THE SURPRISING SIMPLICITY OF OVERPARAMETERIZED DEEP NEURAL NETWORKS

JEFFREY PENNINGTON

GOOGLE BRAIN

STATS 385

10-16-19

COLLABORATION

Jascha Sohl-Dickstein

Surya Ganguli

Lechao Xiao

Jaehoon Lee

Minmin Chen

Bo Chang

Sam Schoenholz

Greg Yang

Yasaman Bahri

Roman Novak

Dar Gilboa

OUTLINE

1. Motivation
2. Functional priors
3. Signal propagation
4. Dynamical isometry
5. Functional posteriors
6. Conclusion

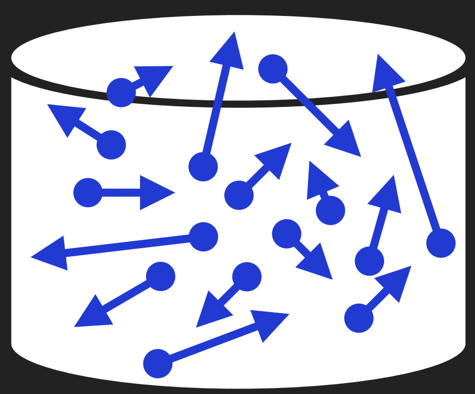
OUTLINE

1. Motivation
2. Functional priors
3. Signal propagation
4. Dynamical isometry
5. Functional posteriors
6. Conclusion

SIMPLICITY IN LARGE NUMBERS: STATISTICAL MECHANICS

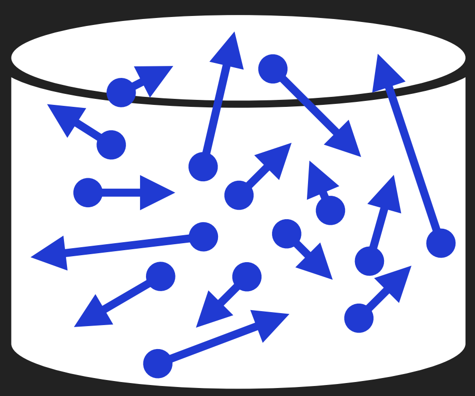
SIMPLICITY IN LARGE NUMBERS: STATISTICAL MECHANICS

Microscopic



SIMPLICITY IN LARGE NUMBERS: STATISTICAL MECHANICS

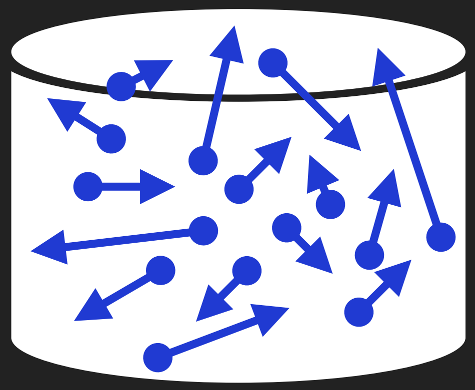
Microscopic



$$\{p_i, x_i\}_{i=1\ldots N}$$

SIMPLICITY IN LARGE NUMBERS: STATISTICAL MECHANICS

Microscopic



$$\{p_i, x_i\}_{i=1\ldots N}$$

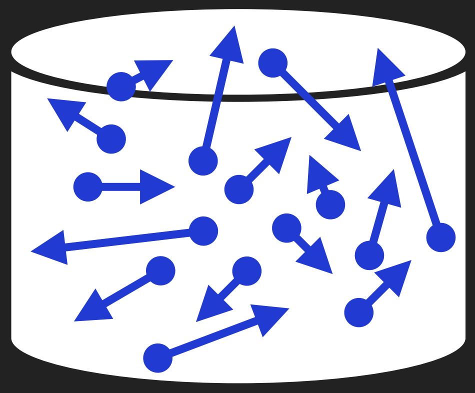
$$H = \frac{1}{2m}\sum_{i=1}^{N} p_i^2 + \sum_{i=1}^{N} V(x_i) + \sum_{i<j} U(x_i - x_j)$$

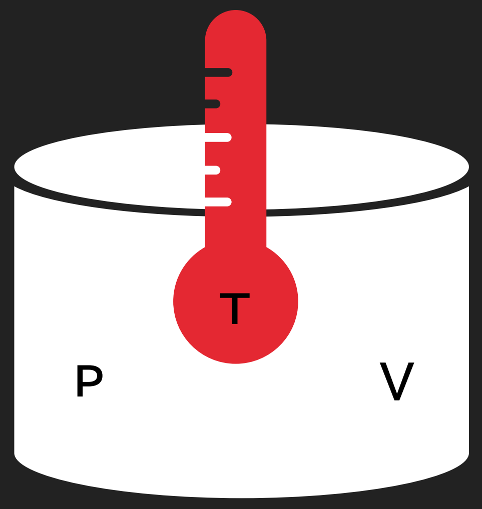
SIMPLICITY IN LARGE NUMBERS: STATISTICAL MECHANICS

Microscopic

N >> 1 →

Macroscopic



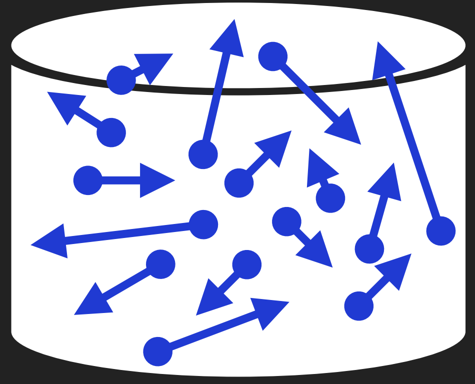


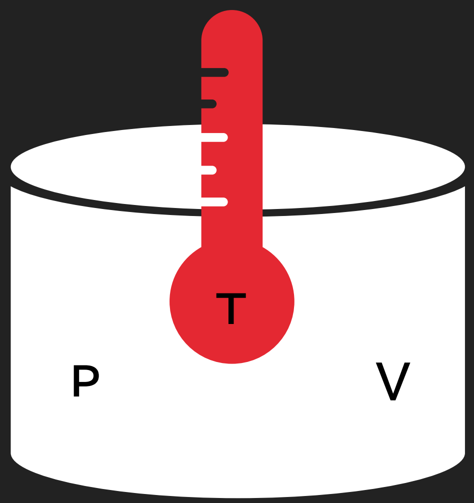
$$\{p_i, x_i\}_{i=1\ldots N}$$

$$H = \frac{1}{2m}\sum_{i=1}^{N} p_i^2 + \sum_{i=1}^{N} V(x_i) + \sum_{i<j} U(x_i - x_j)$$

SIMPLICITY IN LARGE NUMBERS: STATISTICAL MECHANICS

Microscopic $\xrightarrow{N >> 1}$ Macroscopic





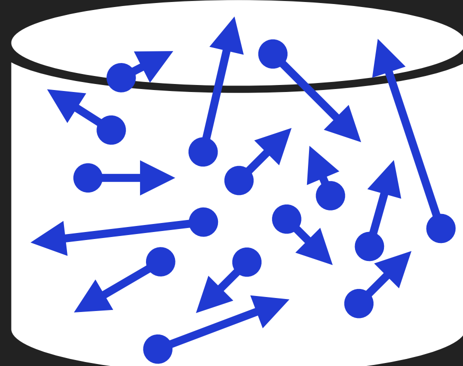
$\{p_i, x_i\}_{i=1\ldots N}$

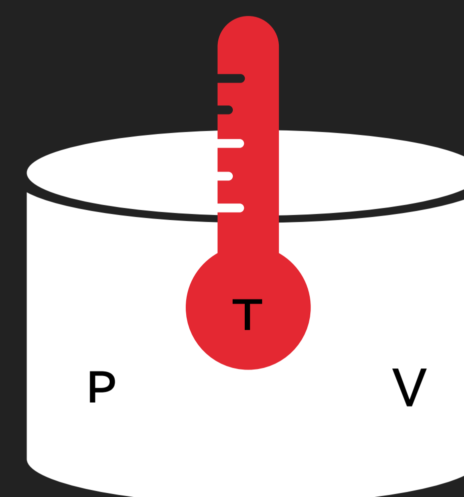
$\{P, V, T\}$

$$H = \frac{1}{2m}\sum_{i=1}^{N} p_i^2 + \sum_{i=1}^{N} V(x_i) + \sum_{i<j} U(x_i - x_j)$$

SIMPLICITY IN LARGE NUMBERS: STATISTICAL MECHANICS

Microscopic $\xrightarrow{N >> 1}$ Macroscopic





$$\{p_i, x_i\}_{i=1\ldots N}$$

$$\{P, V, T\}$$

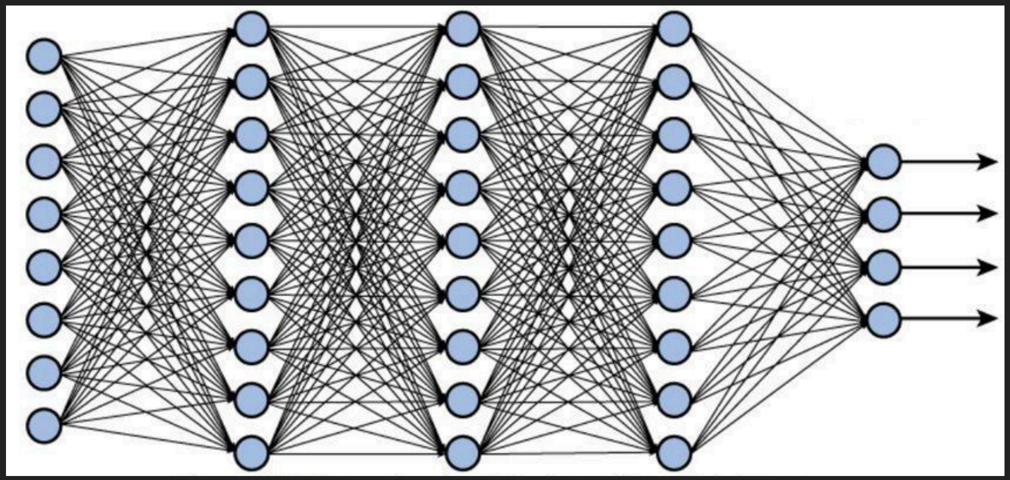
$$H = \frac{1}{2m}\sum_{i=1}^{N} p_i^2 + \sum_{i=1}^{N} V(x_i) + \sum_{i<j} U(x_i - x_j)$$

$$PV = nRT$$

SIMPLICITY IN LARGE NUMBERS: NEURAL NETWORKS

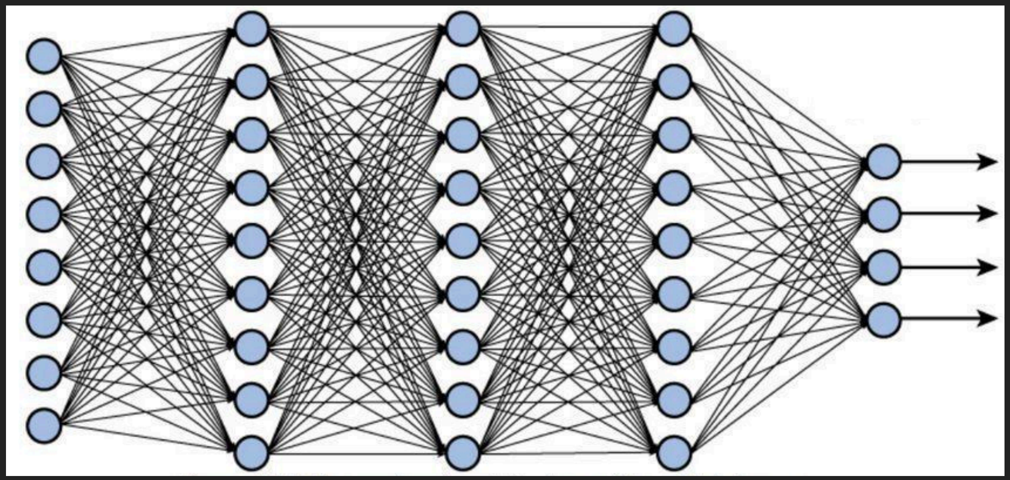
SIMPLICITY IN LARGE NUMBERS: NEURAL NETWORKS

Microscopic



SIMPLICITY IN LARGE NUMBERS: NEURAL NETWORKS

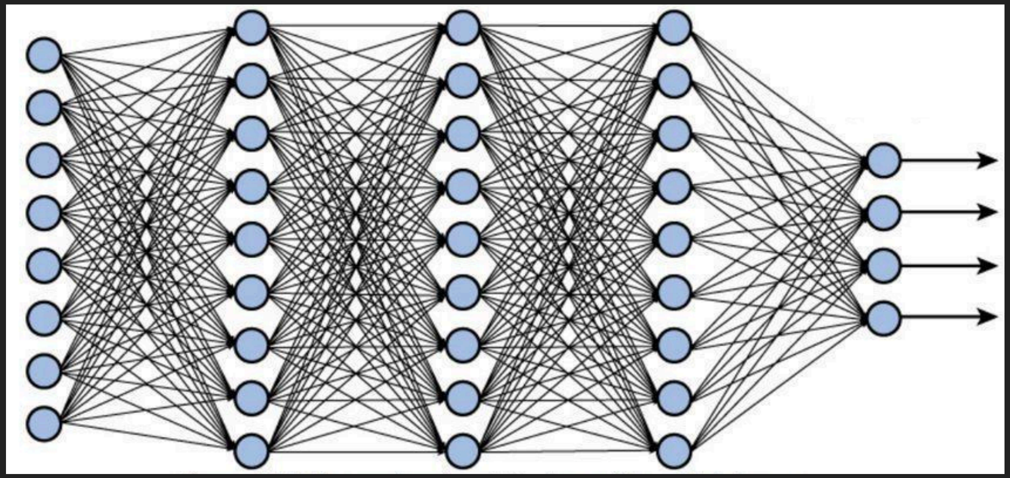
Microscopic



$$\Theta = \{W_{ij}^{l}, b_{i}^{l}\}_{i,j=1\ldots N}^{l=1\ldots L}$$

SIMPLICITY IN LARGE NUMBERS: NEURAL NETWORKS

Microscopic

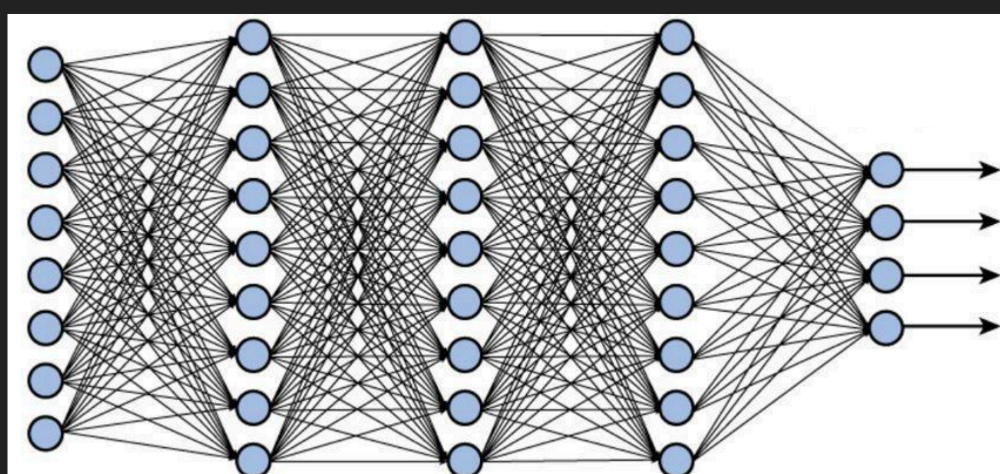


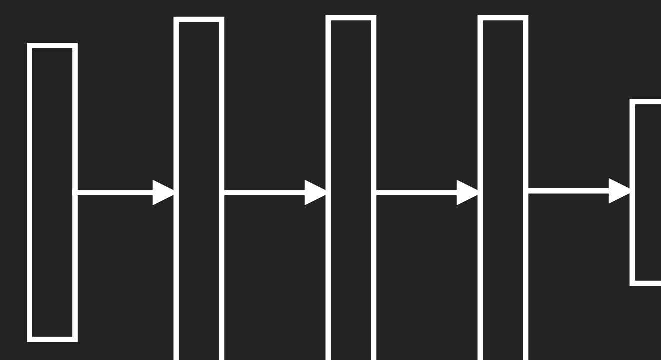
$$\Theta = \{W_{ij}^l, b_i^l\}_{i,j=1\dots N}^{l=1\dots L}$$

$$\partial_t \Theta = -\nabla_\Theta \mathcal{L}$$

SIMPLICITY IN LARGE NUMBERS: NEURAL NETWORKS

Microscopic $\xrightarrow{N >> 1}$ Macroscopic



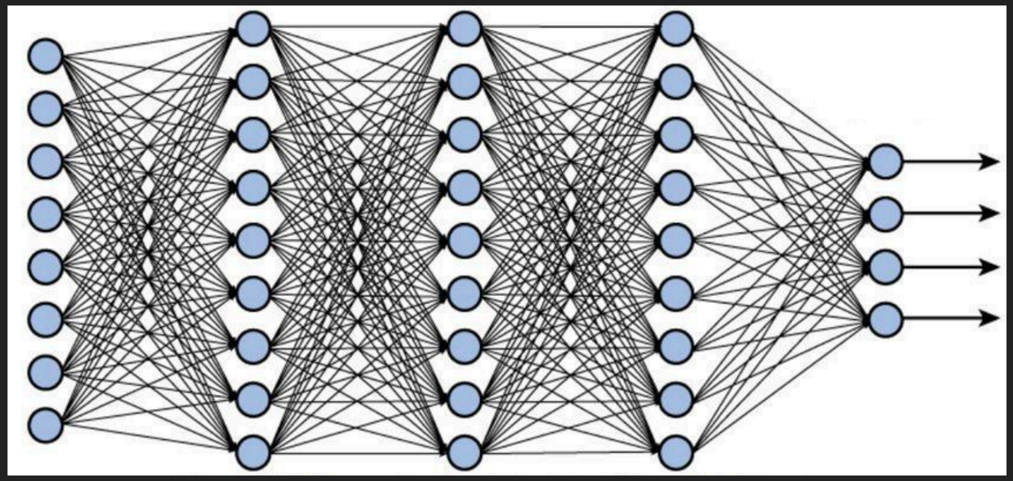


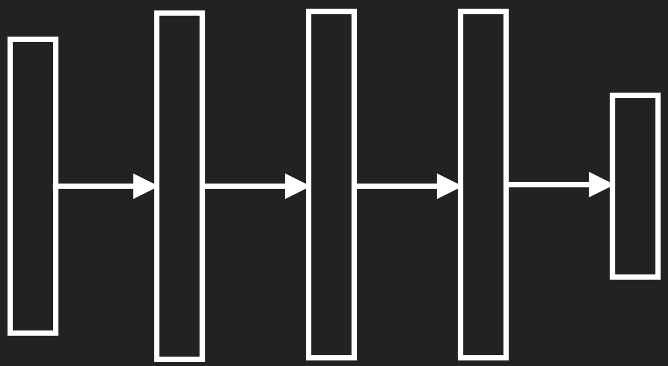
$$\Theta = \{W_{ij}^{l}, b_i^{l}\}_{i,j=1\ldots N}^{l=1\ldots L}$$

$$\partial_t \Theta = -\nabla_\Theta \mathcal{L}$$

SIMPLICITY IN LARGE NUMBERS: NEURAL NETWORKS

Microscopic $\xrightarrow{N >> 1}$ Macroscopic





$$\Theta = \{W_{ij}^l, b_i^l\}_{i,j=1\ldots N}^{l=1\ldots L}$$

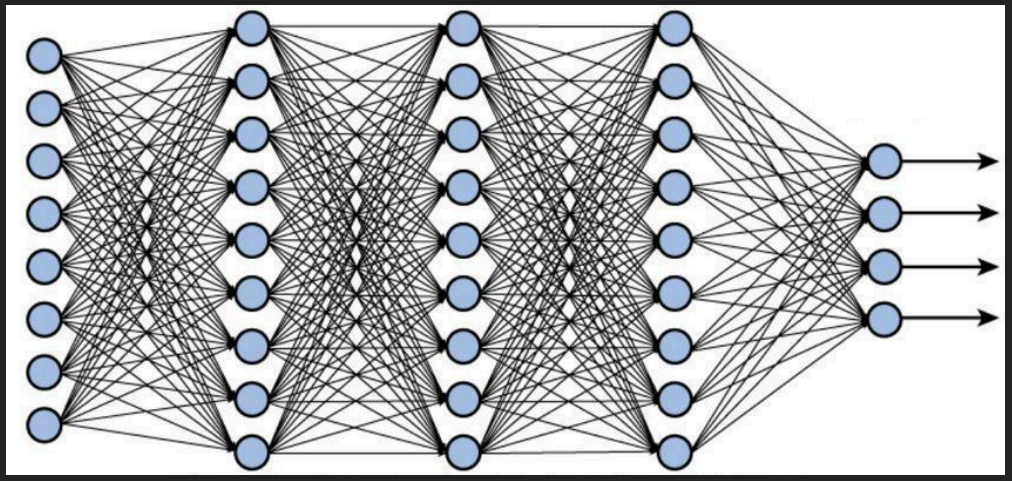
?

