
# 理解神经网络的训练过程

许志钦

上海交通大学

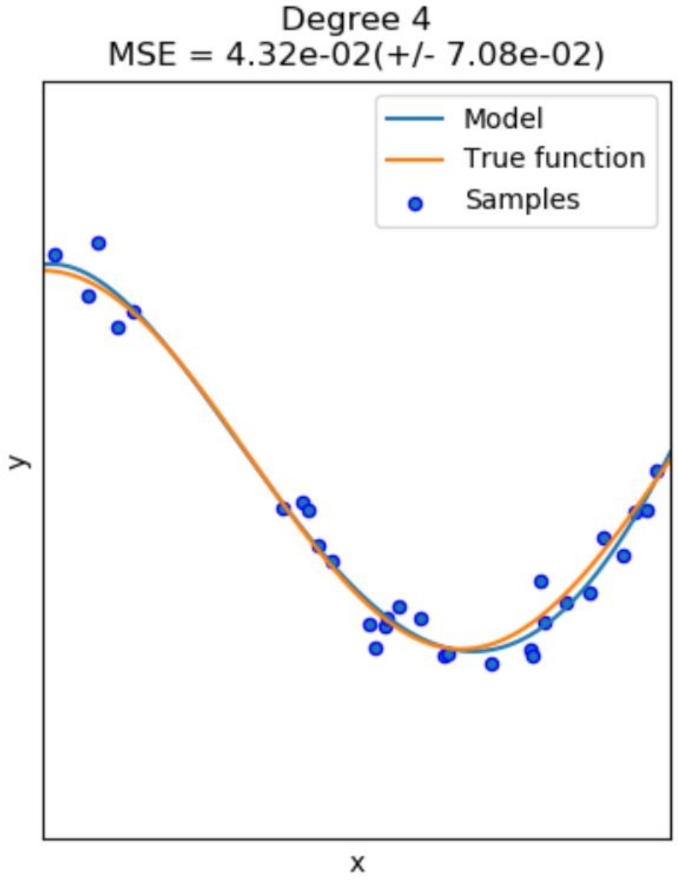
2021.4.17

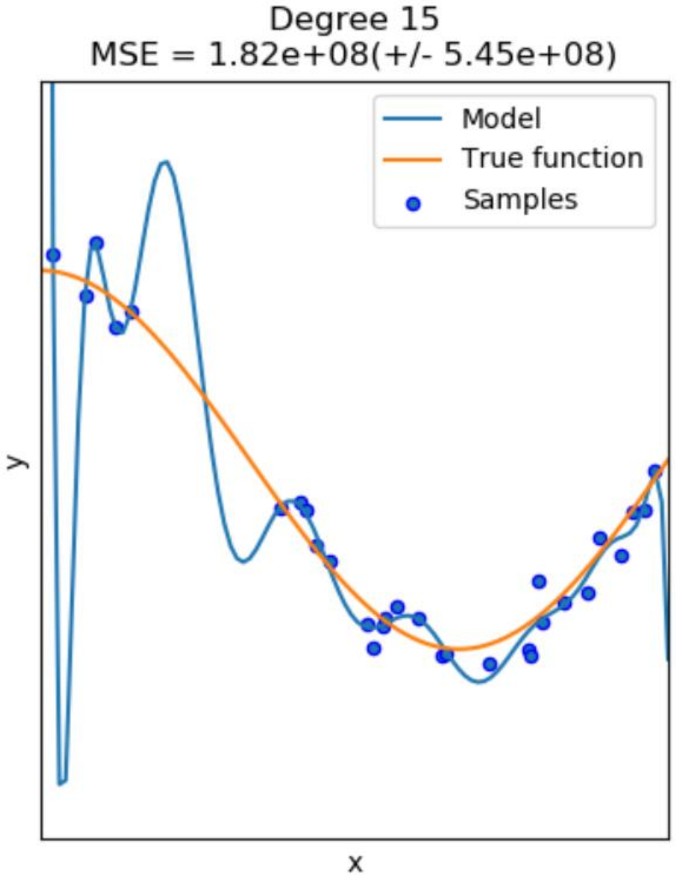
机器学习联合研讨计划

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

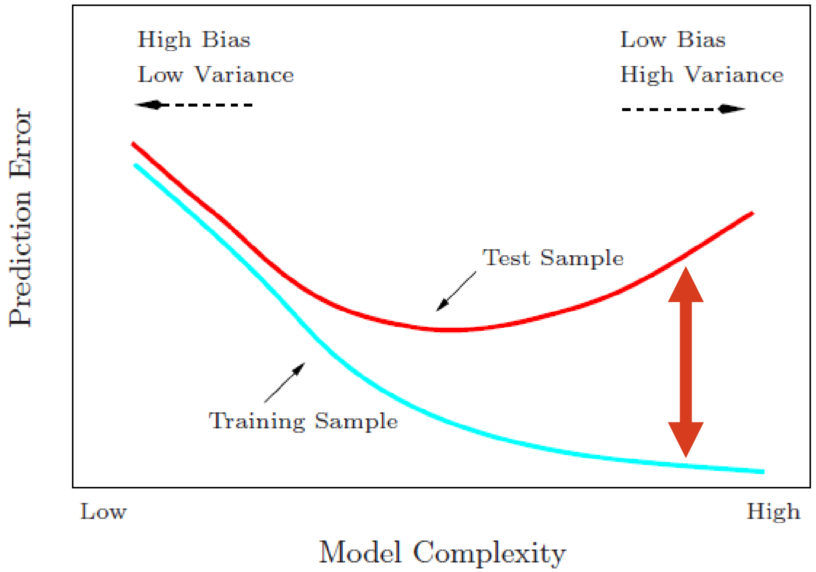
# Start from traditional generalization gap

## Large complexity → Large generalization gap




Runge phenomenon




Generalization Gap

# A generalization puzzle arises in deep learning

**Puzzle: generalize well even # of para >> # of training data**

Table 1: The training and test accuracy (in percentage) of various models on the CIFAR10 dataset. Performance with and without data augmentation and weight decay are compared. The results of fitting random labels are also included.

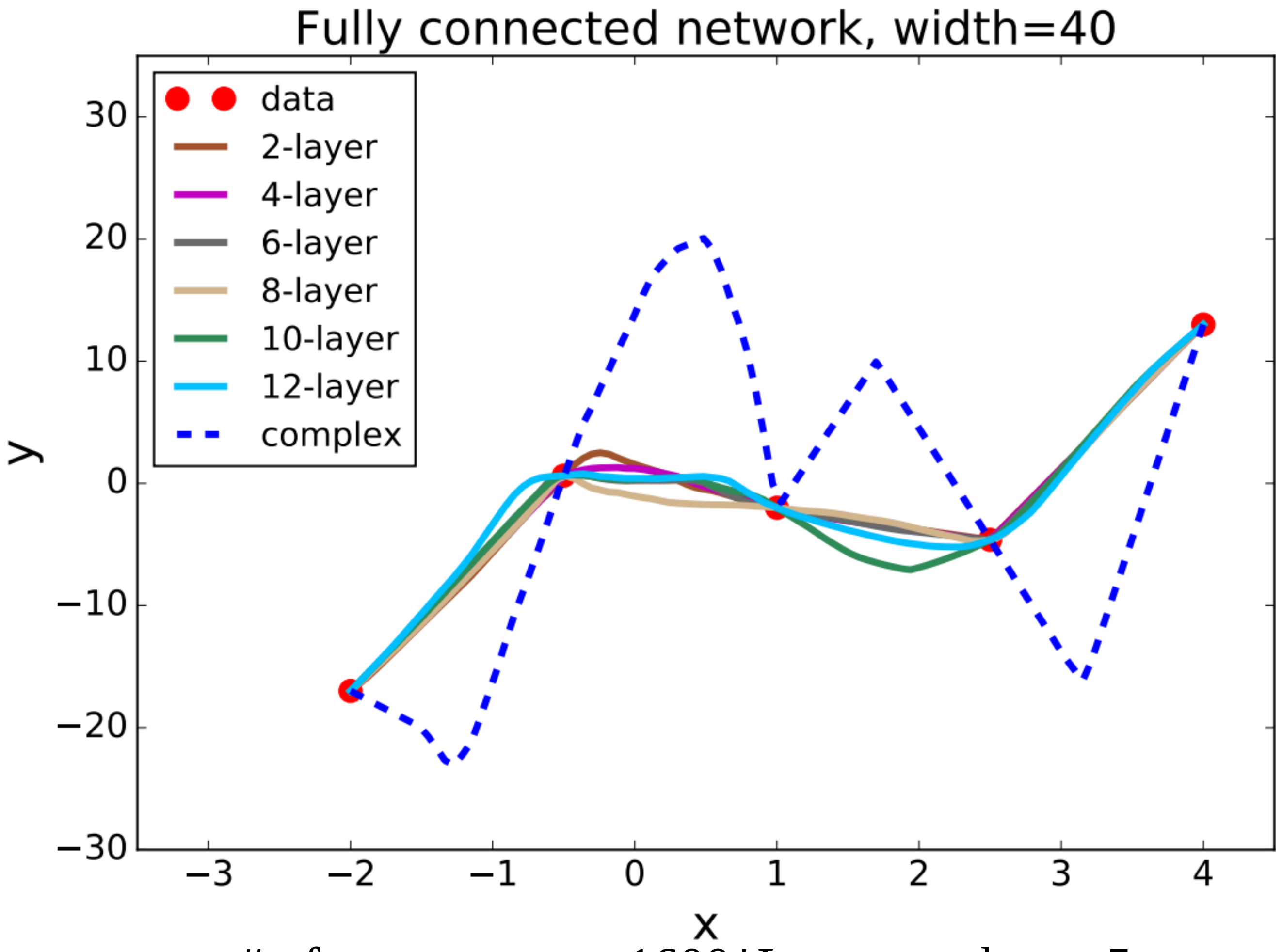
| model | # params | random crop | weight decay | train accuracy | test accuracy |
|---|---|---|---|---|---|
| Inception | 1,649,402 | yes | yes | 100.0 | 89.05 |
| | | yes | no | 100.0 | 89.31 |
| | | no | yes | 100.0 | 86.03 |
| | | no | no | 100.0 | 85.75 |
| (fitting random labels) | | no | no | 100.0 | 9.78 |

60000 32x32 colour images in 10 classes

Zhang et al., 2016

# Generalization puzzle in 1d experiments

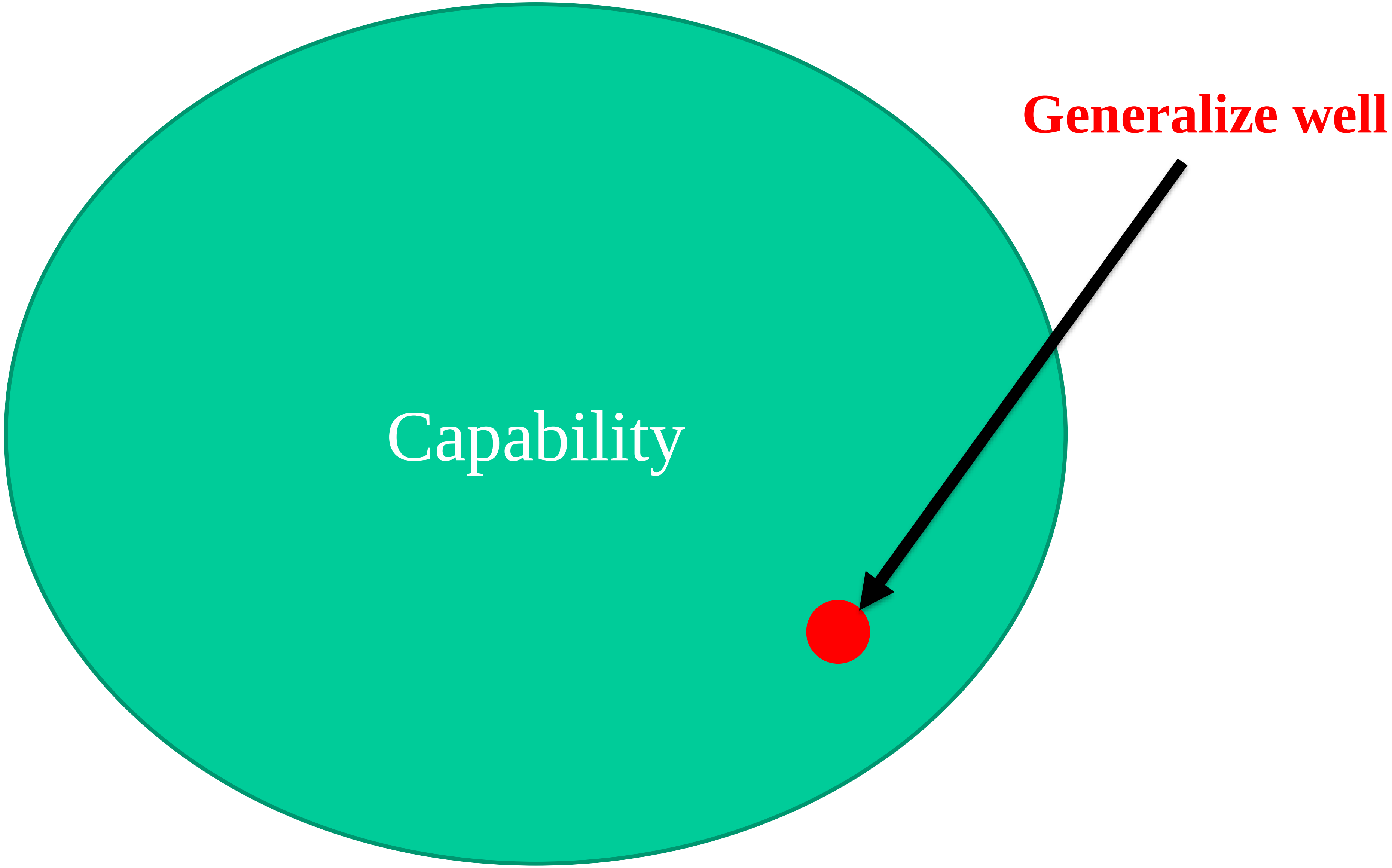
Flat output, no large oscillation




# of parameters: ~ 1600*Layer number>>5

Lei Wu et al., 2017

# Why overparameterized NNs often generalize well?



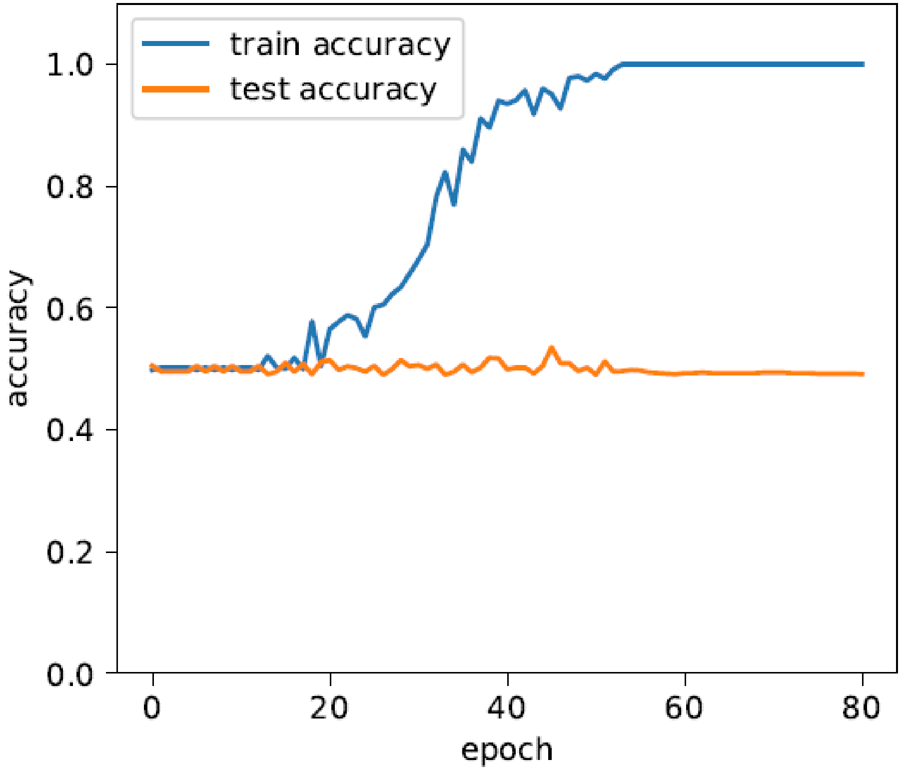
# What DNN cannot do

Parity function:

$$f(\vec{x}) = \prod_{j=1}^{n} x_j$$

$$\vec{x} \in \{-1,1\}^n$$

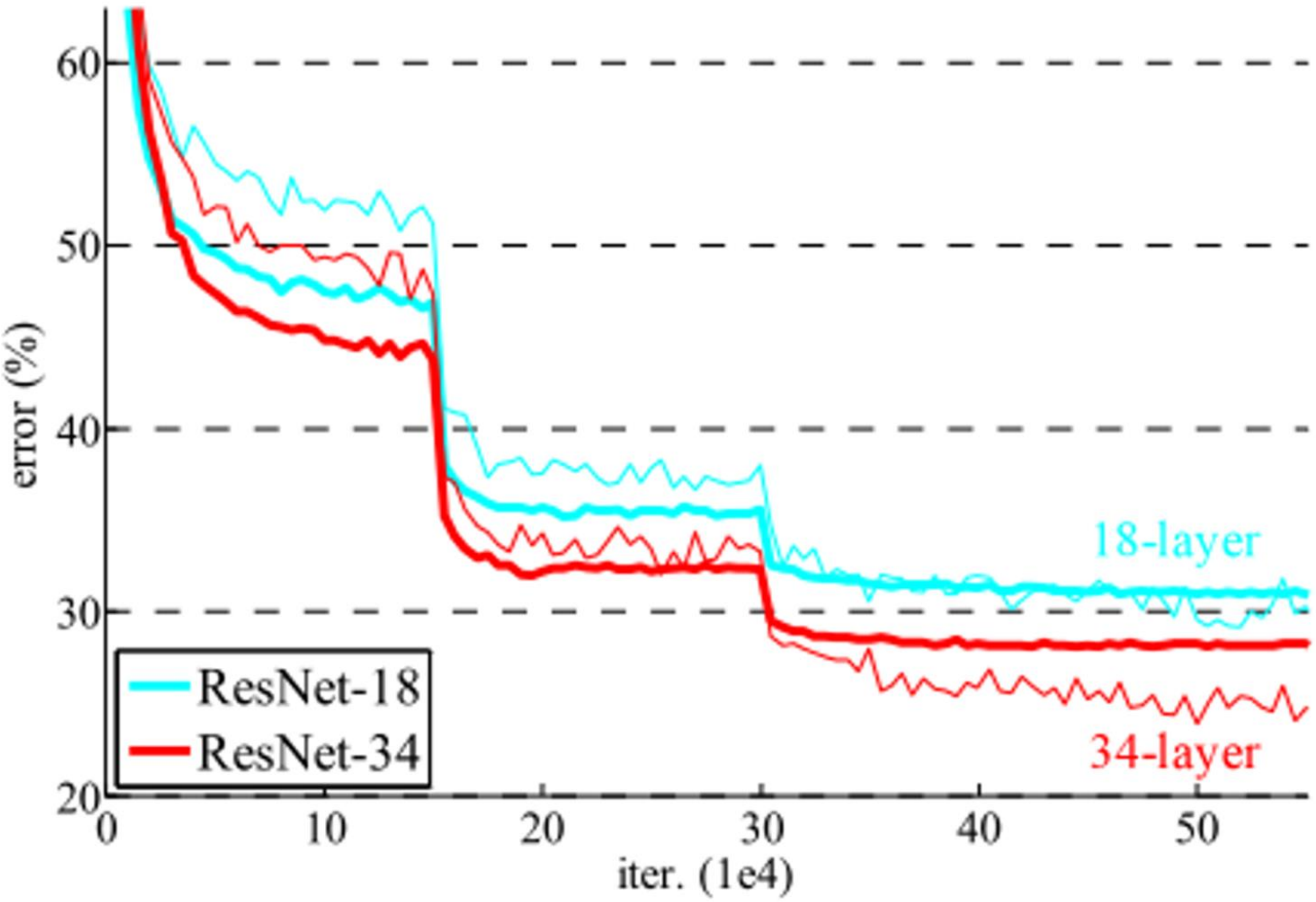
Even '-1' → 1
Odd '-1' → -1




No generalization ability

Abbe and Sandon, 2018

# Depth effect: generalization and speed




Deep Residual Learning for Image Recognition, He et al., 2016

# Some problems

- Overparameterized but often generalize well
- Bad generalization on some problems

# Approximation by Superpositions of a Sigmoidal Function*

G. Cybenko†

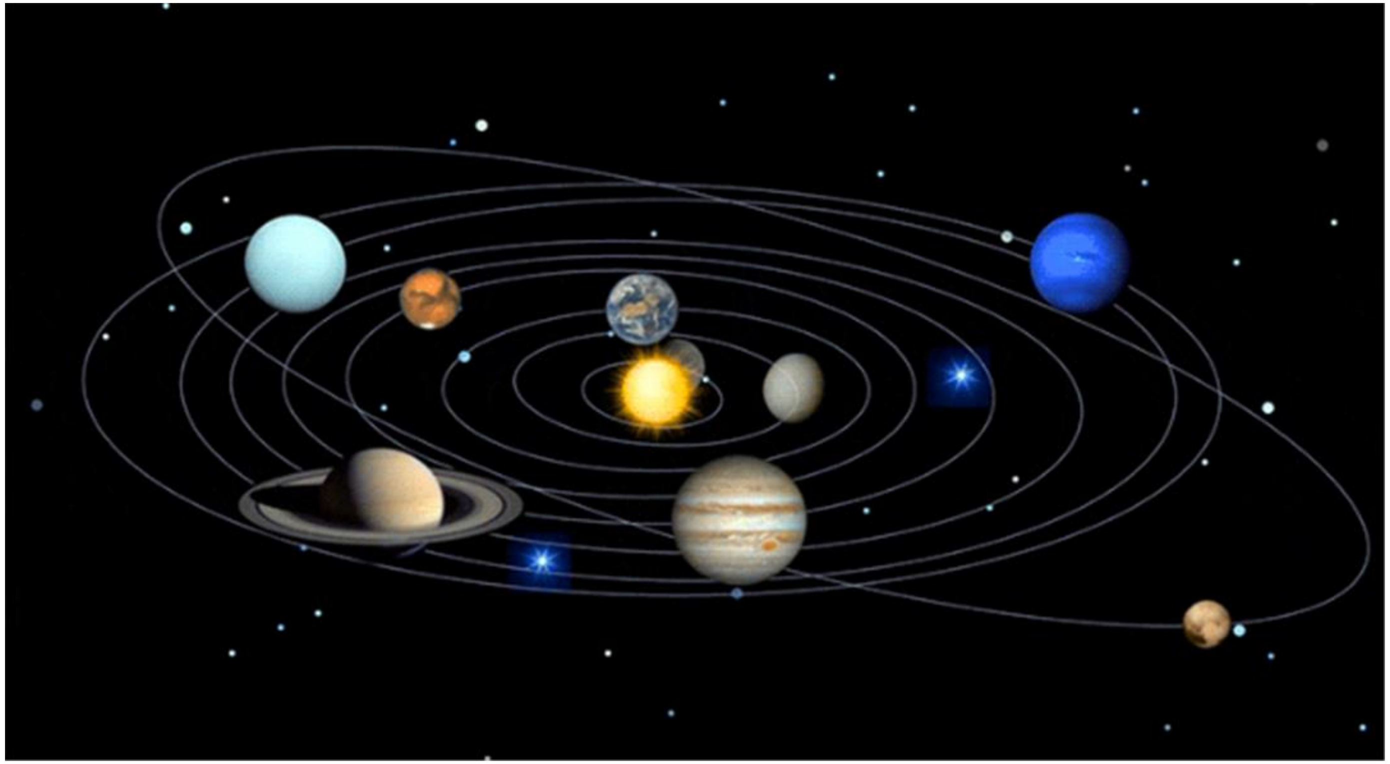
**Abstract.** In this paper we demonstrate that finite linear combinations of compositions of a fixed, univariate function and a set of affine functionals can uniformly approximate any continuous function of $n$ real variables with support in the unit hypercube; only mild conditions are imposed on the univariate function. Our results settle an open question about representability in the class of single hidden layer neural networks. In particular, we show that arbitrary decision regions can be arbitrarily well approximated by continuous feedforward neural networks with only a single internal, hidden layer and any continuous sigmoidal nonlinearity. The paper discusses approximation properties of other possible types of nonlinearities that might be implemented by artificial neural networks.

