Grokking

Tim Kolber [1]

[1]Heidelberg University

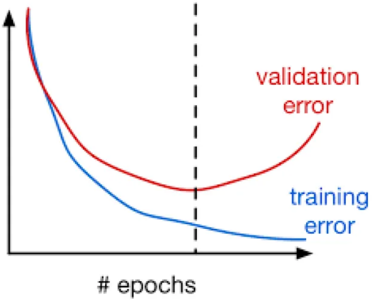
January 20, 2025

Introduction

- *Grokking* comes from *Grok*
- *Grok* is a word made up by Robert A. Heinlin for a science-fiction novel in 1961.
- Means as much as to fully and deeply understand something.

Introduction

In machine learning it describes the phenomenon where models generalize a long time after reaching perfect training accuracy.



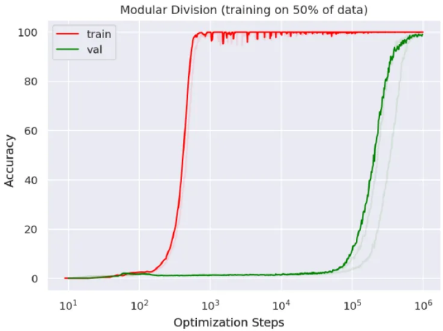
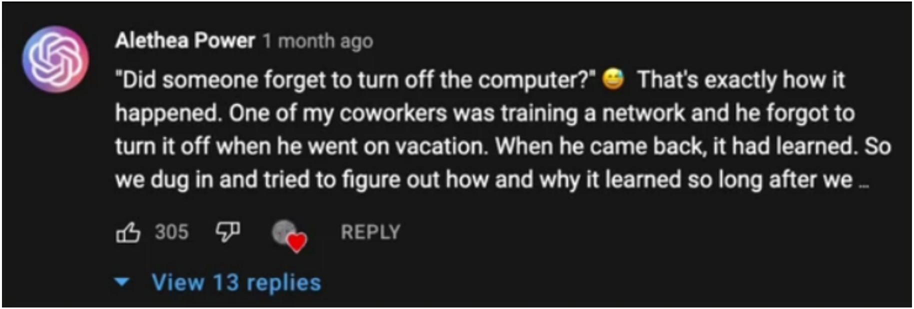


Figure: Left: "normal" model training, Right: Grokking (delayed generalization)

Introduction

Discovered by Power et al. [2022].



Paper Overview

1. **Towards Understanding Grokking: An Effective Theory of Representation Learning** [Liu et al., 2022]

2. **Progress Measures For Grokking Via Mechanistic Interpretability** [Nanda et al., 2023]

Toy Algorithmic Datasets

Task: Learning binary operations such as $a + b$ or $a + b \mod P$.

★	a	b	c	d	e
a	a	d	?	c	d
b	c	d	d	a	c
c	?	e	d	b	d
d	a	?	?	b	c
e	b	b	c	?	a

First paper

Towards Understanding Grokking: An Effective Theory of Representation Learning [Liu et al., 2022]

Research Questions

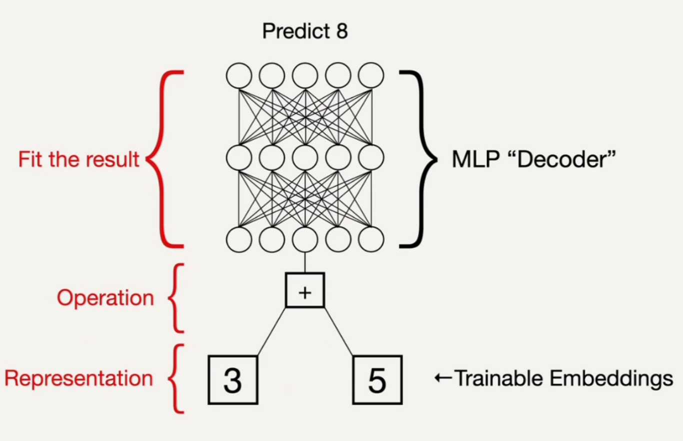
1. The origin of generalization: When trained on the algorithmic datasets where grokking occurs, how do models generalize at all?
   Representation learning
2. The critical training size: Why does the training time needed to "grok" (generalize) diverge as the training set size decreases toward a critical point?
   Training size controls speed of representation learning
3. Delayed generalization: Under what conditions does delayed generalization occur?
   Improper hyperparameters prohibit representation learning.

