Toward Demystifying Grokking

Wei Hu
University of Michigan
https://weihu.me/

Deep Learning and Theory

GPT - 4

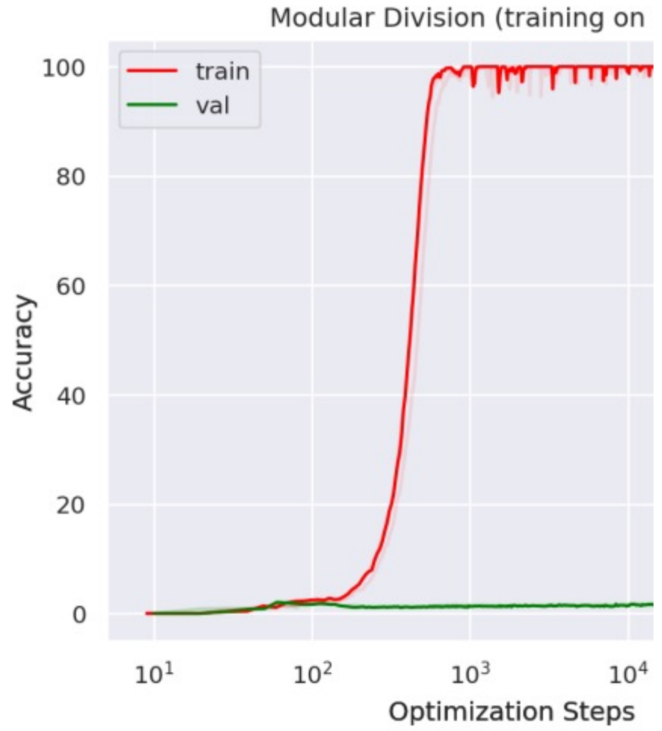
Deep learning has caused theoreticians to ________

GPT-4: reexamine the fundamentals of machine learning…

Neural networks have exhibited a lot of surprising phenomena that defy classical theory

Grokking

"The team member who was training the network went on vacation and forgot to stop the training…"
(https://www.quantamagazine.org/how-do-machines-grok-data-20240412/)



[Power et al. 2022]

★	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

- Originally observed in transformers trained on modular arithmetic problems [Power et al. 2022], e.g., $a + b \bmod p$, $a^2 - b^2 \bmod p$
- Grokking happens for many other tasks (group operations, sparse parity, etc.) and is robust to training methods (architectures, loss functions, optimizers)
- **This talk: theoretical understanding of grokking**

Prior Work

[Liu et al. 2023; Varma et al. 2023]

- Related to parameter norm & initialization scale

[Thilak et al. 2022; Notsawo Jr et al. 2023]

- Related to oscillation in loss curves

[Nanda et al. 2023; Chughtai et al. 2023; Gromov. 2023]

- Found special structure in grokked nets, e.g., sin/cos for modular addition

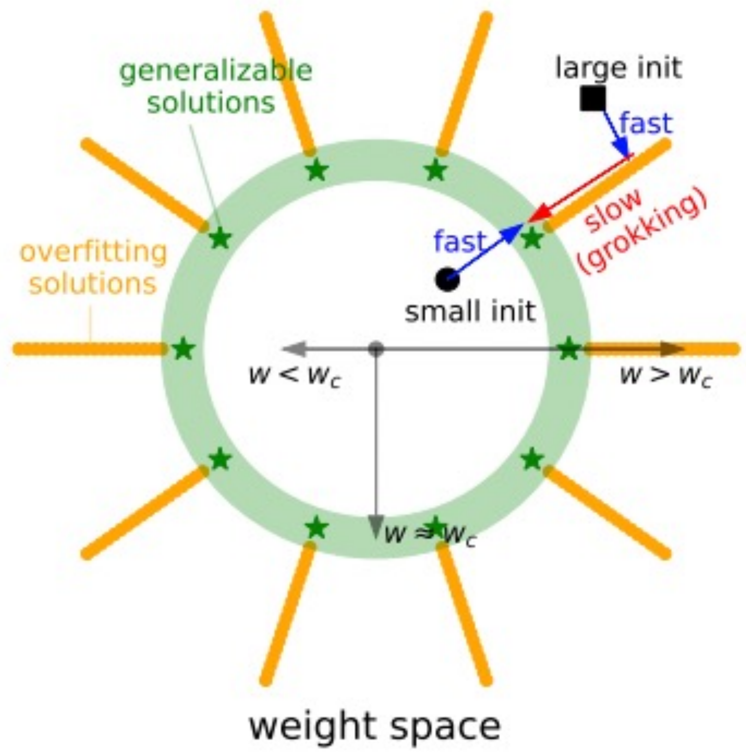
[Žunkovič & Ilievski. 2022; Levi et al. 2024]