Single hidden layer can fit any function

# Fitting is not enough!

How to study?

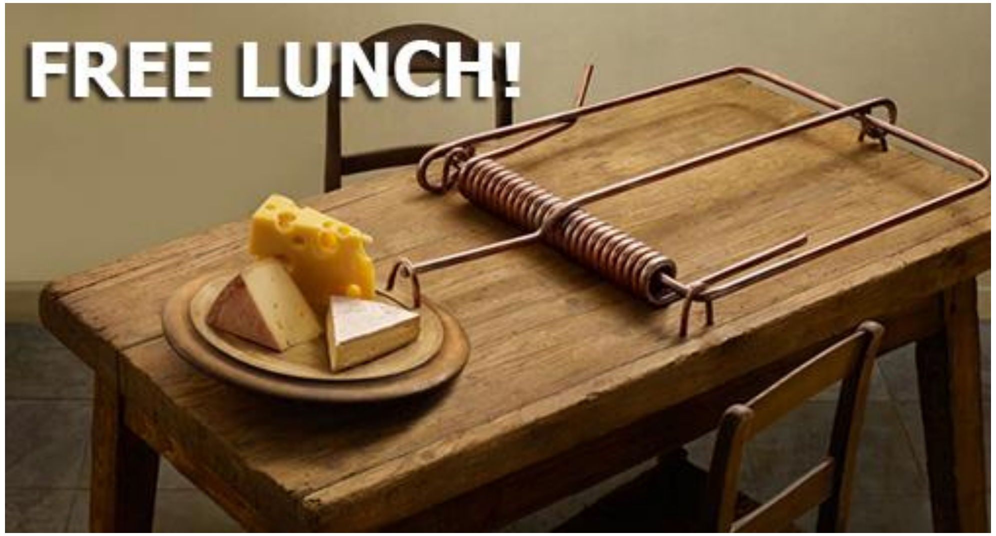
# 扔石头的实验



# 研究过程来理解结论
# Training behavior

# E.g., Generalization error analysis

No free lunch theorem: I can find a dataset that your method generalizes badly.




Data features

Algorithm features

Consistent???

Data features

# Increasing complexity

$x$ is critical sample if there exists $\hat{x}$, close but not same class.

$$\arg\max_i f_i(\mathbf{x}) \neq \arg\max_j f_j(\hat{\mathbf{x}})$$

$$\text{s.t. } \|\mathbf{x} - \hat{\mathbf{x}}\|_\infty \leq r$$

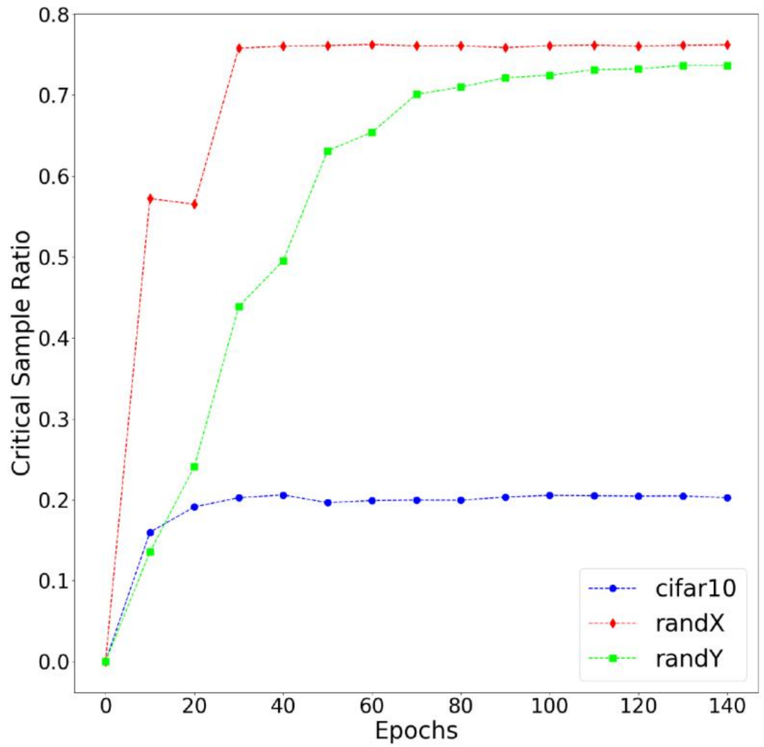



*Figure 9.* Critical sample ratio throughout training on CIFAR-10, random input (randX), and random label (randY) datasets.

Arpit et al., 2017, ICML

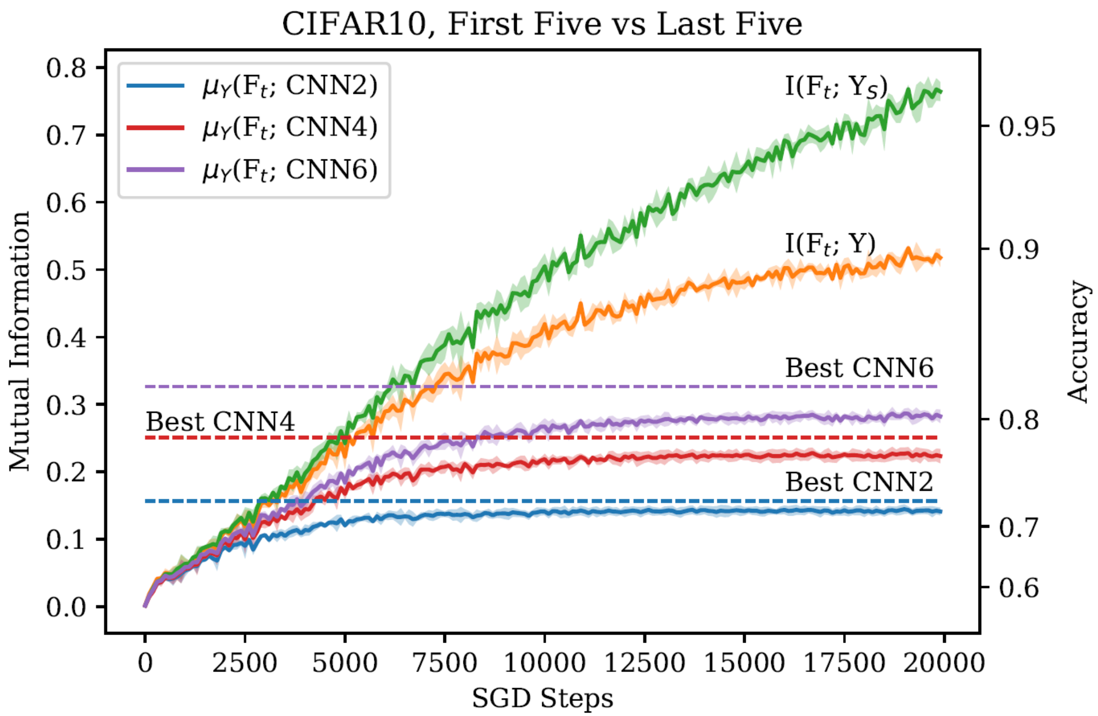
# Increasing complexity

## Performance correlation

**Definition 1.** *For random variables $F, L, Y$ we define the* performance correlation *of $F$ and $L$ as*

$$\mu_Y(F;L) := I(F;Y) - I(F;Y|L) = I(L;Y) - I(L;Y|F) = I(F;L) - I(F;L|Y)\,.$$

The performance correlation is always upper bounded by the minimum of $I(L;Y)$, $I(F;Y)$, and $I(F;L)$.[4] If $\mu_Y(F;L) = I(F;Y)$ then $I(F;Y|L) = 0$ which means that $f$ does not help in predicting $Y$, if we already know $\ell$. Hence, when $\ell$ is a "simpler" model than $f$, we consider $\mu_Y(F;L)$ as denoting the part of $F$'s performance that can be attributed to $\ell$.[5]



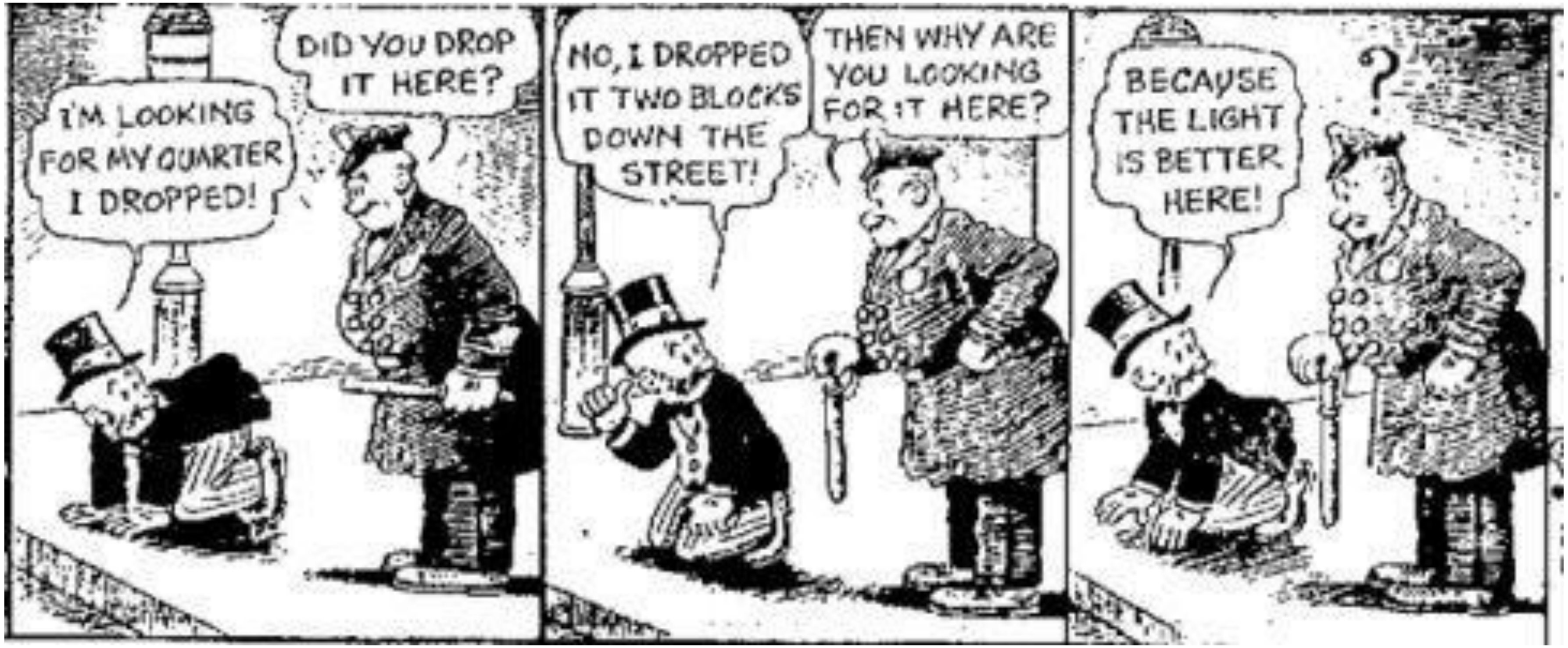

Nakkiran et al., 2019

# Why such studies are difficult for understanding DNN?

- critical sample is difficult to be analyzed
- Performance correlation use black box to characterize black box

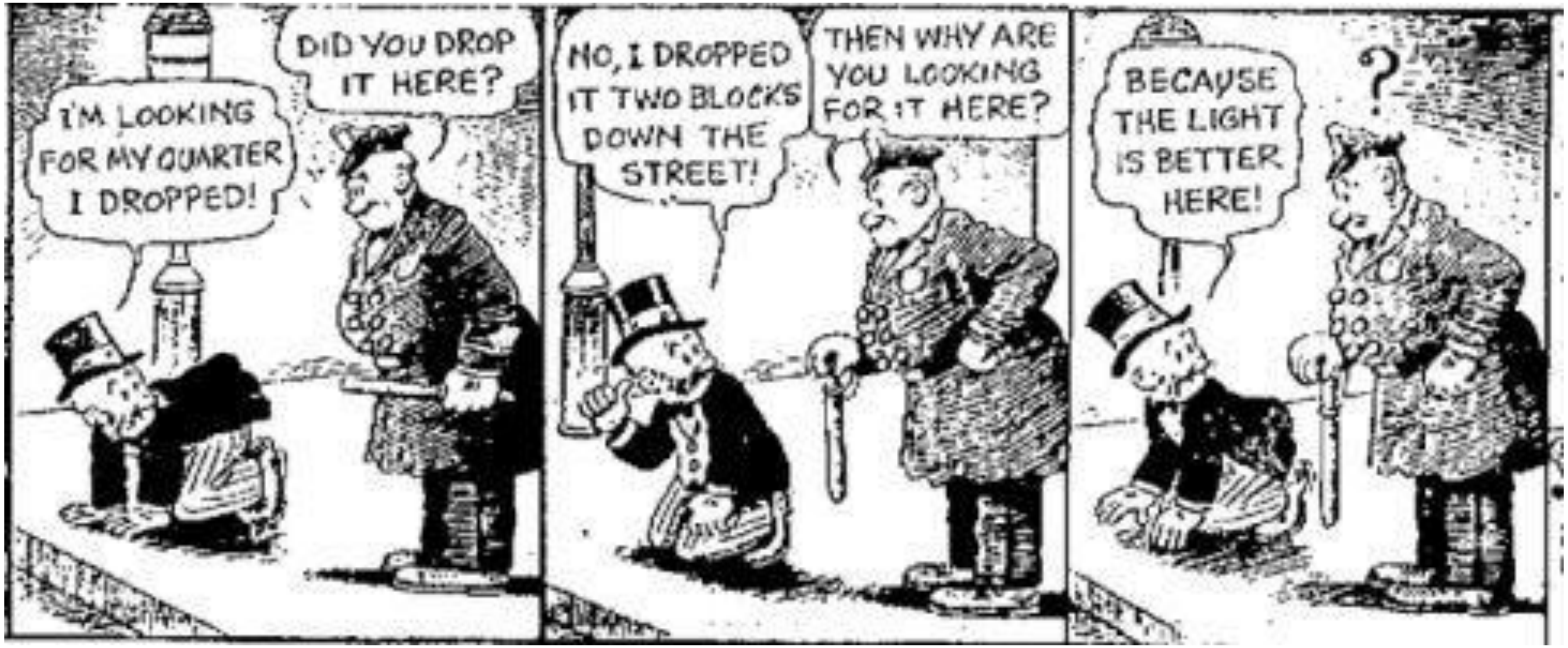
# Philosophy?

A: I am looking for my quarter I dropped.
B: Did you drop it here?
A: No, I dropped it two blocks down the street.
B: Then why are you looking for it here?
A: Because the light is better here.



# Philosophy: from simple to complex

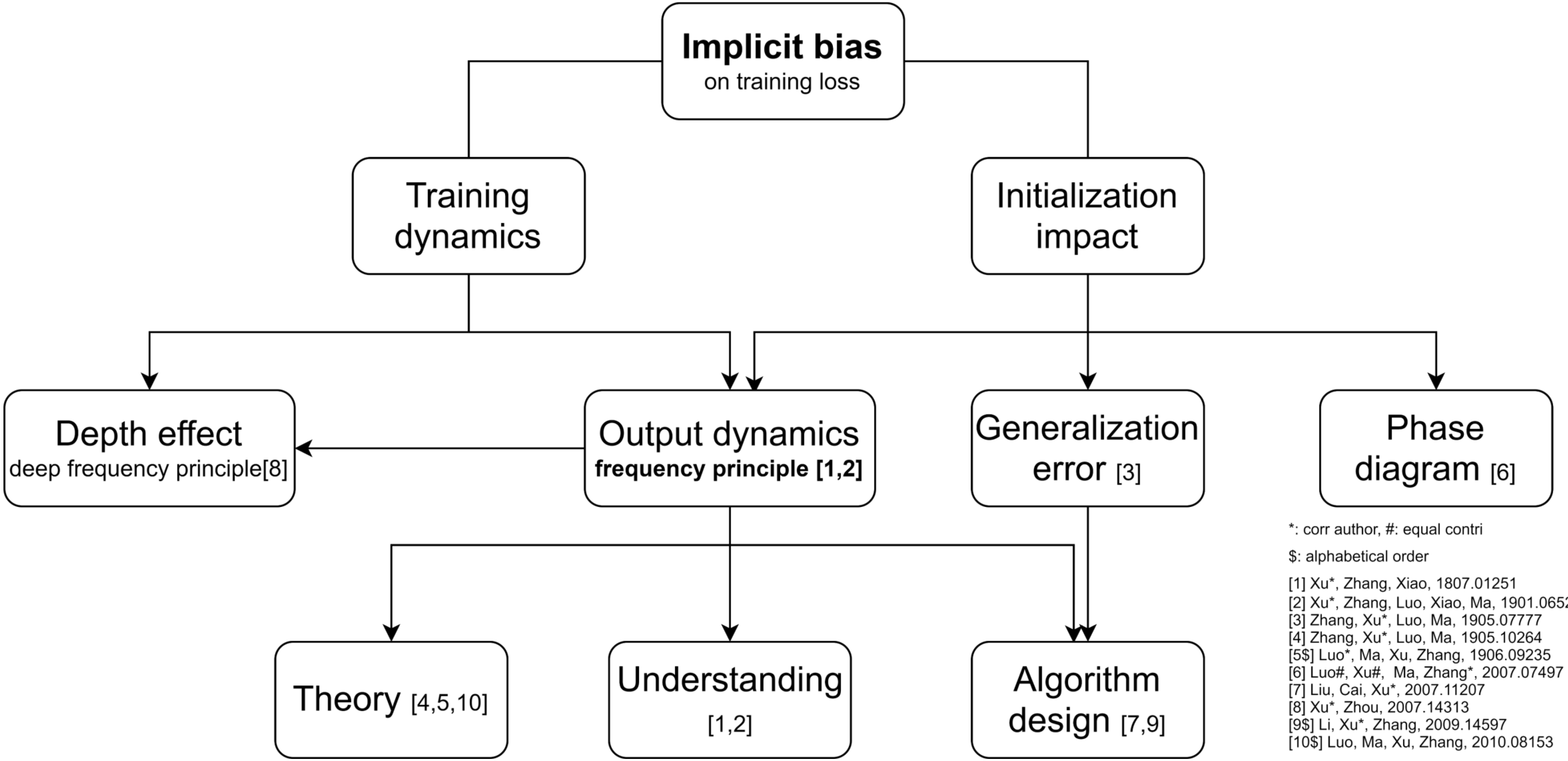
A: I am looking for my quarter I dropped.
B: Did you drop it here?
A: No, I dropped it two blocks down the street.
B: Then why are you looking for it here?
A: **Because I need to get familiar with the road structure first.**




"In the tradition of good old applied mathematics, we will not only give attention to rigorous mathematical results, but also the insight we have gained from careful numerical experiments as well as the analysis of simplified models"

E et al., Towards a Mathematical Understanding of Neural Network-Based Machine Learning: What We Know and What We Don't. CSIAM Trans. Appl. Math, 2020

# A research picture on studying deep neural networks




*: corr author, #: equal contri

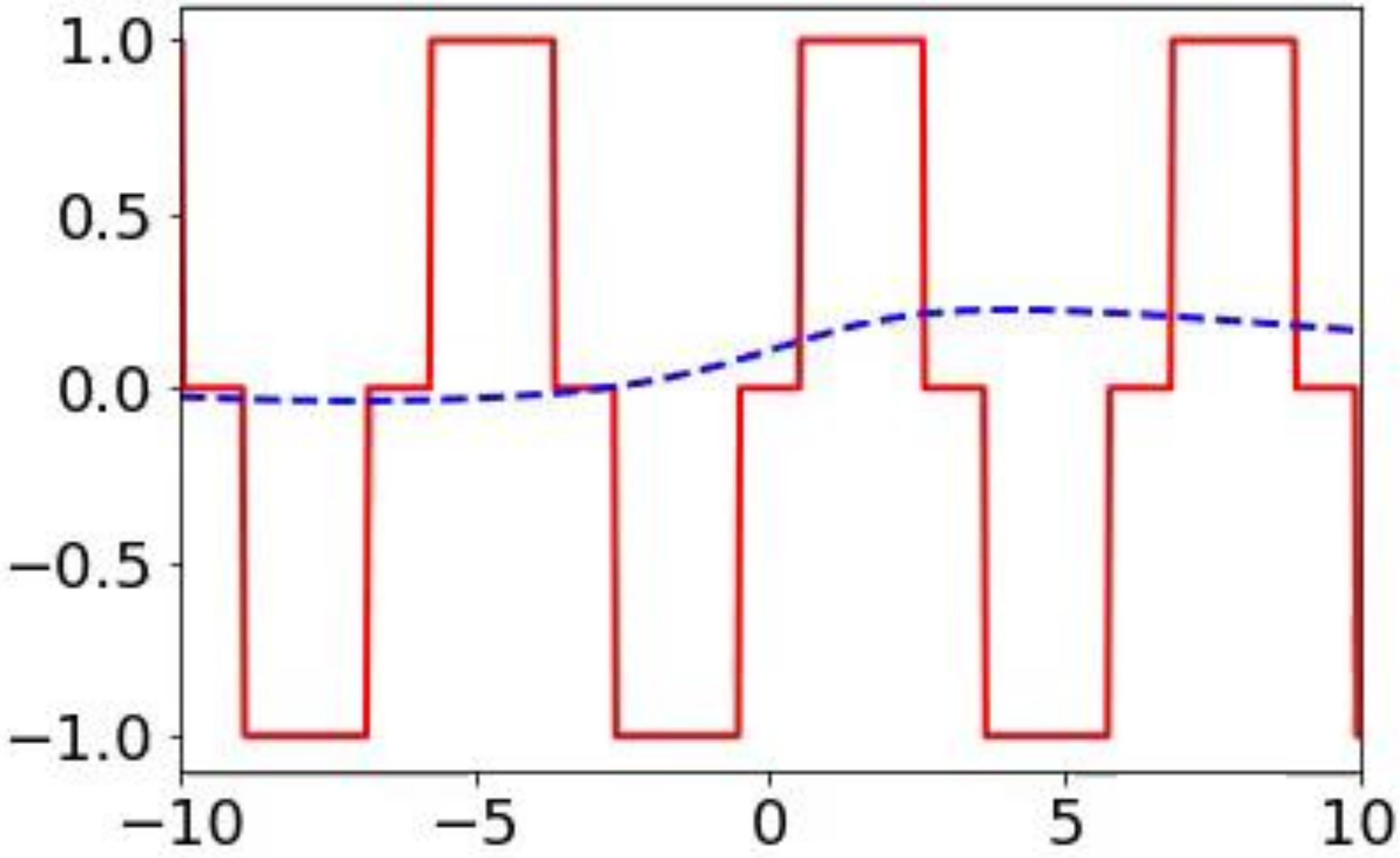
$: alphabetical order


[1] Xu*, Zhang, Xiao, 1807.01251
[2] Xu*, Zhang, Luo, Xiao, Ma, 1901.06523
[3] Zhang, Xu*, Luo, Ma, 1905.07777
[4] Zhang, Xu*, Luo, Ma, 1905.10264
[5$] Luo*, Ma, Xu, Zhang, 1906.09235
[6] Luo#, Xu#, Ma, Zhang*, 2007.07497
[7] Liu, Cai, Xu*, 2007.11207
[8] Xu*, Zhou, 2007.14313
[9$] Li, Xu*, Zhang, 2009.14597
[10$] Luo, Ma, Xu, Zhang, 2010.08153