Setup

- **Input**: Symbols i,j $\in \{0, \dots, p-1\}$
- Mapped to learnable embedding vectors $\mathbb{E}_i, \mathbb{E}_j \in \mathbb{R}^{d_{in}}$
- Sum $\mathbb{E}_i, \mathbb{E}_j$, send result through decoder MLP
- **Output:** $Y_C \in \mathbb{R}^{d_{out}}$ either fixed random vector (regression task) or one-hot vector (classification task)

$$(i, j) \mapsto Dec(\mathbb{E}_i + \mathbb{E}_j)$$

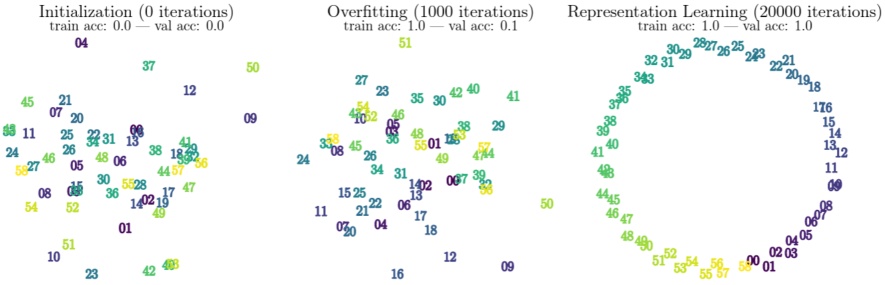
Setup Visualization



How do models generalize?

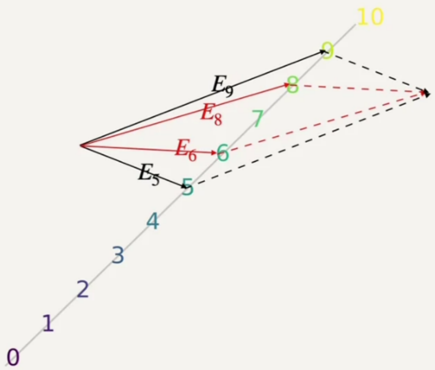
1. The origin of generalization: When trained on the algorithmic datasets where grokking occurs, how do models generalize at all?
   Representation learning

Generalization linked to highly structured embeddings



→ Can we formalize representation quality and use it to predict generalization?

Generalization linked to highly structured embeddings



If $5 + 9 = 14$

is in the train set

then the toy model will generalize to $6 + 8$

Because $E_5 + E_9 = E_6 + E_8$

Definitions

Definition

(i, j, m, n) is a δ-**parallelogram** in the representation $\mathbf{R} \equiv [\mathbf{E}_0, \cdots, \mathbf{E}_{p-1}]$ if

$$|(\mathbf{E}_i + \mathbf{E}_j) - (\mathbf{E}_m + \mathbf{E}_n)| \leq \delta.$$

Representation Quality Index

$$P_0(D) = \{(i, j, m, n) | (i, j) \in D,\ (m, n) \in D,\ i + j = m + n\}.$$

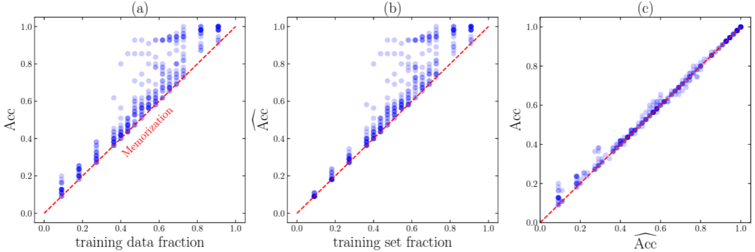
$$P(\mathbf{R}, \delta) = \{(i, j, m, n) | (i, j, m, n) \in P_0, |(\mathbf{E}_i + \mathbf{E}_j) - (\mathbf{E}_m + \mathbf{E}_n)| \leq \delta\}.$$

$$\mathrm{RQI}(\mathbf{R}) = \frac{|P(\mathbf{R})|}{|P_0|} \in [0, 1].$$

Linear representation: RQI$= 1$, random representation: RQI$= 0$

Predicted Accuracy

The predicted accuracy is computed using the training set and the representation R. Are there parallelograms that enable the model to generalize from the training set to the validation set?