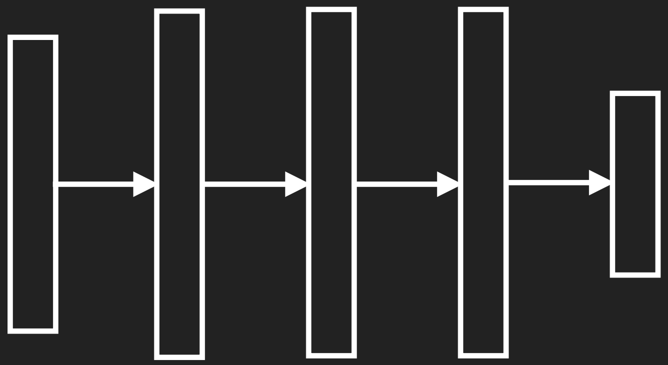
$$\partial_t \Theta = -\nabla_\Theta \mathcal{L}$$

?

SIMPLICITY IN LARGE NUMBERS: NEURAL NETWORKS

Microscopic $\xrightarrow{N >> 1}$ Macroscopic





$$\Theta = \{W^l_{ij}, b^l_i\}^{l=1\ldots L}_{i,j=1\ldots N}$$

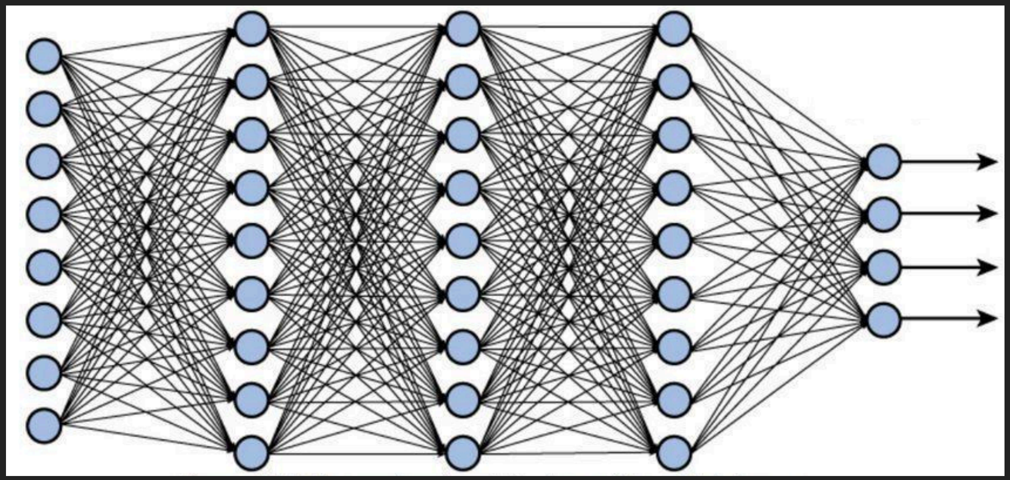
$$\Sigma^l_{ab} = \frac{1}{N}\sum_{i=1}^{N} x^l_{ia} x^l_{ib} \quad (?)$$

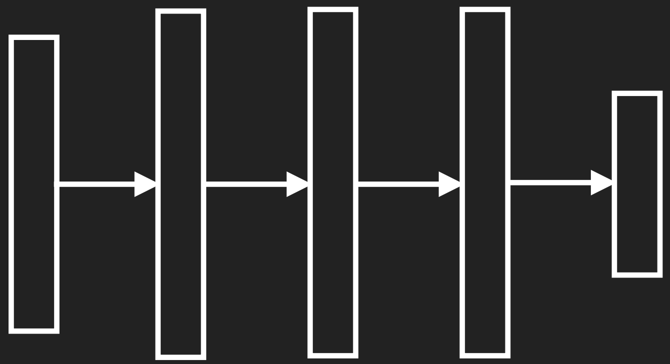
$$\partial_t \Theta = -\nabla_\Theta \mathcal{L}$$

?

SIMPLICITY IN LARGE NUMBERS: NEURAL NETWORKS

Microscopic $\xrightarrow{N >> 1}$ Macroscopic





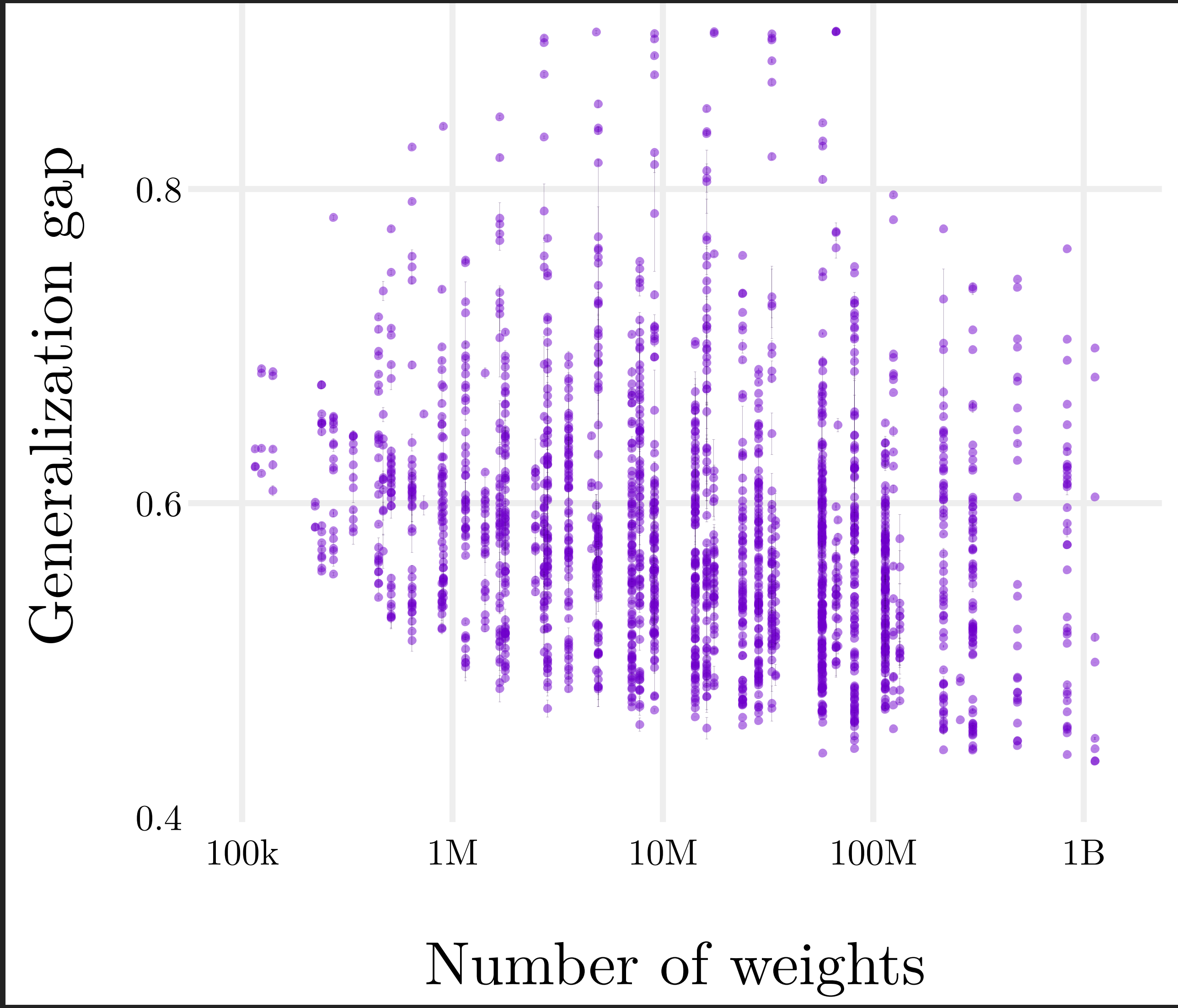
$$\Theta = \{W_{ij}^l, b_i^l\}_{i,j=1\ldots N}^{l=1\ldots L}$$

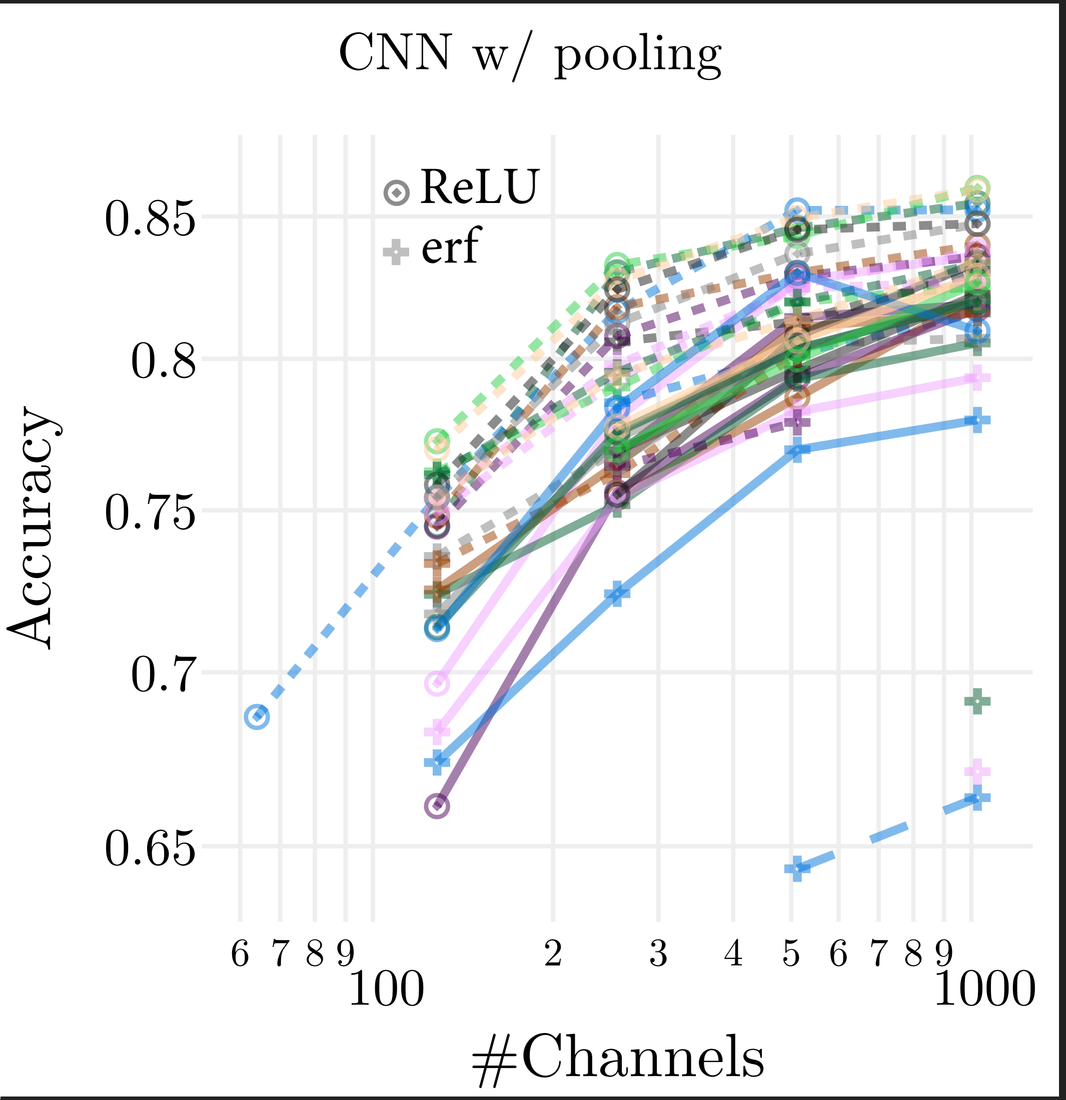
$$\Sigma_{ab}^l = \frac{1}{N}\sum_{i=1}^{N} x_{ia}^l x_{ib}^l \quad (?)$$

$$\partial_t \Theta = -\nabla_\Theta \mathcal{L}$$

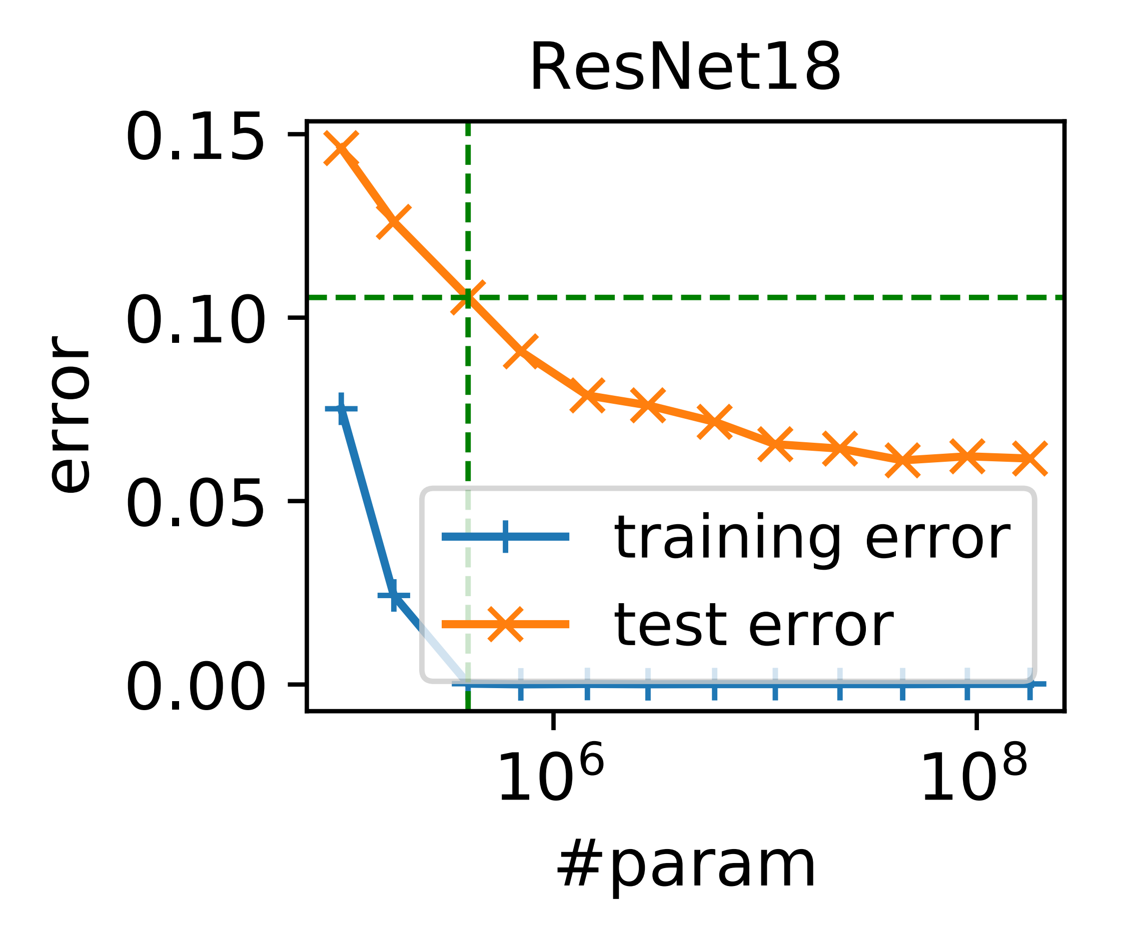
$$\partial_t \Sigma_{ab}^l \approx 0 \quad (?)$$

