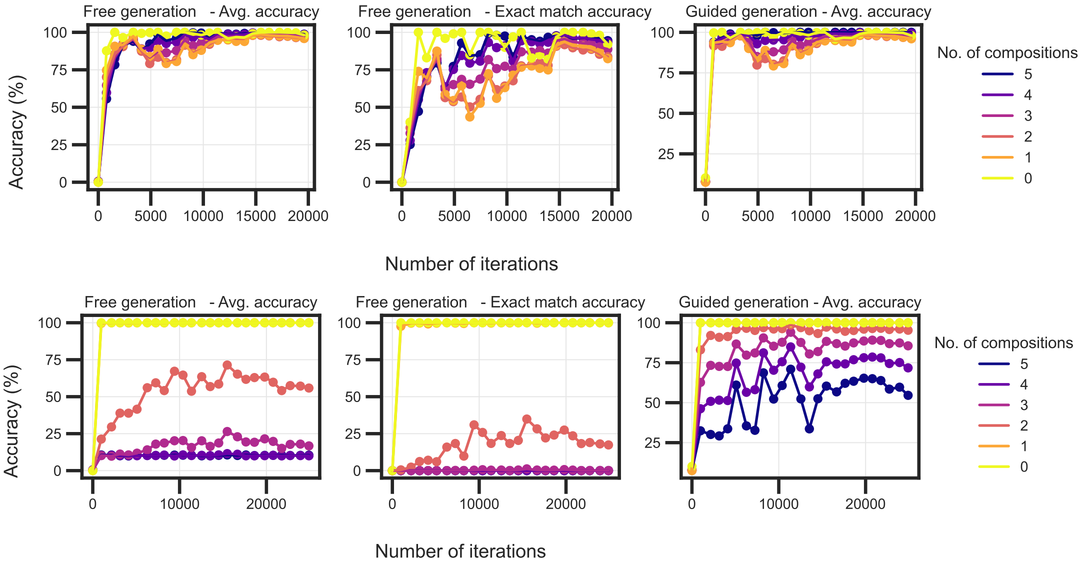


Figure 16: We consider 3 different metrics for evaluating the models. The left column considers the average accuracy when the model generates **The choice of metric doesn't change qualitative trends.** Each sub-plot considers compositions of only size 2, 3, 4, 5, respectively. In each plot, we vary the number of such functions that are present int he training data. **One exception is when we train on compositions of size 2.** In this case, the guided generation accuracy is high, but the free generation accuracy is not.

Extended Abstract Track