Research Questions

2. The critical training size: Why does the training time needed to "grok" (generalize) diverge as the training set size decreases toward a critical point?
   Training size controls speed of representation learning

Effective Loss

Effective loss simplifies training dynamics:

$$\ell_{\text{eff}} = \frac{\ell_0}{Z_0}, \quad \ell_0 \equiv \sum_{(i,j,m,n)\in P_0(D)} |\tilde{\mathbf{E}}_i + \tilde{\mathbf{E}}_j - \tilde{\mathbf{E}}_m - \tilde{\mathbf{E}}_n|^2 / |P_0(D)| \quad ,$$

$$Z_0 \equiv \sum_k |\tilde{\mathbf{E}}_k|^2,$$

$$\frac{d\tilde{\mathbf{E}}_i}{dt} = -\frac{\partial \ell_{\text{eff}}}{\partial \tilde{\mathbf{E}}_i}$$

Theory vs Empirical

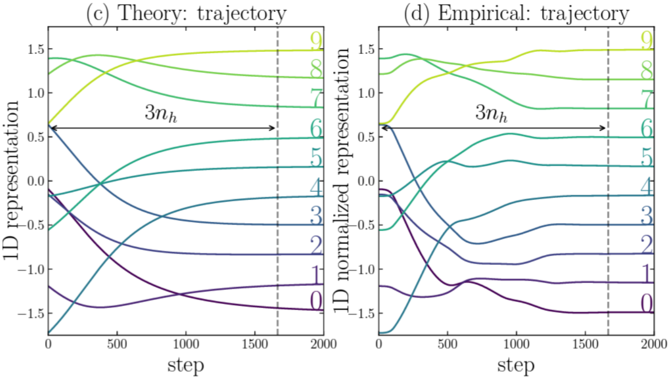


Figure: The 1D representations predicted by the effective theory/obtained from the NN training agree relatively well.

Grokking Rate

- Define Hessian $H_{ij} = \frac{1}{Z_0} \frac{\partial^2 \ell_0}{\partial \mathbf{E}_i \partial \mathbf{E}_j}$ with eigenvalues $\lambda_1 \leq \lambda_2 \leq \lambda_3 \ldots$ with $\lambda_1 = \lambda_2 = 0$
- We can call λ_3 the *grokking rate*, and the grokking time is proportional to $\frac{1}{\lambda_3}$.

Grokking Time

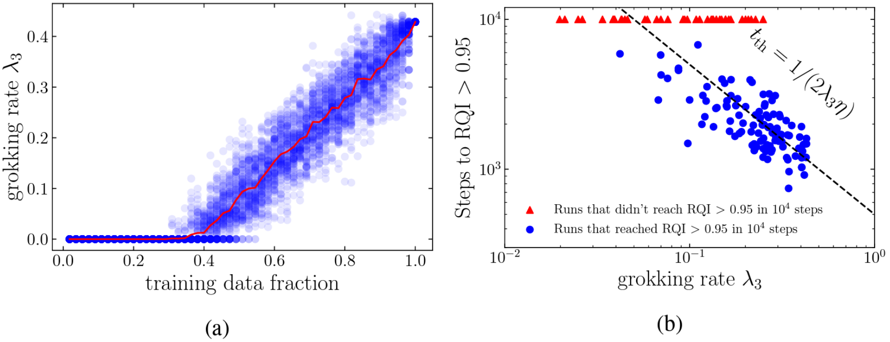


Figure: Training data fraction has a impact on grokking rate and grokking rate has a impact on grokking time.