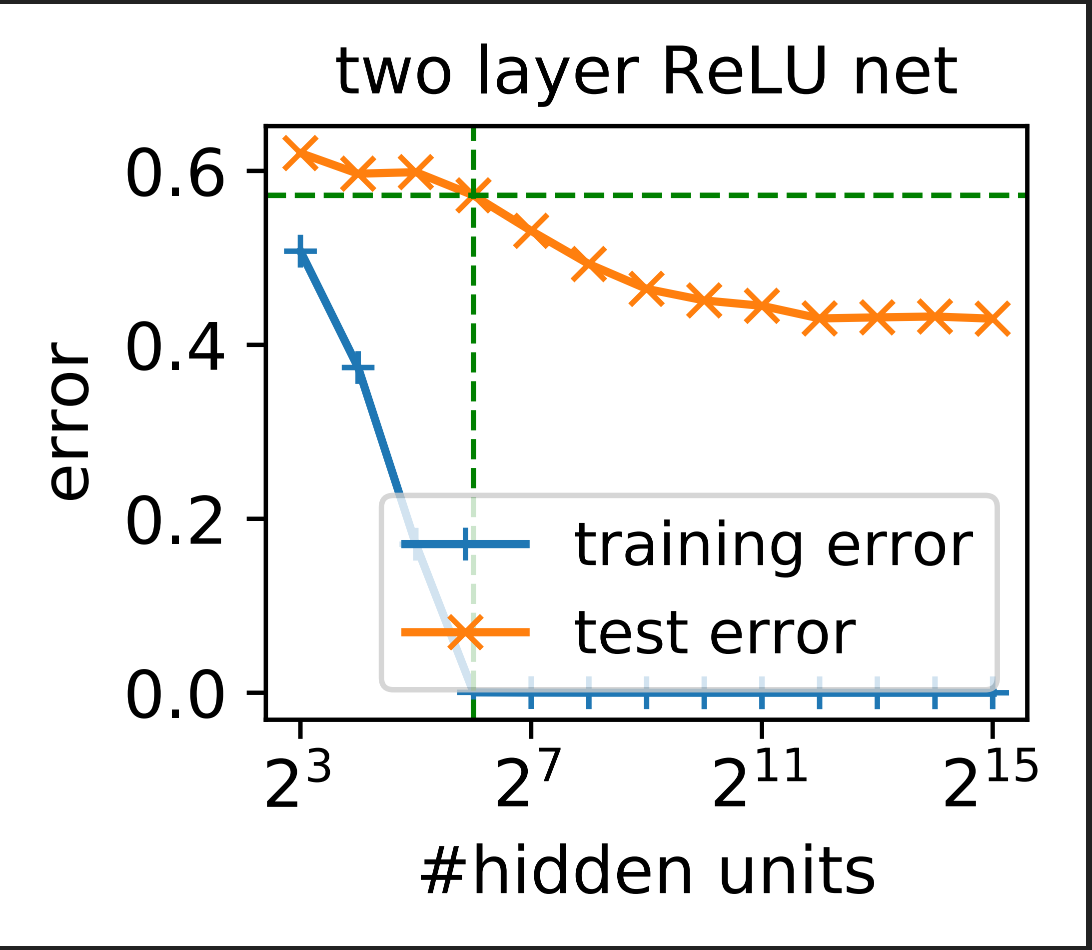
OVERPARAMETERIZED MODELS PERFORM BETTER





OVERPARAMETERIZED MODELS PERFORM BETTER





OUTLINE

1. Motivation
2. Functional priors
3. Signal propagation
4. Dynamical isometry
5. Functional posteriors
6. Conclusion

THE SINGLE HIDDEN LAYER CASE

Fully connected, single hidden layer [Radford Neal '94]

Inputs: $x_a \in \mathbb{R}^{N_0}$ with input index *a*

$$\Sigma^0_{ab} = \frac{1}{N_0}\sum_i x_{ia}x_{ib}$$

Parameters: $W^l_{ij} \in \mathbb{R}^{N_{l-1}\times N_l}$ $\quad b^l_i \in \mathbb{R}^{N_l}$

Prior: $W^l_{ij} \sim \mathcal{N}(0, \sigma^2_w/N_{l-1})$ $\quad b^l_i \sim \mathcal{N}(0, \sigma^2_b)$

Network:

$$z^1_{ia} = \sum_j W^1_{ij} x_{ja} + b^1_i$$

$$y_{ia} = \sum_j W^2_{ij}\phi(z^1_{ja}) + b^2_i$$

THE SINGLE HIDDEN LAYER CASE

Fully connected, single hidden layer [Radford Neal '94]

Inputs: $x_a \in \mathbb{R}^{N_0}$ with input index *a* $\qquad \Sigma^0_{ab} = \frac{1}{N_0}\sum_i x_{ia}x_{ib}$

Parameters: $W^l_{ij} \in \mathbb{R}^{N_{l-1}\times N_l} \qquad b^l_i \in \mathbb{R}^{N_l}$

Prior: $W^l_{ij} \sim \mathcal{N}(0, \sigma^2_w/N_{l-1}) \qquad b^l_i \sim \mathcal{N}(0, \sigma^2_b)$

Network:

Weighted sum of Gaussians

$$z^1_{ia} = \sum_j W^1_{ij}x_{ja} + b^1_i \qquad (z^1_{ia}, z^1_{jb})^T \sim \mathcal{N}(0, \Sigma^1_{ab}\delta_{ij})$$

$$y_{ia} = \sum_j W^2_{ij}\phi(z^1_{ja}) + b^2_i$$

THE SINGLE HIDDEN LAYER CASE

Fully connected, single hidden layer [Radford Neal '94]

Inputs: $x_a \in \mathbb{R}^{N_0}$ with input index *a*

$$\Sigma^0_{ab} = \frac{1}{N_0} \sum_i x_{ia} x_{ib}$$

Parameters: $W^l_{ij} \in \mathbb{R}^{N_{l-1} \times N_l}$ $\quad b^l_i \in \mathbb{R}^{N_l}$

Prior: $W^l_{ij} \sim \mathcal{N}(0, \sigma^2_w / N_{l-1})$ $\quad b^l_i \sim \mathcal{N}(0, \sigma^2_b)$

Network:

Weighted sum of Gaussians

$$z^1_{ia} = \sum_j W^1_{ij} x_{ja} + b^1_i \qquad (z^1_{ia}, z^1_{jb})^T \sim \mathcal{N}(0, \Sigma^1_{ab} \delta_{ij})$$

$$y_{ia} = \sum_j W^2_{ij} \phi(z^1_{ja}) + b^2_i \qquad (y_{ia}, y_{jb})^T \xrightarrow{N_1 \to \infty} \mathcal{N}(0, \Sigma^2_{ab} \delta_{ij})$$

Sum of i.i.d. random variables

THE SINGLE HIDDEN LAYER CASE

Infinitely wide neural networks are Gaussian Processes

$$z^1_{ia} = \sum_j W^1_{ij} x_{ja} + b^1_i \qquad (z^1_{ia}, z^1_{jb})^T \sim \mathcal{N}(0, \Sigma^1_{ab}\delta_{ij})$$

$$y_{ia} = \sum_j W^2_{ij} \phi(z^1_{ja}) + b^2_i \qquad (y_{ia}, y_{jb})^T \xrightarrow{N_1 \to \infty} \mathcal{N}(0, \Sigma^2_{ab}\delta_{ij})$$

Completely defined by a compositional kernel

THE SINGLE HIDDEN LAYER CASE

Infinitely wide neural networks are Gaussian Processes

$$z^1_{ia} = \sum_j W^1_{ij} x_{ja} + b^1_i \qquad (z^1_{ia}, z^1_{jb})^T \sim \mathcal{N}(0, \Sigma^1_{ab}\delta_{ij})$$

$$y_{ia} = \sum_j W^2_{ij} \phi(z^1_{ja}) + b^2_i \qquad (y_{ia}, y_{jb})^T \xrightarrow{N_1 \to \infty} \mathcal{N}(0, \Sigma^2_{ab}\delta_{ij})$$

Completely defined by a compositional kernel

$$\Sigma^1 = \sigma^2_w \Sigma^0 + \sigma^2_b$$

$$\Sigma^2 = \sigma^2_w \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(0,\Sigma^1)}[\phi(\boldsymbol{z})\phi(\boldsymbol{z})^T] + \sigma^2_b$$

Significant simplification

THE SINGLE HIDDEN LAYER CASE

Infinitely wide neural networks are Gaussian Processes

$$z^1_{ia} = \sum_j W^1_{ij} x_{ja} + b^1_i \qquad (z^1_{ia}, z^1_{jb})^T \sim \mathcal{N}(0, \Sigma^1_{ab}\delta_{ij})$$

$$y_{ia} = \sum_j W^2_{ij} \phi(z^1_{ja}) + b^2_i \qquad (y_{ia}, y_{jb})^T \xrightarrow{N_1 \to \infty} \mathcal{N}(0, \Sigma^2_{ab}\delta_{ij})$$

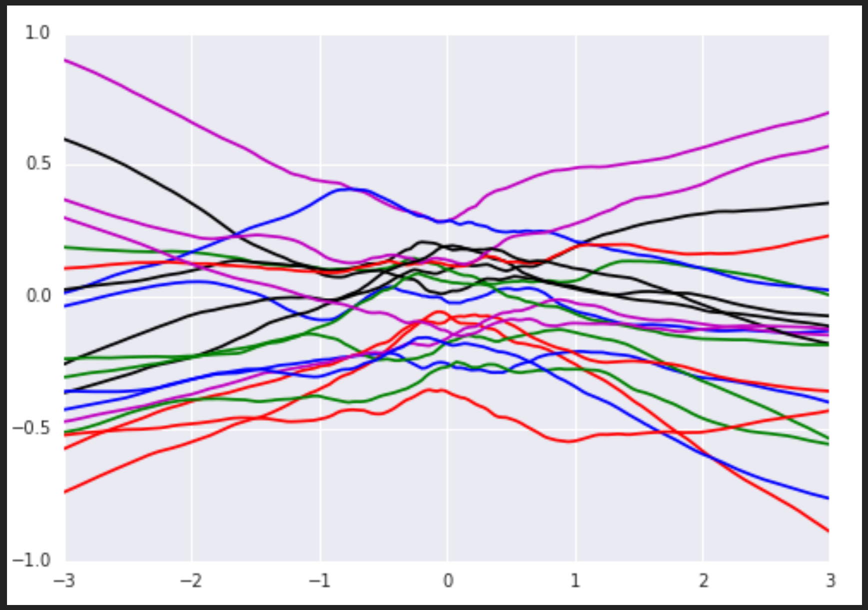
Completely defined by a compositional kernel

$$\Sigma^1 = \sigma^2_w \Sigma^0 + \sigma^2_b$$

$$\Sigma^2 = \sigma^2_w \mathbb{E}_{z \sim \mathcal{N}(0,\Sigma^1)}[\phi(z)\phi(z)^T] + \sigma^2_b$$

Significant simplification

Draws from ReLU-GP



AD, RF, YS ('16)
BP, SL, MR, JSD, SG ('16)
SSS, JG, SG, JSD ('17)
JL, YB et al. ('18)
RN, XLC et al. ('18)

DEEP NETWORKS

Extension to deep networks

$$z^1_{ia} = \sum_j W^1_{ij} x_{ja} + b^1_i \longrightarrow \Sigma^1 = \sigma_w^2 \Sigma^0 + \sigma_b^2$$

AD, RF, YS ('16)
BP, SL, MR, JSD, SG ('16)
SSS, JG, SG, JSD ('17)
JL, YB et al. ('18)
RN, XLC et al. ('18)

DEEP NETWORKS

Extension to deep networks

$$z_{ia}^1 = \sum_j W_{ij}^1 x_{ja} + b_i^1 \longrightarrow \Sigma^1 = \sigma_w^2 \Sigma^0 + \sigma_b^2$$

$$z_{ia}^2 = \sum_j W_{ij}^2 \phi(z_{ja}^1) + b_i^2$$

AD, RF, YS ('16)
BP, SL, MR, JSD, SG ('16)
SSS, JG, SG, JSD ('17)
JL, YB et al. ('18)
RN, XLC et al. ('18)

DEEP NETWORKS

Extension to deep networks

$$z^1_{ia} = \sum_j W^1_{ij} x_{ja} + b^1_i \longrightarrow \Sigma^1 = \sigma_w^2 \Sigma^0 + \sigma_b^2$$

$$z^2_{ia} = \sum_j W^2_{ij} \phi(z^1_{ja}) + b^2_i \xrightarrow{N_1 \to \infty} \Sigma^2 = \sigma_w^2 \mathbb{E}_{\boldsymbol{z} \sim \mathcal{N}(0, \Sigma^1)}[\phi(\boldsymbol{z})\phi(\boldsymbol{z})^T] + \sigma_b^2$$

AD, RF, YS ('16)
BP, SL, MR, JSD, SG ('16)
SSS, JG, SG, JSD ('17)
JL, YB et al. ('18)
RN, XLC et al. ('18)

DEEP NETWORKS

Extension to deep networks

$$
\begin{tikzcd}
z^1_{ia} = \sum_j W^1_{ij} x_{ja} + b^1_i \arrow[r] \arrow[d] & \Sigma^1 = \sigma_w^2 \Sigma^0 + \sigma_b^2 \arrow[d, "\mathcal{C}"] \\
z^2_{ia} = \sum_j W^2_{ij} \phi(z^1_{ja}) + b^2_i \arrow[r, dashed, "N_1 \to \infty"] & \Sigma^2 = \sigma_w^2 \mathbb{E}_{\boldsymbol{z} \sim \mathcal{N}(0, \Sigma^1)} [\phi(\boldsymbol{z}) \phi(\boldsymbol{z})^T] + \sigma_b^2
\end{tikzcd}
$$

AD, RF, YS ('16)
BP, SL, MR, JSD, SG ('16)
SSS, JG, SG, JSD ('17)
JL, YB et al. ('18)
RN, XLC et al. ('18)

DEEP NETWORKS

Extension to deep networks

$$
\begin{tikzcd}
z^1_{ia} = \sum_j W^1_{ij} x_{ja} + b^1_i \arrow[r] \arrow[d] & \Sigma^1 = \sigma_w^2 \Sigma^0 + \sigma_b^2 \arrow[d, "\mathcal{C}"] \\
z^2_{ia} = \sum_j W^2_{ij} \phi(z^1_{ja}) + b^2_i \arrow[r, dashed, "N_1 \to \infty"] \arrow[d] & \Sigma^2 = \sigma_w^2 \mathbb{E}_{\boldsymbol{z} \sim \mathcal{N}(0, \Sigma^1)} [\phi(\boldsymbol{z}) \phi(\boldsymbol{z})^T] + \sigma_b^2 \arrow[d, "\mathcal{C}"] \\
\vdots \arrow[d] & \vdots \arrow[d, "\mathcal{C}"] \\
z^l_{ia} = \sum_j W^l_{ij} \phi(z^{l-1}_{ja}) + b^l_i \arrow[r, dashed, "N_{l-1} \to \infty"] & \Sigma^l = \sigma_w^2 \mathbb{E}_{\boldsymbol{z} \sim \mathcal{N}(0, \Sigma^{l-1})} [\phi(\boldsymbol{z}) \phi(\boldsymbol{z})^T] + \sigma_b^2
\end{tikzcd}
$$

AD, RF, YS ('16)
BP, SL, MR, JSD, SG ('16)
SSS, JG, SG, JSD ('17)
JL, YB et al. ('18)
RN, XLC et al. ('18)

DEEP NETWORKS

Extension to deep networks

$$z^1_{ia} = \sum_j W^1_{ij} x_{ja} + b^1_i \longrightarrow \Sigma^1 = \sigma_w^2 \Sigma^0 + \sigma_b^2$$

$$\Big\downarrow \qquad\qquad \Big\downarrow \mathcal{C}$$

$$z^2_{ia} = \sum_j W^2_{ij} \phi(z^1_{ja}) + b^2_i \xrightarrow{N_1 \to \infty} \Sigma^2 = \sigma_w^2 \mathbb{E}_{\boldsymbol{z} \sim \mathcal{N}(0,\Sigma^1)}\left[\phi(\boldsymbol{z})\phi(\boldsymbol{z})^T\right] + \sigma_b^2$$

$$\Big\downarrow \qquad\qquad \Big\downarrow \mathcal{C}$$

$$\vdots \qquad\qquad \vdots$$

$$\Big\downarrow \qquad\qquad \Big\downarrow \mathcal{C}$$

$$z^l_{ia} = \sum_j W^l_{ij} \phi(z^{l-1}_{ja}) + b^l_i \xrightarrow{N_{l-1} \to \infty} \Sigma^l = \sigma_w^2 \mathbb{E}_{\boldsymbol{z} \sim \mathcal{N}(0,\Sigma^{l-1})}\left[\phi(\boldsymbol{z})\phi(\boldsymbol{z})^T\right] + \sigma_b^2$$

NNGP Kernel

AD, RF, YS ('16)
BP, SL, MR, JSD, SG ('16)
SSS, JG, SG, JSD ('17)
JL, YB et al. ('18)
RN, XLC et al. ('18)

DEEP NETWORKS

Extension to deep networks

$$z_{ia}^1 = \sum_j W_{ij}^1 x_{ja} + b_i^1 \longrightarrow \Sigma^1 = \sigma_w^2 \Sigma^0 + \sigma_b^2$$

$$\downarrow \qquad \qquad \downarrow \mathcal{C}$$

$$z_{ia}^2 = \sum_j W_{ij}^2 \phi(z_{ja}^1) + b_i^2 \xrightarrow{N_1 \to \infty} \Sigma^2 = \sigma_w^2 \mathbb{E}_{\boldsymbol{z} \sim \mathcal{N}(0,\Sigma^1)} [\phi(\boldsymbol{z})\phi(\boldsymbol{z})^T] + \sigma_b^2$$

$$\downarrow \qquad \qquad \downarrow \mathcal{C}$$

$$\vdots \qquad \qquad \vdots$$

$$\downarrow \qquad \qquad \downarrow \mathcal{C}$$

$$z_{ia}^l = \sum_j W_{ij}^l \phi(z_{ja}^{l-1}) + b_i^l \xrightarrow{N_{l-1} \to \infty} \Sigma^l = \sigma_w^2 \mathbb{E}_{\boldsymbol{z} \sim \mathcal{N}(0,\Sigma^{l-1})} [\phi(\boldsymbol{z})\phi(\boldsymbol{z})^T] + \sigma_b^2$$

NNGP Kernel

Neural network induces dynamical system over kernels

Understanding prior equivalent to studying dynamics

OUTLINE

1. Motivation
2. Functional priors
3. Signal propagation
4. Dynamical isometry
5. Functional posteriors
6. Conclusion

DYNAMICS OF SIGNAL PROPAGATION

$$\Sigma^1 \xrightarrow{\mathcal{C}} \Sigma^2 \xrightarrow{\mathcal{C}} \cdots \xrightarrow{\mathcal{C}} \Sigma^l \xrightarrow{\mathcal{C}} \cdots$$

DYNAMICS OF SIGNAL PROPAGATION

$$\Sigma^1 \xrightarrow{\mathcal{C}} \Sigma^2 \xrightarrow{\mathcal{C}} \cdots \xrightarrow{\mathcal{C}} \Sigma^l \xrightarrow{\mathcal{C}} \cdots \quad \Sigma^*$$

Dynamics converge to **universal** fixed point

- Independent of inputs $\Rightarrow$ pathological

DYNAMICS OF SIGNAL PROPAGATION

$$\Sigma^1 \xrightarrow{\mathcal{C}} \Sigma^2 \xrightarrow{\mathcal{C}} \cdots \xrightarrow{\mathcal{C}} \Sigma^l \xrightarrow{\mathcal{C}} \cdots \quad \Sigma^*$$

Dynamics converge to **universal** fixed point

- Independent of inputs $\Rightarrow$ pathological

Rate of convergence determined by behavior near fixed point

DYNAMICS OF SIGNAL PROPAGATION

$$\Sigma^1 \xrightarrow{\mathcal{C}} \Sigma^2 \xrightarrow{\mathcal{C}} \cdots \xrightarrow{\mathcal{C}} \Sigma^l \xrightarrow{\mathcal{C}} \cdots \quad \Sigma^*$$

Dynamics converge to **universal** fixed point

- Independent of inputs $\Rightarrow$ pathological

Rate of convergence determined by behavior near fixed point

$$\epsilon^l = \Sigma^* - \Sigma^l \qquad \Rightarrow \qquad \epsilon^{l+1} = \left.\frac{\partial \mathcal{C}(\Sigma)}{\partial \Sigma}\right|_{\Sigma^*} \epsilon^l$$