# Training process of 1d example in spatial domain

From landscape to detail

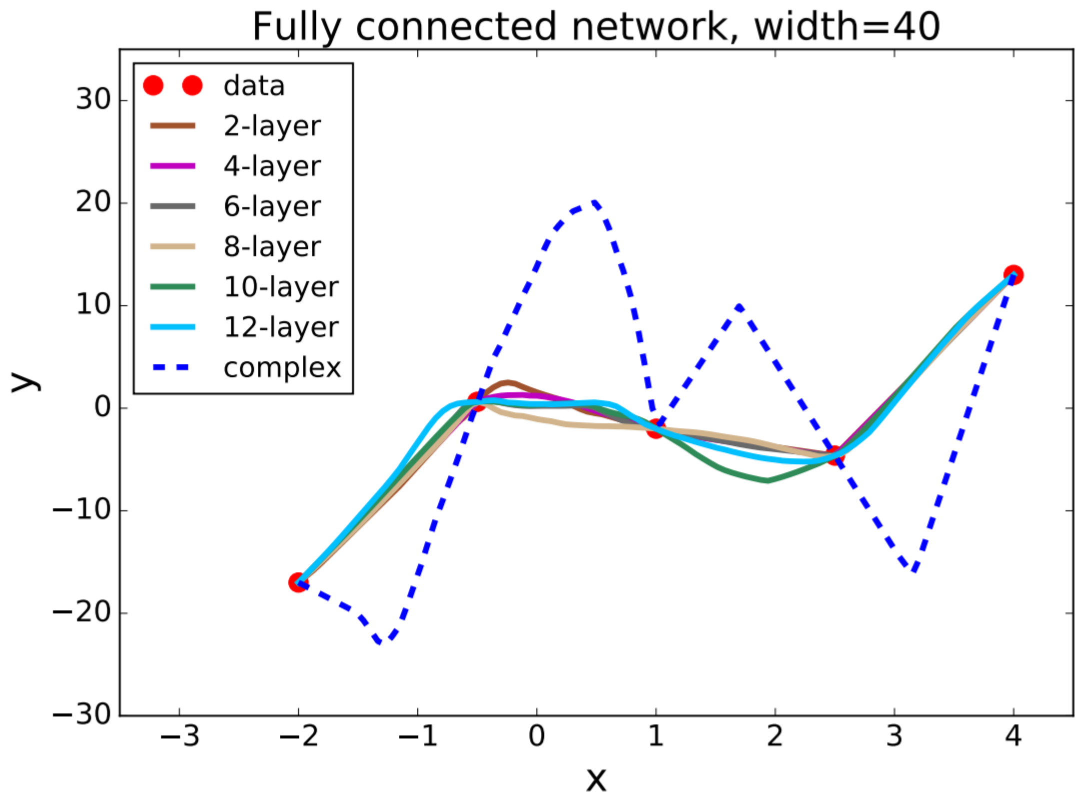
Red:
target function
Blue:
DNN fitting



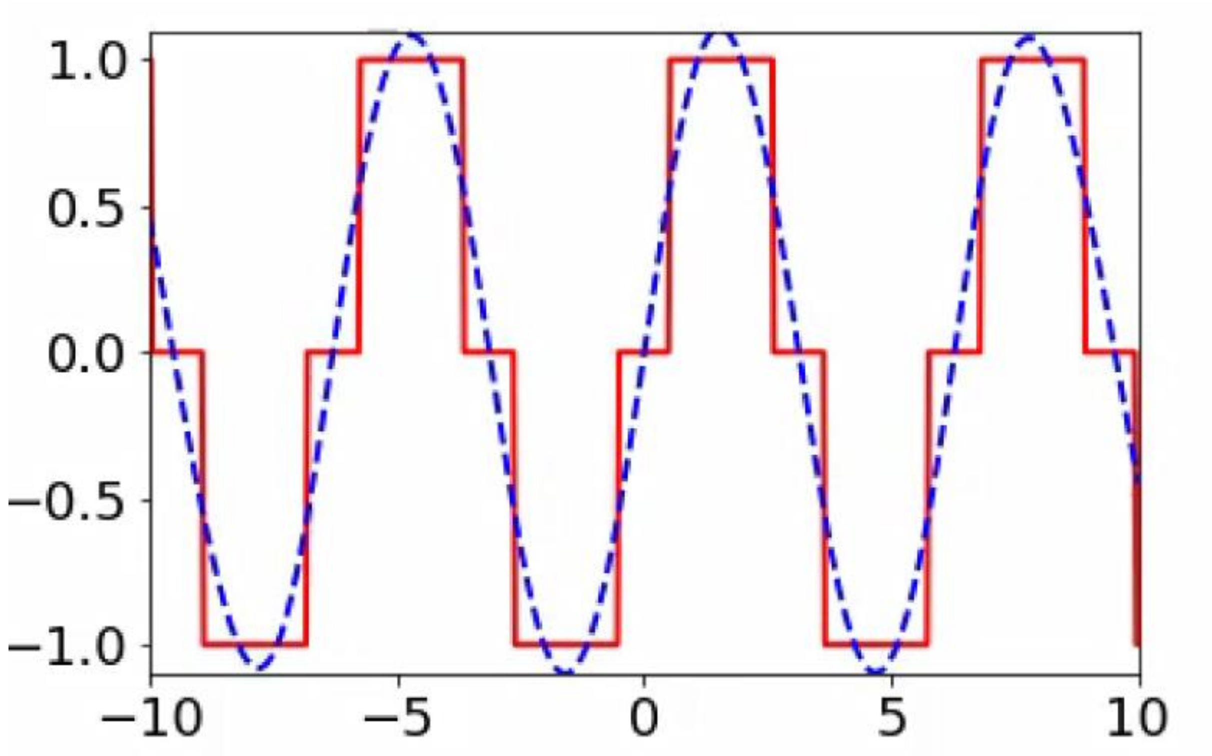

1807.01251

# Features in spatial domain

## Flatness and oscillation




## Landscape and detail

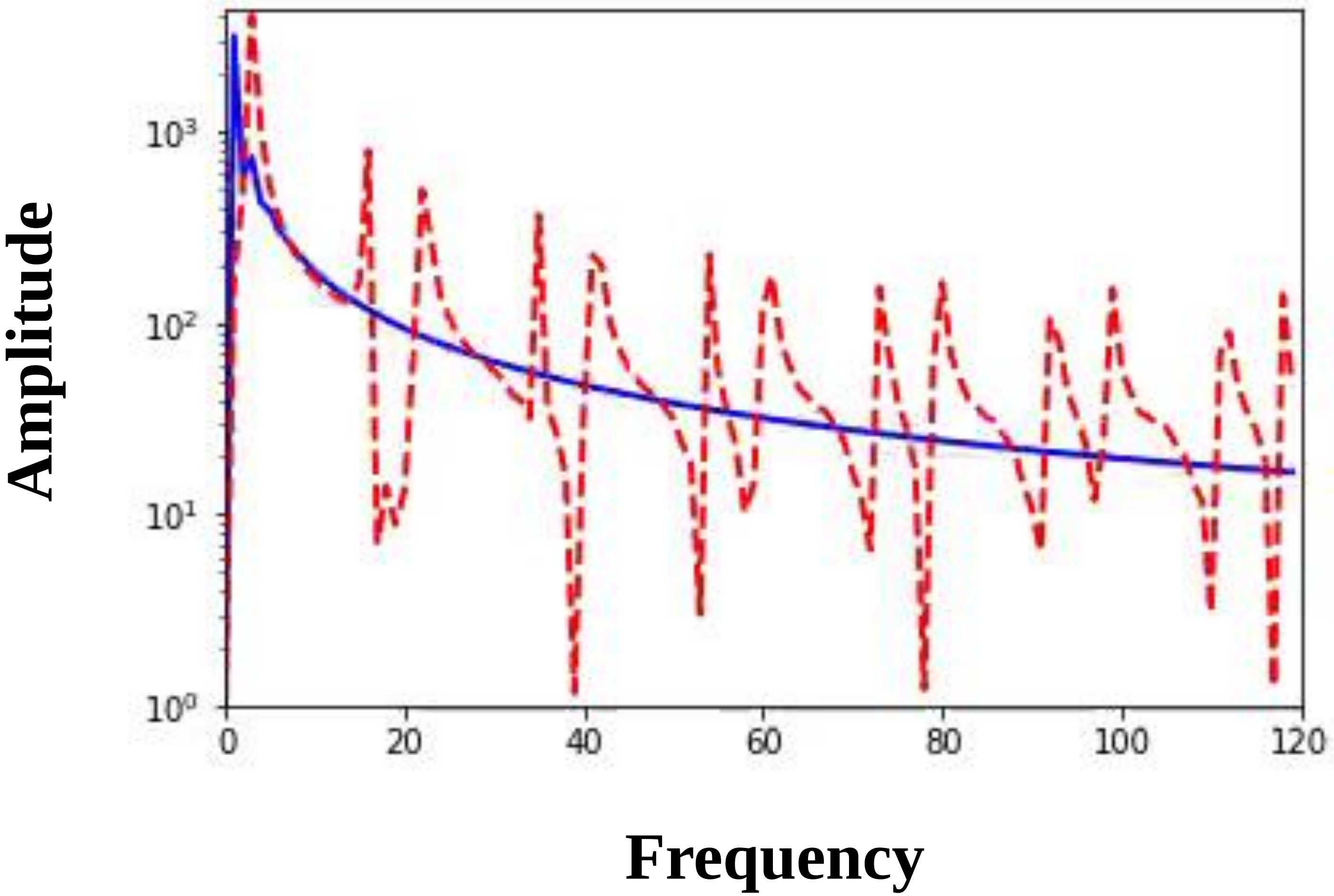



1807.01251

# Training process of 1d example in Fourier domain

Frequency principle: From low frequency to high frequency

Red:
target function
Blue:
DNN fitting




1807.01251

# A Simple Theory Understanding: one hidden layer, infinite width

# Linear approximation for wide NN

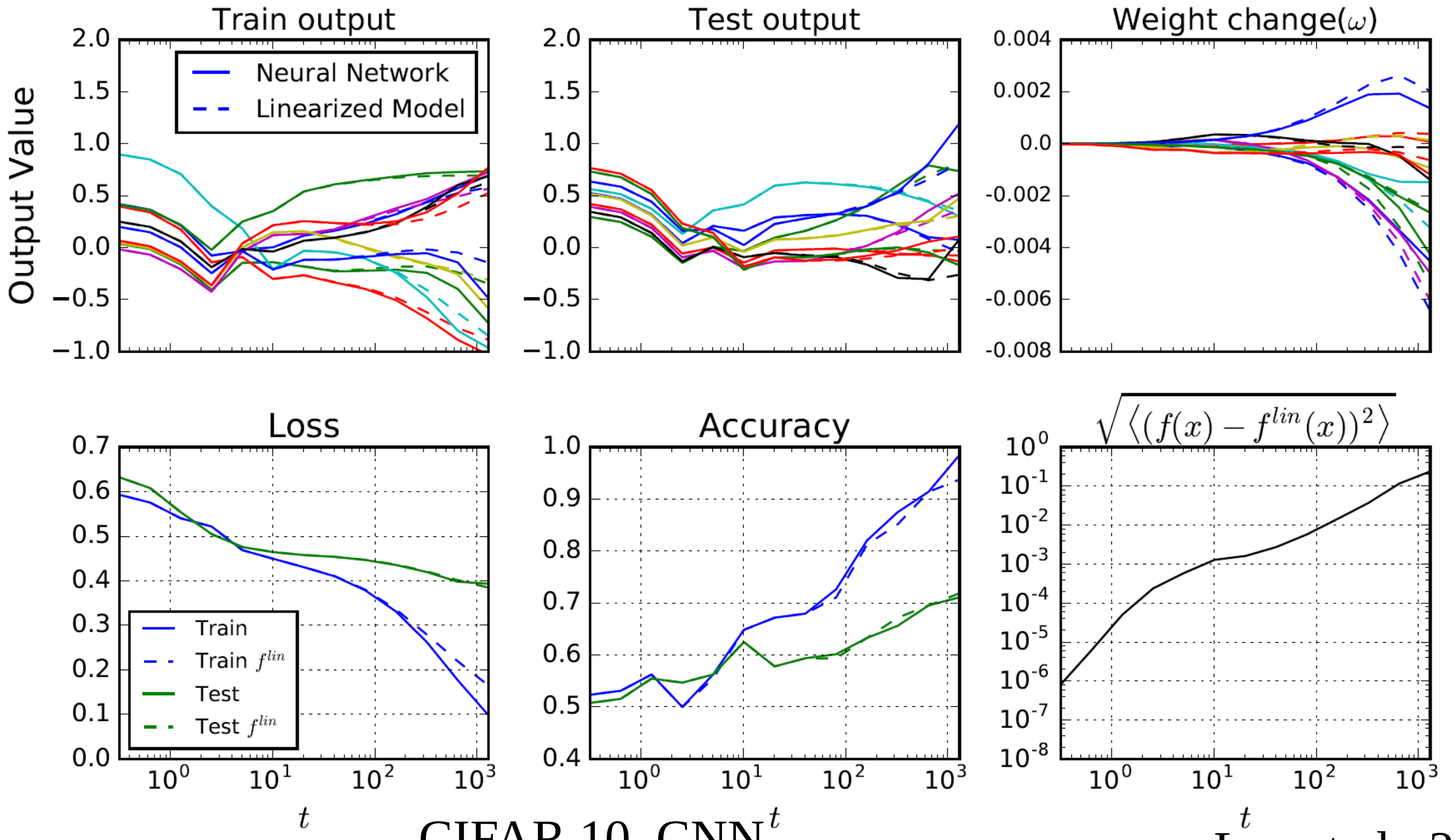
$$h(x,\theta(t)) = h(x,\theta_0) + \nabla_\theta h(x,\theta_0)(\theta(t)-\theta_0) \text{ for any } t>0$$

Jacot et al., 2018

$$\boxed{\frac{d\theta(t)}{dt} = -\nabla_\theta h(X,\theta_0)^T\big(h(X,\theta(t))-Y\big)}$$

$$L(\theta) = \tfrac{1}{2}\|h(X,\theta)-Y\|_2^2$$

$X: [x_i]_{i=1}^n \quad Y: [y_i]_{i=1}^n$




CIFAR 10, CNN

Lee et al., 2019

# Linear F-Principle dynamics

$$\boxed{\frac{d\theta(t)}{dt} = -\nabla_\theta h(X,\theta_0)^T\left(h\left(X,\theta(t)\right) - Y\right)}$$

$$h(\cdot,\theta) = \frac{1}{\sqrt{m}}\sum_{i=1}^{m} a_i\sigma(w_i x + b_i)$$

$m$ sufficiently large, $r = ||w||$

$$\boxed{\partial_t\, \hat{h}(\xi,t) = CE_{a,r}\left[\frac{r^3}{\xi^{d+3}} + \frac{4\pi^2 a^2 r^2}{\xi^{d+1}}\right]\left(\widehat{f_p}(\xi,t) - \widehat{h_p}(\xi,t)\right)}$$

$f$: target function; $(\cdot)_p = (\cdot)p$, where $p(x) = \frac{1}{n}\sum_{i=1}^{n}\delta(x - x_i)$;

$\hat{\cdot}$: Fourier transform; $\xi$: frequency

## For simplicity, d=1

$$\partial_t\, \hat{u}(\xi,t) = -\left[\frac{\langle r^3\rangle}{\xi^4} + \frac{4\pi^2\langle r^2 a^2\rangle}{\xi^2}\right]\widehat{u_p}(\xi,t)$$

$$u(x,t) = h(x,t) - f(x)$$

LMXZ (1905.10264, 2010.08153)

# Preference induced by LFP dynamics

$$\partial_t \hat{h}(\xi,t) = -\left[\frac{\langle r^3\rangle}{\xi^4} + \frac{4\pi^2\langle r^2 a^2\rangle}{\xi^2}\right]\left(\widehat{h_p}(\xi,t) - \widehat{f_p}(\xi,t)\right)$$

$$\updownarrow$$

low frequency preference

$$\boxed{\min_{h\in F_\gamma} \int \left[\frac{\langle r^3\rangle}{\xi^4} + \frac{4\pi^2\langle r^2 a^2\rangle}{\xi^2}\right]^{-1} \left|\hat{h}(\xi)\right|^2 \mathrm{d}\xi}$$

s.t. $h(x_i) = y_i$ for $i = 1,\cdots,n$

Case 1: $\xi^{-4}$ dominant

- $\min \int \xi^4 \left|\hat{h}(\xi)\right|^2 \mathrm{d}\xi \sim \min \int |h''(x)|^2\, \mathrm{d}\xi \to$ **cubic spline**

Case 2: $\xi^{-2}$ dominant

- $\min \int \xi^2 \left|\hat{h}(\xi)\right|^2 \mathrm{d}\xi \sim \min \int |h'(x)|^2\, \mathrm{d}\xi \to$ **linear spline**

LMXZ (1905.10264, 2010.08153)

# Limit of the frequency bias

$$\min_{h\in\mathcal{H}} Q_\alpha[h] = \int_{\mathbb{R}^d} \langle\boldsymbol{\xi}\rangle^\alpha \left|\mathcal{F}[h](\boldsymbol{\xi})\right|^2 \mathrm{d}\boldsymbol{\xi},$$
$$\text{s.t.}\quad h(\boldsymbol{x}_i) = y_i, \quad i = 1, \cdots, n,$$

**Theorem 1 (non-existence)** *Suppose that $\boldsymbol{Y} \neq \mathbf{0}$. For $\alpha < d$, there is no function $\phi^* \in \mathcal{A}_{\boldsymbol{X},\boldsymbol{Y}}$ satisfying*

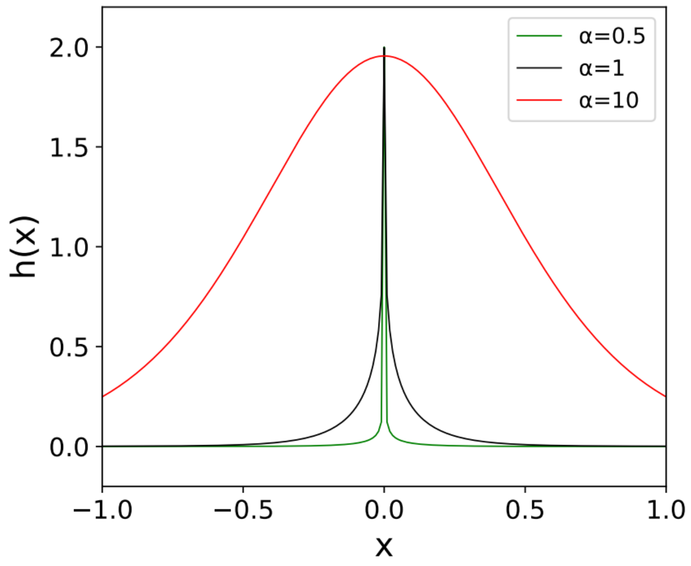
$$\phi^* \in \arg\min_{\phi\in\mathcal{A}_{\boldsymbol{X},\boldsymbol{Y}}} \|\mathcal{F}^{-1}[\phi]\|^2_{H^{\frac{\alpha}{2}}}.$$

*In other words, there is no solution to the Problem 1.*

**Theorem 2 (existence)** *For $\alpha > d$, there exists $\phi^* \in \mathcal{A}_{\boldsymbol{X},\boldsymbol{Y}}$ satisfying*

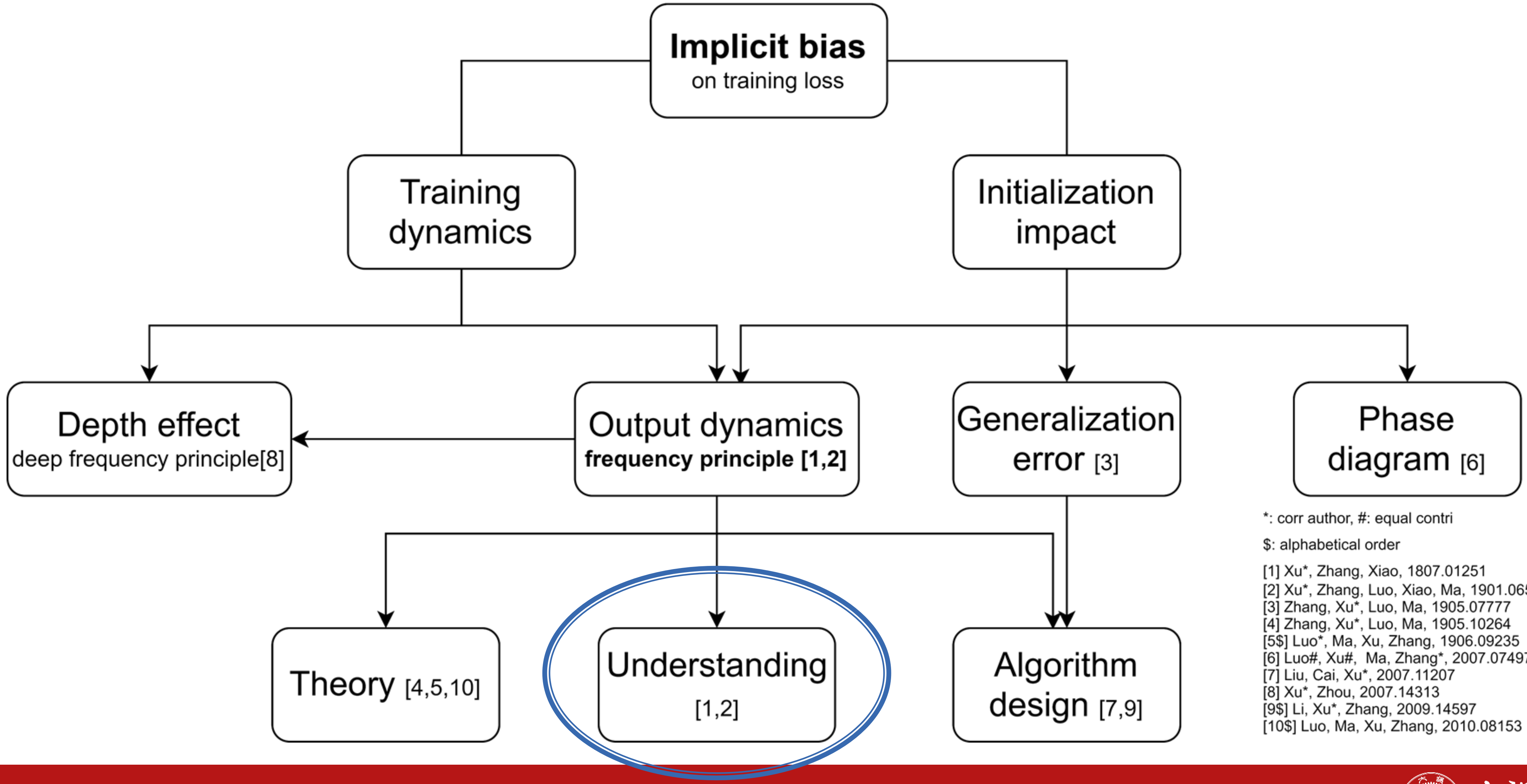
$$\phi^* \in \arg\min_{\phi\in\mathcal{A}_{\boldsymbol{X},\boldsymbol{Y}}} \|\mathcal{F}^{-1}[\phi]\|^2_{H^{\frac{\alpha}{2}}}.$$

