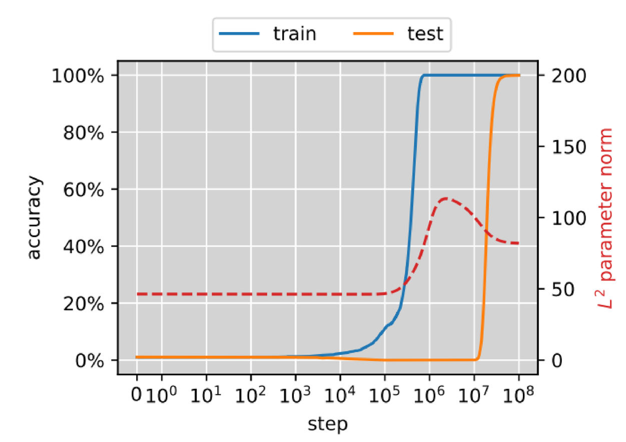
Dichotomy of Early and Late Phase Implicit Biases Can Provably Induce Grokking

Kaifeng Lyu*, Jikai Jin*, Zhiyuan Li, Simon S. Du, Jason D. Lee, Wei Hu

[1]Princeton University [2]Stanford University [3]Toyota Technological Institute at Chicago [4]University of Washington [5]University of Michigan

What is Grokking?



For some algorithmic datasets…

- **Example (Modular Addition):** Given one-hot representations of a, b, output $(a+b) \bmod p$
- **Other Examples:** learning group operations, sparse parity, GCD…

training neural nets leads to grokking (Power et al., 2022):

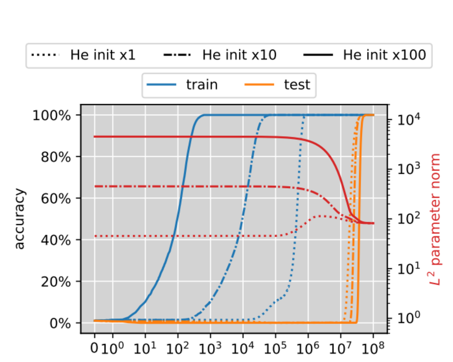
- **Phase 1:** train acc = 100% but test acc is very low
- But after training for sufficiently longer …
- **Phase 2:** test acc "suddenly" jumps to 100%

Many recent works aim to explain grokking: Liu et al. (2023); Varma et al. (2023); Thilak et al. (2022); Notsawo Jr et al. (2023); Nanda et al. (2023); Chughtai et al. (2023); Gromov (2023); Žunkovič & Ilievski (2022); Levi et al. (2024).

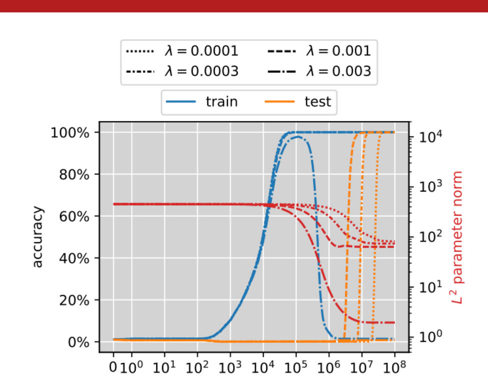
But …

1. No theoretical analysis for neural nets
2. No quantitative explanation for why the transition is sharp in grokking

Ablation Studies: Init Scale and WD



Grokking time is delayed with larger init



Grokking time is delayed with smaller WD

"large init & small WD makes grokking more significant"

Prior work (Liu et al., 2023): With large init and small WD, grokking may happen even on natural tasks (e.g., image/sentiment classification)

Main Results

Our Goal: Give some examples of grokking with precise theoretical characterization.

Classification with L2 Regularization (Weight Decay):

$$\mathcal{L}_\lambda(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i f(\boldsymbol{\theta}; \boldsymbol{x}_i)) + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2$$

We use the exponential loss $\ell(q) = e^{-q}$ for simplicity

Gradient Flow:

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = -\nabla \mathcal{L}_\lambda(\boldsymbol{\theta})$$

(= gradient descent with infinitesimal learning rate)

Large Initialization, Small Weight Decay

"large init & small WD makes grokking more significant"

⇒ Take $\boldsymbol{\theta}(0) = \alpha\bar{\boldsymbol{\theta}}$, $\alpha \to +\infty$, $\lambda \to 0$.