DYNAMICS OF SIGNAL PROPAGATION

$$\Sigma^1 \xrightarrow{\mathcal{C}} \Sigma^2 \xrightarrow{\mathcal{C}} \cdots \xrightarrow{\mathcal{C}} \Sigma^l \xrightarrow{\mathcal{C}} \cdots \quad \Sigma^*$$

Dynamics converge to **universal** fixed point

- Independent of inputs $\Rightarrow$ pathological

Rate of convergence determined by behavior near fixed point

$$\epsilon^l = \Sigma^* - \Sigma^l \qquad \Rightarrow \qquad \epsilon^{l+1} = \boxed{\left.\frac{\partial \mathcal{C}(\Sigma)}{\partial \Sigma}\right|_{\Sigma^*}} \epsilon^l$$

$\lambda_{\max} > 1$ — Unstable fixed point

$\lambda_{\max} \leq 1$ — Stable fixed point

DYNAMICS OF SIGNAL PROPAGATION

$$\Sigma^1 \xrightarrow{\mathcal{C}} \Sigma^2 \xrightarrow{\mathcal{C}} \cdots \xrightarrow{\mathcal{C}} \Sigma^l \xrightarrow{\mathcal{C}} \cdots \quad \Sigma^*$$

Dynamics converge to **universal** fixed point

- Independent of inputs $\Rightarrow$ pathological

Rate of convergence determined by behavior near fixed point

$$\epsilon^l = \Sigma^* - \Sigma^l \qquad \Rightarrow \qquad \epsilon^{l+1} \quad \approx \lambda_{\mathrm{max}}^l$$

$\lambda_{\mathrm{max}} > 1$

Unstable fixed point

$\lambda_{\mathrm{max}} \leq 1$

Stable fixed point

DYNAMICS OF SIGNAL PROPAGATION

$$\Sigma^1 \xrightarrow{\mathcal{C}} \Sigma^2 \xrightarrow{\mathcal{C}} \cdots \xrightarrow{\mathcal{C}} \Sigma^l \xrightarrow{\mathcal{C}} \cdots \quad \Sigma^*$$

Dynamics converge to **universal** fixed point

- Independent of inputs $\Rightarrow$ pathological

Rate of convergence determined by behavior near fixed point

Exponential convergence

$$\epsilon^l \approx e^{-l/\xi}$$

with rate

$$\xi = -1/\log \lambda_{\max}$$

$\Leftarrow$

$$\epsilon^{l+1} \approx \lambda_{\max}^l$$

$\lambda_{\max} > 1$ → Unstable fixed point

$\lambda_{\max} \leq 1$ → Stable fixed point

DYNAMICS OF SIGNAL PROPAGATION

$$\Sigma^1 \xrightarrow{\mathcal{C}} \Sigma^2 \xrightarrow{\mathcal{C}} \cdots \xrightarrow{\mathcal{C}} \Sigma^l \xrightarrow{\mathcal{C}} \cdots \quad \Sigma^*$$

Dynamics converge to **universal** fixed point

- Independent of inputs $\Rightarrow$ pathological

Rate of convergence determined by behavior near fixed point

Exponential convergence

$$\epsilon^l \approx e^{-l/\xi}$$

with rate

$$\xi = -1/\log \lambda_{\max}$$

$\Leftarrow$

$$\epsilon^{l+1} \approx \lambda_{\max}^l$$

$\lambda_{\max} > 1$ — Unstable fixed point

$\lambda_{\max} \leq 1$ — Stable fixed point

How can we adjust the hyperparameters to delay convergence?

FIXED POINT ANALYSIS

The fixed point satisfies

$$\Sigma^* = \mathcal{C}(\Sigma^*) = \sigma_w^2\, \mathbb{E}_{z\sim\mathcal{N}(0,\Sigma^*)}[\phi(z)\phi(z)^\top] + \sigma_b^2$$

BP, SL, MR, JSD, SG ('16)
SSS, JG, SG, JSD ('17)

FIXED POINT ANALYSIS

The fixed point satisfies

$$\Sigma^* = \mathcal{C}(\Sigma^*) = \sigma_w^2 \, \mathbb{E}_{z \sim \mathcal{N}(0, \Sigma^*)}[\phi(z)\phi(z)^\top] + \sigma_b^2$$

One solution is perfect correlation,

$$\Sigma^* = q^* \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

Variance

Correlation

BP, SL, MR, JSD, SG ('16)
SSS, JG, SG, JSD ('17)

FIXED POINT ANALYSIS

The fixed point satisfies

$$\Sigma^* = \mathcal{C}(\Sigma^*) = \sigma_w^2 \, \mathbb{E}_{z\sim\mathcal{N}(0,\Sigma^*)}[\phi(z)\phi(z)^\top] + \sigma_b^2$$

One solution is perfect correlation,

$$\Sigma^* = q^* \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

Variance Correlation

Is this fixed point stable?

BP, SL, MR, JSD, SG ('16)
SSS, JG, SG, JSD ('17)

FIXED POINT ANALYSIS

The fixed point satisfies

$$\Sigma^* = \mathcal{C}(\Sigma^*) = \sigma_w^2 \, \mathbb{E}_{z\sim\mathcal{N}(0,\Sigma^*)}[\phi(z)\phi(z)^\top] + \sigma_b^2$$

One solution is perfect correlation,

$$\Sigma^* = \underbrace{q^*}_{\text{Variance}} \underbrace{\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}}_{\text{Correlation}}$$

Is this fixed point stable?

- Depends on hyperparameters

BP, SL, MR, JSD, SG ('16)
SSS, JG, SG, JSD ('17)

FIXED POINT ANALYSIS

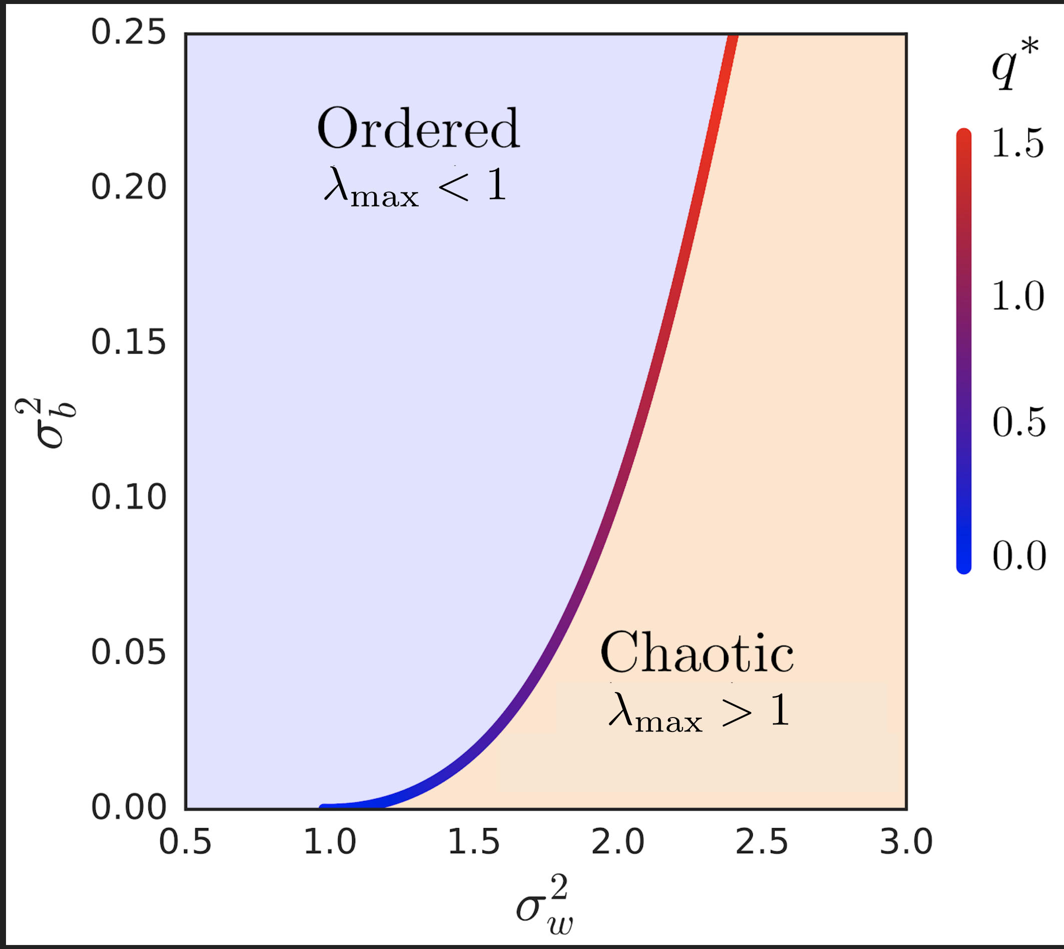
The fixed point satisfies

$$\Sigma^* = \mathcal{C}(\Sigma^*) = \sigma_w^2 \, \mathbb{E}_{z\sim\mathcal{N}(0,\Sigma^*)}[\phi(z)\phi(z)^\top] + \sigma_b^2$$

One solution is perfect correlation,

$$\Sigma^* = q^* \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

Variance　　Correlation



Is this fixed point stable?

- Depends on hyperparameters

BP, SL, MR, JSD, SG ('16)
SSS, JG, SG, JSD ('17)

FIXED POINT ANALYSIS

The fixed point satisfies

$$\Sigma^* = \mathcal{C}(\Sigma^*) = \sigma_w^2\, \mathbb{E}_{z\sim\mathcal{N}(0,\Sigma^*)}[\phi(z)\phi(z)^\top] + \sigma_b^2$$

One solution is perfect correlation,

$$\Sigma^* = q^* \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

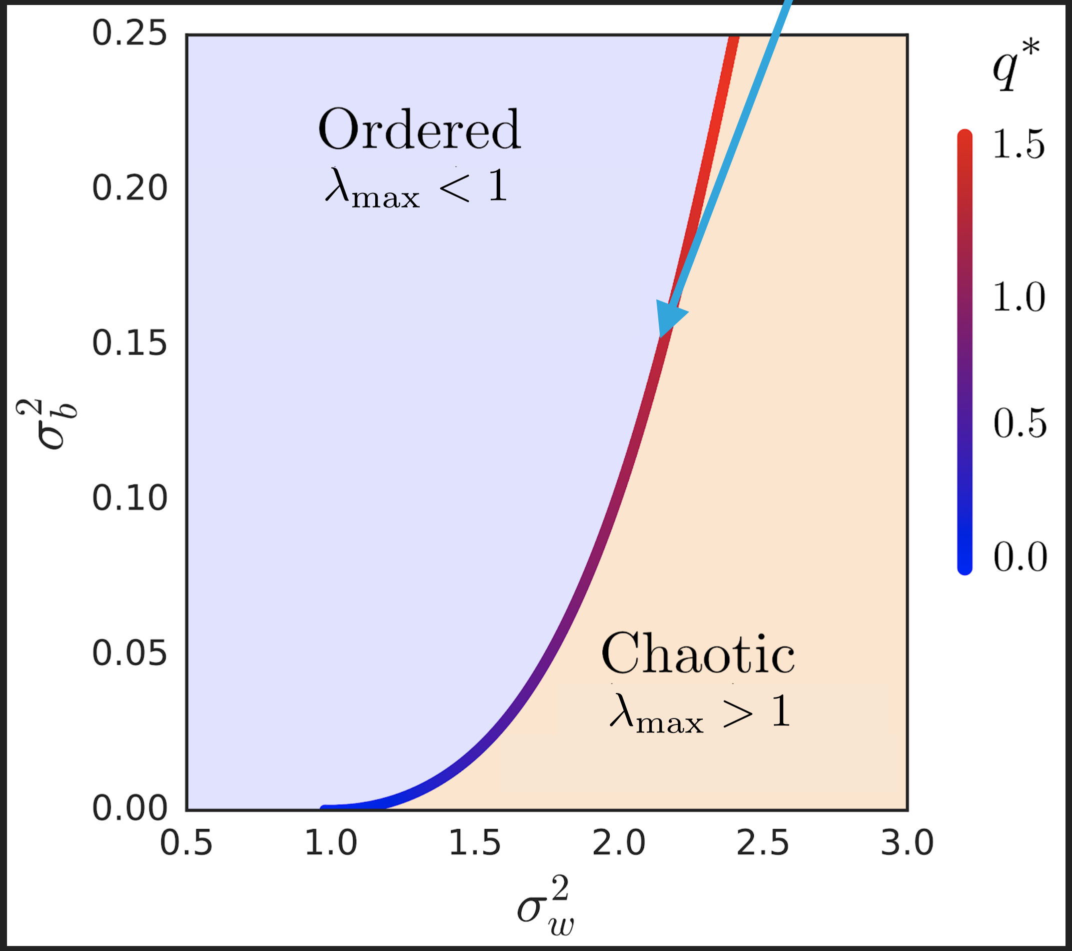
Variance

Correlation

Is this fixed point stable?

- Depends on hyperparameters

$\lambda_{\max} = 1$
$\xi = \infty$



BP, SL, MR, JSD, SG ('16)
SSS, JG, SG, JSD ('17)

THE EDGE OF CHAOS

For **deep** signal propagation, initialize on the "edge of chaos"

BP, SL, MR, JSD, SG ('16)
SSS, JG, SG, JSD ('17)

THE EDGE OF CHAOS

For **deep** signal propagation, initialize on the "edge of chaos"

- Analyze the fixed point and set $\lambda_{\max} = 1$

BP, SL, MR, JSD, SG ('16)
SSS, JG, SG, JSD ('17)

THE EDGE OF CHAOS

For **deep** signal propagation, initialize on the "edge of chaos"

- Analyze the fixed point and set $\lambda_{\max} = 1$

$$\Sigma^* = q^* \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \qquad\qquad \lambda_{\max}\left(\left. \frac{\partial \mathcal{C}(\Sigma)}{\partial \Sigma} \right|_{\Sigma^*} \right) = \chi(\sigma_w, \sigma_b)$$

THE EDGE OF CHAOS

For **deep** signal propagation, initialize on the "edge of chaos"

- Analyze the fixed point and set $\lambda_{\max} = 1$

$$\Sigma^* = q^* \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

$$\lambda_{\max}\left(\left. \frac{\partial \mathcal{C}(\Sigma)}{\partial \Sigma} \right|_{\Sigma^*} \right) = \chi(\sigma_w, \sigma_b)$$

$$\chi(\sigma_w, \sigma_b) = \sigma_w^2 \int dz \, \frac{e^{-\frac{1}{2}z^2}}{\sqrt{2\pi}} \phi'(\sqrt{q^*}z)^2$$

BACKPROPAGATED GRADIENTS

Given a loss $\mathcal{L}$, back-propagation gives

$$\frac{\partial \mathcal{L}}{\partial W_{ij}^l} = \delta_i^l \phi(z_j^{l-1}) \qquad \delta_i^l = \frac{\partial \mathcal{L}}{\partial z_i^l} \qquad \delta_i^l = \phi'(z_i^l) \sum_j \delta_j^{l+1} W_{ji}^{l+1}$$

BACKPROPAGATED GRADIENTS

Given a loss $\mathcal{L}$, back-propagation gives

$$\frac{\partial \mathcal{L}}{\partial W_{ij}^l} = \delta_i^l \phi(z_j^{l-1}) \qquad \delta_i^l = \frac{\partial \mathcal{L}}{\partial z_i^l} \qquad \delta_i^l = \phi'(z_i^l) \sum_j \delta_j^{l+1} W_{ji}^{l+1}$$

Gradients scale like

$$\mathbb{E}[(\delta_1^l)^2] = \mathbb{E}[(\delta_1^{l+1})^2] \sigma_w^2 \mathbb{E}[\phi'(z_1^{l+1})^2]$$

BACKPROPAGATED GRADIENTS

Given a loss $\mathcal{L}$, back-propagation gives

$$\frac{\partial \mathcal{L}}{\partial W_{ij}^l} = \delta_i^l \phi(z_j^{l-1}) \qquad \delta_i^l = \frac{\partial \mathcal{L}}{\partial z_i^l} \qquad \delta_i^l = \phi'(z_i^l) \sum_j \delta_j^{l+1} W_{ji}^{l+1}$$

Gradients scale like

$$\mathbb{E}[(\delta_1^l)^2] = \mathbb{E}[(\delta_1^{l+1})^2] \underbrace{\sigma_w^2 \mathbb{E}[\phi'(z_1^{l+1})^2]}_{\chi(\sigma_w, \sigma_b)}$$

$$\mathbb{E}[(\delta_1^1)^2] = \mathbb{E}[(\delta_1^L)^2] \chi(\sigma_w, \sigma_b)^L$$

BP, SL, MR, JSD, SG ('16)
SSS, JG, SG, JSD ('17)

BACKPROPAGATED GRADIENTS

Given a loss $\mathcal{L}$, back-propagation gives

$$\frac{\partial \mathcal{L}}{\partial W_{ij}^l} = \delta_i^l \phi(z_j^{l-1}) \qquad \delta_i^l = \frac{\partial \mathcal{L}}{\partial z_i^l} \qquad \delta_i^l = \phi'(z_i^l) \sum_j \delta_j^{l+1} W_{ji}^{l+1}$$

Gradients scale like

$$\mathbb{E}[(\delta_1^l)^2] = \mathbb{E}[(\delta_1^{l+1})^2] \underbrace{\sigma_w^2 \mathbb{E}[\phi'(z_1^{l+1})^2]}_{\chi(\sigma_w, \sigma_b)}$$

$$\mathbb{E}[(\delta_1^1)^2] = \mathbb{E}[(\delta_1^L)^2] \chi(\sigma_w, \sigma_b)^L$$

Gradients explode/vanish unless

$$\boxed{\chi(\sigma_w, \sigma_b) = 1}$$

BP, SL, MR, JSD, SG ('16)
SSS, JG, SG, JSD ('17)

BACKPROPAGATED GRADIENTS

Given a loss $\mathcal{L}$, back-propagation gives

$$\frac{\partial \mathcal{L}}{\partial W_{ij}^l} = \delta_i^l \phi(z_j^{l-1}) \qquad \delta_i^l = \frac{\partial \mathcal{L}}{\partial z_i^l} \qquad \delta_i^l = \phi'(z_i^l) \sum_j \delta_j^{l+1} W_{ji}^{l+1}$$