*In other words, there exists a solution to the Problem 1.*




WLMXZ, 2012.03238

# A research picture on studying deep neural networks

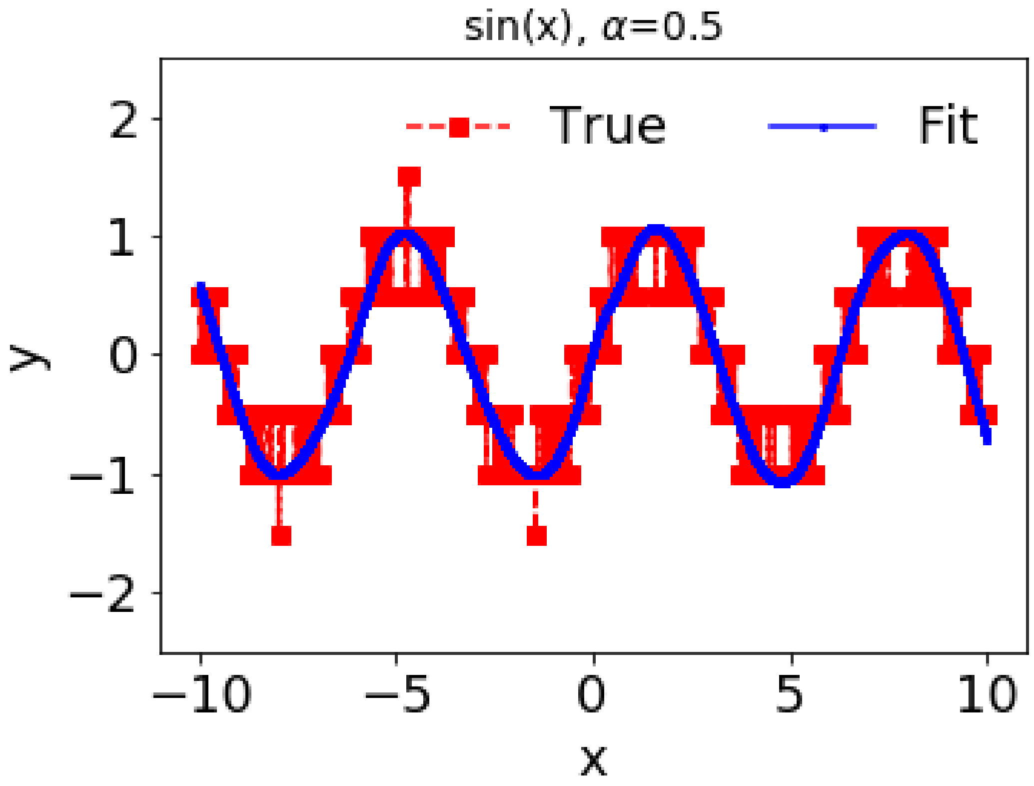


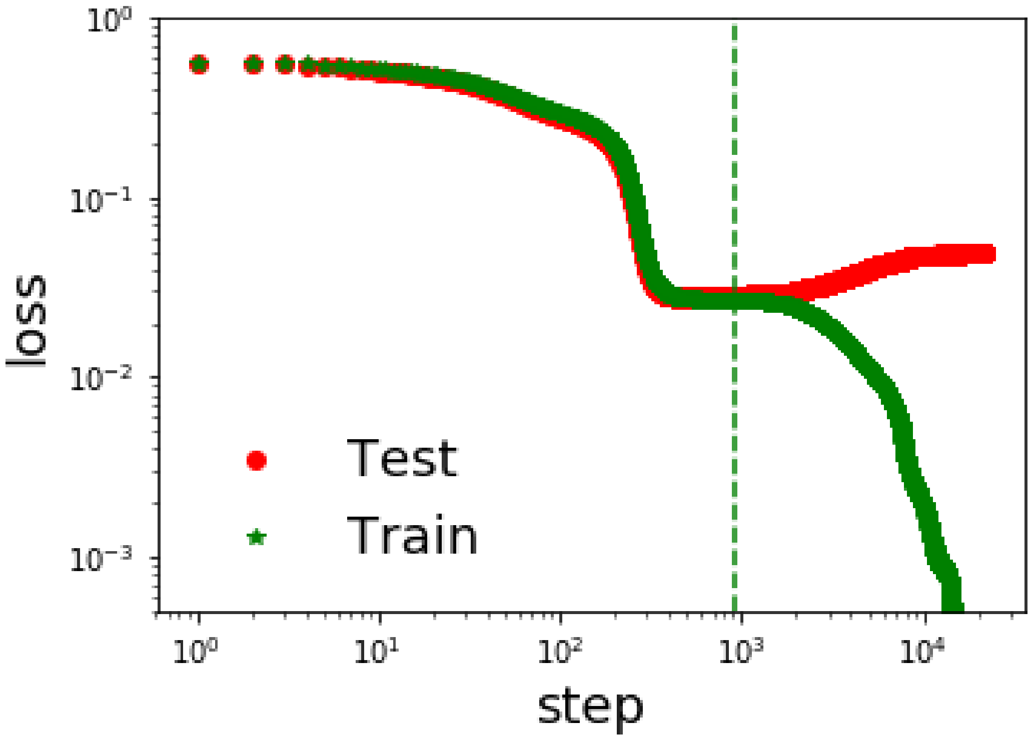

*: corr author, #: equal contri

$: alphabetical order


[1] Xu*, Zhang, Xiao, 1807.01251
[2] Xu*, Zhang, Luo, Xiao, Ma, 1901.06523
[3] Zhang, Xu*, Luo, Ma, 1905.07777
[4] Zhang, Xu*, Luo, Ma, 1905.10264
[5$] Luo*, Ma, Xu, Zhang, 1906.09235
[6] Luo#, Xu#, Ma, Zhang*, 2007.07497
[7] Liu, Cai, Xu*, 2007.11207
[8] Xu*, Zhou, 2007.14313
[9$] Li, Xu*, Zhang, 2009.14597
[10$] Luo, Ma, Xu, Zhang, 2010.08153

# Effect of early stopping

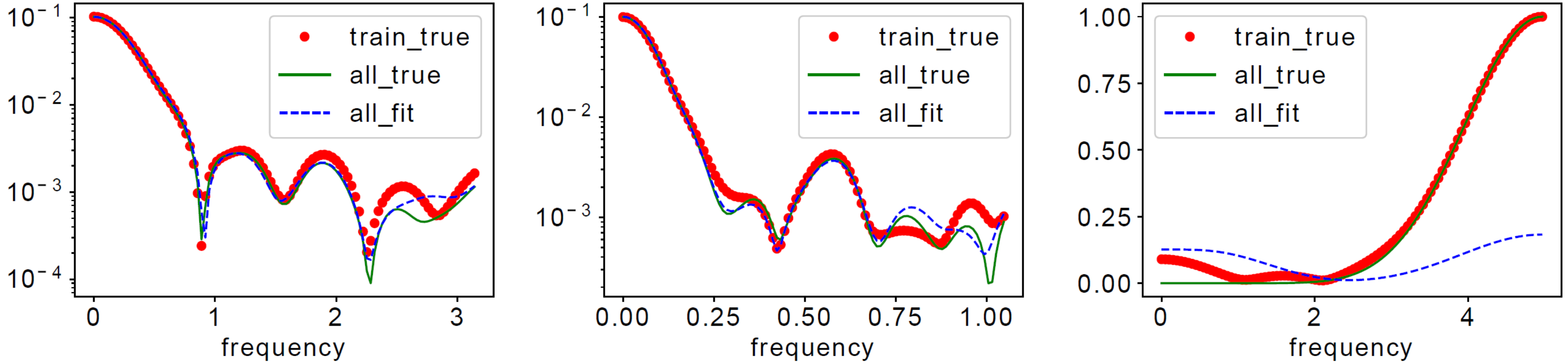


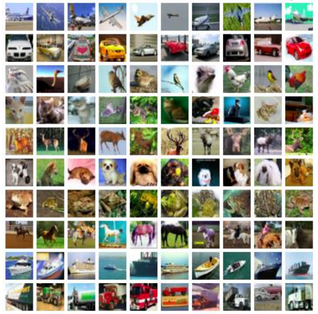

1807.01251

# Generalization difference

F-Principe: DNN prefers low frequencies




(a) MNIST (b) CIFAR10 (c) parity



For $\vec{x} \in \{-1,1\}^n$

$\boldsymbol{f(\vec{x}) = \prod_{j=1}^{n} x_j}$,

Even #‘-1’ → 1;

Odd #‘-1’ → -1.

Test accuracy: 96.3%>>10%

Test accuracy: 72% %>>10%

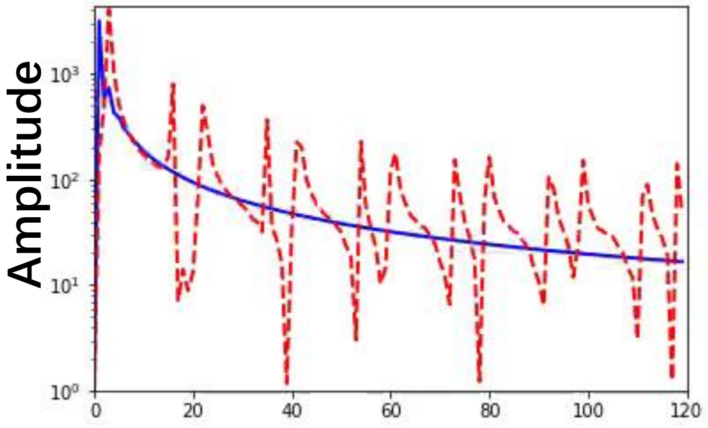
Test accuracy: ~50%, random guess

1901.06523

# Frequency Principle

## Beginning

Red: FFT of target function
Blue: FFT of DNN fitting
Each frame is one training step




**Xu, Zhang, Xiao, ICONIP, 2019**
**Xu, Zhang, Luo, Xiao, Ma, CiCP, 2019**
Rahaman et al., ICML, 2019

## Theory: regularity of activation function

**General theory**:
**Luo, Ma, Xu, Zhang, 2019**
E, Ma, Wu. Science China Mathematics, 2020
**Wide two-layer ReLU network**:
**Zhang, Xu, Luo, Ma, 2019**
Basri et al., NeurIPS, 2019
Cao, Fang, Wu, Zhou, Gu. 2019
Bordelon, Canatar, Pehlevan, ICML, 2020.
**Zhang, Xu, Luo, Ma, 2020**

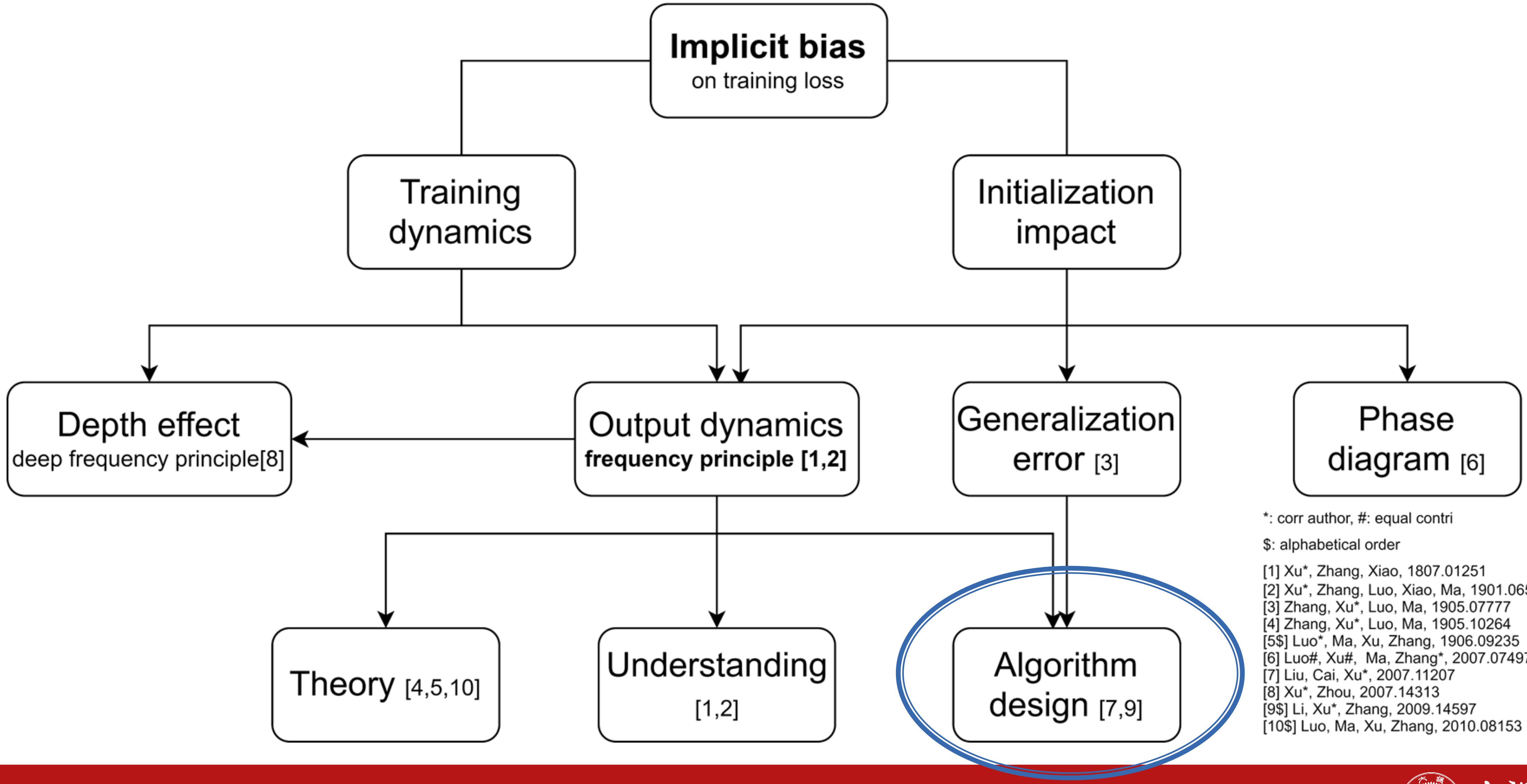
## Algorithms: Fast capture high-frequency

Cai., Li, Liu, PhaseDNN, SIAM J. Scientific Computing, 2019
**Liu, Cai, Xu, MscaleDNN. CiCP, 2020.**
Jagtap, & Karniadakis, Adaptive activation, J. Comput. Phys, 2020
Wang et al., Inverse problems, Scientific reports, 2018
Biland et al., Frequency-aware reconstruction of fluid. 2019.
Dziedzic et al., Band-limited Training for CNN, ICML, 2020

## Understanding

Wang et al., High frequency helps explain the generalization of CNN, CVPR, 2020
You et al., Drawing Early-Bird Tickets, ICLR, 2020
Chakrabarty & Maji, The Spectral Bias of the Deep Image Prior, NeurIPS, 2019
Jin, Lu, Tang, Karniadakis, Quantifying the generalization, Neural Networks, 2019
Stamatescu, McDonnell, Diagnosing CNN, DICTA, 2018
Rabinowitz, Meta-learners' learning dynamics are unlike learners, 2019
Zhang, Wu, Rethink Generalization, Memorization and the Spectral Bias of DNNs, 2020
Ma, Wu, E, The slow deterioration of the generalization error, MSML, 2020
**Xu, Zhou, Deep frequency principle, 2020**

# A research picture on studying deep neural networks




*: corr author, #: equal contri

$: alphabetical order

[1] Xu*, Zhang, Xiao, 1807.01251
[2] Xu*, Zhang, Luo, Xiao, Ma, 1901.06523
[3] Zhang, Xu*, Luo, Ma, 1905.07777
[4] Zhang, Xu*, Luo, Ma, 1905.10264
[5$] Luo*, Ma, Xu, Zhang, 1906.09235
[6] Luo#, Xu#, Ma, Zhang*, 2007.07497
[7] Liu, Cai, Xu*, 2007.11207
[8] Xu*, Zhou, 2007.14313
[9$] Li, Xu*, Zhang, 2009.14597
[10$] Luo, Ma, Xu, Zhang, 2010.08153

# Convergence behavior

Poisson Equation: A Finite Difference Approach

$$-\partial_x^2 u(x) = g(x), \quad u(0) = u(1) = 0.$$

The finite difference scheme

$$-\frac{u(x_{j+1}) - 2u(x_j) + u(x_{j-1})}{h^2} = g(x_j)$$

yields a linear system

$$Au = g,$$

where

$$A = \frac{1}{h^2}\begin{pmatrix} -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \end{pmatrix}.$$

Iterative solver: **lower frequency converges slower.**

# Poisson Equation: DNN Approach

A $1d$ Poisson equation on $(0,1)$ with Dirichlet boundary condition,

$$-\Delta u(x) = g(x), \quad u(0) = u(1) = 0.$$

$u(x)$ can be solved by the following variational problem,

$$\min_{u \in H^1(0,1)} \int_0^1 \left( \frac{1}{2} |\partial_x u(x)|^2 - g(x)u(x) \right) \mathrm{d}x + \beta \left( u(0)^2 + u(1)^2 \right).$$

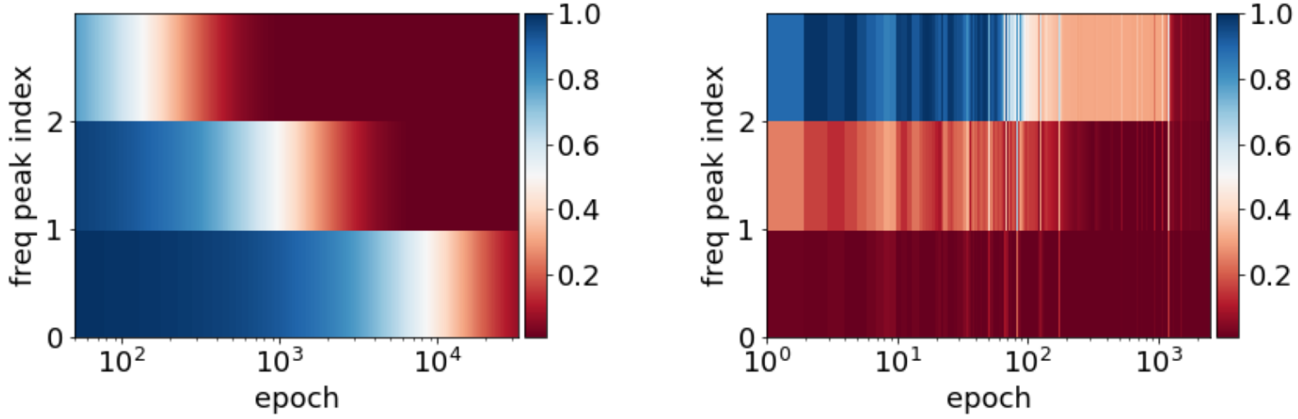
Here we can parametrize $u(x)$ using **deep neural network (DNN)**:

- **Input**: $x$.
- **Output**: $u(x) = u_{\mathrm{Net}}(x)$.
- **Train**: stochastic gradient decent.