Asymptotics: $\lambda = \Theta(\alpha^{-p})$ for some positive $p = \Theta(1)$.

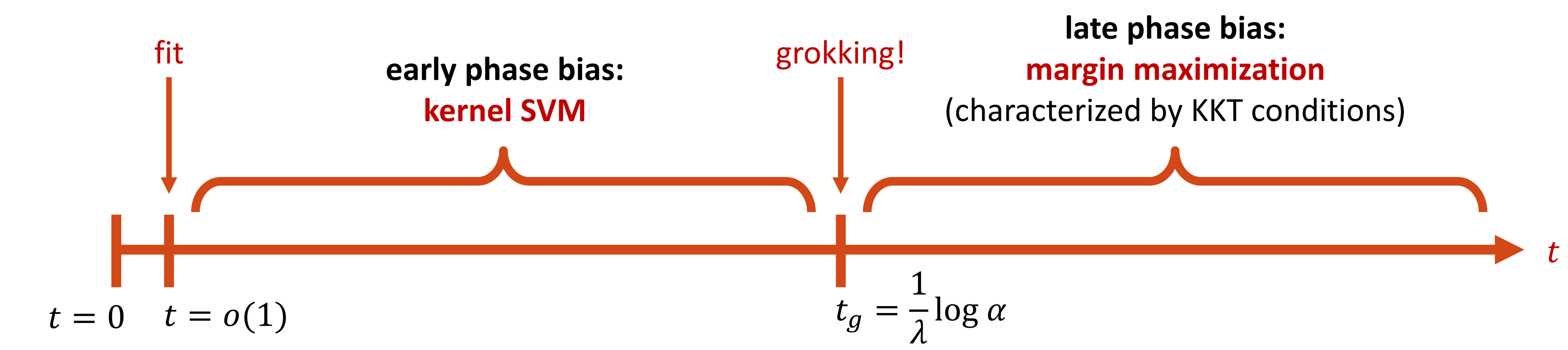
Assumption 1: Homogeneous Nets (Lyu & Li, 2020; Ji & Telgarsky, 2020, …)
The model is L-homogeneous wrt $\boldsymbol{\theta}$:
$f(c\boldsymbol{\theta}; \boldsymbol{x}) = c^L f(\boldsymbol{\theta}; \boldsymbol{x})$ for all $c > 0$

Assumption 2: Smoothness
$f(\boldsymbol{\theta}; \boldsymbol{x})$ is $\mathcal{C}^2$-smooth wrt $\boldsymbol{\theta}$.
(just for simplicity; the proof should be extendable to non-smooth cases)

Assumption 3: Zero initial output
$\bar{\boldsymbol{\theta}}$ satisfies $f(\bar{\boldsymbol{\theta}}; \boldsymbol{x}) = 0$ for all $\boldsymbol{x}$.

Justification: A common assumption for studying NTK. Make the proof much simpler (Chizat et al., 2019; Hu et al., 2020).

- can be done by symmetrized init or "difference trick"
- NTK init + width → ∞ ⇒ approximately true. Proof is usually extendable to this case.



Theorem 1. For $c \in (0,1)$, at $t = (1-c)t_g$, the model outputs the same as the kernel SVM with $K(\boldsymbol{x}, \boldsymbol{x}') = \langle \nabla f(\bar{\boldsymbol{\theta}}; \boldsymbol{x}), \nabla f(\bar{\boldsymbol{\theta}}; \boldsymbol{x}')\rangle$ when $\alpha \to +\infty$.

$$\min \ \|\boldsymbol{h}\|_2^2 \quad \text{s.t.} \quad y_i\langle \nabla f(\bar{\boldsymbol{\theta}}; \boldsymbol{x}_i), \boldsymbol{h}\rangle \geq 1$$

Theorem 2. For $c > 0$, at $t = (1+c)t_g$, the model attains first-order optimal conditions (KKT conditions) for the following margin maximization problem:

$$\min \ \|\boldsymbol{\theta}\|_2^2 \quad \text{s.t.} \quad y_i f(\boldsymbol{\theta}; \boldsymbol{x}_i) \geq 1$$