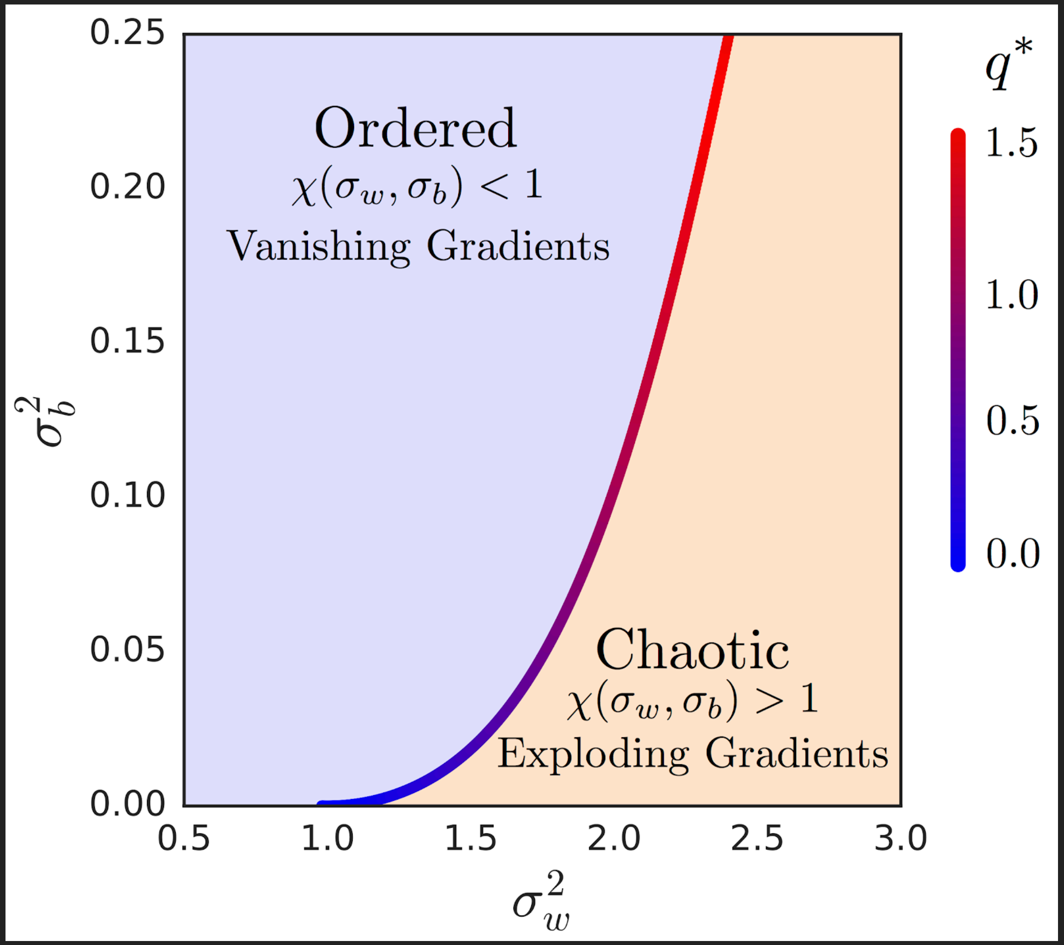
Gradients scale like

$$\mathbb{E}[(\delta_1^l)^2] = \mathbb{E}[(\delta_1^{l+1})^2] \underbrace{\sigma_w^2 \mathbb{E}[\phi'(z_1^{l+1})^2]}_{\chi(\sigma_w, \sigma_b)}$$

$$\mathbb{E}[(\delta_1^1)^2] = \mathbb{E}[(\delta_1^L)^2] \chi(\sigma_w, \sigma_b)^L$$

Gradients explode/vanish unless

$$\boxed{\chi(\sigma_w, \sigma_b) = 1}$$



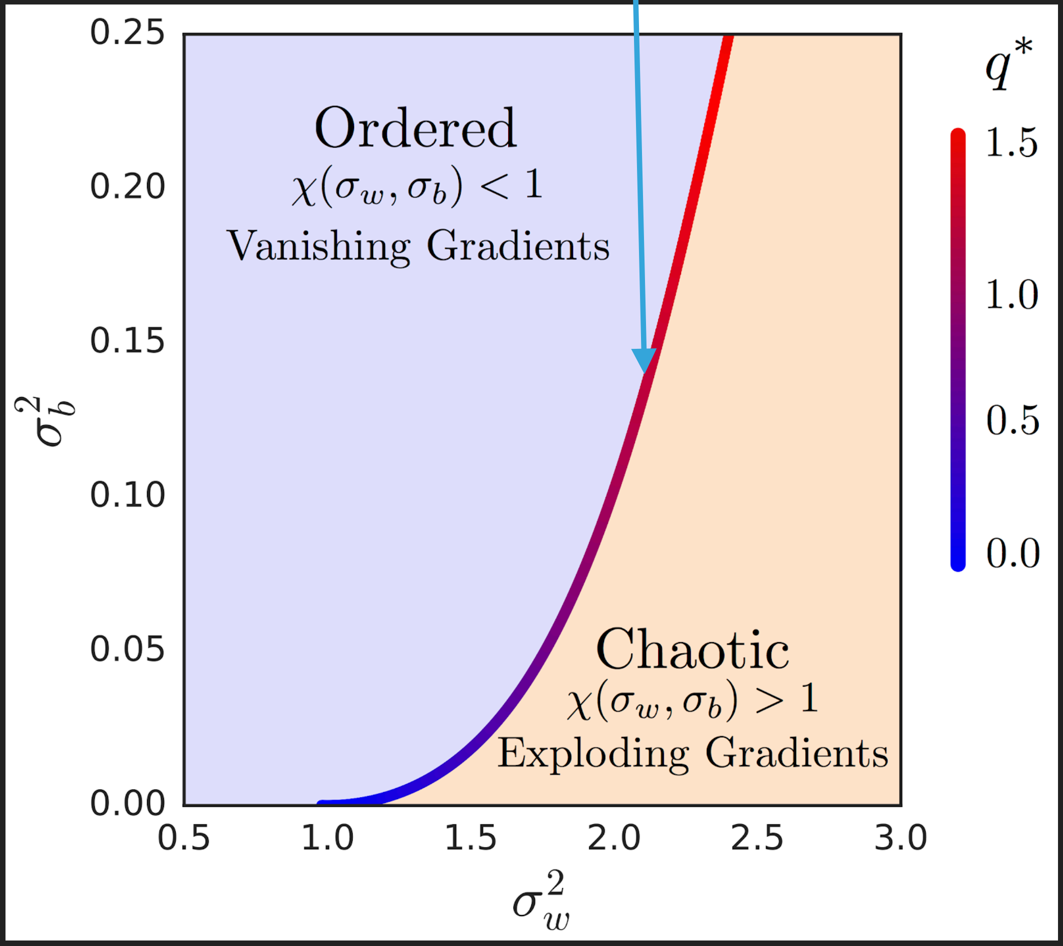
BP, SL, MR, JSD, SG ('16)
SSS, JG, SG, JSD ('17)

CRITICAL INITIALIZATION

Critical line

Critical initialization:

In order for signals to propagate forward and backward through a deep network, the initialization hyperparameters should lie on the critical line

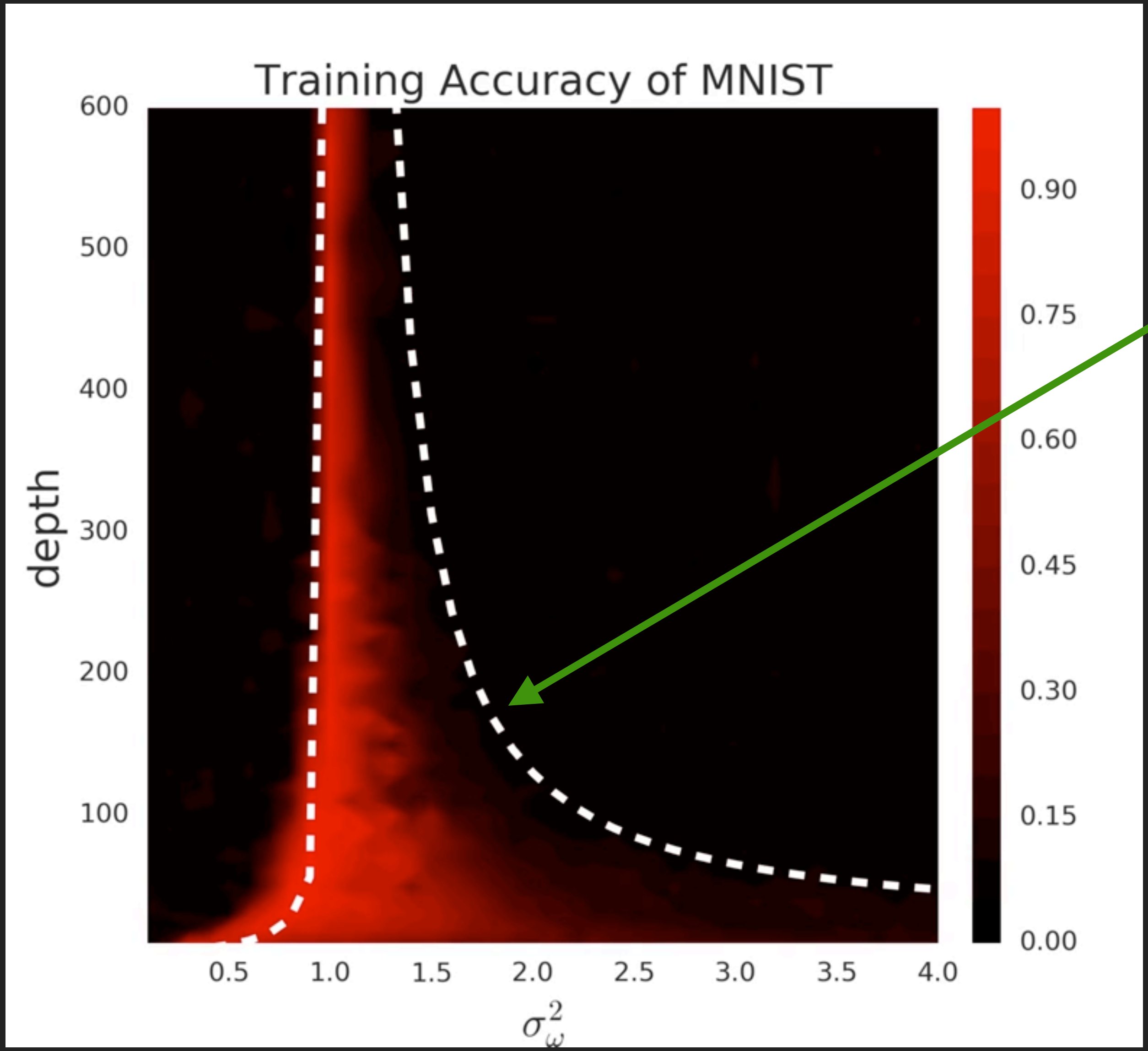


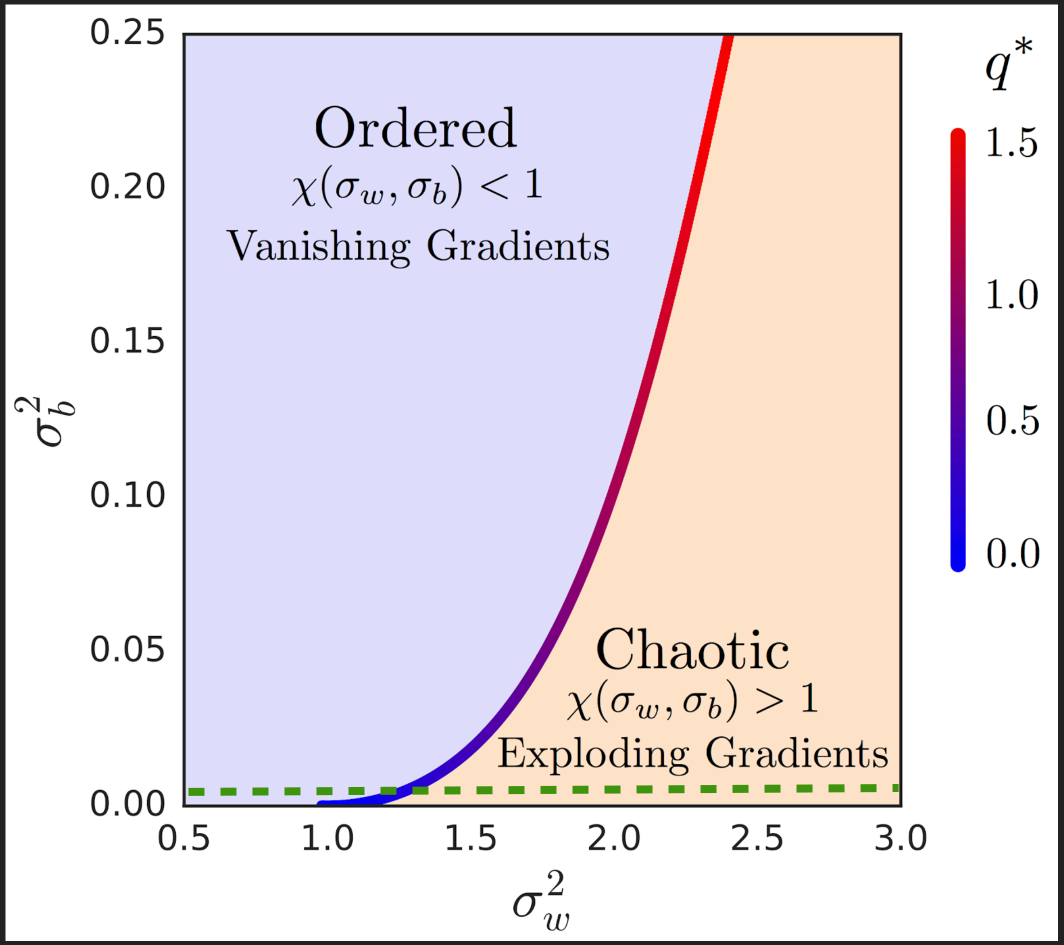
BP, SL, MR, JSD, SG ('16)
SSS, JG, SG, JSD ('17)

PREDICTING TRAINABLE DEPTH

$$\epsilon^l \approx e^{-l/\xi}$$

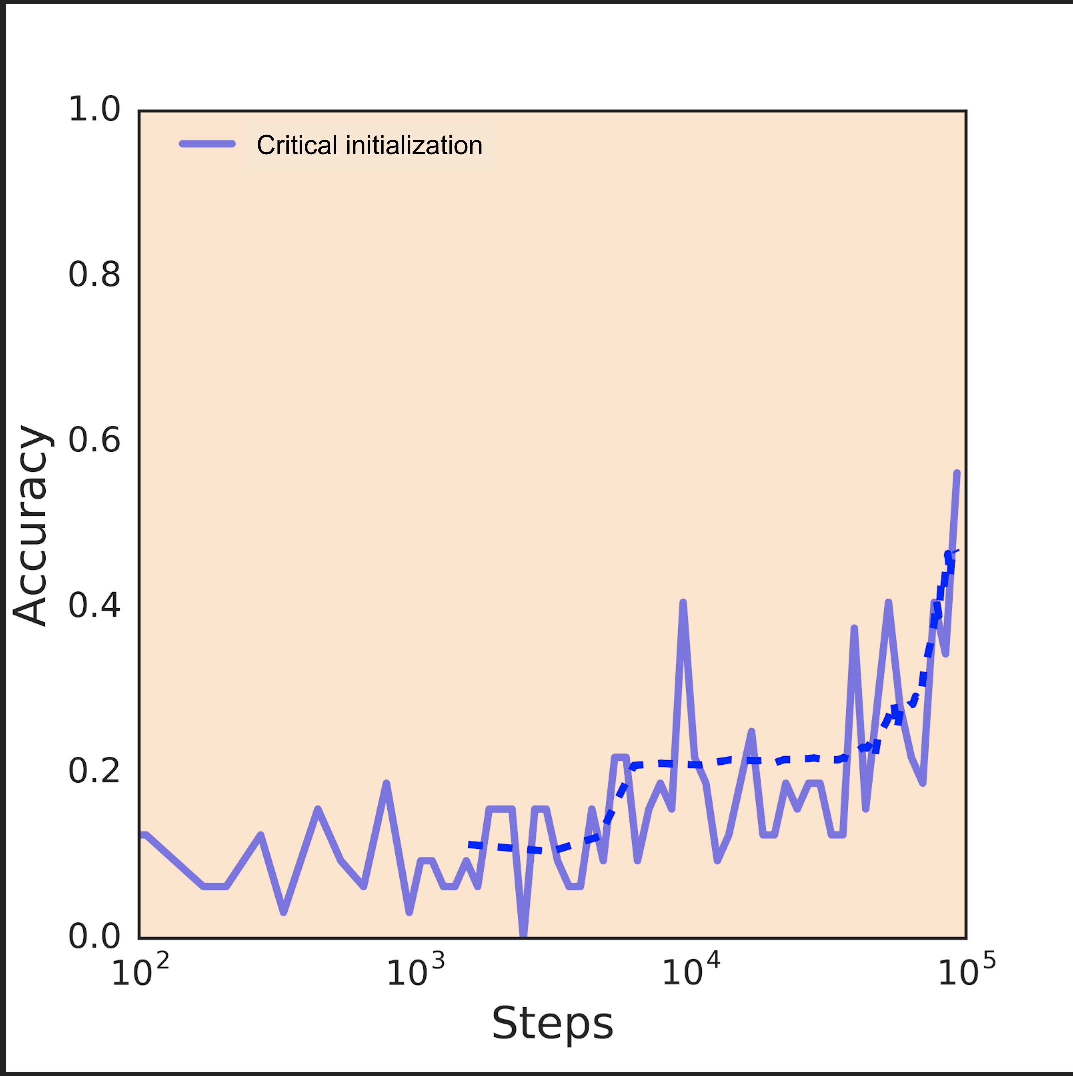
$$\boxed{\xi(\sigma_w) = -1/\log \lambda_{\max}(\sigma_w)}$$