E, Yu, *The Deep Ritz Method: A Deep Learning-Based Numerical Algorithm for Solving Variational Problems*, Communications in Mathematics and Statistics, 2018

**Suffer from high-frequency curse**

$$g(x) = \sin(x) + 4\sin(4x) - 8\sin(8x) + 16\sin(24x).$$



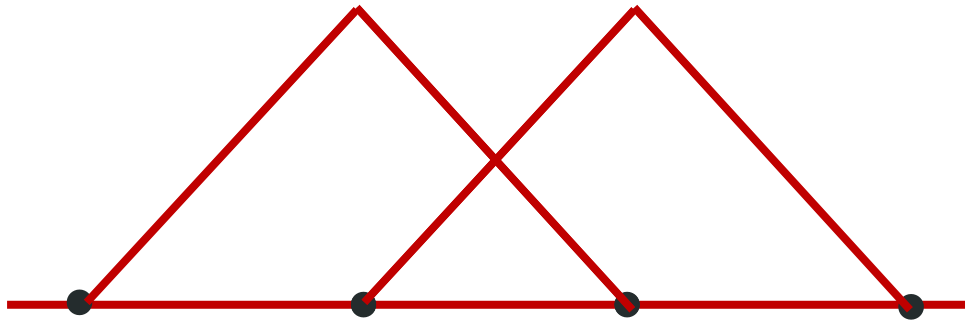

(a) Jacobi (b) DNN

**F-Principle**: *A DNN tends to learn a target function from low to high frequencies during the training.*

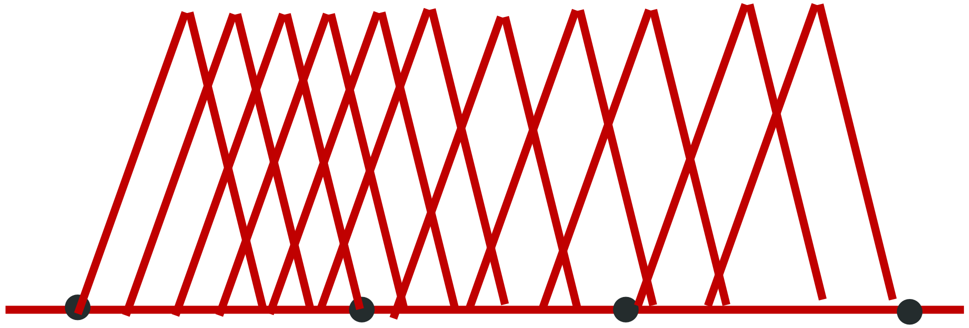
Xu, Zhang, Luo, Xiao, Ma, *Frequency Principle: Fourier Analysis Sheds Light on Deep Neural Networks*, 1901.06523, 2019

# Learning Results

- **Consider a control experiment:**
  - $m$: **Neuron number in DNN= Basis number in FEM**
  - $m > n$ (**$n$: grid points)**
  - $-\Delta u(x) = f(x)$, $f(x)$ **is only given in finite points (NOTE: not common)**



FEM case

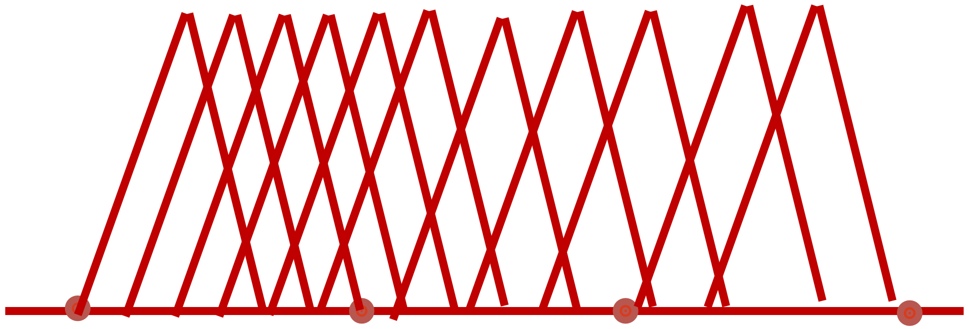


DNN case

# Learning Results: FEM as $m \to \infty$

**Theorem 1.** When $m \to \infty$, the numerical method (2.8) is solving the problem

$$\begin{cases} -\Delta u(\boldsymbol{x}) = \dfrac{1}{n}\sum_{i=1}^{n}\delta(\boldsymbol{x}-\boldsymbol{x}_i)f(\boldsymbol{x}_i), & \boldsymbol{x}\in\Omega, \\ u(\boldsymbol{x}) = 0, & \boldsymbol{x}\in\partial\Omega, \end{cases} \tag{3.1}$$



**Remark:** For the 1D case, the analytic solution to problem (3.1) defined in $[a,b]$ can be given as a piecewise linear function, namely

$$u(x) = \frac{1}{n}\sum_{i=1}^{n} f(x_i)(b-x_i)\frac{x-a}{b-a} - \frac{1}{n}\sum_{i=1}^{n} f(x_i)(x-x_i)H(x-x_i), \tag{3.2}$$

where $H(x)$ is the Heaviside step function

$$H(x) = \begin{cases} 0, & x<0, \\ 1, & x\geq 0. \end{cases}$$

For the 2D case, [12] gives the exact solution in $[0,a]\times[0,b]$ by Green's function

$$u(x,y) = \frac{4}{nab}\sum_{i=1}^{n} f_i \sum_{k=1}^{\infty}\sum_{l=1}^{\infty}\frac{\sin(p_k x)\sin(q_l y)\sin(p_k x_i)\sin(q_l y_i)}{p_k^2+q_l^2}. \tag{3.3}$$

where $f_i = f(x_i,y_i), p_k = \pi k/a, q_l = \pi l/b$. We can prove that this series diverges at the sampling point $(x_i,y_i)(i=1,2,\cdots,n)$ and converges at other points. Therefore, the 2d exact solution $u(x,y)$ is highly singular.

Wang, Xu, Zhang, Zhang, 2020, 2002.07989

# Learning Results: FEM as $m \to \infty$ for d=2

$$\begin{cases} -\Delta u(\boldsymbol{x}) = f(\boldsymbol{x}), & \boldsymbol{x} \in (0,1)^2, \\ u(\boldsymbol{x}) = 0, & \boldsymbol{x} \in \partial(0,1)^2, \end{cases}$$

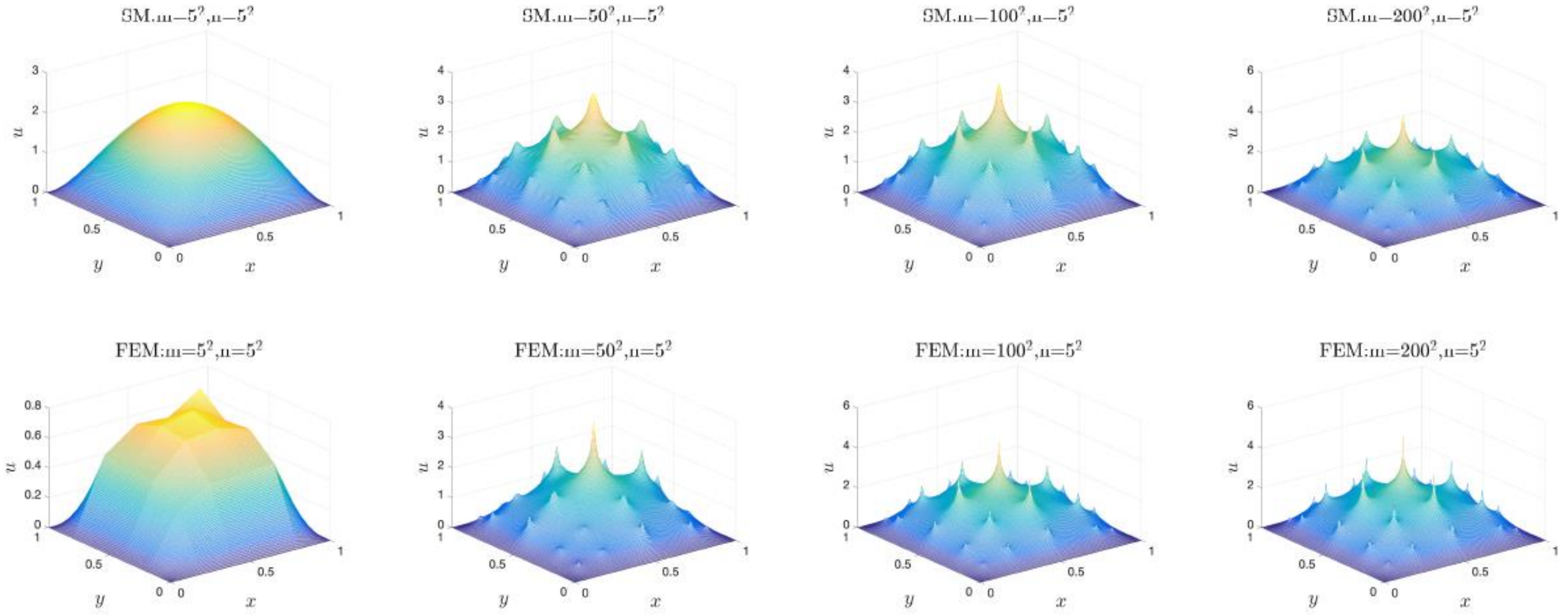



Figure 6: (Example 3): R-G solutions with different $m$. The basis functions for the first and the second row are Legendre basis function and piecewise linear basis function, respectively.

Wang, Xu, Zhang, Zhang, 2020, 2002.07989

# Learning Results: FEM as $m \to \infty$ for d=2

$$\begin{cases} -\Delta u(\boldsymbol{x}) = f(\boldsymbol{x}), & \boldsymbol{x} \in (0,1)^2, \\ u(\boldsymbol{x}) = 0, & \boldsymbol{x} \in \partial(0,1)^2, \end{cases}$$

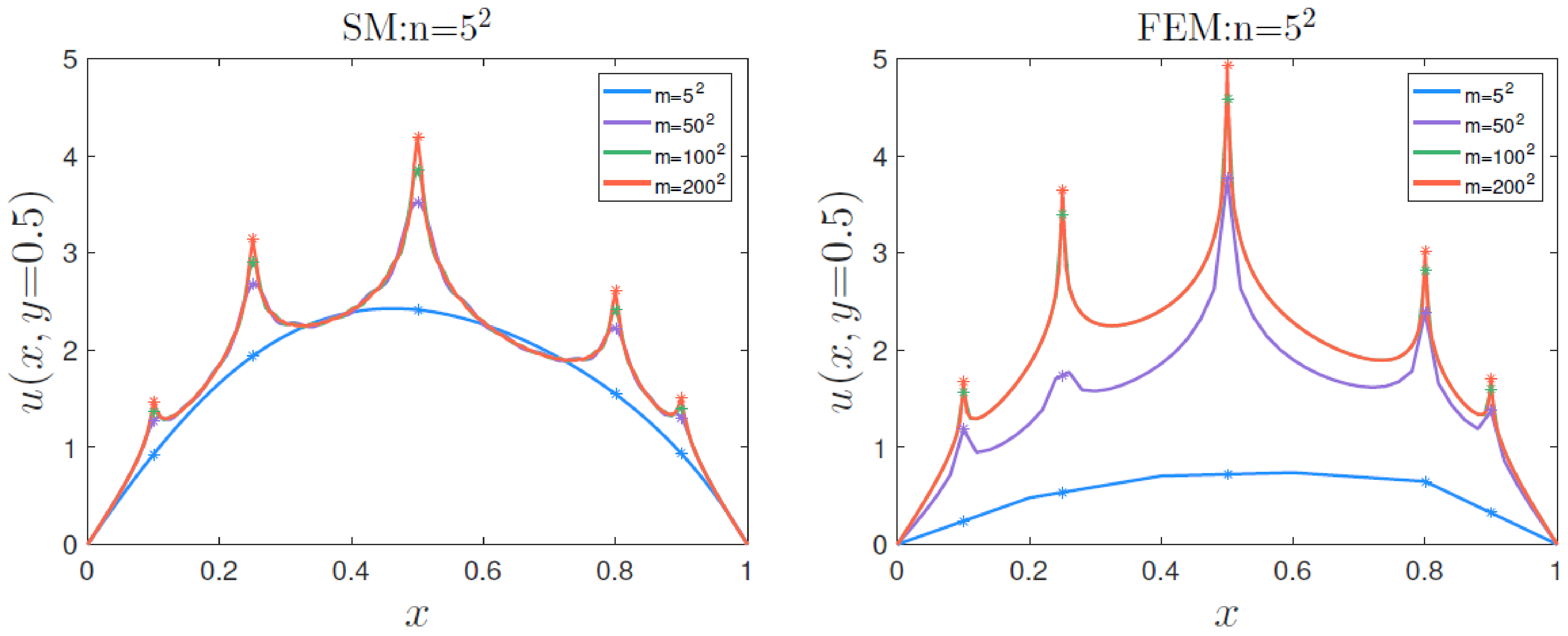



Figure 7: (Example 3): Profile of R-G solutions with different $m$.

Wang, Xu, Zhang, Zhang, 2020, 2002.07989

# Learning Results: DNN as $m \to \infty$ for d=2

$$\begin{cases} -\Delta u(\boldsymbol{x}) = f(\boldsymbol{x}), & \boldsymbol{x} \in (0,1)^2, \\ u(\boldsymbol{x}) = 0, & \boldsymbol{x} \in \partial(0,1)^2, \end{cases}$$

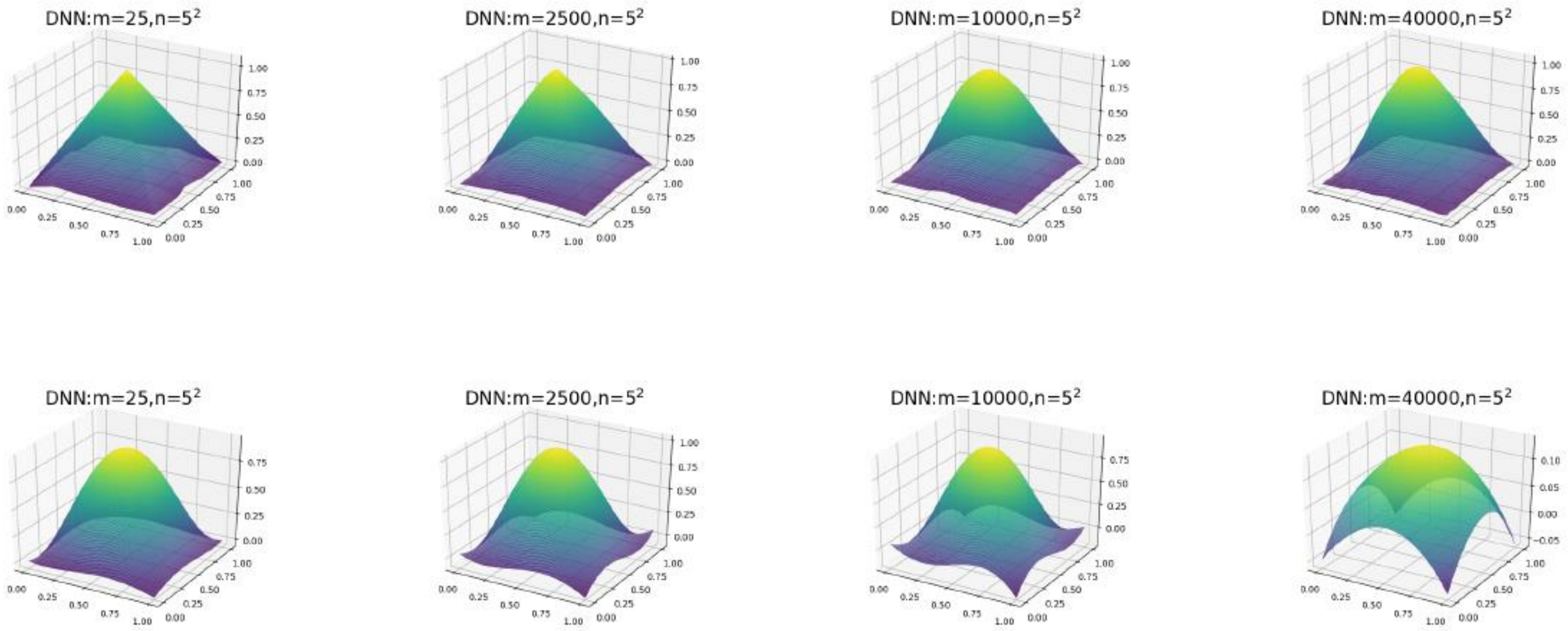



Figure 8: (Example 3): DNN solutions with different $m$. The activation functions for the first and the second row are $\mathrm{ReLU}(x)$ and $\sin(x)$, respectively.

Wang, Xu, Zhang, Zhang, 2020, 2002.07989

# MscaleDNN: A multi-scale DNN for high-D and frequency PDEs

- Use radial scaling in k-space to convert high frequency learning to low frequency learning, applicable to high-D problem

- Use compact support activation function (i.e. scaling and wavelet functions in wavelet theory)

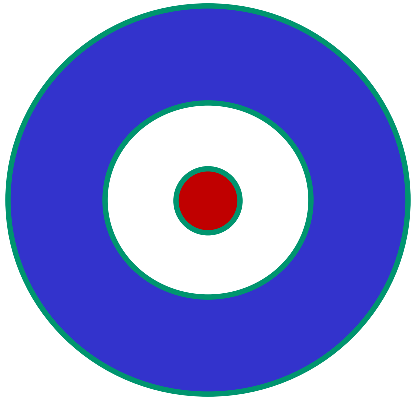
2007.11207

# Radial scaling in k-space

- Consider a band-limited function in R^d

$$\sup \widehat{f}(\mathbf{k}) \subset B(K_{\max}) = \{\mathbf{k} \in R^d, |\mathbf{k}| \leq K_{\max}\}.$$

$$B(K_{\max}) = \bigcup_{i=1}^{M} A_i$$



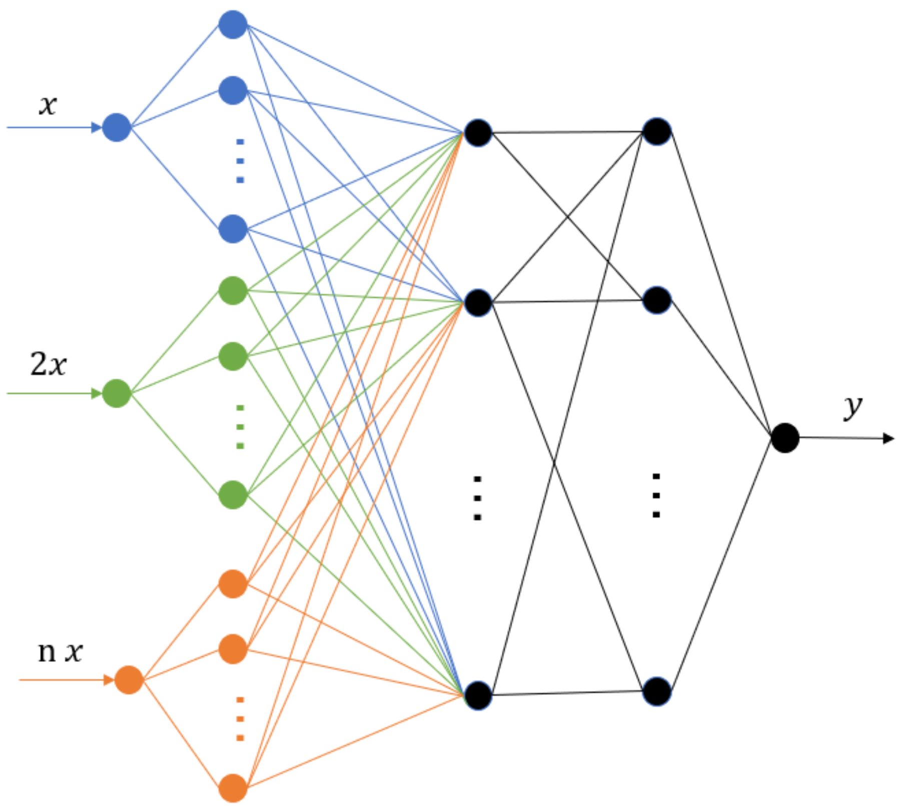
Rings in k-space: Red: low frequency
Blue: a high frequency ring A_i

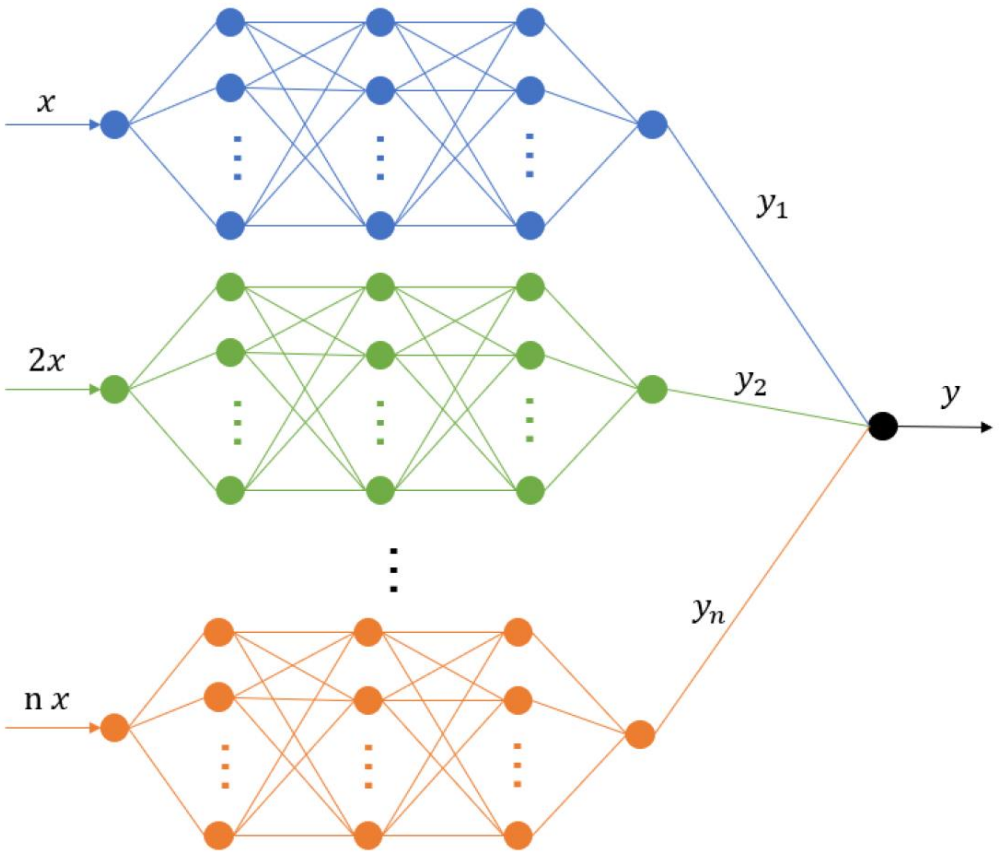
$$A_i = \{\mathbf{k} \in R^d, (i-1)K_0 \leq |\mathbf{k}| \leq iK_0\}, K_0 = K_{\max}/M, 1 \leq i \leq M$$

2007.11207

# MscaleDNN structures

Giving a MscaleDNN $$f(\mathbf{r}) \sim \sum_{i=1}^{M} h_i(\alpha_i \mathbf{r}, \theta^{n_i}).$$