- Dynamical analysis for linear models + data assumptions

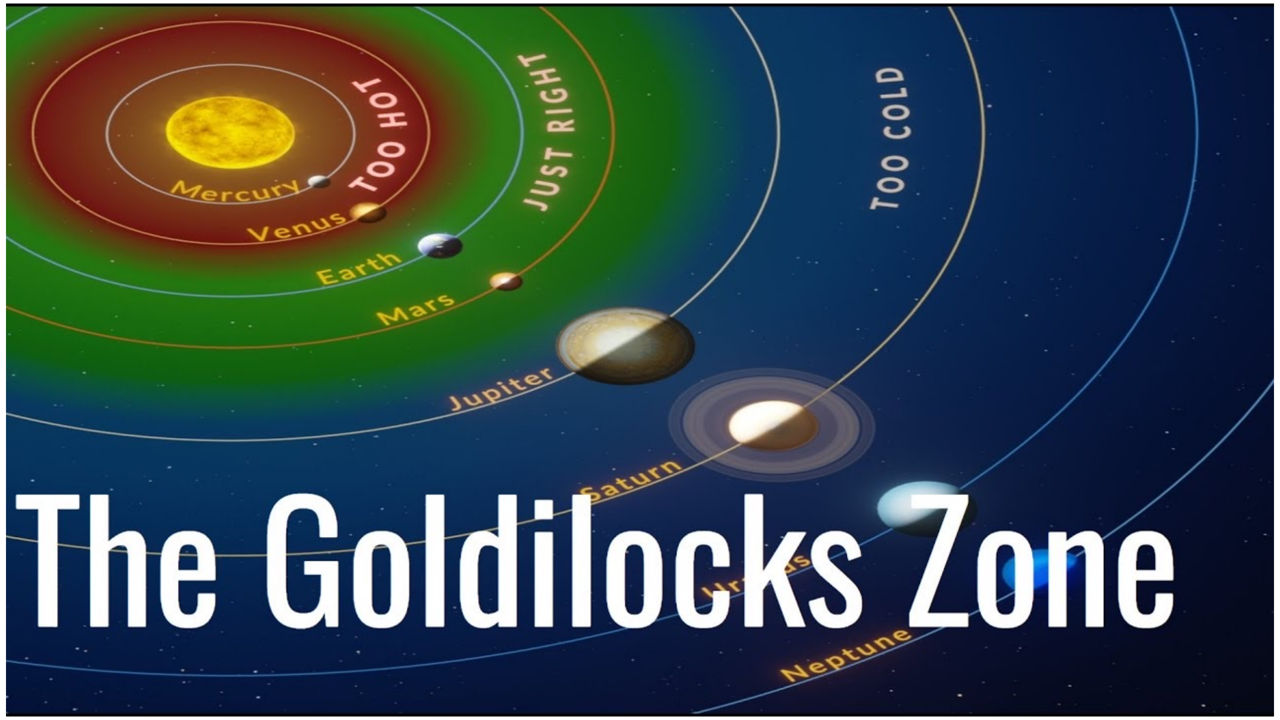
Limitations:

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

Prior Work: Goldilocks Zone

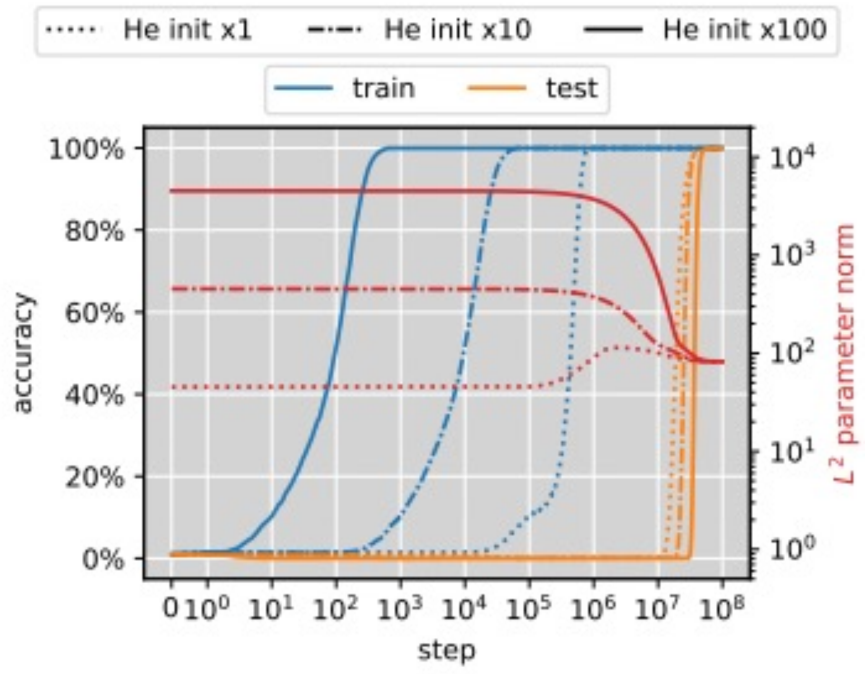


[Liu et al. 2023]

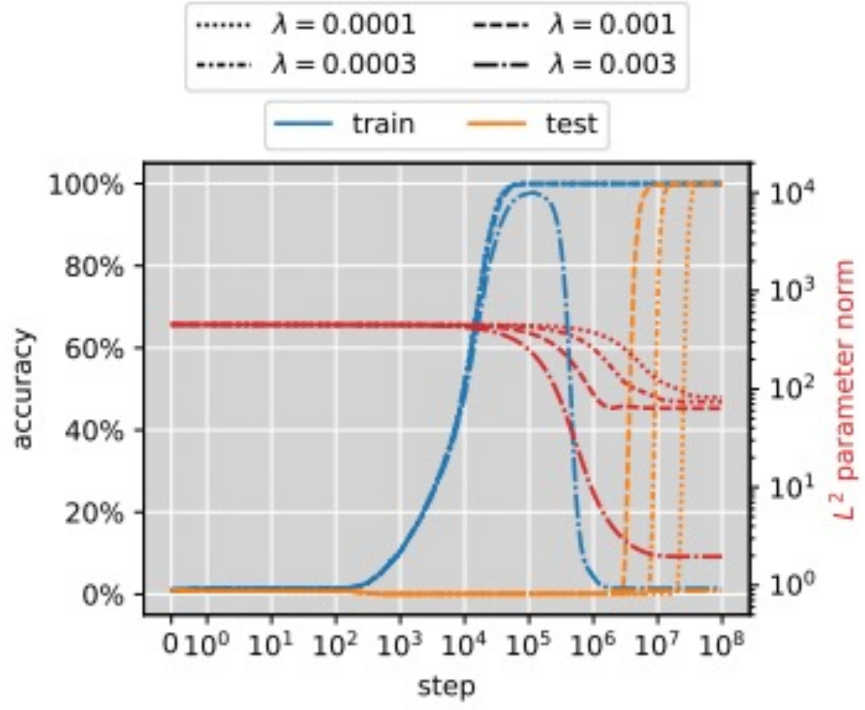


Questions:
1. What determines the norm of the Goldilocks zone?
2. Why is the Goldilocks zone narrow?
3. Many neural networks are homogeneous wrt the weights… Why does the norm matter?