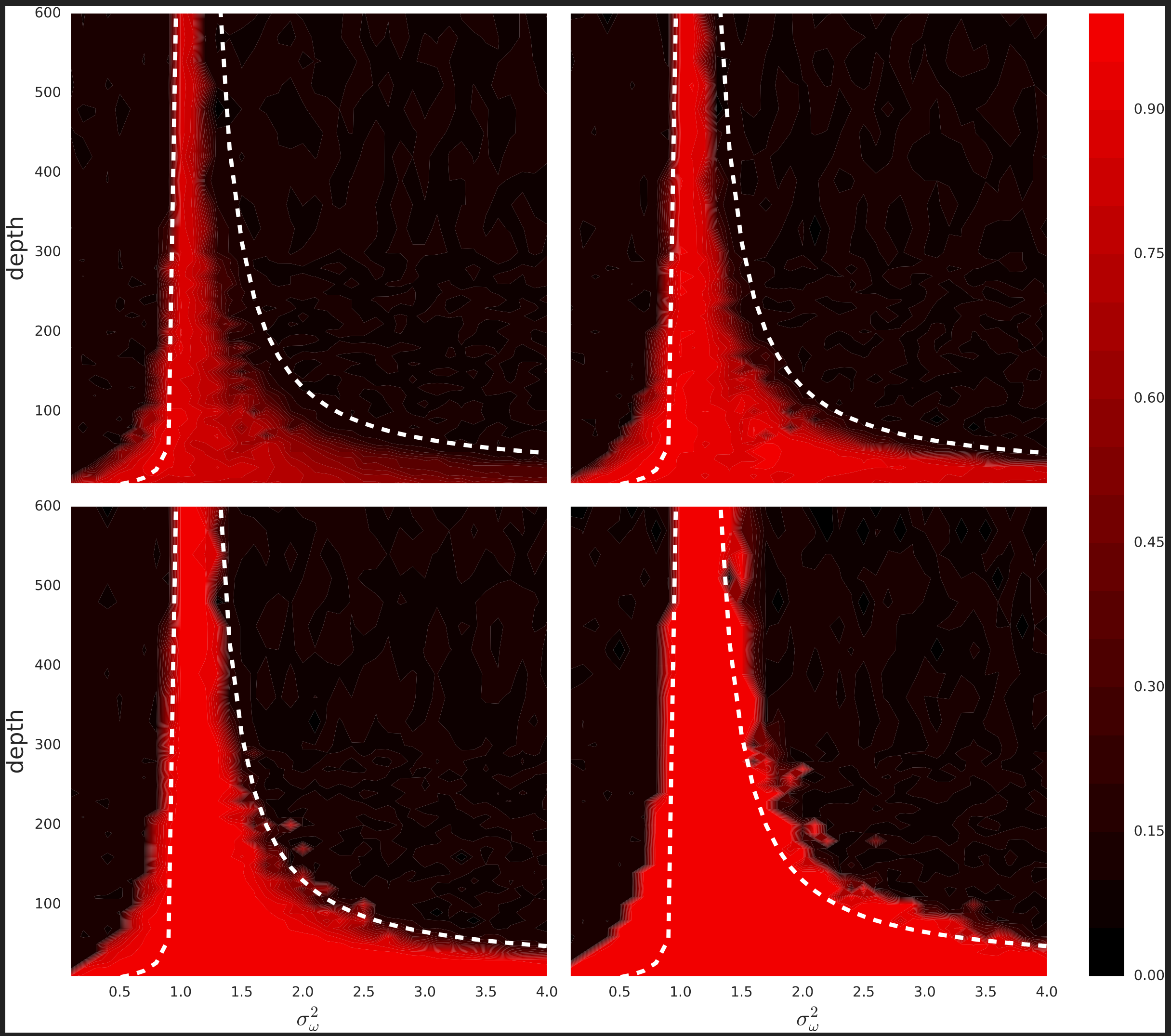


TRAINABILITY OF VERY DEEP NETWORKS

4000-layer CNN on MNIST



Trainability heat maps



OUTLINE

DYNAMICAL ISOMETRY

Study the end-to-end Jacobian

$$\boldsymbol{J} = \frac{\partial \boldsymbol{z}^L}{\partial \boldsymbol{z}^0} = \prod_l \boldsymbol{D}^l \boldsymbol{W}^l \qquad D_{ij}^l = \phi'(z_i^l)\delta_{ij}$$

Diagonal Matrix

DYNAMICAL ISOMETRY

Study the end-to-end Jacobian

$$\boldsymbol{J} = \frac{\partial \boldsymbol{z}^L}{\partial \boldsymbol{z}^0} = \prod_l \boldsymbol{D}^l \boldsymbol{W}^l \qquad D^l_{ij} = \phi'(z^l_i)\delta_{ij}$$

Diagonal Matrix

A few relations that make this interesting

$$\boldsymbol{\delta}^0 = \boldsymbol{J}\boldsymbol{\delta}^L \qquad f(\boldsymbol{x}+\boldsymbol{\delta}) \approx f(\boldsymbol{x}) + \boldsymbol{J}^T\boldsymbol{\delta} \qquad G = \boldsymbol{J}^T\boldsymbol{J}$$

Gradients Linear Response Induced Metric

DYNAMICAL ISOMETRY

Study the end-to-end Jacobian

$$\boldsymbol{J} = \frac{\partial \boldsymbol{z}^L}{\partial \boldsymbol{z}^0} = \prod_l \boldsymbol{D}^l \boldsymbol{W}^l \qquad D^l_{ij} = \phi'(z^l_i)\delta_{ij}$$

Diagonal Matrix

A few relations that make this interesting

$$\boldsymbol{\delta}^0 = \boldsymbol{J}\boldsymbol{\delta}^L \qquad f(\boldsymbol{x} + \boldsymbol{\delta}) \approx f(\boldsymbol{x}) + \boldsymbol{J}^T\boldsymbol{\delta} \qquad G = \boldsymbol{J}^T\boldsymbol{J}$$

Gradients — Linear Response — Induced Metric

We have worked out behavior of gradients on average:

Criticality $\Leftrightarrow$ $\mathbb{E}[\mathrm{tr}(J^T J)] = \chi(\sigma_w, \sigma_b)^L = 1$

DYNAMICAL ISOMETRY

Study the end-to-end Jacobian

$$\boldsymbol{J} = \frac{\partial \boldsymbol{z}^L}{\partial \boldsymbol{z}^0} = \prod_l \boldsymbol{D}^l \boldsymbol{W}^l \qquad D^l_{ij} = \phi'(z^l_i)\delta_{ij}$$

Diagonal Matrix

A few relations that make this interesting

$$\boldsymbol{\delta}^0 = \boldsymbol{J}\boldsymbol{\delta}^L \qquad f(\boldsymbol{x}+\boldsymbol{\delta}) \approx f(\boldsymbol{x}) + \boldsymbol{J}^T\boldsymbol{\delta} \qquad G = \boldsymbol{J}^T\boldsymbol{J}$$

Gradients — Linear Response — Induced Metric

We have worked out behavior of gradients on average:

Criticality $\Leftrightarrow$ $\mathbb{E}[\mathrm{tr}(J^TJ)] = \chi(\sigma_w, \sigma_b)^L = 1$

But what is a good prior for the whole spectrum?

DYNAMICAL ISOMETRY

Study the end-to-end Jacobian

$$\boldsymbol{J} = \frac{\partial \boldsymbol{z}^L}{\partial \boldsymbol{z}^0} = \prod_l \boldsymbol{D}^l \boldsymbol{W}^l \qquad D^l_{ij} = \phi'(z^l_i)\delta_{ij}$$

Diagonal Matrix

A few relations that make this interesting

$$\delta^0 = \boldsymbol{J}\boldsymbol{\delta}^L \qquad f(\boldsymbol{x}+\boldsymbol{\delta}) \approx f(\boldsymbol{x}) + \boldsymbol{J}^T\boldsymbol{\delta} \qquad G = \boldsymbol{J}^T\boldsymbol{J}$$

Gradients — Linear Response — Induced Metric

We have worked out behavior of gradients on average:

Criticality $\Leftrightarrow$ $\mathbb{E}[\mathrm{tr}(J^TJ)] = \chi(\sigma_w, \sigma_b)^L = 1$

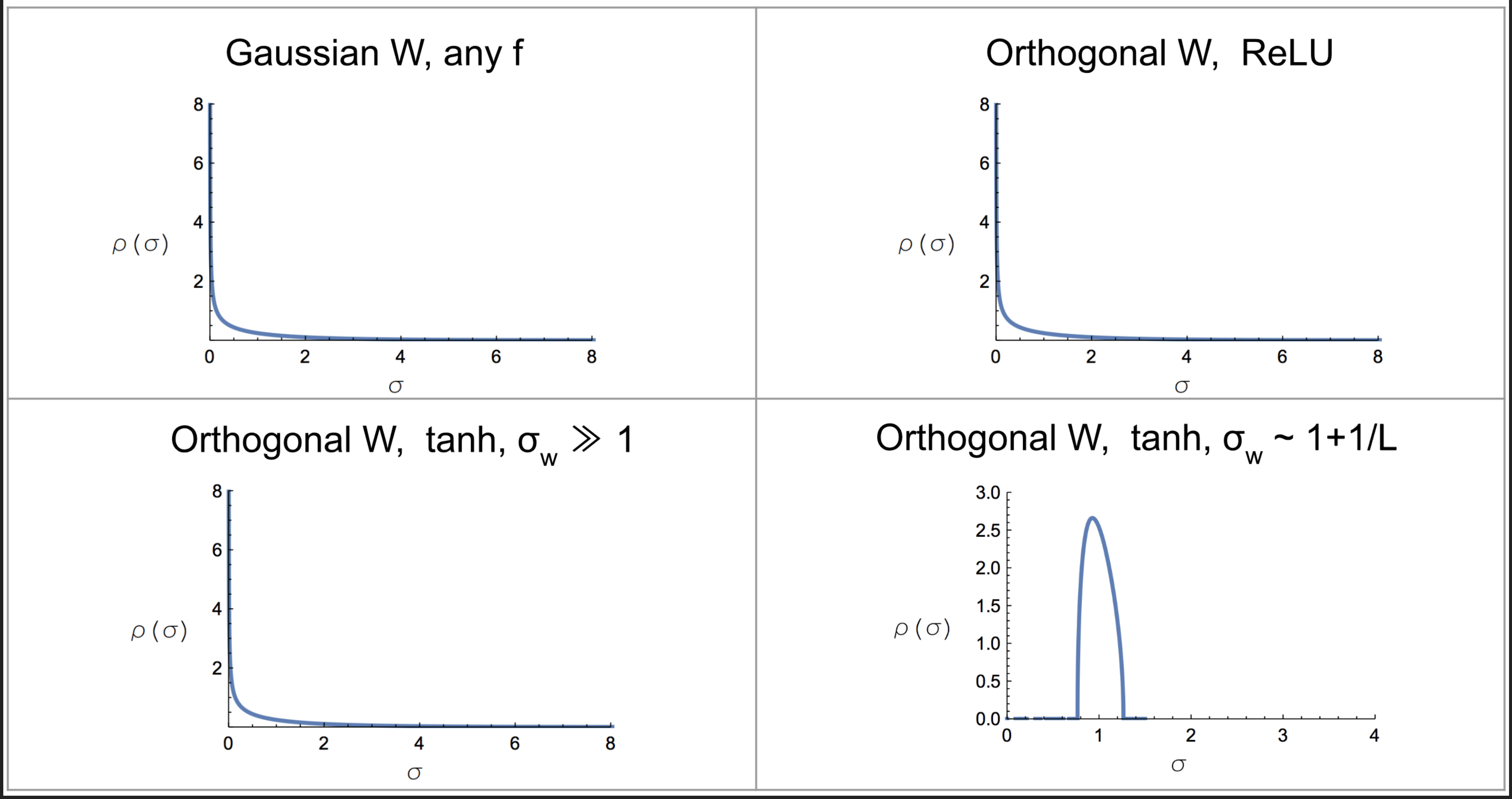
But what is a good prior for the whole spectrum?

- Isometry: all singular values ≈ 1

JP, SSS, SG ('17); JP SS, SG ('18)

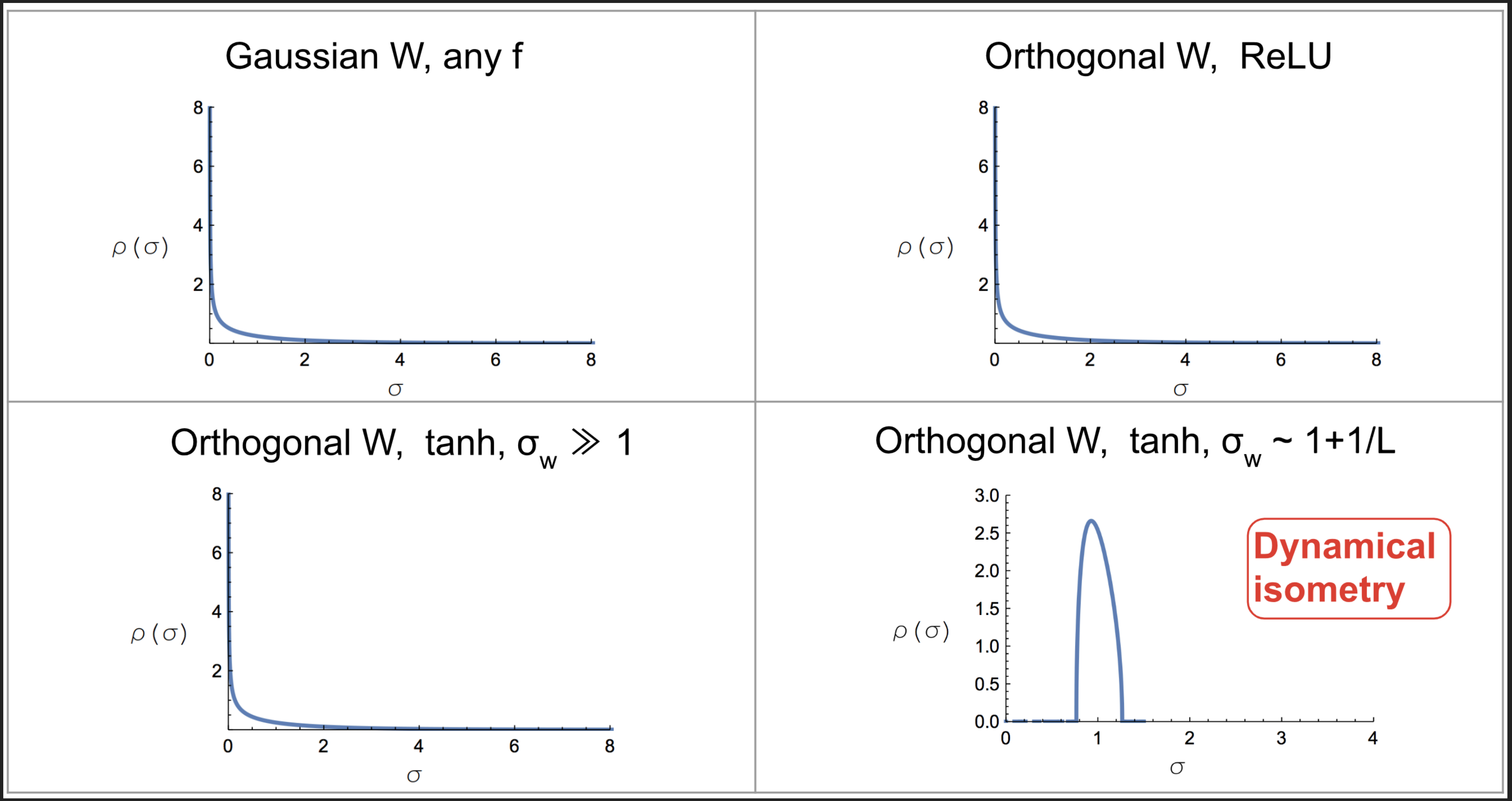
COMPUTING THE SPECTRUM

Using tools from random matrix theory (free probability), can compute spectrum analytically:



COMPUTING THE SPECTRUM

Using tools from random matrix theory (free probability), can compute spectrum analytically:

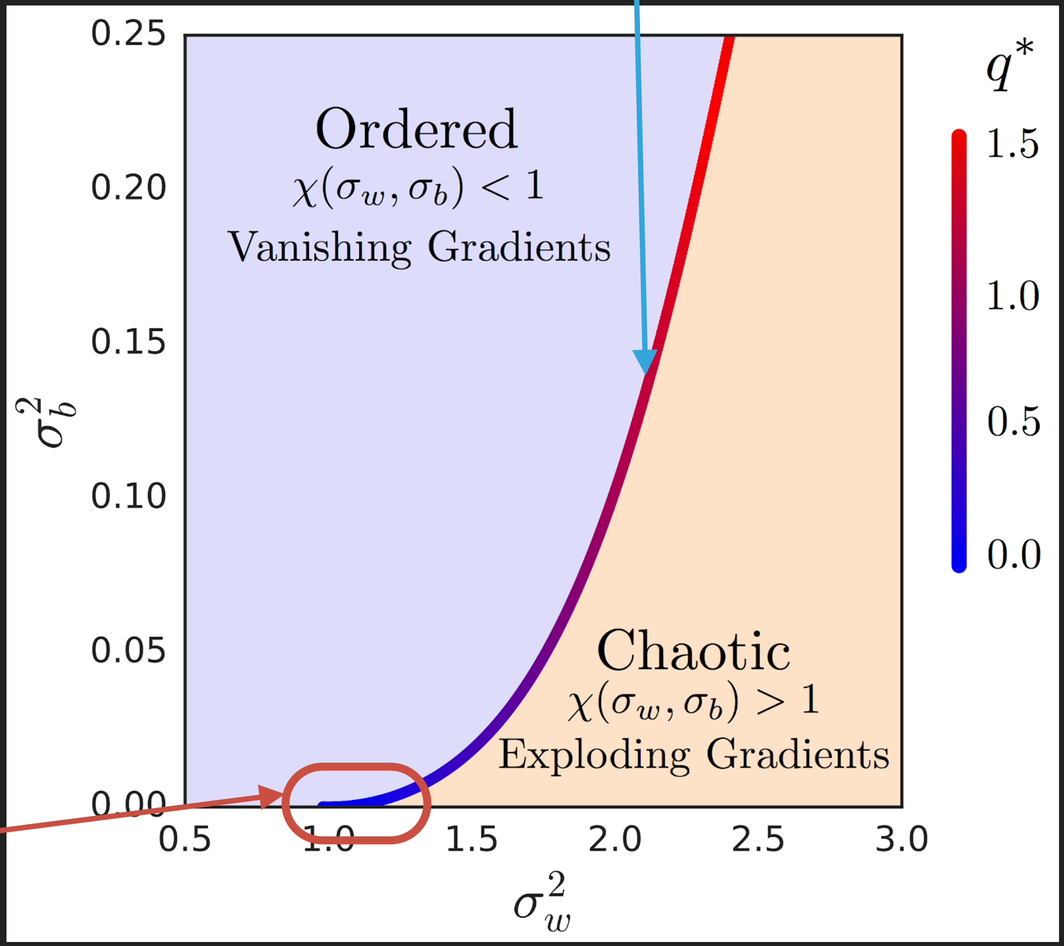


BP, SL, MR, JSD, SG ('16)
SSS, JG, SG, JSD ('17)

DYNAMICAL ISOMETRY

Not every point on the critical line is equally favorable for gradient propagation. For activation functions that are linear near the origin, dynamical isometry (i.e. well-conditioned Jacobians) can be achieved with small bias variance.