2007.11207

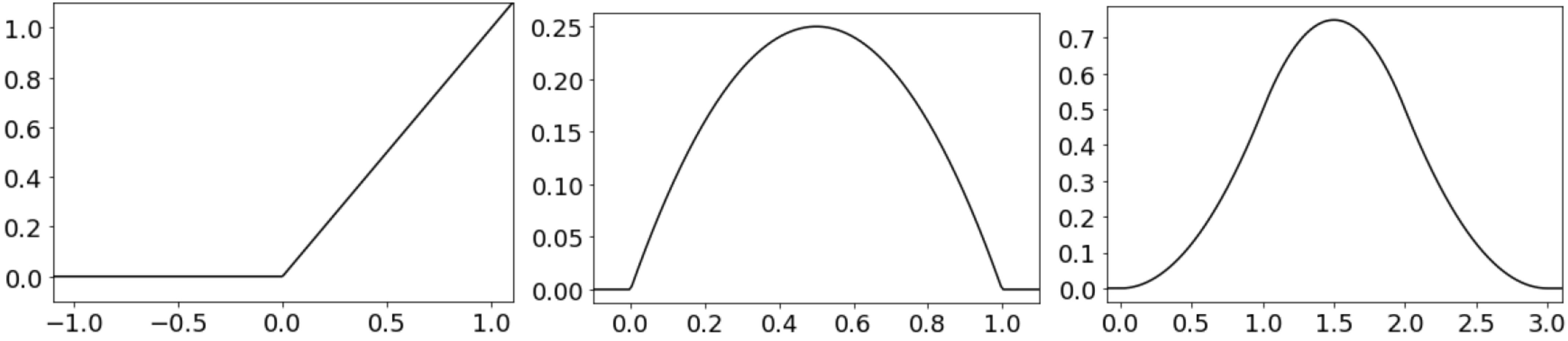
# Compact supported activation function

In order to produce scale separation and identification capability of the MscaleDNN, we take the hint from the theory of compact mother scaling function in the wavelet theory

$$\mathrm{sReLU}(x)=\mathrm{ReLU}(-(x-1))\times\mathrm{ReLU}(x)=(x)_+(1-x)_+,$$

$$\phi(x)=(x-0)_+^2-3(x-1)_+^2+3(x-2)_+^2-(x-3)_+^2.$$

$$\text{sin-sReLU}(x)=\sin(2\pi x)*\mathrm{ReLU}(x)*\mathrm{ReLU}(1-x)$$




(a) ReLU (b) sReLU (c) $\phi$

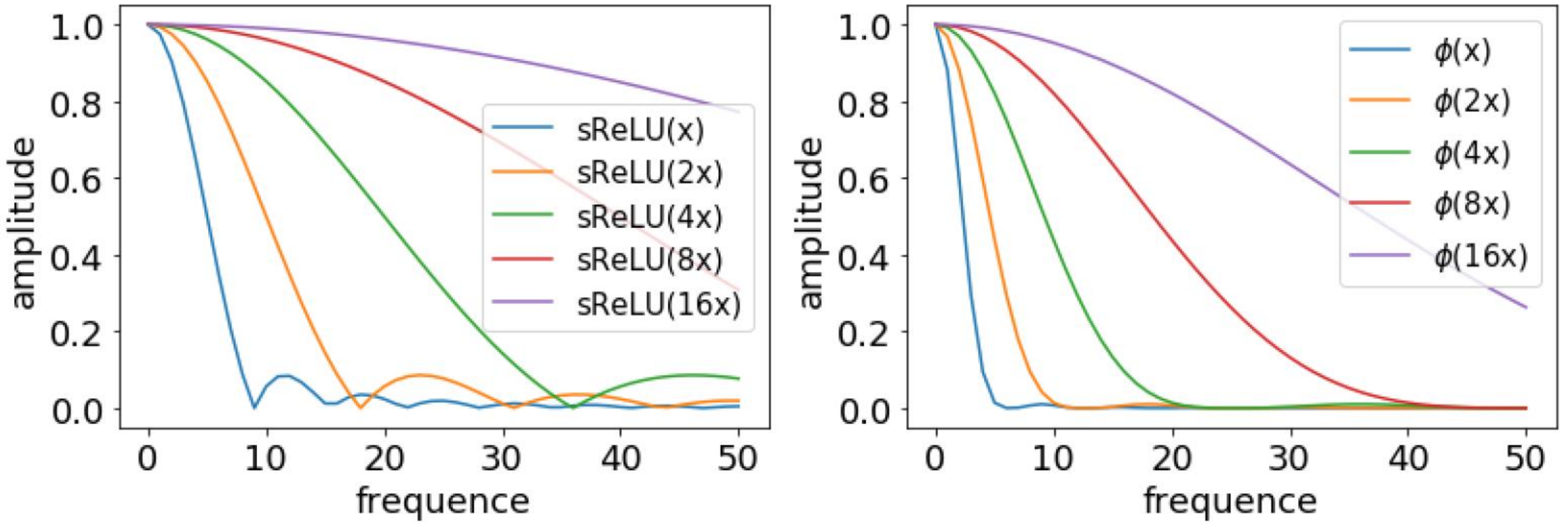
2007.11207

# Compact supported activation function

In order to produce scale separation and identification capability of the MscaleDNN, we take the hint from the theory of compact mother scaling function in the wavelet theory

$$\mathrm{sReLU}(x) = \mathrm{ReLU}(-(x-1)) \times \mathrm{ReLU}(x) = (x)_+(1-x)_+,$$

$$\phi(x) = (x-0)_+^2 - 3(x-1)_+^2 + 3(x-2)_+^2 - (x-3)_+^2.$$




(a) sReLU (b) $\phi$

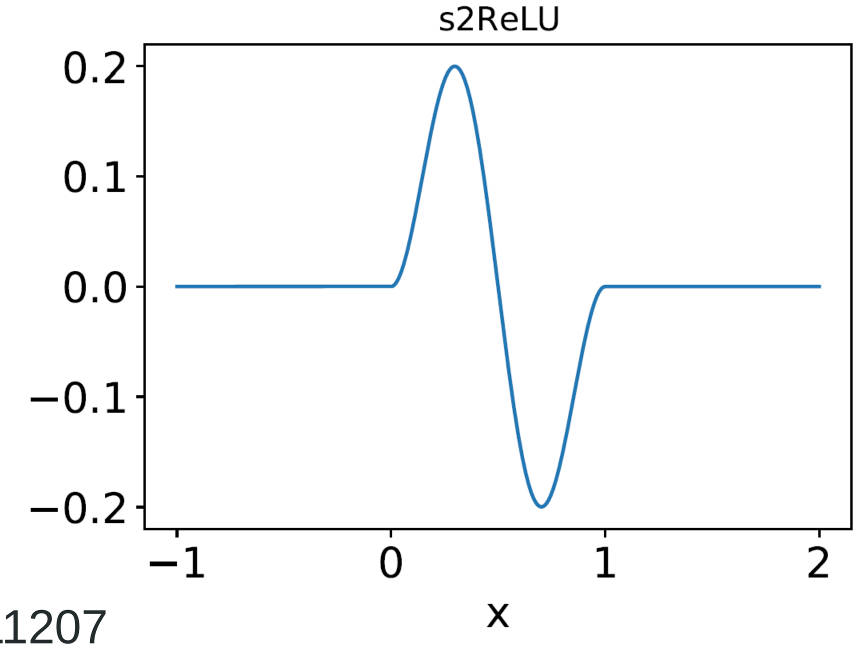
2007.11207

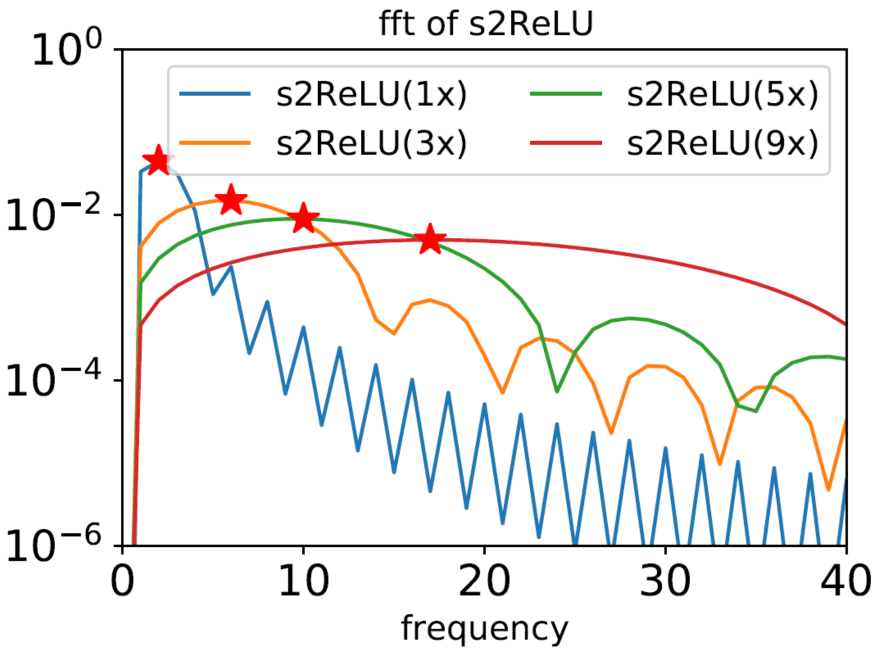
# Compact supported activation function

In order to produce scale separation and identification capability of the MscaleDNN, we take the hint from the theory of compact mother scaling function in the wavelet theory

$$\mathrm{sReLU}(x) = \mathrm{ReLU}(-(x-1)) \times \mathrm{ReLU}(x) = (x)_+(1-x)_+,$$

$$\text{sin-sReLU}(x) = \sin(2\pi x) * \mathrm{ReLU}(x) * \mathrm{ReLU}(1-x)$$




2007.11207

SHANGHAI JIAO TONG UNIVERSITY

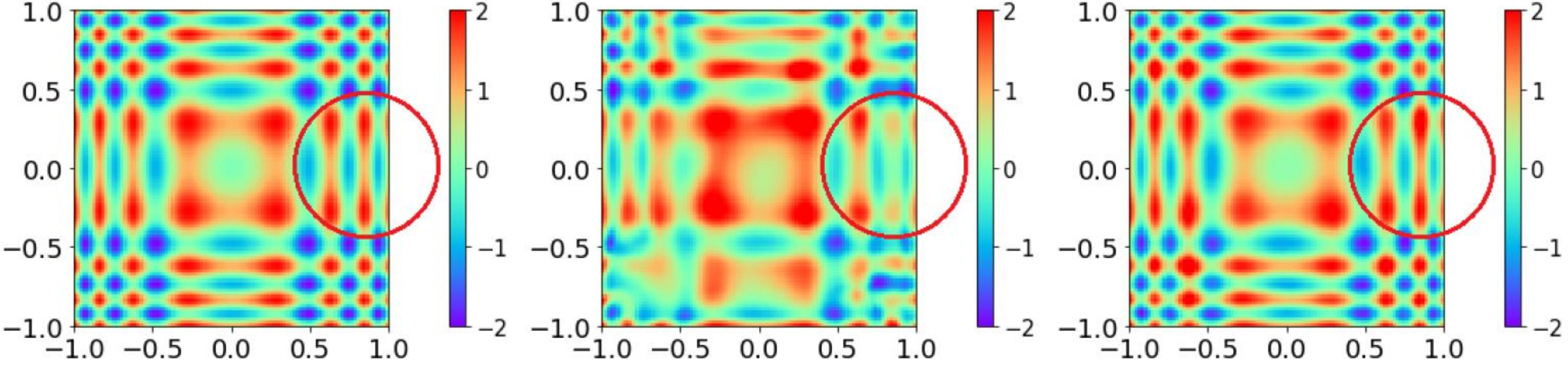
# Two dim case: not fixed frequency

$$-\Delta u(\boldsymbol{x}) = f(\boldsymbol{x}), \quad \Omega = [-1,1]^d$$

$$f(\boldsymbol{x}) = \sum_{i=1}^{d} 4\mu^2 x_i^2 \sin(\mu x_i^2) - 2\mu\cos(\mu x_i^2).$$

$$u(\boldsymbol{x}) = \sum_{i=1}^{d} \sin(\mu x_i^2)$$

$$\mu = 15$$




(a) exact (b) normal (c) Mscale

1. a fully-connected DNN with size **1-1000-1000-1000-1** (**normal**).

2. a MscaleDNN-2 with five subnetworks with size **1-200-200-200-1**, and scale coefficients $\{1,2,4,8,16\}$. (**Mscale**).

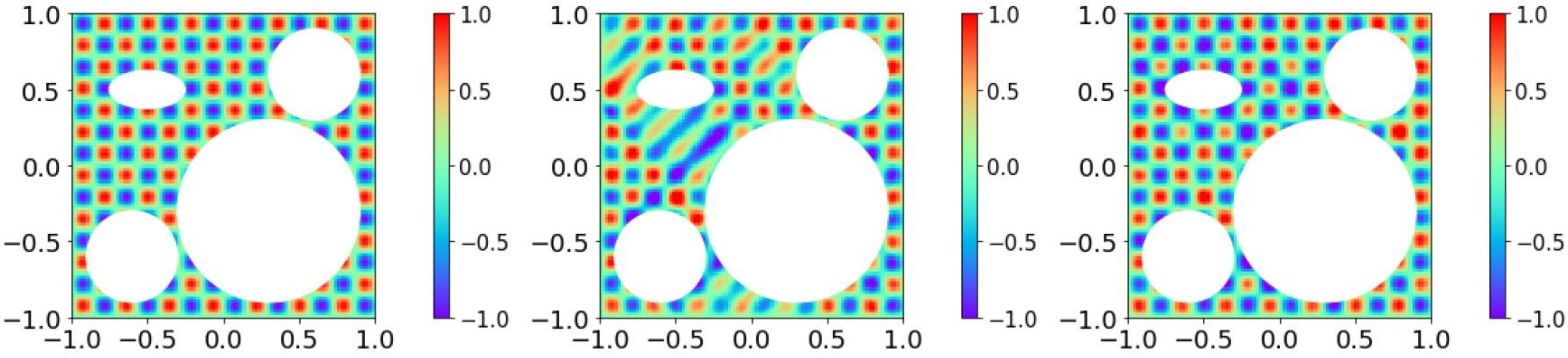
2007.11207

# Two dim case: complex domain

$$-\Delta u(\boldsymbol{x}) = f(\boldsymbol{x}), \quad \Omega = [-1,1]^d$$

$$u(\boldsymbol{x}) = \sin \mu x_1 \sin \mu x_2$$

$$f(\boldsymbol{x}) = 2\mu^2 \sin \mu x_1 \sin \mu x_2, \mu = 7\pi$$




(a) exact (b) normal (c) Mscale

2007.11207

# High-dim case

**Example 4.6.** We consider the following p-Laplacian problem in domain $\Omega=[0,1]^5$

$$\begin{cases} -\mathrm{div}\left(\kappa(x_1,x_2,\cdots,x_5)|\nabla u(x_1,x_2,\cdots,x_5)|^{p-2}\nabla u(x_1,x_2,\cdots,x_5)\right)=f(x_1,x_2,\cdots,x_5), \\ u(0,x_2,\cdots,x_5)=u(1,x_2,\cdots,x_5)=0, \\ \qquad\qquad \cdots\cdots \\ u(x_1,x_2,\cdots,0)=u(x_1,x_2,\cdots,1)=0. \end{cases} \tag{4.8}$$

In this example, we take $p=2$ and

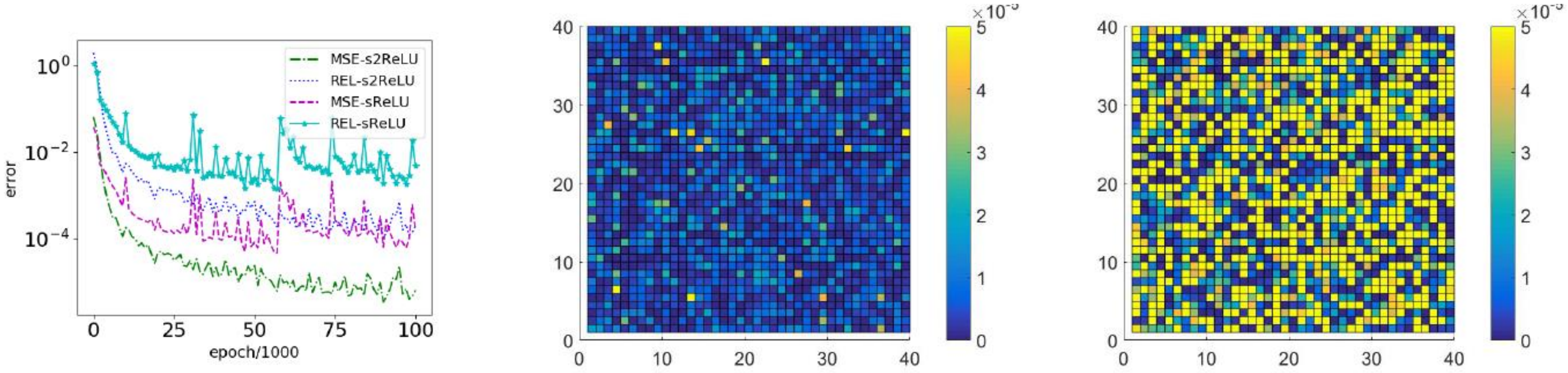
$$\kappa(x_1,x_2,\cdots,x_5)=1+\cos(\pi x_1)\cos(2\pi x_2)\cos(3\pi x_3)\cos(2\pi x_4)\cos(\pi x_5).$$

We choose the forcing term $f$ such that the exact solution is

$$u(x_1,x_2,\cdots,x_5)=\sin(\pi x_1)\sin(\pi x_2)\sin(\pi x_3)\sin(\pi x_4)\sin(\pi x_5).$$

2009.14597

# High-dim case

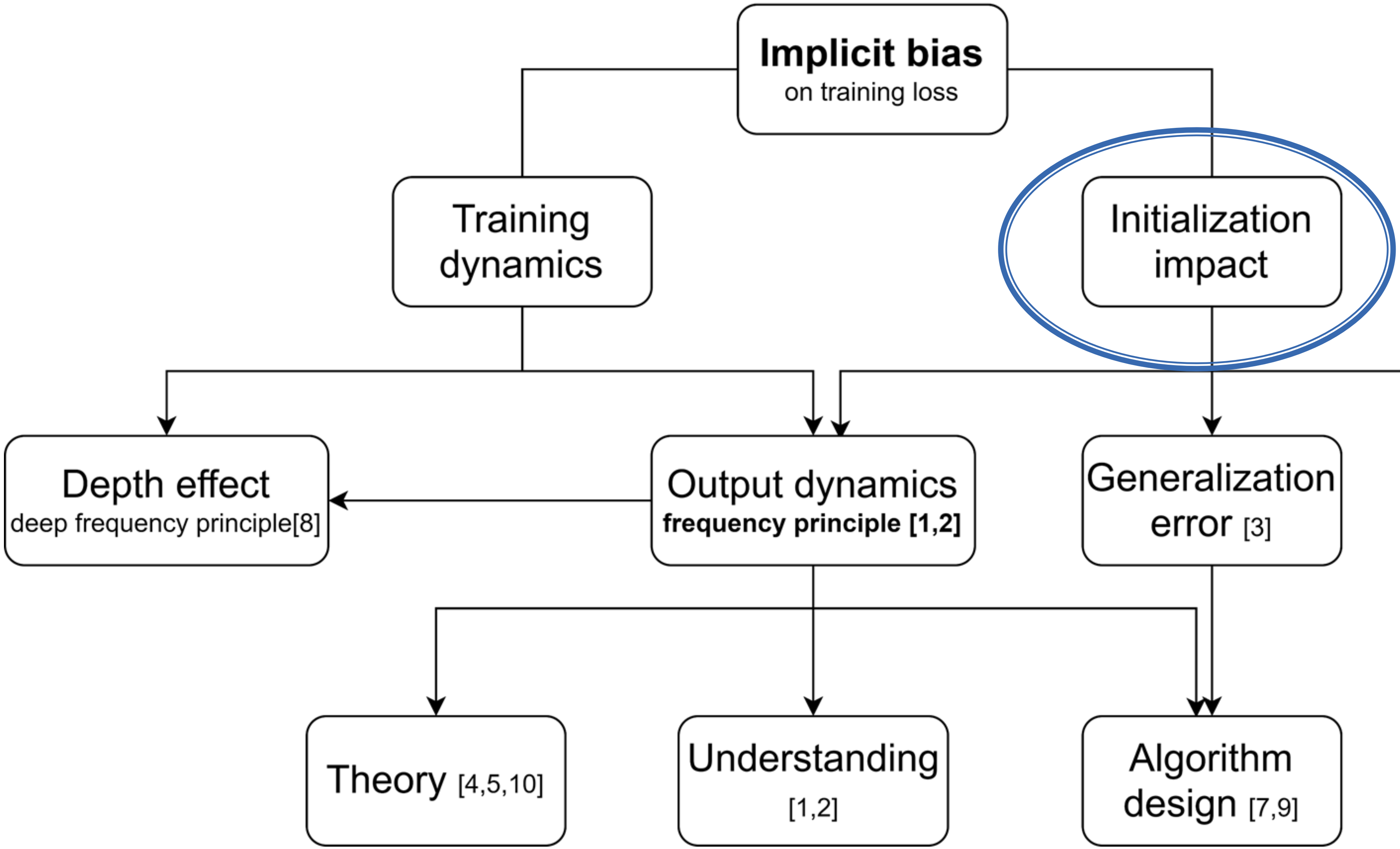



(a) MSE and REL (b) point-wise error (c) point-wise error

Figure 11: Testing results for Example 4.6. 11(a): Mean square error and relative error for s2ReLU and sReLU, respectively. 11(b): Point-wise square error for s2ReLU. 11(c): Point-wise square error for sReLU.

2009.14597

# A research picture on studying deep neural networks

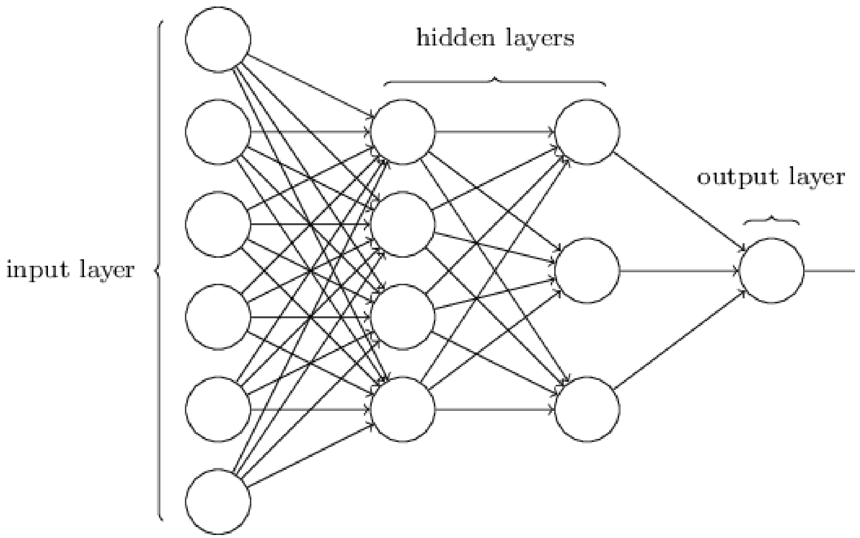



*: corr author, #: equal contri

$: alphabetical order

[1] Xu*, Zhang, Xiao, 1807.01251
[2] Xu*, Zhang, Luo, Xiao, Ma, 1901.06523
[3] Zhang, Xu*, Luo, Ma, 1905.07777
[4] Zhang, Xu*, Luo, Ma, 1905.10264
[5$] Luo*, Ma, Xu, Zhang, 1906.09235
[6] Luo#, Xu#, Ma, Zhang*, 2007.07497
[7] Liu, Cai, Xu*, 2007.11207
[8] Xu*, Zhou, 2007.14313
[9$] Li, Xu*, Zhang, 2009.14597
[10$] Luo, Ma, Xu, Zhang, 2010.08153

## Deep Neural Network




$$h(x;\boldsymbol{\theta}) = h^{[H]}$$

$$h^{[j]} = \sigma\left(W^{[j]}h^{[j-1]} + b^{[j]}\right)$$

$$\boldsymbol{\theta}: \left[W^{[j]}, b^{[j]}\right]_{j=1,\cdots,H}$$

Example: Two-layer NN

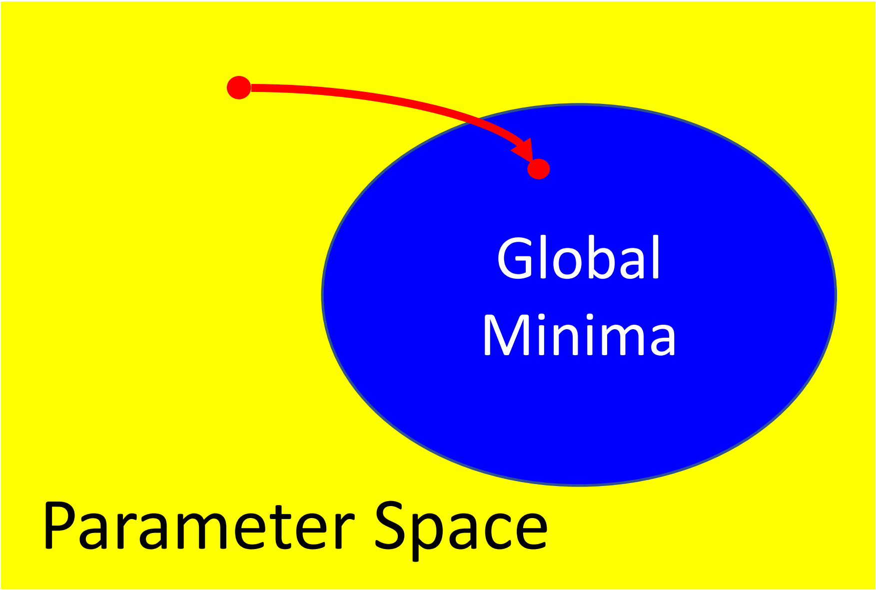
$$h_{\boldsymbol{\theta}}(x) = \sum_{i=1}^{m_1} w_i^{[2]}\sigma(w_i^{[1]}x + b_i^{[1]})$$