Motivating Experiment



(b) Effect of Initialization Scale

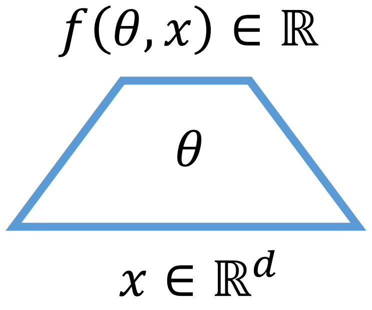


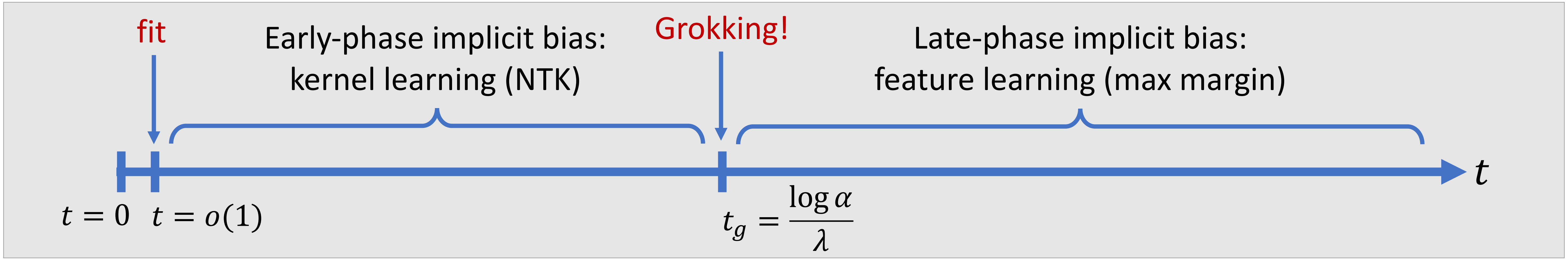
(c) Effect of Weight Decay

Grokking is most prominent with large initialization and small weight decay

Our Result: Dichotomy of Early and Late Phase Implicit Biases

- GD on Logistic loss with L2 regularization/weight decay: $\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i f(\theta, x_i)) + \frac{\lambda}{2}\|\theta\|^2$
- Take limit (large init and small weight decay): $\|\theta_{\text{init}}\| = \alpha \to \infty, \lambda \to 0$





Theorem. At $t = 0.999\frac{\log \alpha}{\lambda}$, the network implements kernel SVM with $K(x, x') = \langle \nabla f(\theta_{\text{init}}, x), \nabla f(\theta_{\text{init}}, x') \rangle$

Theorem. At $t = 1.001\frac{\log \alpha}{\lambda}$, the network attains first-order optimal conditions for margin maximization

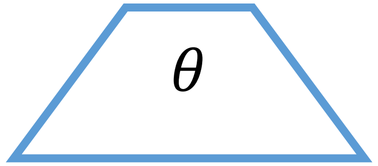
$$\min \ \|\theta\|_2^2 \quad \text{s.t.} \quad y_i f(\theta, x_i) \geq 1, \forall i$$

[Lyu*, Jin*, Li, Du, Lee, H. 2024]

Background: Homogeneous Networks

A neural net is L-homogeneous if $f(c\theta, x) = c^L f(\theta, x)$ for any $c > 0$

- e.g., L-layer ReLU networks and CNNs (without bias terms)



θ: $\boxed{x} \xrightarrow[\text{linear}]{W_1} \boxed{W_1 x} \xrightarrow[\text{ReLU}]{} \boxed{\sigma(W_1 x)} \xrightarrow[\text{linear}]{W_2} \boxed{W_2 \sigma(W_1 x)}$

$c\theta$: $\boxed{x} \xrightarrow[\text{linear}]{cW_1} \boxed{cW_1 x} \xrightarrow[\text{ReLU}]{} \boxed{c\sigma(W_1 x)} \xrightarrow[\text{linear}]{cW_2} \boxed{c^2 W_2 \sigma(W_1 x)}$

"2-homogeneous"

Property: Only the direction of θ matters for classification

Neural Tangent Kernel (NTK) Regime

A series of work: [Jacot et al. 2018; Du et al. 2019; Allen-Zhu et al. 2019; Lee et al. 2019; Arora, Du, H, Li, Salakhudinov, Wang. 2019; Chizat et al. 2019; ...]

Setting 1: Random initialization & width of hidden layers $\to \infty$
Setting 2: large initialization $||\theta_{\text{init}}|| \to \infty$

"Kernel regime":

- Gradient descent converges to global min while the network stays close to its *linearization*:
$$f(\theta, x) \approx f(\theta_{\text{init}}, x) + \langle \nabla f(\theta_{\text{init}}, x), \theta - \theta_{\text{init}} \rangle$$
- Little change in network weights:
$$\frac{||\theta - \theta_{\text{init}}||}{||\theta_{\text{init}}||} \approx 0 \qquad (*)$$
- Neural network at convergence $\Leftrightarrow$ linear model on gradient feature $\phi(x) = \nabla f(\theta_{\text{init}}, x)$ $\Leftrightarrow$ kernel learning with the NTK: $K(x, x') = \langle \phi(x), \phi(x') \rangle$

What's new in our result?

- Due to L2 regularization, the weights will change a lot, so (*) doesn't hold anymore
- We need to carefully track how the norm and direction of θ evolves; turns out the norm will decrease exponentially, but the direction doesn't change much up to time $t = 0.999 \frac{\log \alpha}{\lambda}$

Margin Maximization in Homogeneous Networks

Intention: Hope to understand generalization of deep neural networks using similar ideas as classical linear models like SVMs.

Margin maximization

$$\min \ \|\theta\|_2^2 \quad \text{s.t.} \quad y_i f(\theta, x_i) \geq 1, \forall i$$

Homogeneous model

$$\max \ \min_i y_i f(\theta, x_i) \quad \text{s.t.} \quad \|\theta\| \leq 1$$

cf. SVM: $\min \ \|\theta\|_2^2 \quad \text{s.t.} \quad y_i \theta^\top x_i \geq 1, \forall i$

Prior work [Wei et al. 2019]: If $\lambda \to 0$, the *global minimum* of logistic loss with L2 regularization $\frac{1}{n}\sum_{i=1}^{n} \ell(y_i f(\theta, x_i)) + \frac{\lambda}{2}\|\theta\|^2$ is the max-margin solution.