Key Insights

Kernel regime:

- Unlike the usual kernel regime, where $\boldsymbol{\theta}(t) \approx \boldsymbol{\theta}(0)$, here the parameter itself changes a lot but direction does not, i.e., $\frac{\boldsymbol{\theta}(t)}{\|\boldsymbol{\theta}(t)\|_2} \approx \frac{\boldsymbol{\theta}(0)}{\|\boldsymbol{\theta}(0)\|_2}$.
- Analyze the change in direction and norm very carefully.
- Some calculation ⇒ As long as the norm is not $o\left(\left(\log\frac{1}{\lambda}\right)^{1/L}\right)$, $\boldsymbol{\theta}$ is in the kernel regime.

Rich regime:

- Once the norm decays to this level, only $o\left(\frac{1}{\lambda}\log\frac{1}{\lambda}\right)$ time is needed to reach the KKT.
- Turns out to be very short compared to the time to decay the norm!

Example: Two-layer Diagonal Nets

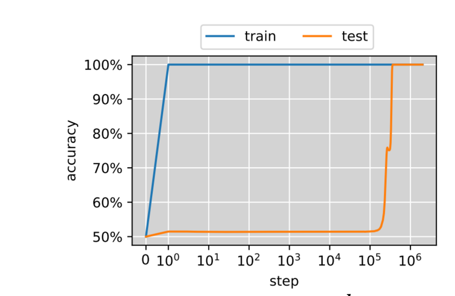
Two-layer Diagonal Net: A reparameterization of linear model (Woodworth et al., 2020)

$$f(\boldsymbol{\theta}; \boldsymbol{x}) = \langle \boldsymbol{u} \odot \boldsymbol{u} - \boldsymbol{v} \odot \boldsymbol{v}, \boldsymbol{x}\rangle, \qquad \boldsymbol{\theta} = (\boldsymbol{u}, \boldsymbol{v}).$$

Alternatively: $\boldsymbol{w} = \boldsymbol{u} \odot \boldsymbol{u} - \boldsymbol{v} \odot \boldsymbol{v}$, $f(\boldsymbol{\theta}; \boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x}\rangle$.

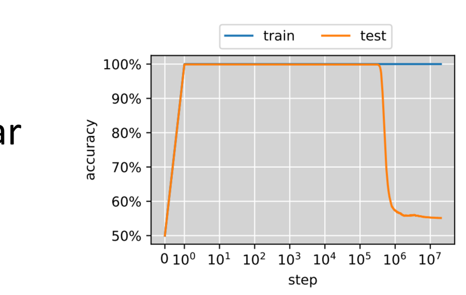
Initialization: $\boldsymbol{u} = \boldsymbol{v} = (1, 1, \dots, 1)$.
Kernel SVM = L2 max-margin linear classifier
Max-margin solution = L1 max-margin linear classifier (encouraging sparsity)

Sparse Linear Regression
⇒ Grokking



$\boldsymbol{x} \sim U(\{\pm 1\}^d)$
$y = \mathrm{sgn}(x_1 + x_2 + x_3)$

Labels are generated by a linear classifier with large L2 margin
⇒ "Misgrokking"



$\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
$y = \mathrm{sgn}(\langle \boldsymbol{z}, \boldsymbol{w}^*\rangle), \boldsymbol{x} = \boldsymbol{z} + \frac{\gamma}{2} y \boldsymbol{w}^*$

Example: Completing Multiplication Tables

(Our results can be extended to regression settings; see our paper)

Overparameterized Matrix Completion:
Parameterize $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{U}^\top - \boldsymbol{V}\boldsymbol{V}^\top$, $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{d \times d}$.
Use MSE loss on observed entries.
Related to learning two-layer nets with quadratic activation.

$$\begin{bmatrix} 1 & ? & 3 \\ ? & ? & 6 \\ 3 & 6 & ? \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & 3 \\ 0 & 0 & 6 \\ 3 & 6 & 0 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{bmatrix}$$

Observed | **Early Phase:** Kernel Regression = Fill in 0 | **Late phase:** Norm minimization = Find a matrix with min nuclear norm (encouraging low-rank)