Research Questions

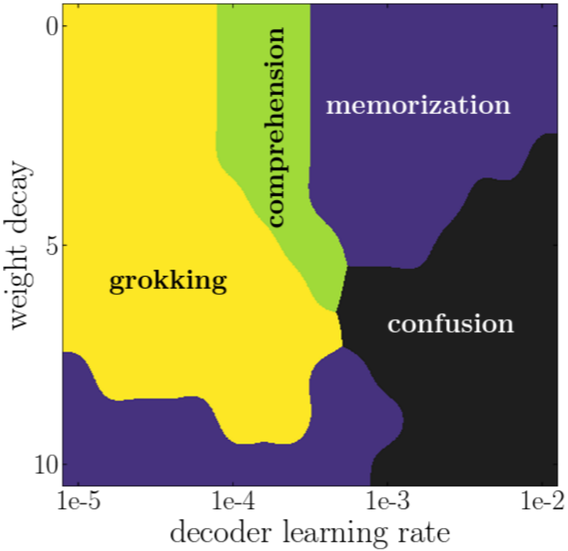
3. Delayed generalization: Under what conditions does delayed generalization occur?
   Improper hyperparameters prohibit representation learning.

Definition of phases

Table 1: Definitions of the four phases of learning

Phase	criteria		
	training acc > 90% within 10^5 steps	validation acc > 90% within 10^5 steps	step(validation acc>90%) −step(training acc>90%)<10^3
Comprehension	Yes	Yes	Yes
Grokking	Yes	Yes	No
Memorization	Yes	No	Not Applicable
Confusion	No	No	Not Applicable

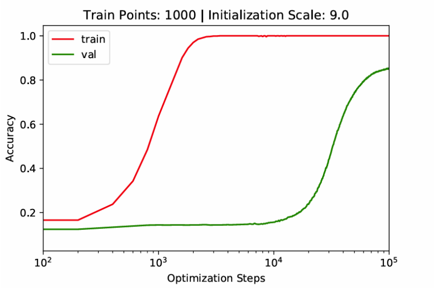
Phase diagrams



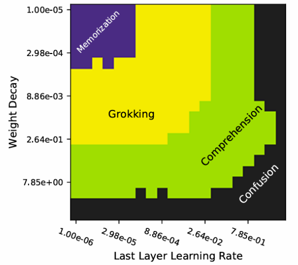
Beyond the toy example: Grokking in MNIST

Figure: MNIST Examples
https://upload.wikimedia.org/wikipedia/commons/thumb/2/27/MnistExamples.png
MnistExamples.png

Beyond the toy example: Grokking in MNIST



(a)



(b)

Conclusion

1. Generalization can be attributed to learning a good (=structured) representation.
2. Developed effective theory of representation learning dynamics (in toy setting) $\rightarrow$ shows dependence of learning on training data fraction
3. Phase diagrams show how learning depends on hyperparameters, which allow control over the grokking effect.
4. Grokking happens when good representations are formed too slowly.

Second paper

- **Progress Measures For Grokking Via Mechanistic Interpretability** [Nanda et al., 2023]

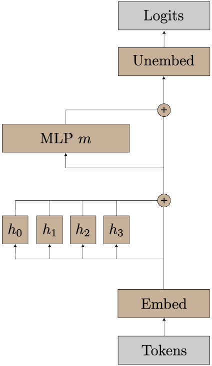
Key Idea

- Solve the mystery of grokking using Mechanistic Interpretability.
- **Hypothesis:** Models learn human-comprehensible algorithms and can therefore be understood if they are made "legible"
- Reverse engineer the learned algorithm by the model.

Setup

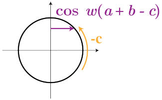
- Trained a 1 layer model to do modular addition ($a + b \mod P$).
- **Input:** $''a\ b =''$
- a and b P-dimensional one-hot-encoded vector.
- $P = 113$

Reverse-engineered Algorithm



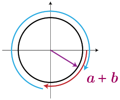
Computes logits using further trig identities:

$$\mathrm{Logit}(c) \propto \cos(w(a+b-c))$$
$$= \cos(w(a+b))\cos(wc) + \sin(w(a+b))\sin(wc)$$



Calculates sine and cosine of $a+b$ using trig identities:

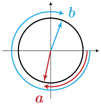
$$\sin(w(a+b)) = \sin(wa)\cos(wb) + \cos(wa)\sin(wb)$$
$$\cos(w(a+b)) = \cos(wa)\cos(wb) - \sin(wa)\sin(wb)$$



Translates one-hot a, b to Fourier basis:

$a \to \sin(wa), \cos(wa)$

$b \to \sin(wb), \cos(wb)$



Evidence

1. Suggestive Evidence: Surprising Periodicity
2. Mechanistic Evidence: Composing Model Weights
3. Zooming In: Approximating Neurons with Sines and Cosines
4. Correctness checks: Ablations

Suggestive Evidence: Surprising Periodicity

1. Suggestive Evidence: Surprising Periodicity

Fourier Transform

Transforms signal/function into its constituent components and frequencies.

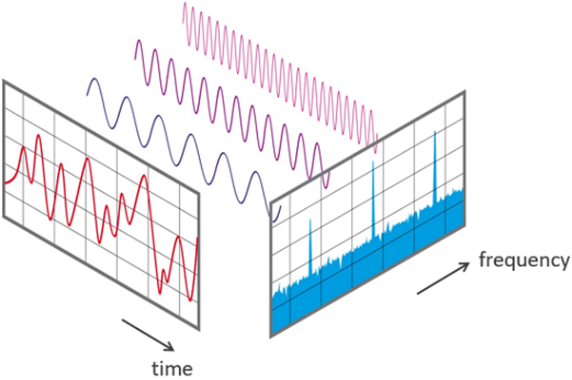


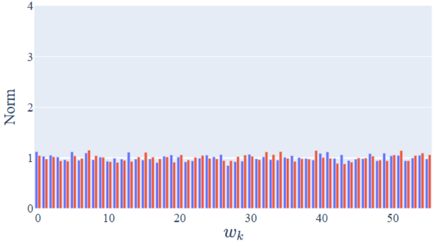
Figure:
https://www.nti-audio.com/en/support/know-how/fast-fourier-transform-fft

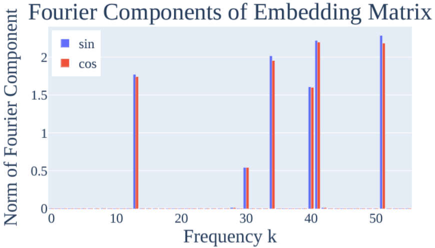
Key Frequencies

- Apply Fourier-transform along the input dimension of the embedding matrix W_E
- Compute ℓ_2-norm along the other dimension.
- W_E sparse in the Fourier basis, 6 frequencies
- The model has learned to embed the different inputs as a linear combination of *sin* and *cosine* terms of 6 frequencies.
- 5 are used throughout the model: $k \in \{14, 35, 41, 42, 52\} \rightarrow$ *key frequencies*

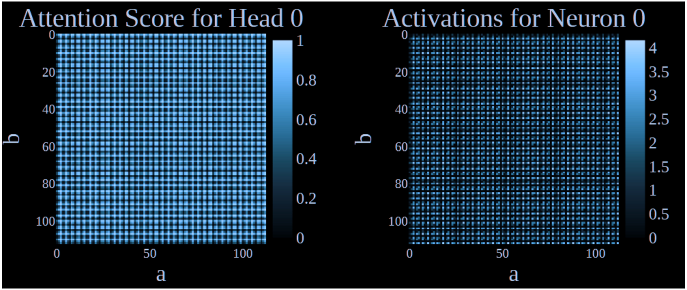
Suggestive Evidence: Surprising Periodicity

Fourier components before and after training. The sparsity of W_E in the Fourier basis is evidence that the network is operating in this basis.