What's new in our result?

- We precisely characterize the GD convergence time to the max-margin solution: $t = 1.001 \frac{\log \alpha}{\lambda}$
- We only show the first-order optimality (KKT) conditions of margin maximization instead of global optimality

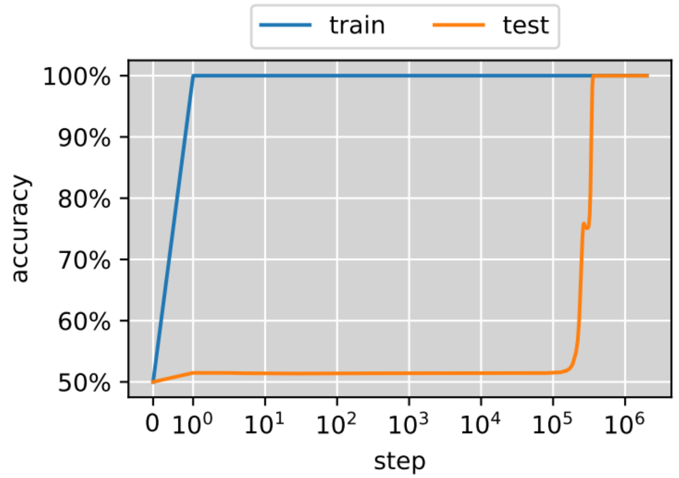
Concrete Example I: Two-Layer Diagonal Linear Network

Two-layer Diagonal Net: A reparameterization of linear model [Woodworth et al. 2020]

- $f(\theta, x) = \langle \theta \odot \theta, x \rangle$

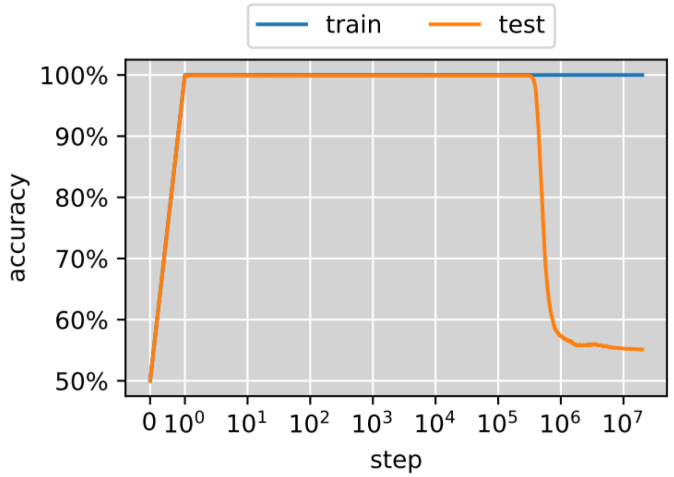
Consider training a two-layer diagonal net on linearly separable data

- Early phase: Kernel SVM $\Longleftrightarrow$ L2 max-margin linear classifier
- Late phase: Max-margin solution $\Longleftrightarrow$ L1 max-margin linear classifier (good for learning sparse models)



$y = \mathrm{sign}(x_1 + x_2 + x_3)$

$\Rightarrow$ grokking



the ground-truth data has large L2 margin

$\Rightarrow$ “misgrokking”

Concrete Example II: Matrix Completion / Multiplication Table

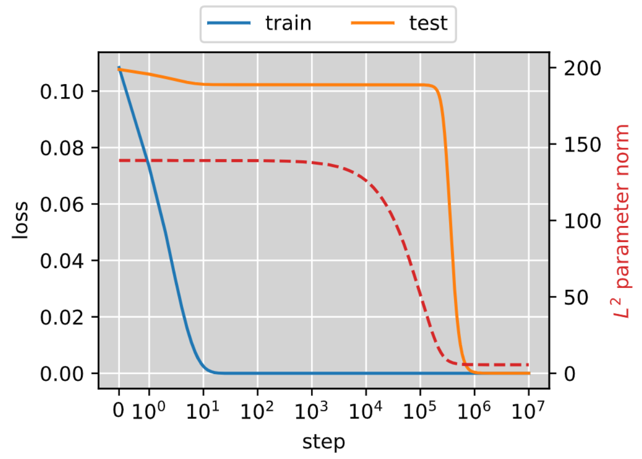
Matrix Completion: given a partially observed low-rank matrix, complete the matrix

- Two-layer model: Parameterize $W = UU^T - VV^T, U, V \in \mathbb{R}^{d\times d}$. Use MSE loss on observed entries.

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

Early phase: NTK

Late phase: min nuclear norm

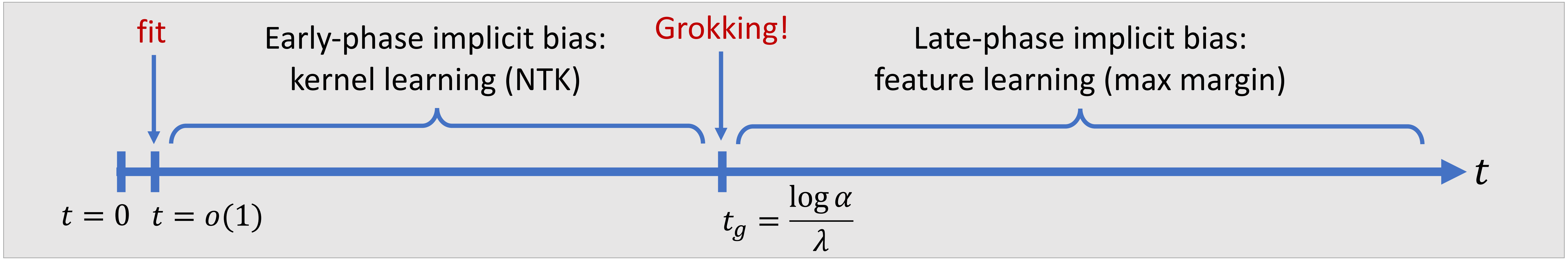


(Extended our result to regression)