## Dynamics (regression)

Data: $\{(x_i, y_i)\}_{i=1}^{n}$

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{n}(h_{\boldsymbol{\theta}}(x_i) - y_i)^2$$

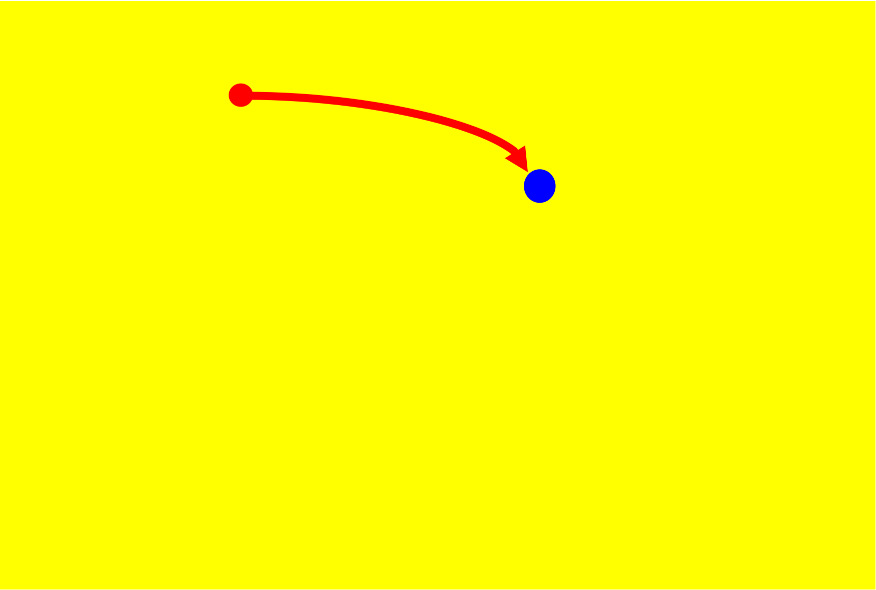
$$\dot{\boldsymbol{\theta}} = -\nabla_{\boldsymbol{\theta}}L(\boldsymbol{\theta})$$



**Solution is determined by**

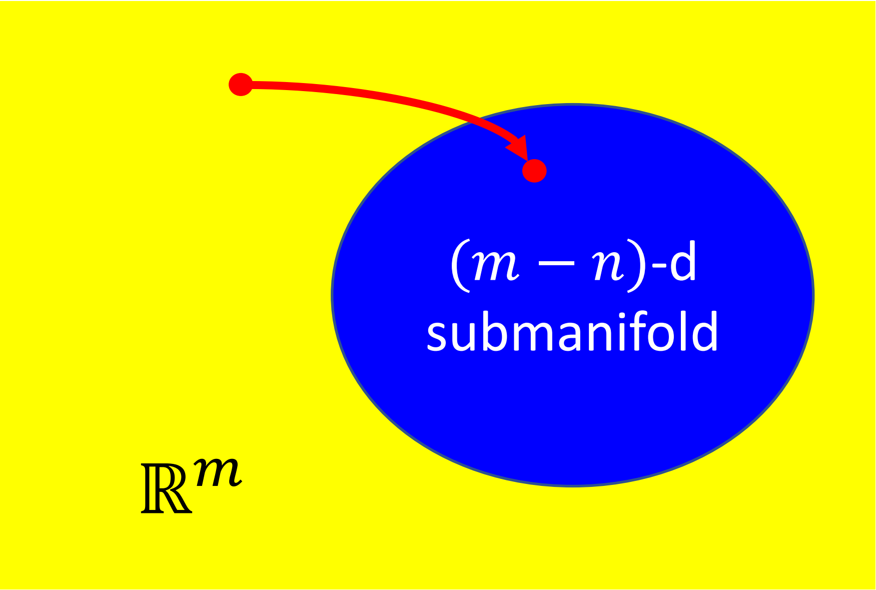
**Conventional optimization**

Loss + L1/L2/…



**Deep learning**

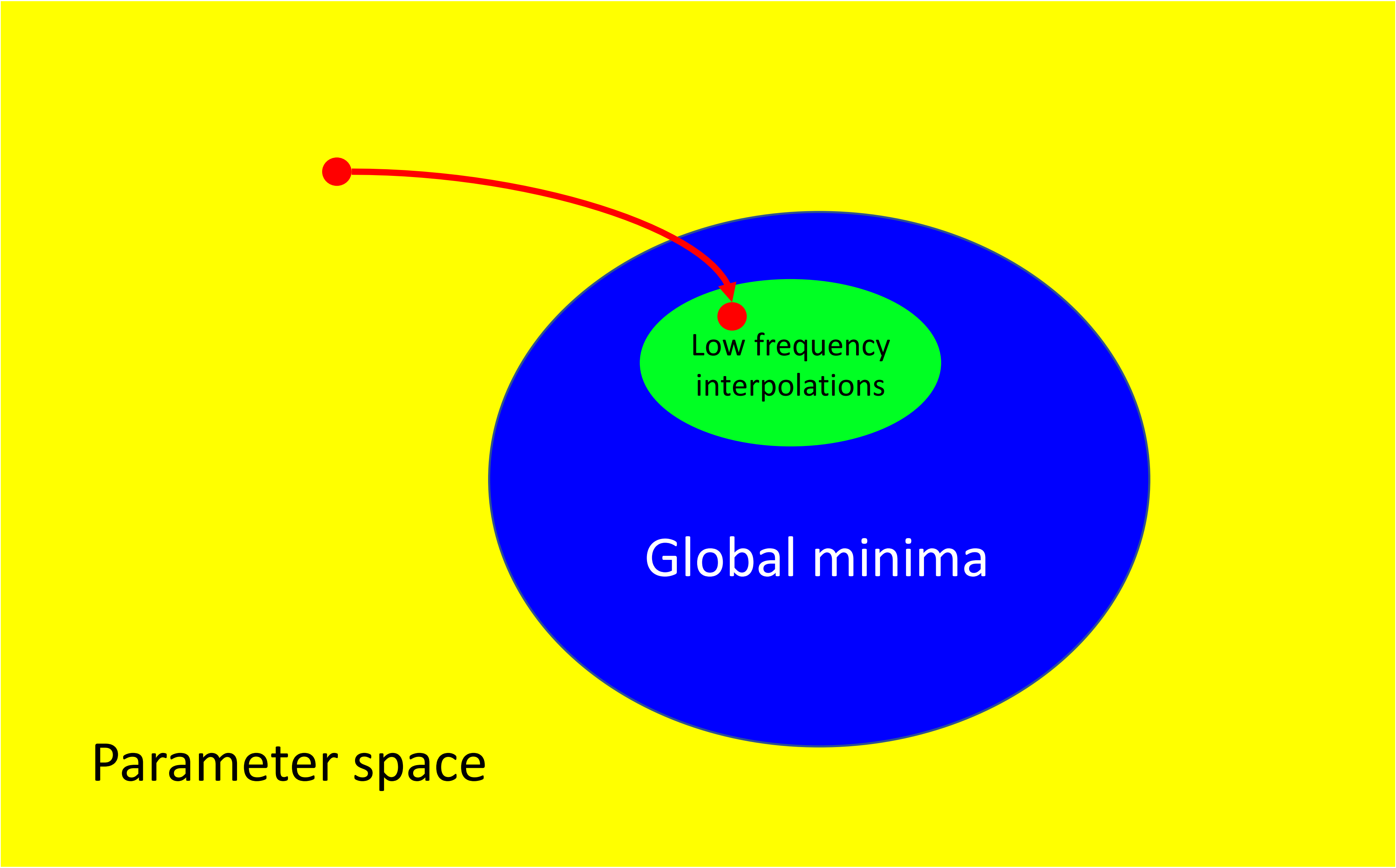
loss + initialization + dynamics




Yaim Cooper, 2018

# Picture of deep learning with frequency principle

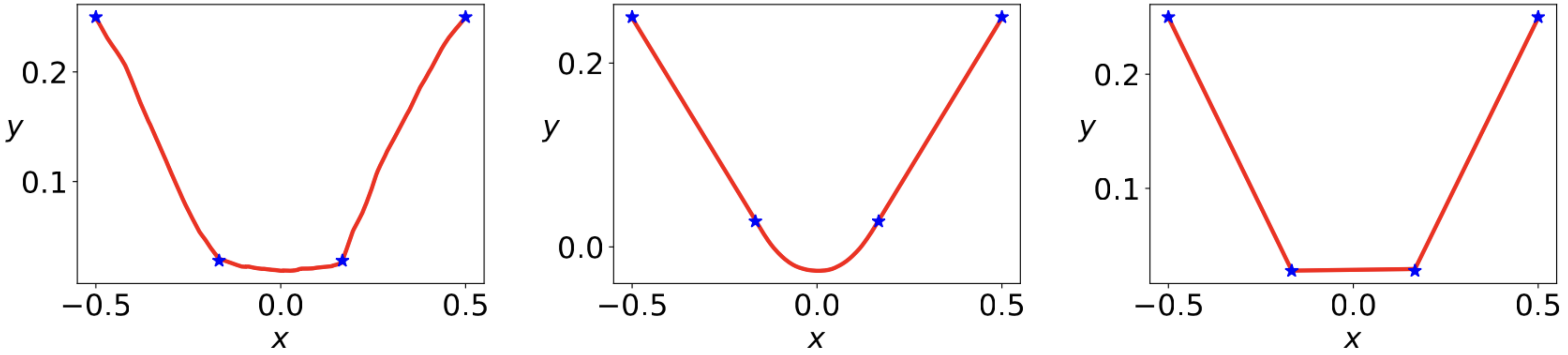
loss
initialization
dynamics



# Impact of initialization on generalization via training dynamics

T Luo, ZQJ Xu, Z Ma, Y Zhang, *Phase diagram for two-layer ReLU neural networks at infinite-width limit, JMLR*, 2020

# Motivation




T Luo, ZQJ Xu, Z Ma, Y Zhang, *Phase diagram for two-layer ReLU neural networks at infinite-width limit*, 2020

# Setup

- Two layer ReLU network at infinite-width limit

$$f_{\boldsymbol{\theta}}^{\alpha}(\boldsymbol{x}) = \frac{1}{\boxed{\alpha}} \sum_{k=1}^{m} a_k \sigma(\boldsymbol{w}_k^{\intercal} \boldsymbol{x}) \qquad a_k^0 \sim N(0, \boxed{\beta_1^2}),\ \boldsymbol{w}_k^0 \sim N(0, \boxed{\beta_2^2} \boldsymbol{I}_d)$$

$$x = [x^T, 1]^T$$
$$w_k = [w_k^T, b_k]^T$$

- Normalized gradient flow

$$\bar{a}_k = \beta_1^{-1} a_k, \quad \bar{\boldsymbol{w}}_k = \beta_2^{-1} \boldsymbol{w}_k, \quad \bar{t} = \frac{1}{\beta_1 \beta_2} t,$$

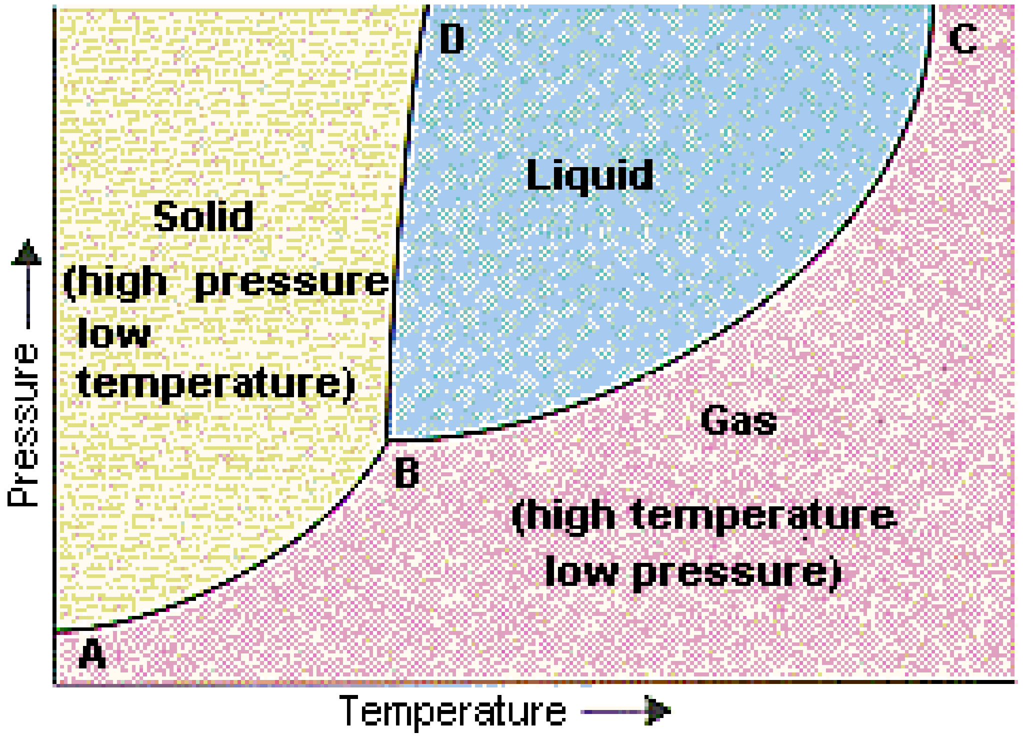
$$\frac{\mathrm{d}\bar{a}_k}{\mathrm{d}\bar{t}} = -\frac{\beta_2}{\beta_1} \frac{1}{n} \sum_{i=1}^{n} \frac{\beta_1 \beta_2}{\alpha} \sigma(\bar{\boldsymbol{w}}_k^{\intercal} \boldsymbol{x}_i) \left( \frac{\beta_1 \beta_2}{\alpha} \sum_{k=1}^{m} \bar{a}_k \sigma(\bar{\boldsymbol{w}}_k^{\intercal} \boldsymbol{x}_i) - y_i \right),$$

$$\frac{\mathrm{d}\bar{\boldsymbol{w}}_k}{\mathrm{d}\bar{t}} = -\frac{\beta_1}{\beta_2} \frac{1}{n} \sum_{i=1}^{n} \frac{\beta_1 \beta_2}{\alpha} \bar{a}_k \sigma'(\bar{\boldsymbol{w}}_j^{\intercal} \boldsymbol{x}_i) \boldsymbol{x}_i \left( \frac{\beta_1 \beta_2}{\alpha} \sum_{k=1}^{m} \bar{a}_k \sigma(\bar{\boldsymbol{w}}_k^{\intercal} \boldsymbol{x}_i) - y_i \right).$$

- Scaling parameters and infinite-width limit

$$\kappa := \frac{\beta_1 \beta_2}{\alpha}, \quad \kappa' := \frac{\beta_1}{\beta_2}, \qquad \gamma = \lim_{m \to \infty} -\frac{\log \kappa}{\log m}, \quad \gamma' = \lim_{m \to \infty} -\frac{\log \kappa'}{\log m},$$

# Phase diagram




- **Phase diagram for matter**

distinctive states of matter <-> environment
(phase transition happens at infinite size limit)
solid, liquid, gas <-> pressure, temperature

- **Phase diagram for two-layer ReLU NN**

distinctive training dynamics <-> initialization
$(m \to \infty)$
? <->?

**Identification of coordinates of phase diagram (in analogy to pressure, temperature)**

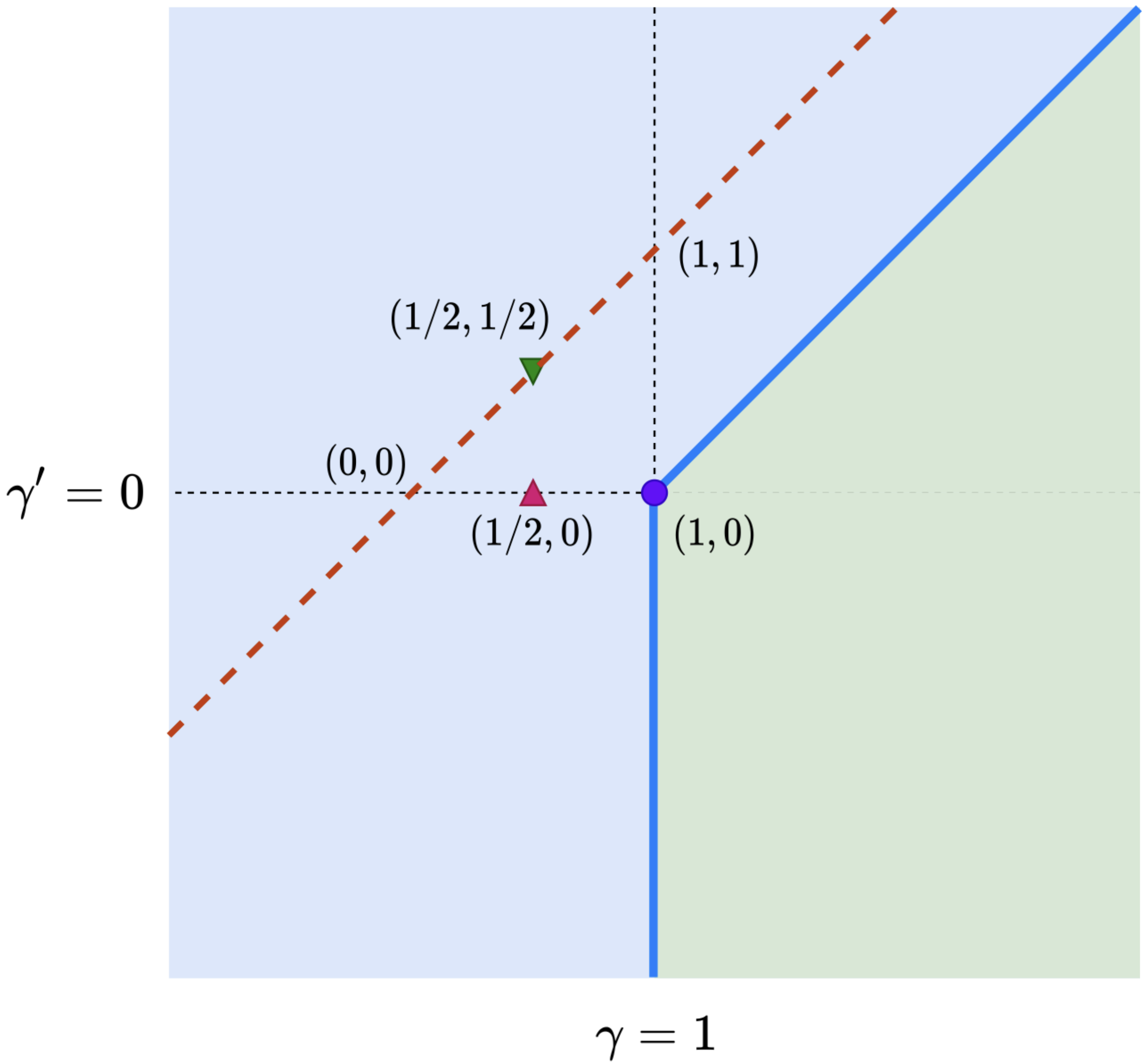
1. Effectively independent
2. Dynamical similarity
3. Differentiation capability

$$\gamma = \lim_{m\to\infty} -\frac{\log \beta_1\beta_2/\alpha}{\log m}, \quad \gamma' = \lim_{m\to\infty} -\frac{\log \beta_1/\beta_2}{\log m}$$

# Initialization methods with their scaling parameters

| Name (related works) | $\alpha$ | $\beta_1$ | $\beta_2$ | $\kappa$ $\left(\frac{\beta_1\beta_2}{\alpha}\right)$ | $\kappa'$ $\left(\frac{\beta_1}{\beta_2}\right)$ | $\gamma$ $\left(\lim_{m\to\infty}\frac{\log 1/\kappa}{\log m}\right)$ | $\gamma'$ $\left(\lim_{m\to\infty}\frac{\log 1/\kappa'}{\log m}\right)$ |
|---|---|---|---|---|---|---|---|
| LeCun (LeCun et al., 2012) | $1$ | $\sqrt{\frac{1}{m}}$ | $\sqrt{\frac{1}{d}}$ | $\sqrt{\frac{1}{md}}$ | $\sqrt{\frac{d}{m}}$ | $\frac{1}{2}$ | $\frac{1}{2}$ |
| He (He et al., 2015) | $1$ | $\sqrt{\frac{2}{m}}$ | $\sqrt{\frac{2}{d}}$ | $\sqrt{\frac{4}{md}}$ | $\sqrt{\frac{d}{m}}$ | $\frac{1}{2}$ | $\frac{1}{2}$ |
| Xavier (Glorot and Bengio, 2010) | $1$ | $\sqrt{\frac{2}{m+1}}$ | $\sqrt{\frac{2}{m+d}}$ | $\sqrt{\frac{4}{(m+1)(m+d)}}$ | $\sqrt{\frac{m+d}{m+1}}$ | $1$ | $0$ |
| NTK (Jacot et al., 2018) | $\sqrt{m}$ | $1$ | $1$ | $\sqrt{\frac{1}{m}}$ | $1$ | $\frac{1}{2}$ | $0$ |
| Mean-field (Mei et al., 2018) (Sirignano and Spiliopoulos, 2020) (Rotskoff and Vanden-Eijnden, 2018) | $m$ | $1$ | $1$ | $\frac{1}{m}$ | $1$ | $1$ | $0$ |
| E et al. (E et al., 2020) | $1$ | $\beta$ | $1$ | $\beta$ | $\beta$ | $\lim_{m\to\infty}\frac{\log 1/\beta}{\log m}$ | $\lim_{m\to\infty}\frac{\log 1/\beta}{\log m}$ |