Suggestive Evidence: Surprising Periodicity



Mechanistic Evidence: Composing Model Weights

Mechanistic Evidence: Composing Model Weights

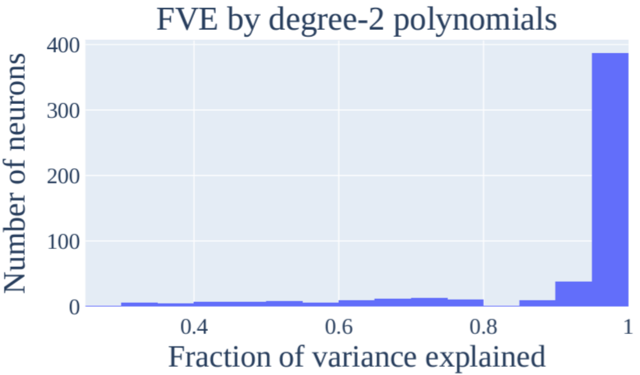
- Logits can be approximated by the sum $\sum_k \alpha_k \cos(\omega_k(a+b-c))$ for $k \in \{14, 35, 41, 42, 52\}$
- α_k coefficients can be fitted using least squares
- Resulting approximation explain 95% of the variance in the original logits.
- Evaluate test loss using this approximation: $\rightarrow$ improvement!

Zooming In: Approximating Neurons with Sines and Cosines

3. Zooming In: Approximating Neurons with Sines and Cosines

Zooming In: Approximating Neurons with Sines and Cosines

Most neurons are well-approximated by degree-2 polynomials of a single frequency.



Correctness checks: Ablations

4. Correctness checks: Ablations

Correctness checks: Ablations

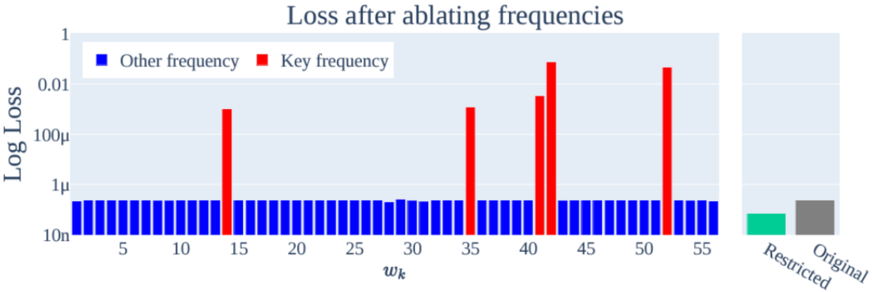


Figure: Ablating key frequencies causes a performance drop, while the other ablations do not harm performance.

Progress measures

- A progress measure is a **smooth** metric that can identify previously hidden progress.
- Goal is to use the mechanistic knowledge gained to derive these measures.

Restricted Loss

- The final network uses a sparse set of frequencies
- Idea: How well does the model do throughout the epochs using only those frequencies?
- Set the terms corresponding to the key frequencies to 0.

Excluded Loss

- Instead of keeping the key frequencies, for the excluded loss they only remove the key frequencies.
- "How much of the performance comes from the algorithm vs. memorization?"

The phases that lead to grokking

The phases that lead to grokking

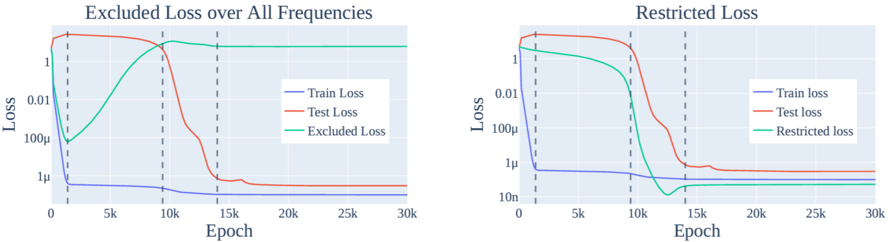


Figure: Memorization, Circuit Formation, Cleanup

Conclusion

- Reverse-engineer the algorithm learned by a model that was tasked with learning modular addition.
- Use this algorithm to derive progress measures that shows the model making continuous progress prior to the grokking phase.

$\rightarrow$ Proof of Concept that mechanistic interpretability can be used to solve machine learning mysteries.

Questions/Comments

- Any questions or comments?

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization Beyond Overfitting On Small Algorithmic Datasets. 2022.

Ziming Liu, Ouail Kitouni, Niklas Nolte, Eric J Michaud, Max Tegmark, and Mike Williams. Towards Understanding Grokking: An Effective Theory of Representation Learning. 2022.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. 2023.