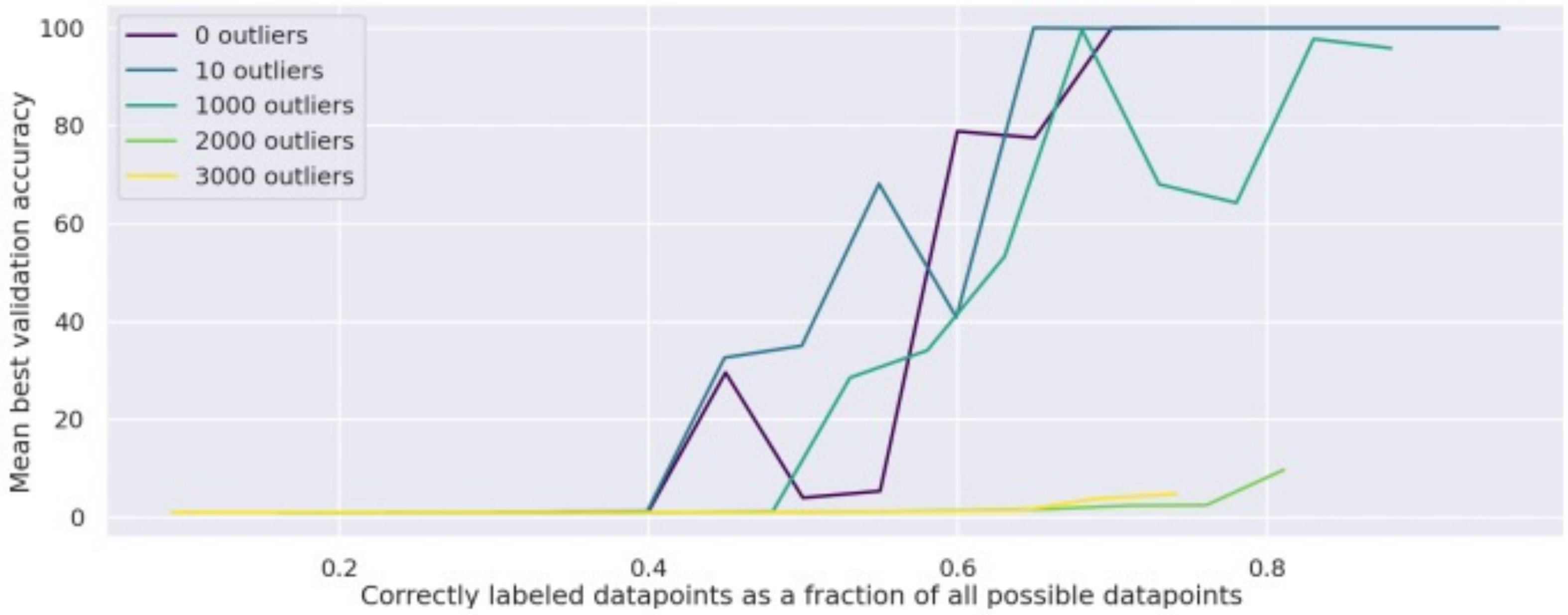
Summary for Part I

A dichotomy of early and late phase implicit biases can provably induce grokking

- Sharp transition from kernel learning to margin maximization



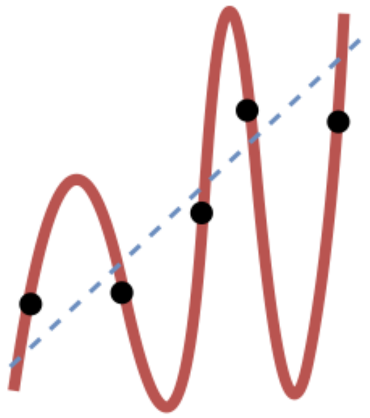
Part II: Generalization While Fitting Noisy Labels



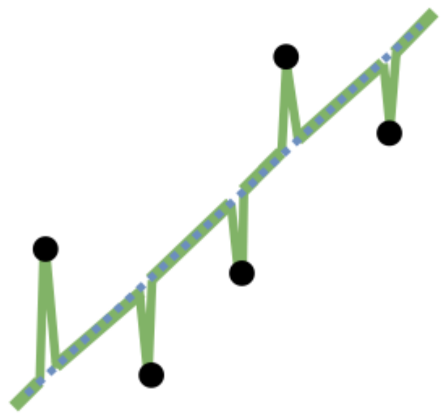
[Power et al. 2022]

The neural network still groks to perfect generalization even if some of the training datapoints have *random labels*

Benign Overfitting



Classical overfitting



Benign overfitting

Benign overfitting: a model interpolates noisily labeled training data, but still achieves near-optimal generalization

- Linear regression and classification: [Bartlett et al. 2020; Chatterji and Long 2021; Wang and Thrampoulidis, 2021; …]
- Neural network trained on linearly separable data: [Frei et al. 2022; Xu and Gu 2023; Cao et al. 2022; Kou et al. 2023; …]
- Open question: Neural network trained on non-linearly separable data???