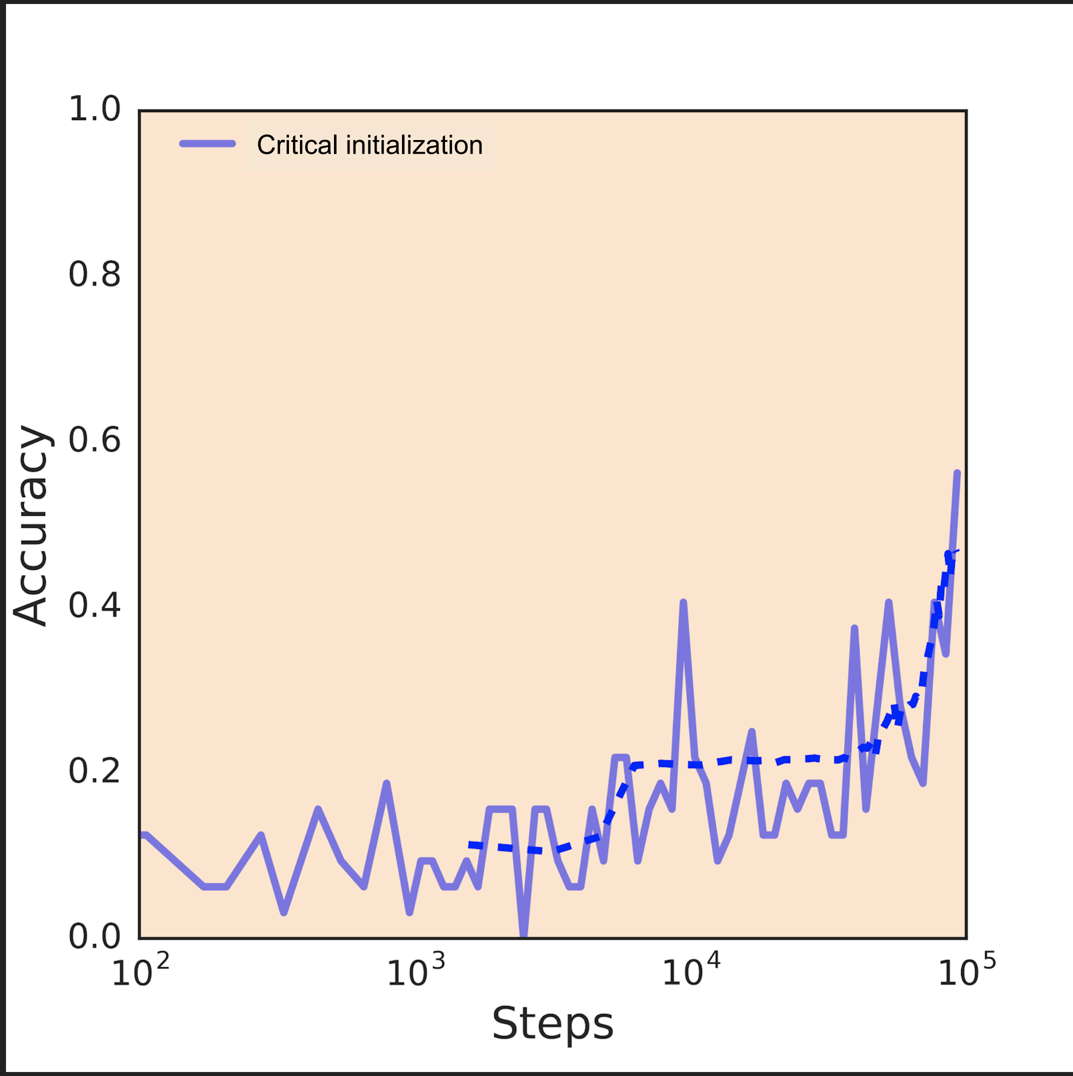
Critical line



Dynamical isometry

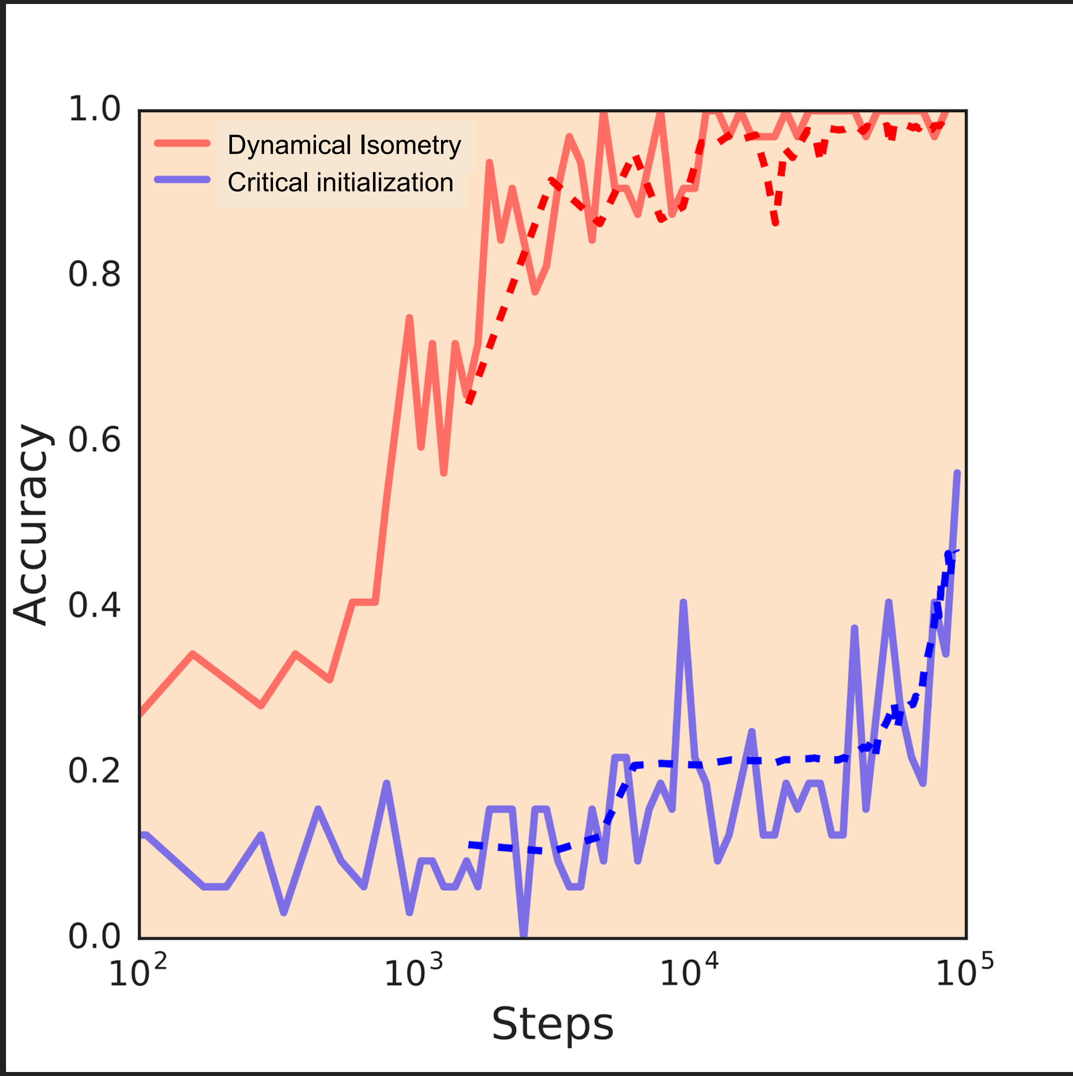
THE BENEFITS OF A BETTER PRIOR

4000-layer CNN on MNIST



THE BENEFITS OF A BETTER PRIOR

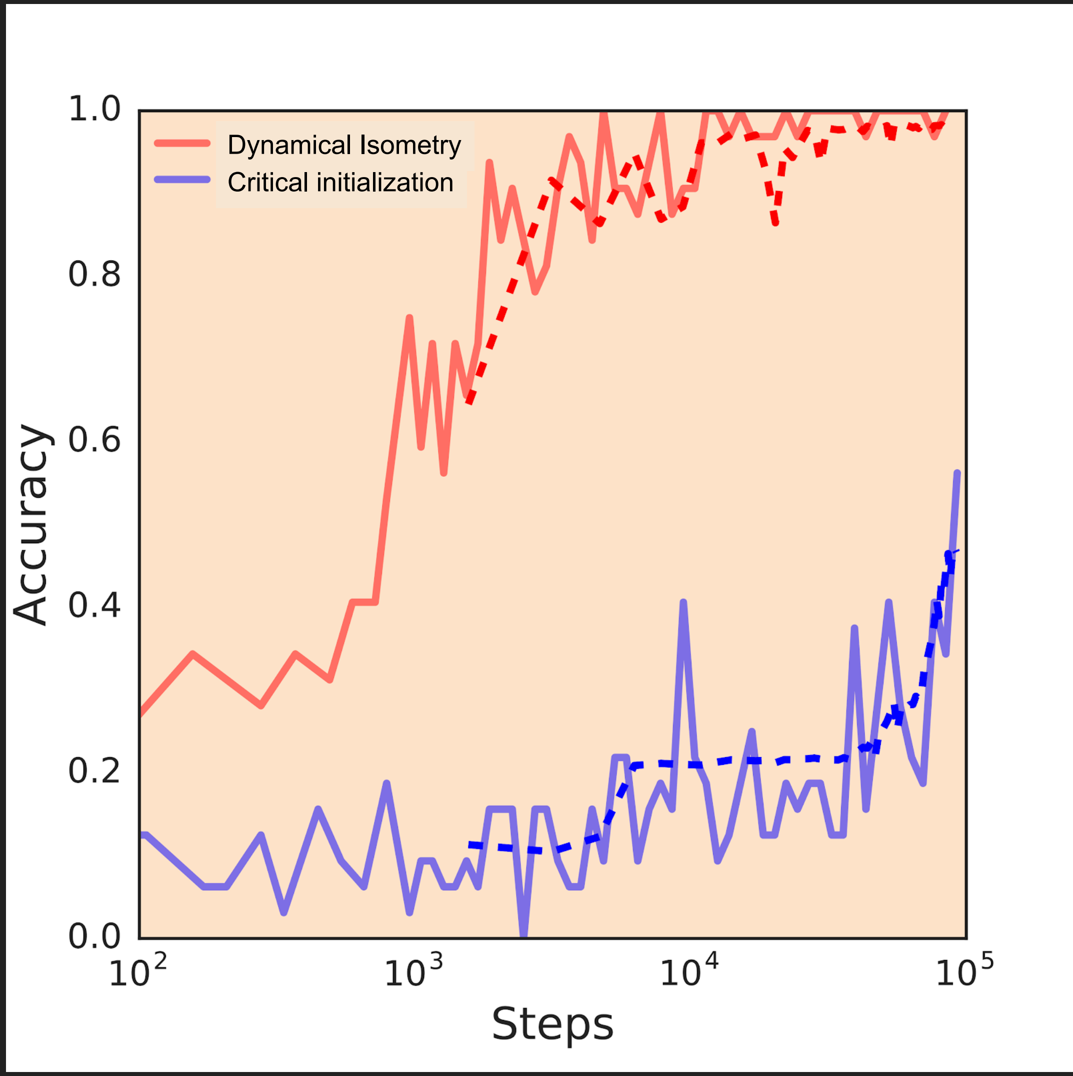
4000-layer CNN on MNIST



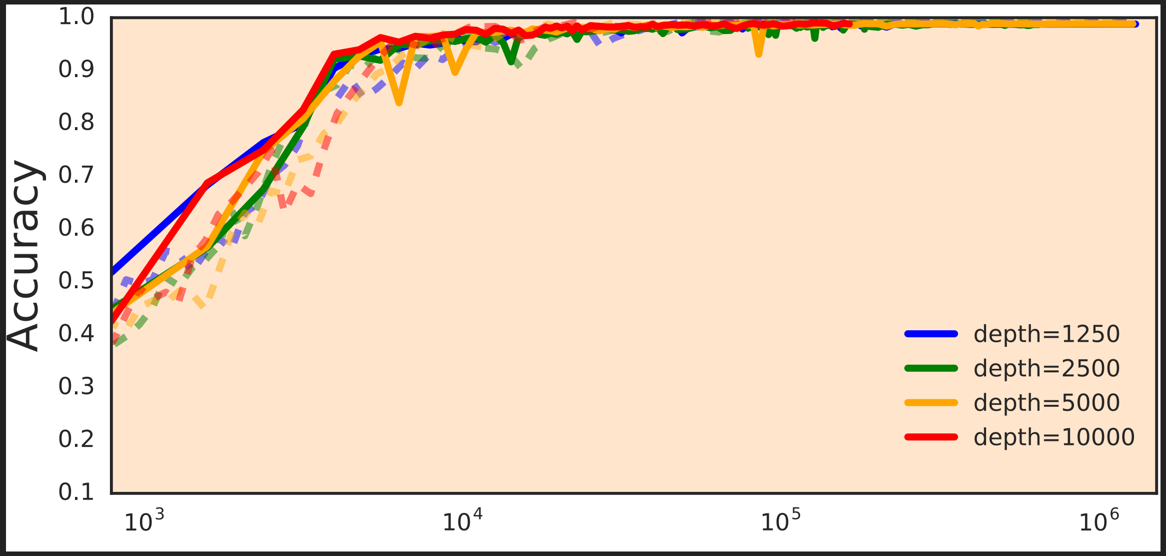
LC, YB, JSD, SSS, JP ('18)

THE BENEFITS OF A BETTER PRIOR

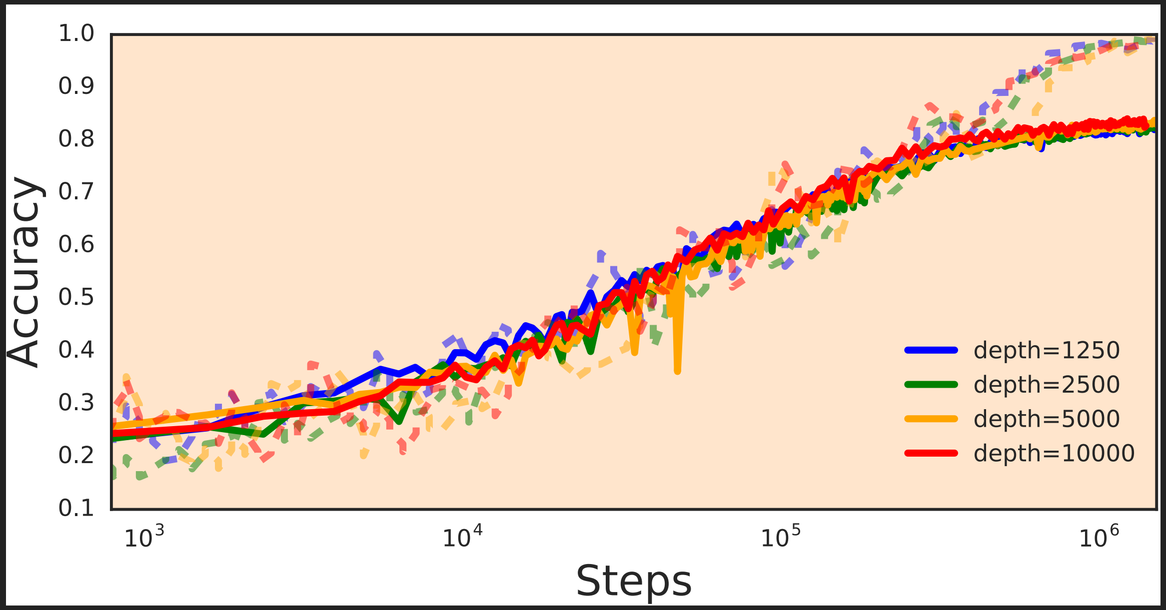
4000-layer CNN on MNIST



MNIST



CIFAR-10



OUTLINE

1. Motivation
2. Functional priors
3. Signal propagation
4. Dynamical isometry
5. Functional posteriors
6. Conclusion

AJ, FG, CH ('18)
JL, XLC, SSS, YB, JSD, JP ('19)

WIDE, DEEP, NETWORKS EVOLVE AS LINEAR MODELS

Consider a FC neural network, $f(x;\theta(\underset{\uparrow}{t}))$

Gradient Descent Time

AJ, FG, CH ('18)
JL, XLC, SSS, YB, JSD, JP ('19)

WIDE, DEEP, NETWORKS EVOLVE AS LINEAR MODELS

Consider a FC neural network, $f(x; \theta(\underbrace{t}_{\text{Gradient Descent Time}}))$

As the width of the network grows, parameters move less during gradient descent

AJ, FG, CH ('18)
JL, XLC, SSS, YB, JSD, JP ('19)

WIDE, DEEP, NETWORKS EVOLVE AS LINEAR MODELS

Consider a FC neural network, $f(x;\theta(t))$

Gradient Descent Time

As the width of the network grows, parameters move less during gradient descent

Motivates a linear approximation,

$$f(x;\theta(t)) \approx f(x;\theta(0)) + \sum_\alpha \frac{\partial f(x;\theta(0))}{\partial \theta_\alpha(0)}(\theta(t)-\theta(0)) + \mathcal{O}((\theta(t)-\theta(0))^2)$$

Function at Initialization

Jacobian at Initialization

This becomes exact as $N \to \infty$

AJ, FG, CH ('18)
JL, XLC, SSS, YB, JSD, JP ('19)

WIDE, DEEP, NETWORKS EVOLVE AS LINEAR MODELS

Consider a FC neural network, $f(x; \theta(t))$

Gradient Descent Time

As the width of the network grows, parameters move less during gradient descent

Motivates a linear approximation,

Jacobian at Initialization

$$f_t(x) \approx f_0(x) + J_0(x)\omega(t) \qquad \omega(t) = \theta(t) - \theta(0)$$

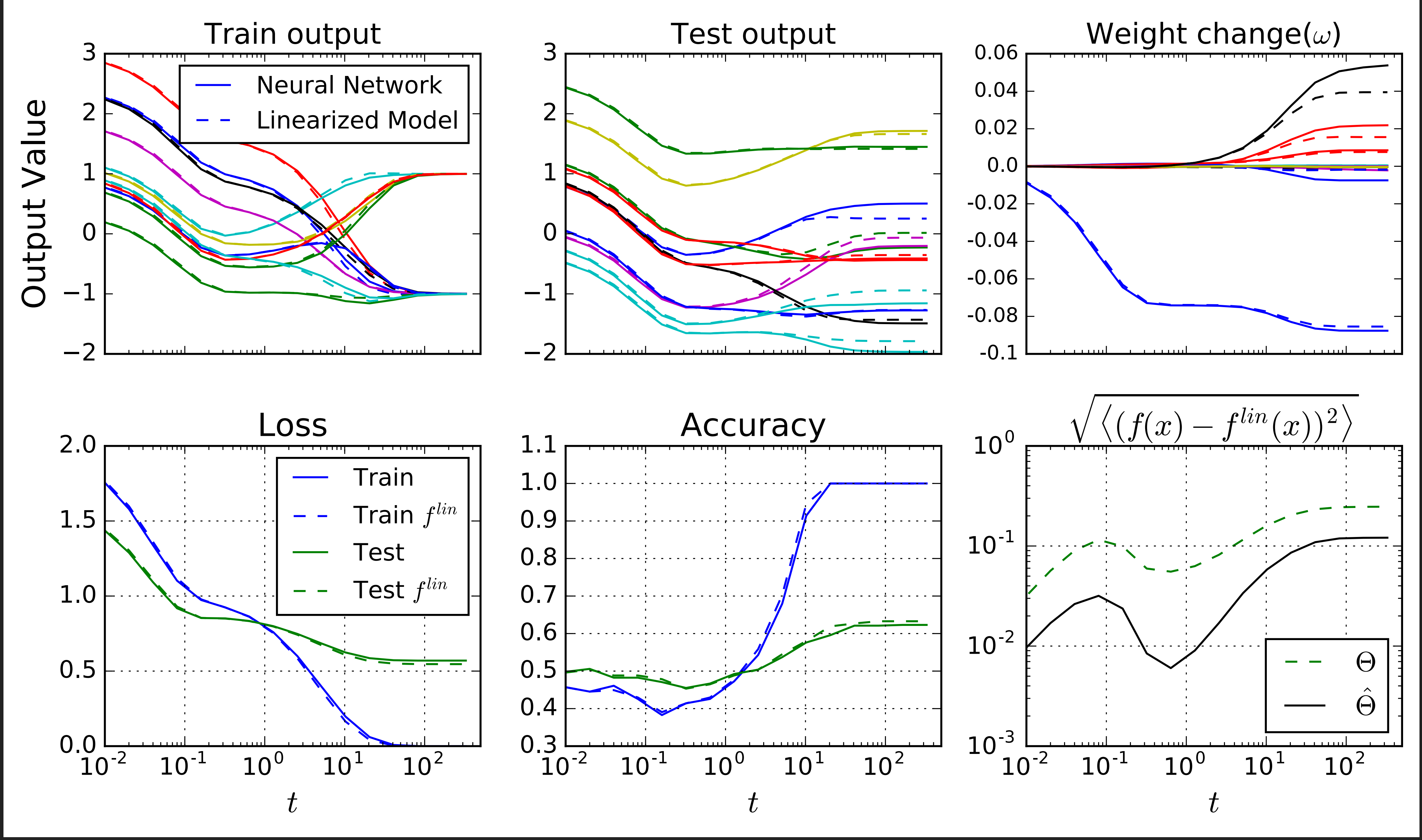
Function at Initialization

This becomes exact as $N \to \infty$

AJ, FG, CH ('18)
JL, XLC, SSS, YB, JSD, JP ('19)