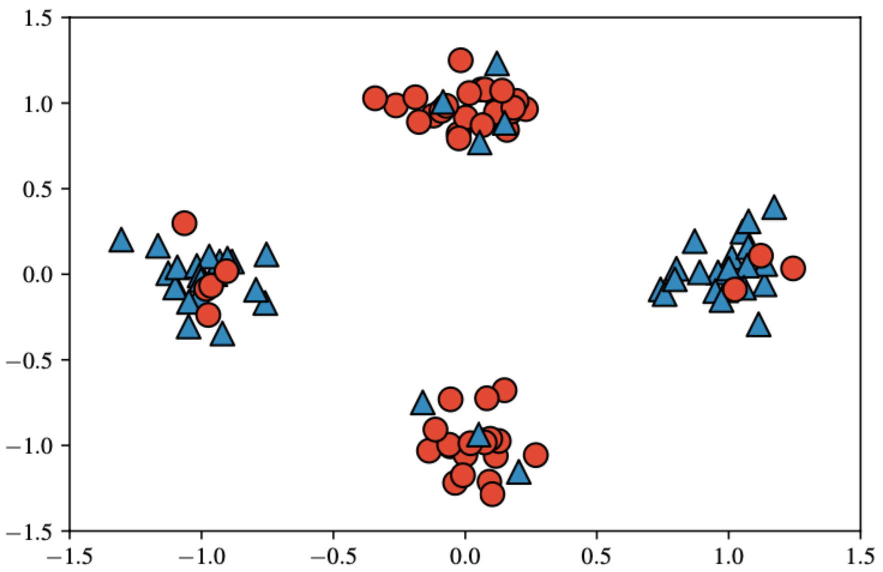
Coming Next

- We characterize a synthetic setting in which benign overfitting and grokking provably occur
  - A two-layer ReLU network trained on XOR cluster data

- The first theoretical characterization of benign overfitting in a truly non-linear setting

XOR Cluster Data

- Binary classification
  - Class $+1$: $\frac{1}{2}\mathcal{N}(\mu_1, I) + \frac{1}{2}\mathcal{N}(-\mu_1, I)$
  - Class -1: $\frac{1}{2}\mathcal{N}(\mu_2, I) + \frac{1}{2}\mathcal{N}(-\mu_2, I)$
  - $\mu_1, \mu_2 \in \mathbb{R}^p$ are orthogonal
- Flip label $y \mapsto -y$ with probability η
- n i.i.d. examples $\{(x_i, y_i)\}_{i=1}^n$



Not linearly separable

Key assumptions (C is a large constant)

- Norm of the mean satisfies

$$Cn^{0.51}p \le \|\mu_1\|^2 = \|\mu_2\|^2 \le \frac{p}{Cn^2}$$

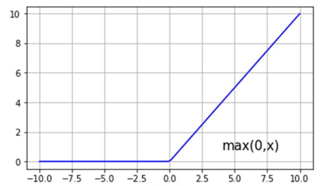
- Label flipping rate satisfies

$$\eta \le \frac{1}{C}$$

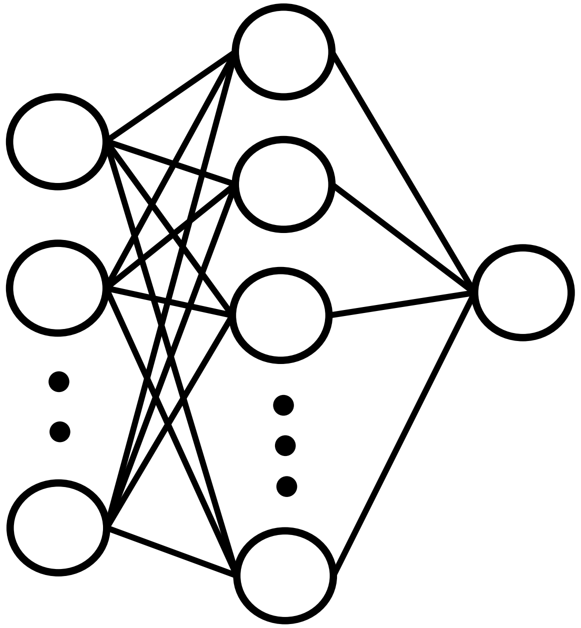
Key property: Near orthogonality

$$\frac{|\langle x_i, x_j \rangle|}{\|x_i\| \cdot \|x_j\|} \ll 1, \qquad \forall i \ne j$$

Two-Layer ReLU Neural Network



ReLU: $\phi(z) = \max\{z, 0\}$



$$f(x; W) = \sum_{j=1}^{m} a_j \phi(\langle w_j, x \rangle)$$

- Second layer $a_j \sim \mathrm{unif}(\pm 1/\sqrt{m})$ i.i.d., not trained
- First layer W is trained by gradient descent on logistic loss:

$$w_j^{(t+1)} = w_j^{(t)} - \alpha \nabla_{w_j} L(W^{(t)}), \qquad j \in [m]$$

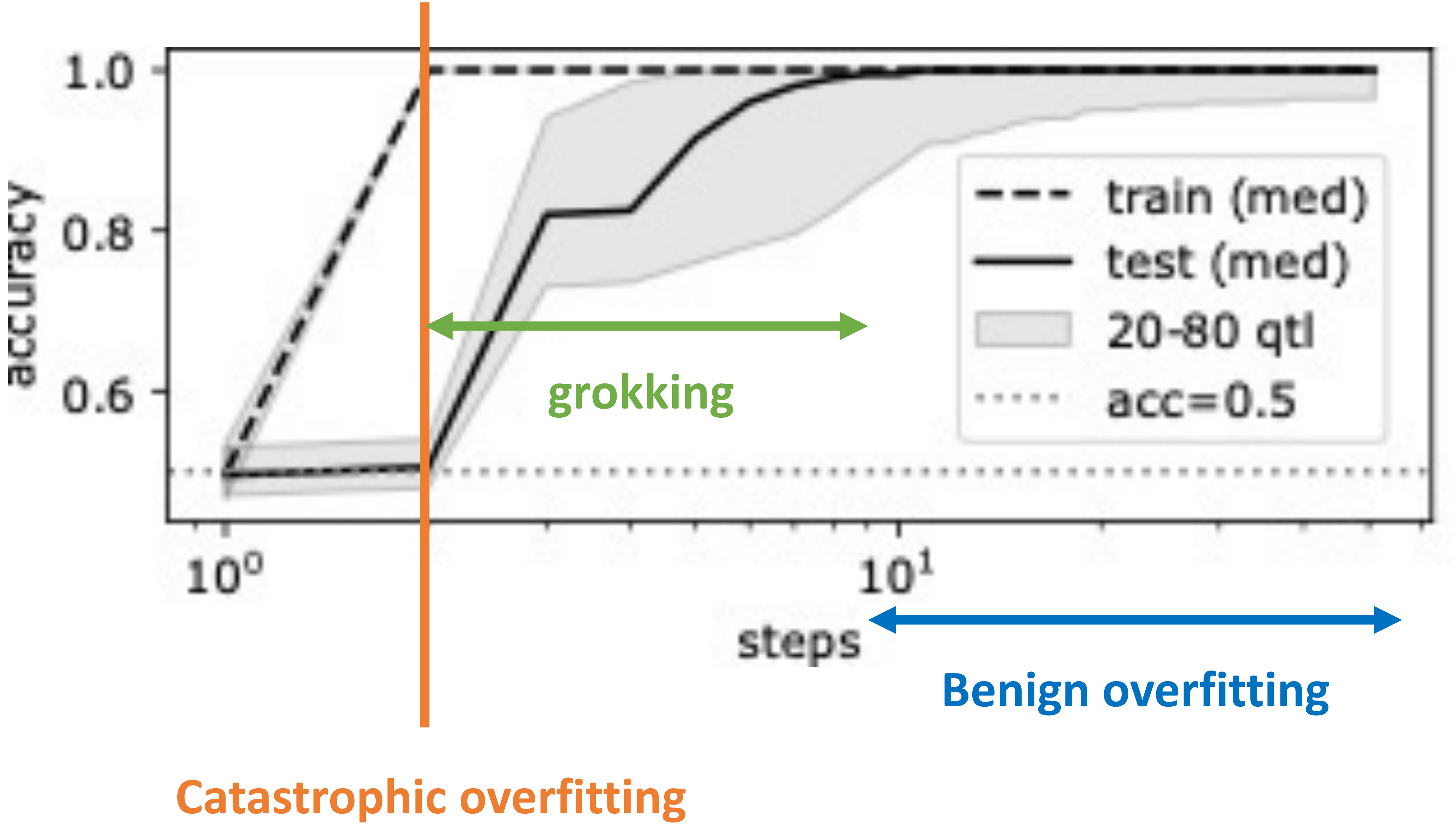
$$L(W) = \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i f(x_i; W)))$$

- Initialization $w_j^{(0)} \sim \mathcal{N}(0, \omega^2 I)$ i.i.d.

Key assumptions (C is a large constant)

- Small step size: $\alpha \leq 1/(Cnp)$
- Small initialization scale: $\omega \leq \frac{\alpha \|\mu_1\|^2}{nm^{3/2}p}$
- Not so small width: $m \geq Cn^{0.02}$

Empirical Behavior



Theoretical Results

Theorem [Xu, Wang, Frei, Vardi, H. 2024]

With probability $1 - o(1)$ over the dataset and the random initialization of the weights:

1. Perfect fitting to noisy training data points for $1 \leq t \leq \sqrt{n}$:
$$y_i = \operatorname{sign}\left(f(x_i; W^{(t)})\right), \qquad \forall i \in [n]$$

2. Near-random test error at $t = 1$:
$$\Pr_{(x,y)\sim P_{\text{clean}}}\left[y \neq \operatorname{sign}\left(f(x; W^{(1)})\right)\right] = \frac{1}{2} \pm o(1)$$

3. Near-optimal generalization for $Cn^{0.01} \leq t \leq \sqrt{n}$:
$$\Pr_{(x,y)\sim P_{\text{clean}}}\left[y \neq \operatorname{sign}\left(f(x; W^{(t)})\right)\right] = e^{-\Omega(n^2)}$$

Proof Sketch

At $t = 1$, the network approximately learns a linear classifier

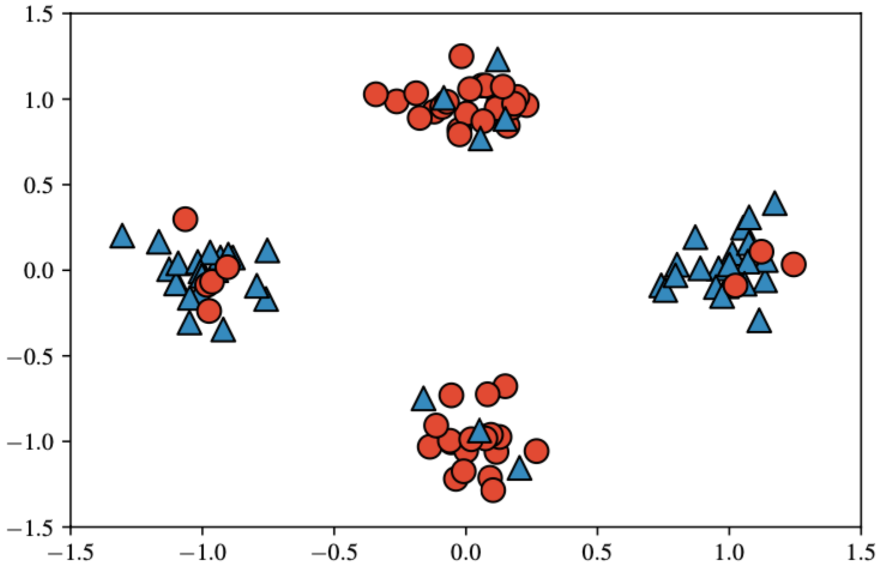
$$f\left(x; W^{(1)}\right) = \sum_{j=1}^{m} a_j \phi\left(\left\langle w_j^{(1)}, x\right\rangle\right) \approx \frac{\alpha}{8n}\left\langle \sum_{i=1}^{n} y_i x_i \, , x\right\rangle$$

- This linear classifier can *perfectly fit* all the training data $\{(x_i, y_i)\}_{i=1}^{n}$, but only *achieves* $\sim 50\%$ *accuracy* on the XOR distribution