WIDE, DEEP, NETWORKS EVOLVE AS LINEAR MODELS

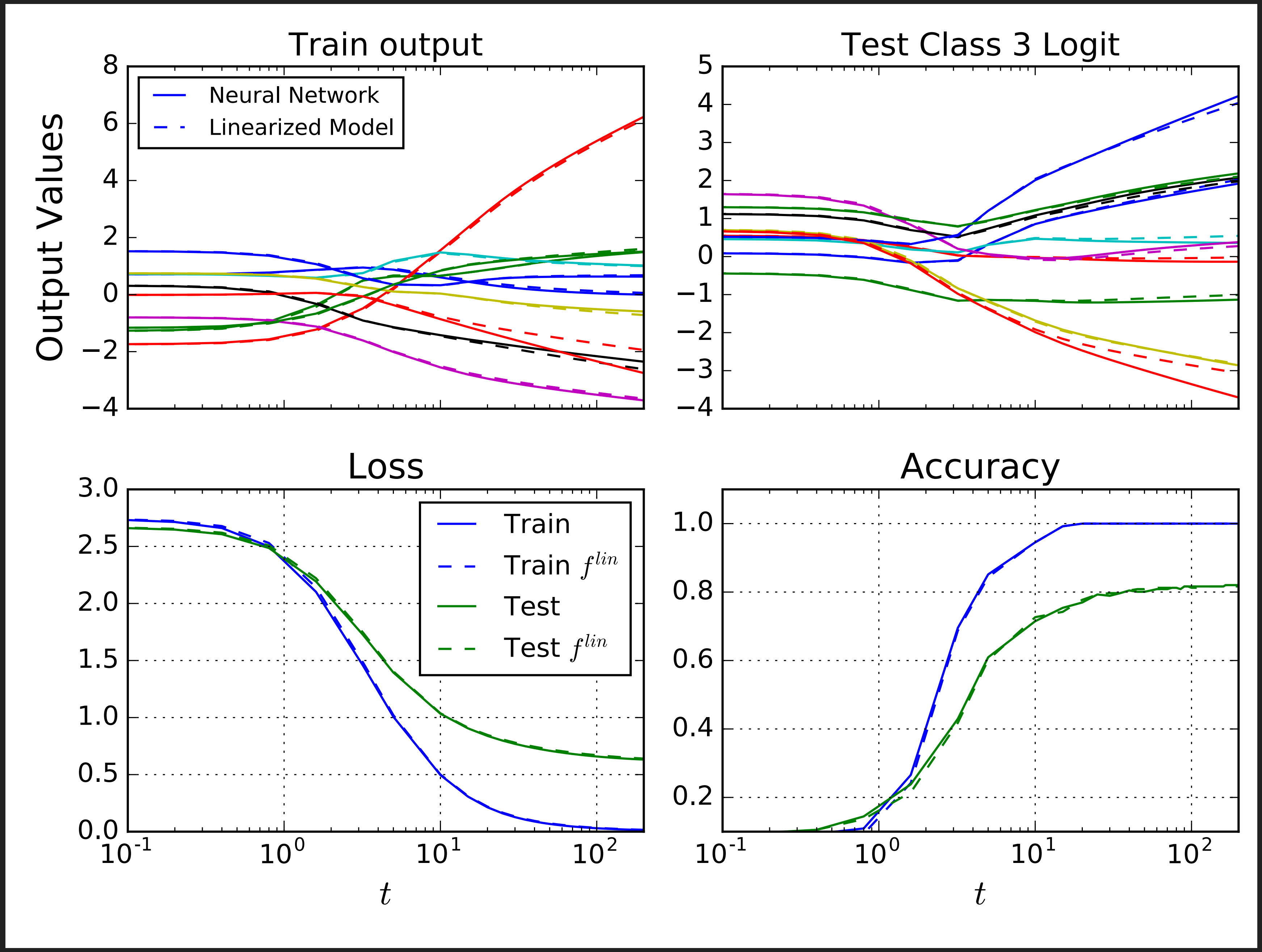
Fully Connected, N=2048, Single Output, MSE Loss, Gradient Descent



AJ, FG, CH ('18)
JL, XLC, SSS, YB, JSD, JP ('19)

WIDE, DEEP, NETWORKS EVOLVE AS LINEAR MODELS

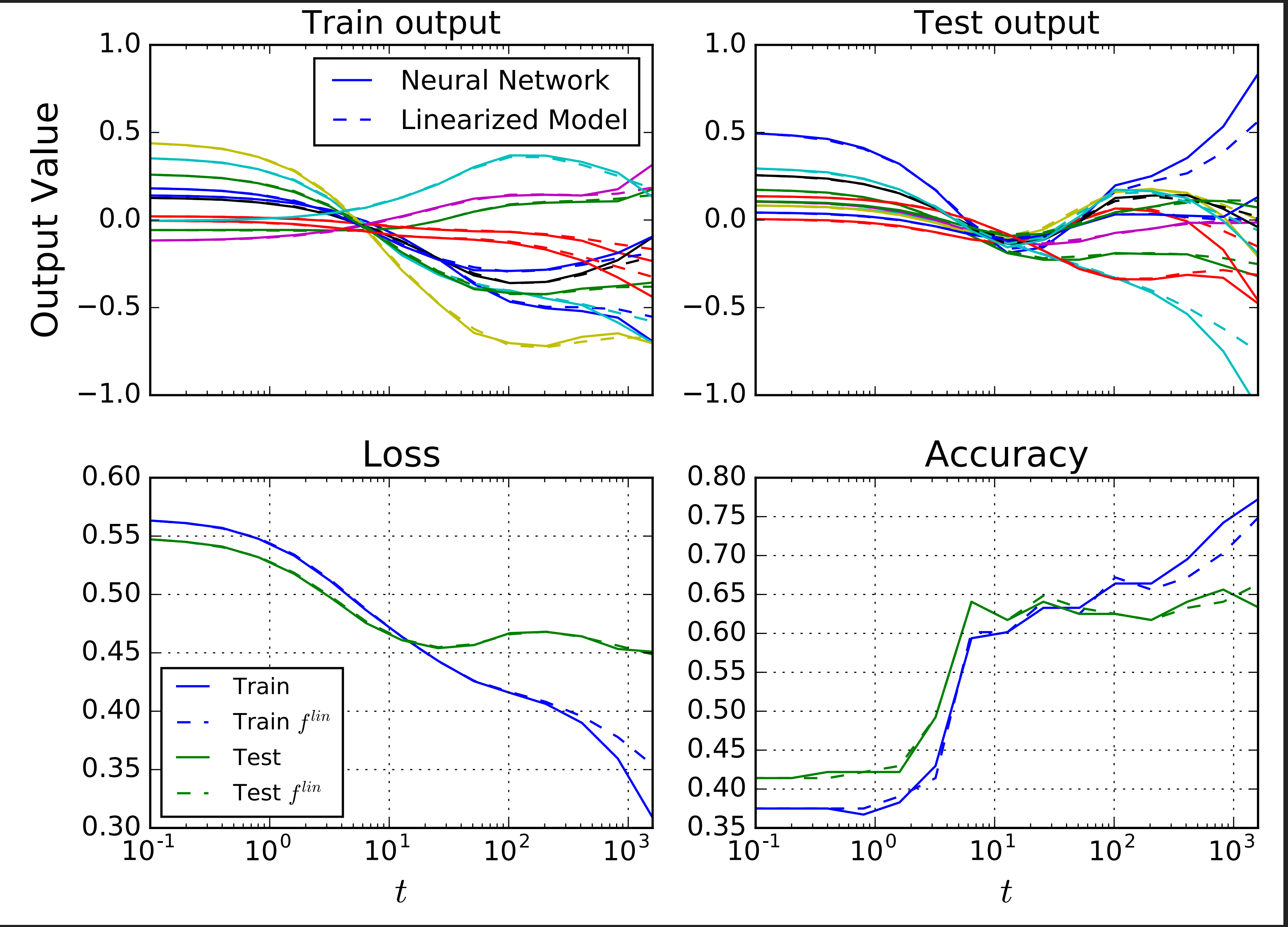
Fully Connected, N=1024, 10-Class, Cross Entropy Loss, Momentum



AJ, FG, CH ('18)
JL, XLC, SSS, YB, JSD, JP ('19)

WIDE, DEEP, NETWORKS EVOLVE AS LINEAR MODELS

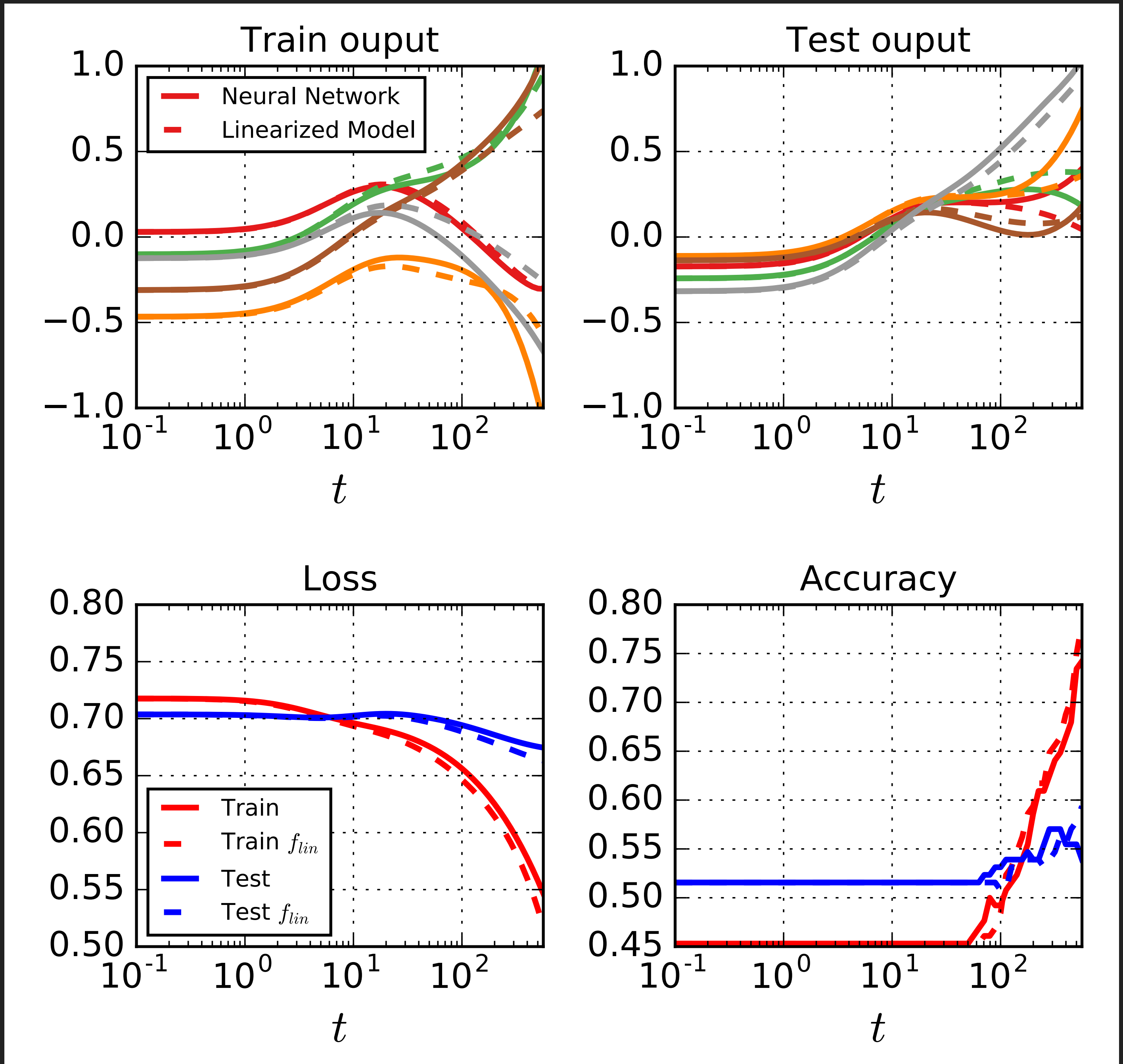
CNN, C=256, 2-Class, MSE Loss, GD



AJ, FG, CH ('18)
JL, XLC, SSS, YB, JSD, JP ('19)

WIDE, DEEP, NETWORKS EVOLVE AS LINEAR MODELS

Wide Resnet (10-layers), C=1024, 2-Class, Cross Entropy Loss, Momentum



AJ, FG, CH ('18)
JL, XLC, SSS, YB, JSD, JP ('19)

IMPLICATIONS FOR THE POSTERIOR

For MSE Loss,

$$\partial_t f_t(X) = -\Theta(X,X)(f_t(X) - Y) \qquad \Theta(X,Y) = \frac{1}{M} J_0(X) J_0(Y)^T$$

Neural Tangent Kernel

AJ, FG, CH ('18)
JL, XLC, SSS, YB, JSD, JP ('19)

IMPLICATIONS FOR THE POSTERIOR

For MSE Loss,

$$\partial_t f_t(X) = -\Theta(X,X)(f_t(X) - Y) \qquad \Theta(X,Y) = \frac{1}{M} J_0(X) J_0(Y)^T$$

Neural Tangent Kernel

This allows us to compute the "posterior" after *t* steps of GD,

$$\mu(x) = \Theta(x,X)\Theta^{-1}(I - e^{-\eta\Theta t})Y$$

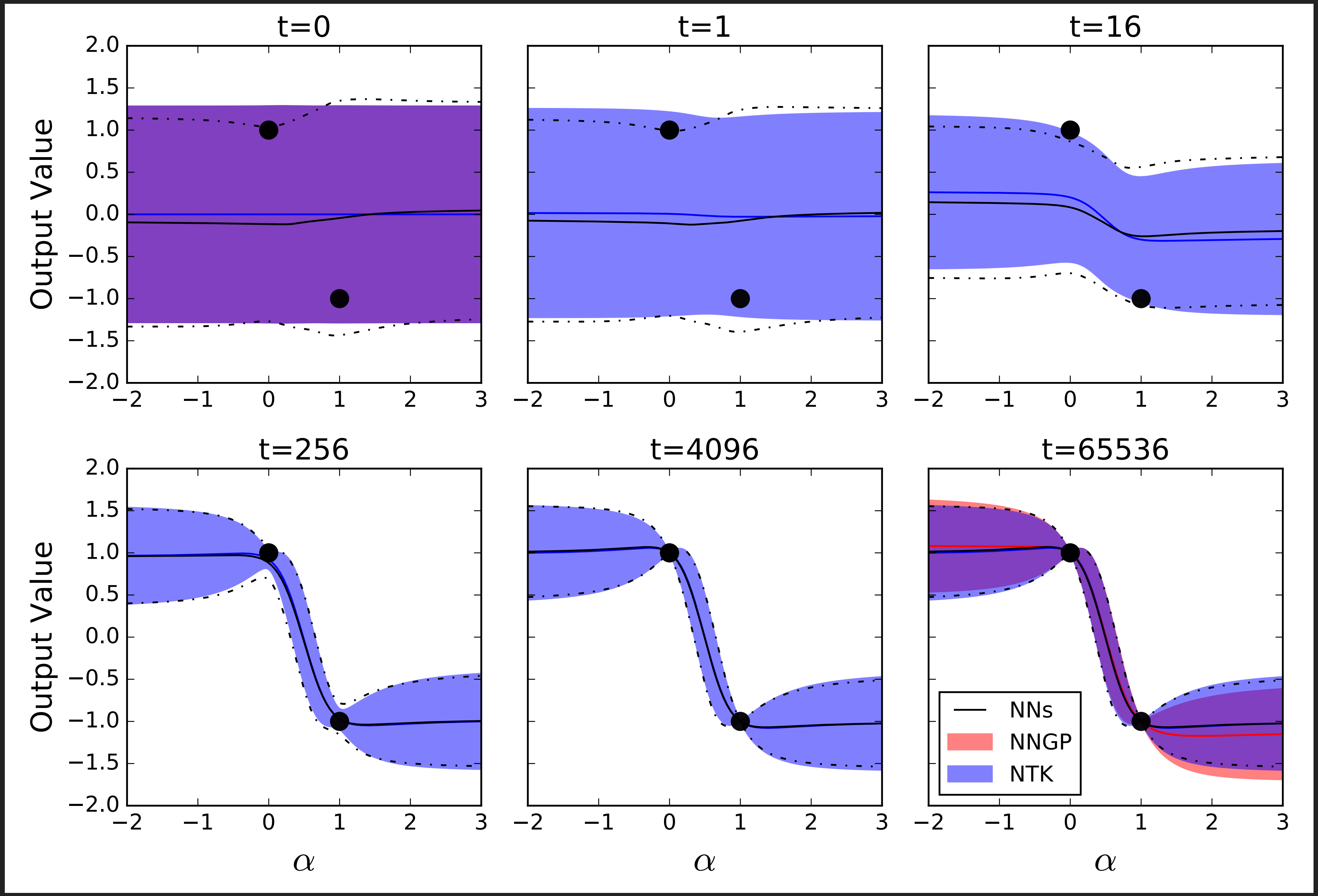
$$\Sigma(x) = K(x,x) - 2\Theta(x,X)\Theta^{-1}(I - e^{-\eta\Theta t})K(x,X)^T + \Theta(x,X)\Theta^{-1}(I - e^{-\eta\Theta t})K\Theta^{-1}(I - e^{-\eta\Theta t})\Theta(x,X)^T$$

NNGP Kernel

AJ, FG, CH ('18)
JL, XLC, SSS, YB, JSD, JP ('19)

IMPLICATIONS FOR THE POSTERIOR

FC Network, N=8192, MNIST, MSE Loss



CONCLUSIONS

Overparameterized models are simple!

The prior over functions can be computed analytically

Properties of the prior are intimately related to trainability

Wide neural networks are almost linear models

Overall, a powerful framework is emerging for theoretically analyzing overparameterized neural networks