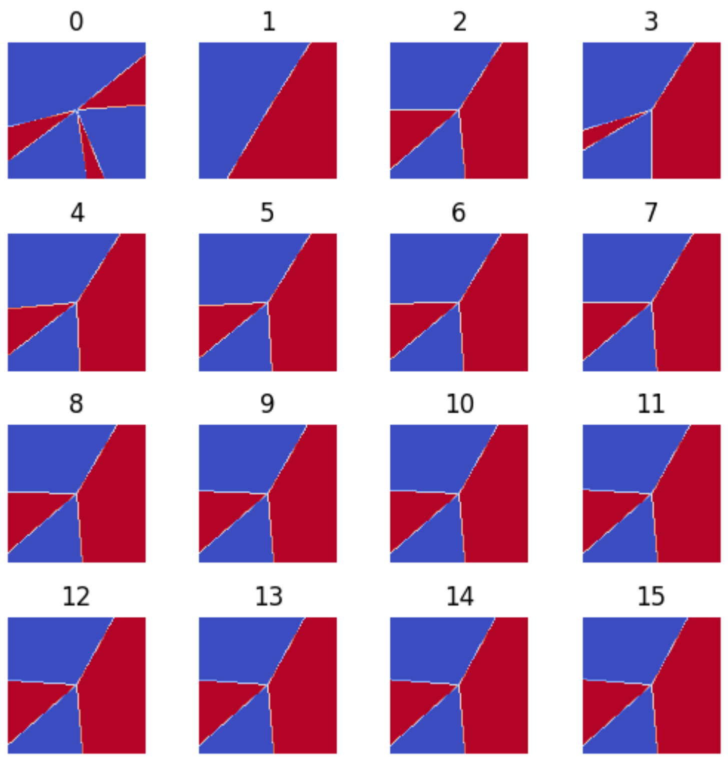
At a later time, the network learns the correct features

- If $a_j > 0$, $w_j^{(t)}$ aligns with $\pm\mu_1$
- If $a_j < 0$, $w_j^{(t)}$ aligns with $\pm\mu_2$

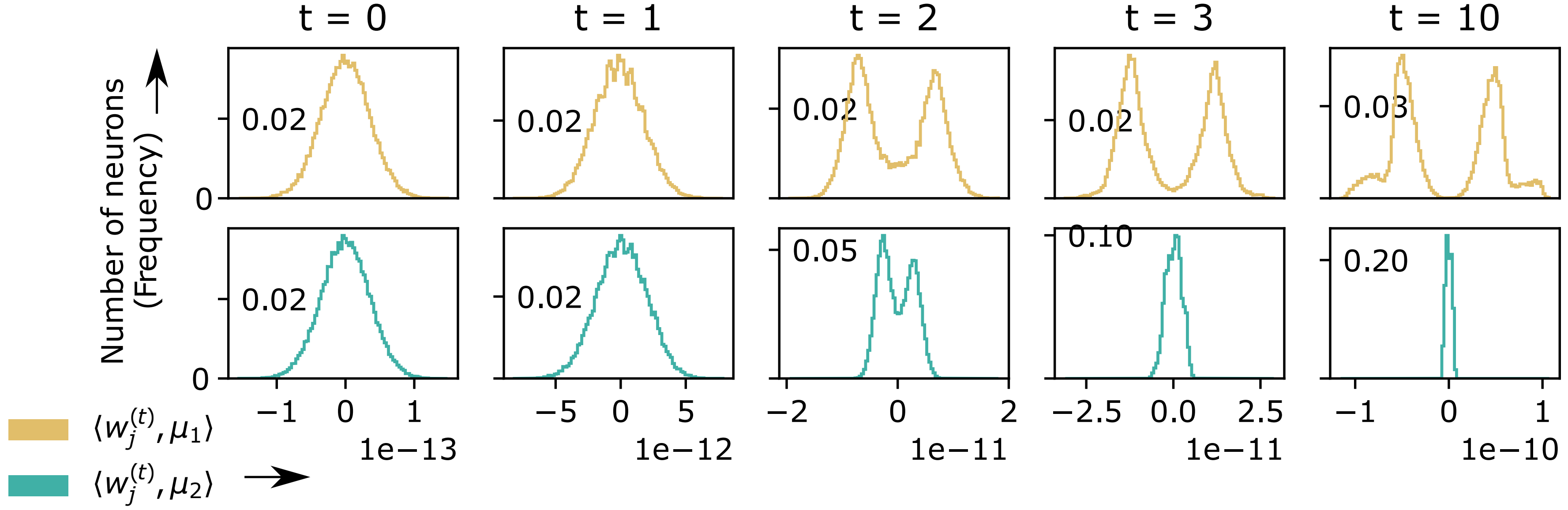
$$\sqrt{m} f\left(x; W^{(t)}\right) = \sum_{j:a_j>0} \phi\left(\left\langle w_j^{(t)}, x\right\rangle\right) - \sum_{j:a_j<0} \phi\left(\left\langle w_j^{(t)}, x\right\rangle\right)$$



Visualizations



Decision boundary in span$\{\mu_1, \mu_2\}$



Distributions of $w_j^{(t)}$'s projections on μ_1 and μ_2

Closing Thoughts

- Deep learning exhibits intriguing phenomena that defy classical statistical wisdom, e.g., grokking & benign overfitting
- Understanding these phenomena likely requires "opening the black box" of neural networks and their training processes
- Questions
  - Understanding grokking in Transformers and modular arithmetic problems
  - How to make neural network inductive biases better "aligned" with the tasks of interest?

DICHOTOMY OF EARLY AND LATE PHASE IMPLICIT BIASES CAN PROVABLY INDUCE GROKKING

Kaifeng Lyu* Princeton University klyu@cs.princeton.edu
Jikai Jin* Stanford University jkjin@stanford.edu
Zhiyuan Li Toyota Technological Institute at Chicago zhiyuanli@ttic.edu
Simon S. Du University of Washington ssdu@cs.washington.edu
Jason D. Lee Princeton University jasonlee@princeton.edu
Wei Hu University of Michigan vvh@umich.edu

BENIGN OVERFITTING AND GROKKING IN RELU NETWORKS FOR XOR CLUSTER DATA

Zhiwei Xu†, Yutong Wang†, Spencer Frei‡, Gal Vardi◇, Wei Hu†
†University of Michigan, ‡University of California, Davis, ◇TTI-Chicago and Hebrew University
{zhiweixu, yutongw, vvh}@umich.edu, sfrei@ucdavis.edu, galvardi@ttic.edu