# Phase Diagram

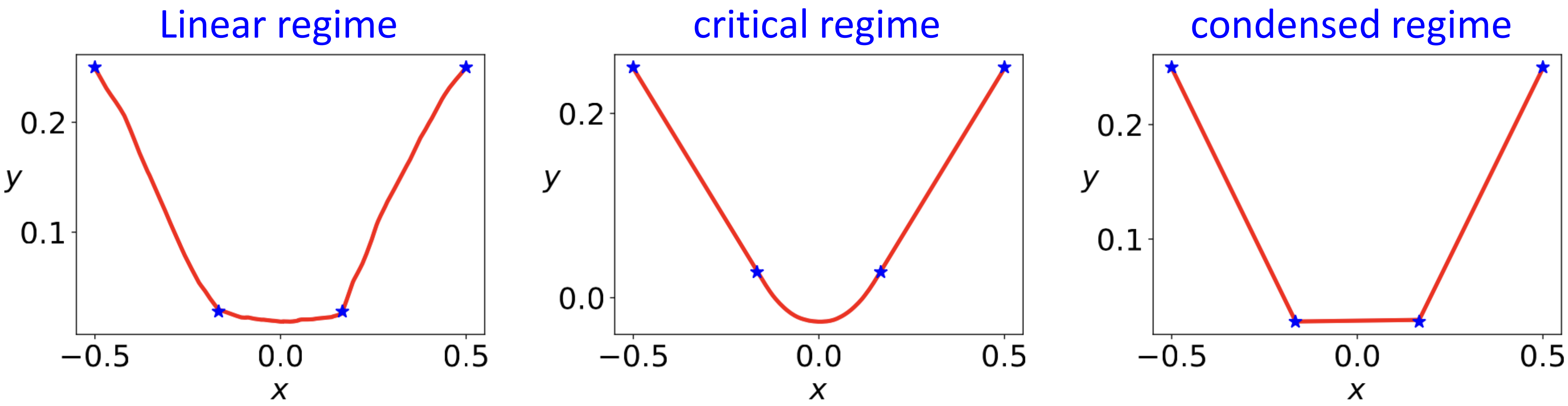



$$a_k^0 \sim N(0, \beta_1^2), \quad \mathbf{w}_k^0 \sim N(0, \beta_2^2 \mathbf{I}_d)$$

$$\gamma = \lim_{m \to \infty} -\frac{\log \beta_1 \beta_2 / \alpha}{\log m}, \quad \gamma' = \lim_{m \to \infty} -\frac{\log \beta_1 / \beta_2}{\log m}$$

# Typical cases across the phase diagram

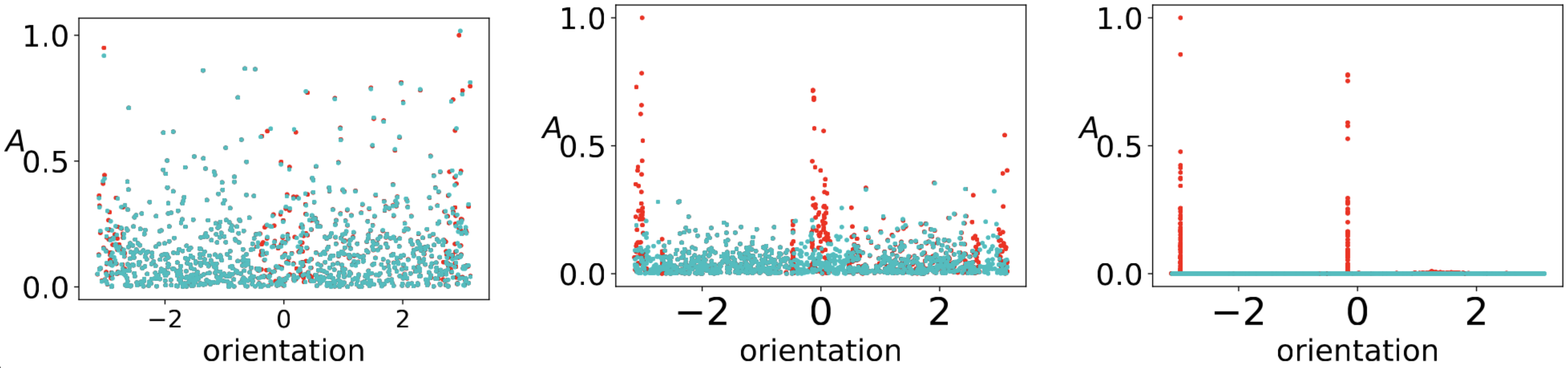
$\gamma' = 0$




(a) $\gamma = 0.5$ (b) $\gamma = 1$ (c) $\gamma = 1.75$

$$\{(A_k, \hat{\boldsymbol{w}}_k)\}_{k=1}^{m}$$

$$A = |a|\|\boldsymbol{w}\|_2$$




(d) $\gamma = 0.5$ (e) $\gamma = 1$ (f) $\gamma = 1.75$

$$f_{\boldsymbol{\theta}}^{\alpha}(\boldsymbol{x}) = \frac{1}{\alpha} \sum_{k=1}^{m} a_k \sigma(\boldsymbol{w}_k^{\intercal} \boldsymbol{x})$$

# Regime identification

- Linear regime (with ASI)

$$f_{\boldsymbol{\theta}}^{\mathrm{lin}} = \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}(0)} \cdot (\boldsymbol{\theta}(t) - \boldsymbol{\theta}(0)).$$

- Relative distance

$$\mathrm{RD}(\boldsymbol{\theta}_{\boldsymbol{w}}(t)) = \frac{\|\theta_{\boldsymbol{w}}(t) - \theta_{\boldsymbol{w}}(0)\|_2}{\|\theta_{\boldsymbol{w}}(0)\|_2}.$$

$$f_{\boldsymbol{\theta}}^{\alpha}(\boldsymbol{x}) = \frac{1}{\alpha} \sum_{k=1}^{m} a_k \sigma(\boldsymbol{w}_k^{\intercal} \boldsymbol{x})$$

$$\boldsymbol{\theta}_{\boldsymbol{w}} = \mathrm{vec}(\{\boldsymbol{w}_k\}_{k=1}^{m})$$

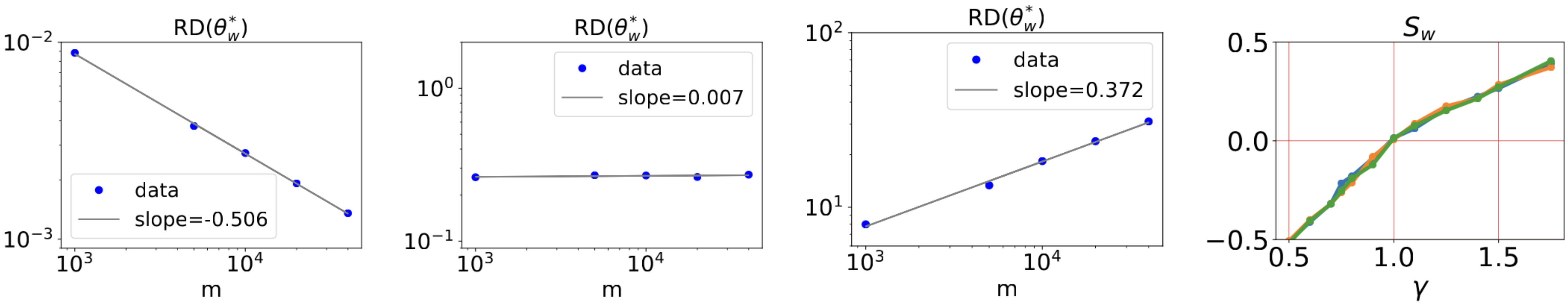
As $m \to \infty$,

- Linear regime: $\sup_{t \in [0,+\infty)} \mathrm{RD}(\boldsymbol{\theta}_{\boldsymbol{w}}(t)) \to 0$
- Condensed regime: $\sup_{t \in [0,+\infty)} \mathrm{RD}(\boldsymbol{\theta}_{\boldsymbol{w}}(t)) \to +\infty$
- Critical regime: $\sup_{t \in [0,+\infty)} \mathrm{RD}(\boldsymbol{\theta}_{\boldsymbol{w}}(t)) \to O(1)$.

# Regime identification through experiments

$\gamma' = 0$

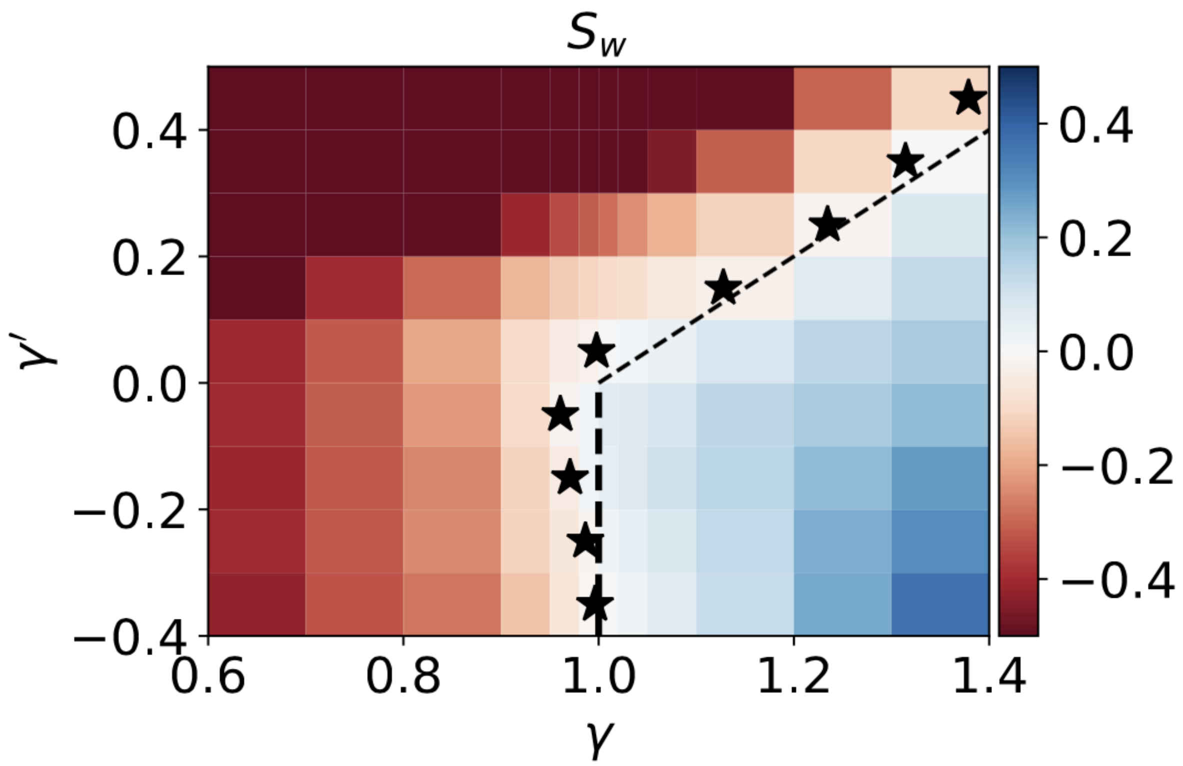
$$S_{\boldsymbol{w}} = \lim_{m\to\infty} \frac{\log \mathrm{RD}(\boldsymbol{\theta}^*_{\boldsymbol{w}})}{\log m}$$



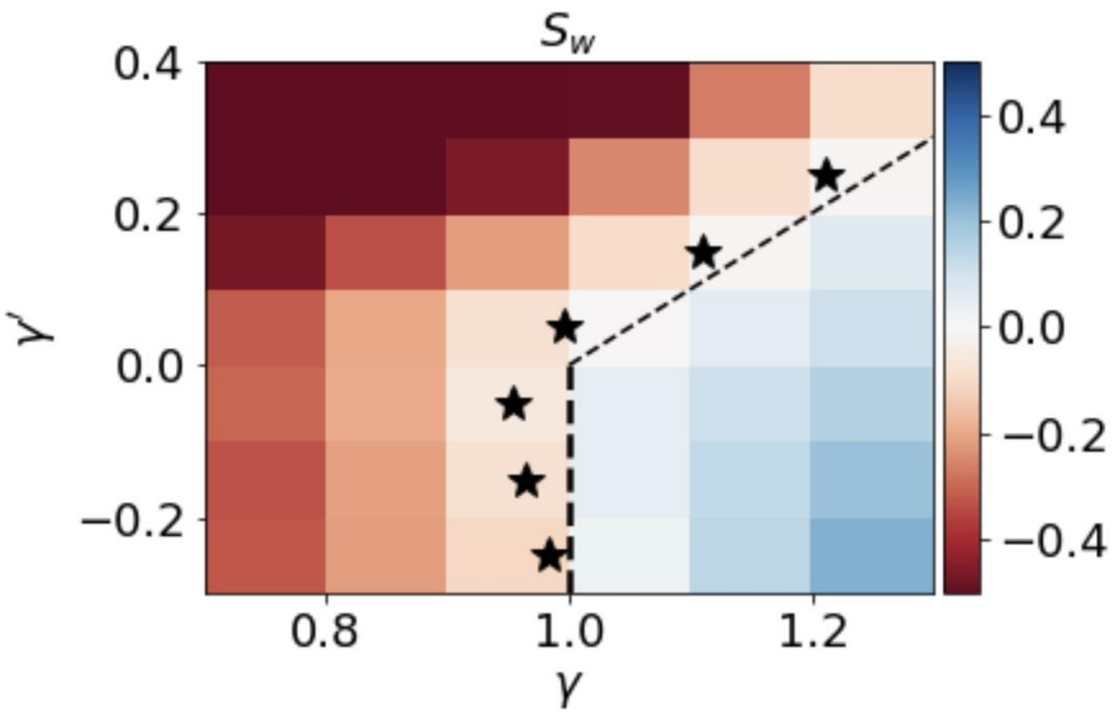

(a) $\gamma = 0.5$ (b) $\gamma = 1$ (c) $\gamma = 1.75$ (d) $S_w$ vs. $\gamma$

# Regime identification through experiments

Synthetic data




MNIST data



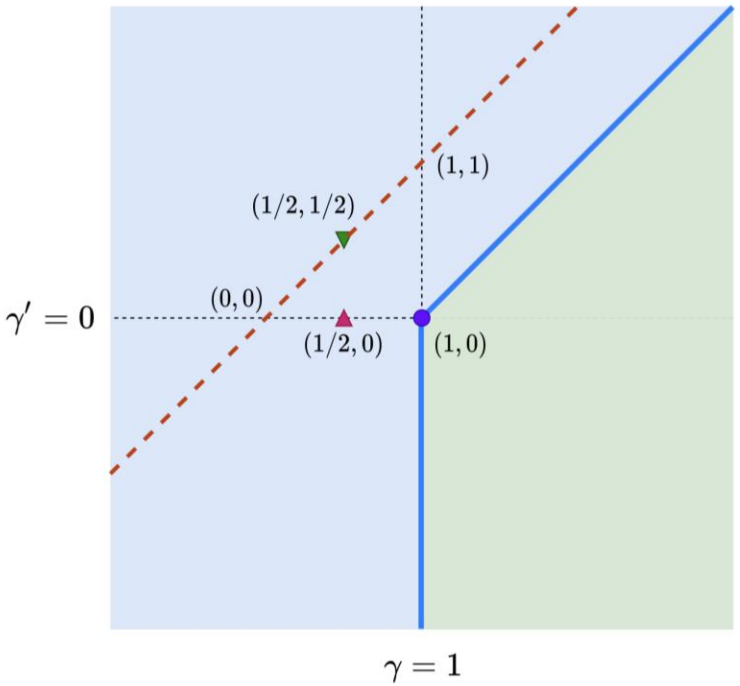
# Regime separation -- theorems

**Theorem 1*.** *(Informal statement of Theorem 6) If $\gamma < 1$ or $\gamma' > \gamma - 1$, then with a high probability over the choice of $\boldsymbol{\theta}^0$, we have*

$$\lim_{m \to +\infty} \sup_{t \in [0, +\infty)} \mathrm{RD}(\boldsymbol{\theta}_{\boldsymbol{w}}(t)) = 0. \tag{20}$$

**Theorem 2*.** *(Informal statement of Theorem 8) If $\gamma > 1$ and $\gamma' < \gamma - 1$, then with a high probability over the choice of $\boldsymbol{\theta}^0$, we have*

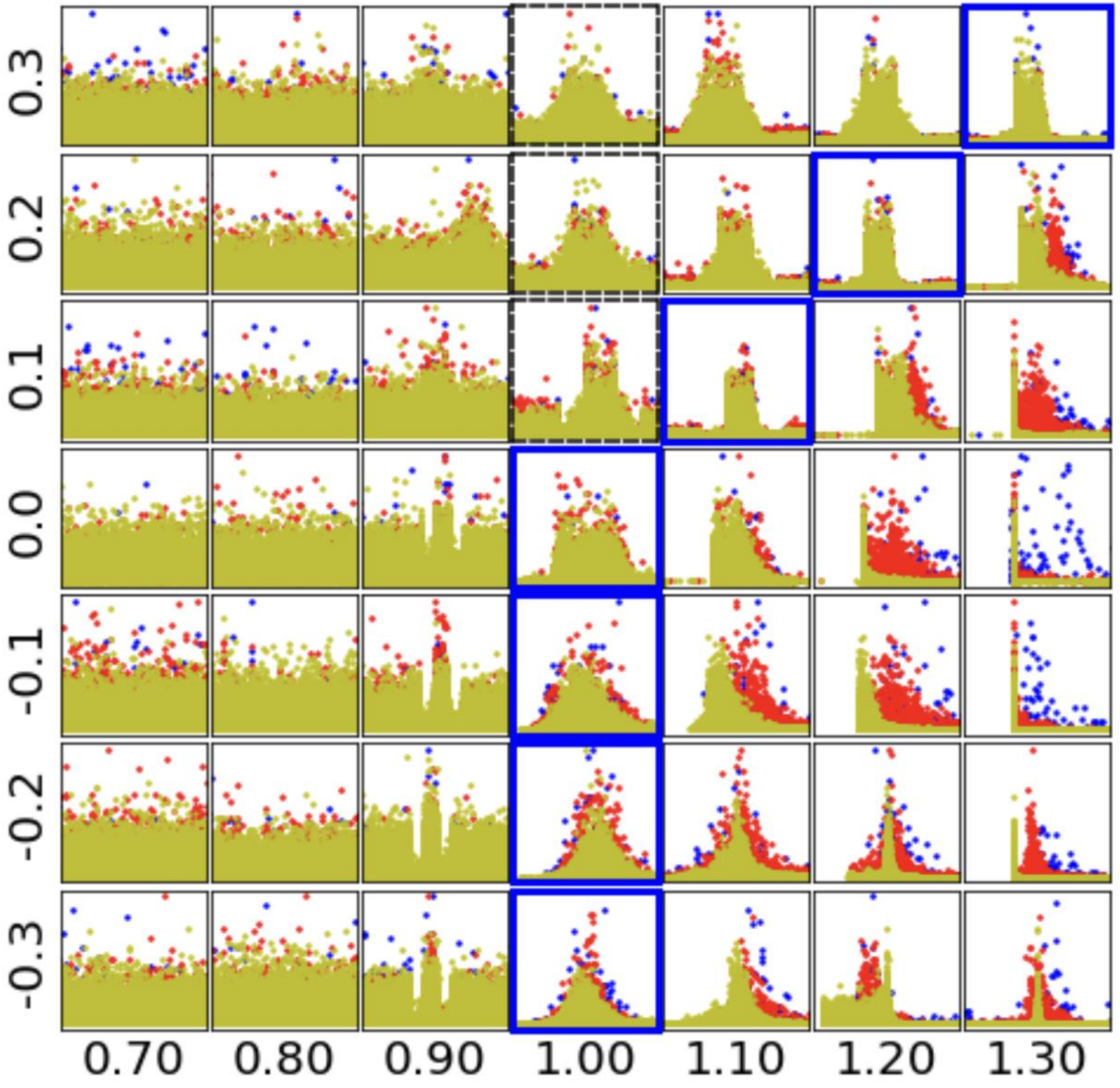
$$\lim_{m \to +\infty} \sup_{t \in [0, +\infty)} \mathrm{RD}(\boldsymbol{\theta}_{\boldsymbol{w}}(t)) = +\infty. \tag{21}$$



# Feature distribution at the condensed regime

$$\{(A_k, \hat{\boldsymbol{w}}_k)\}_{k=1}^{m}$$

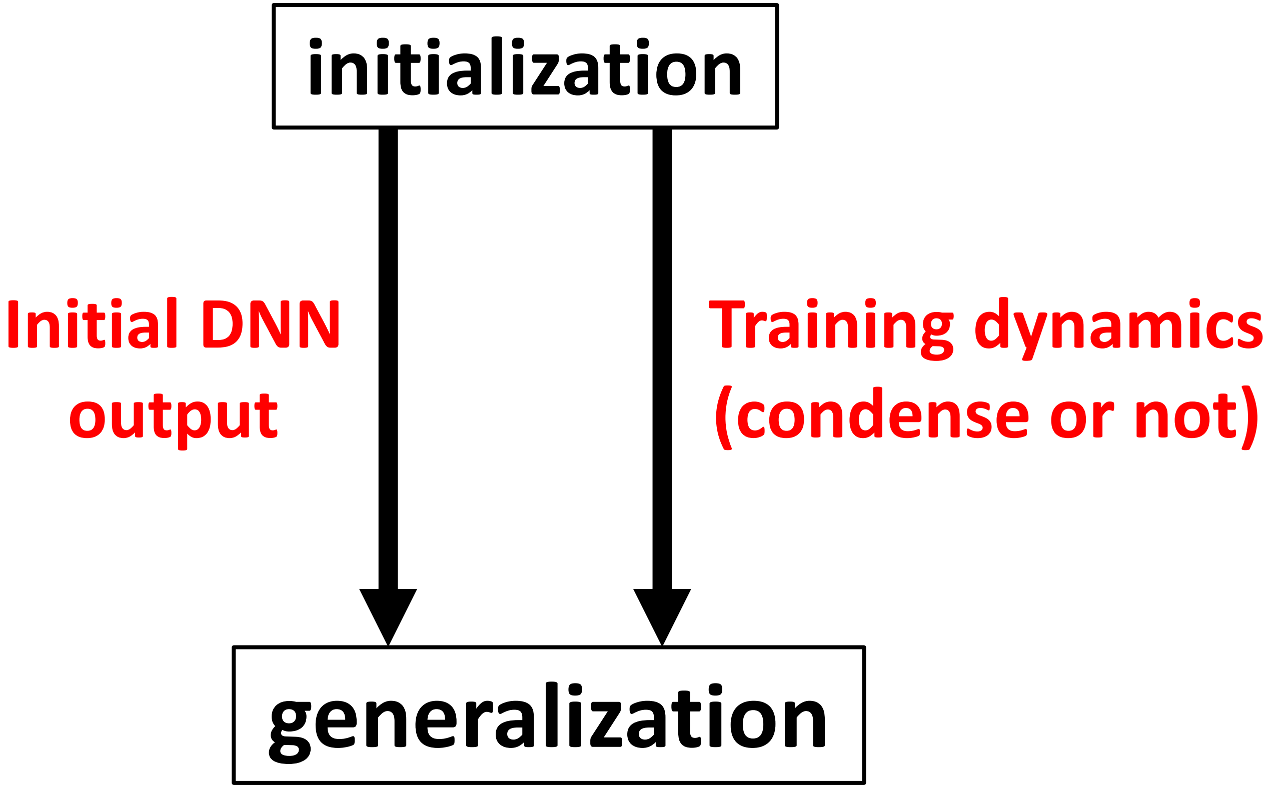
$$A = |a|\|\boldsymbol{w}\|_2$$




Blue: $m = 10^3$
red: $m = 10^4$
Yellow: $m = 10^6$

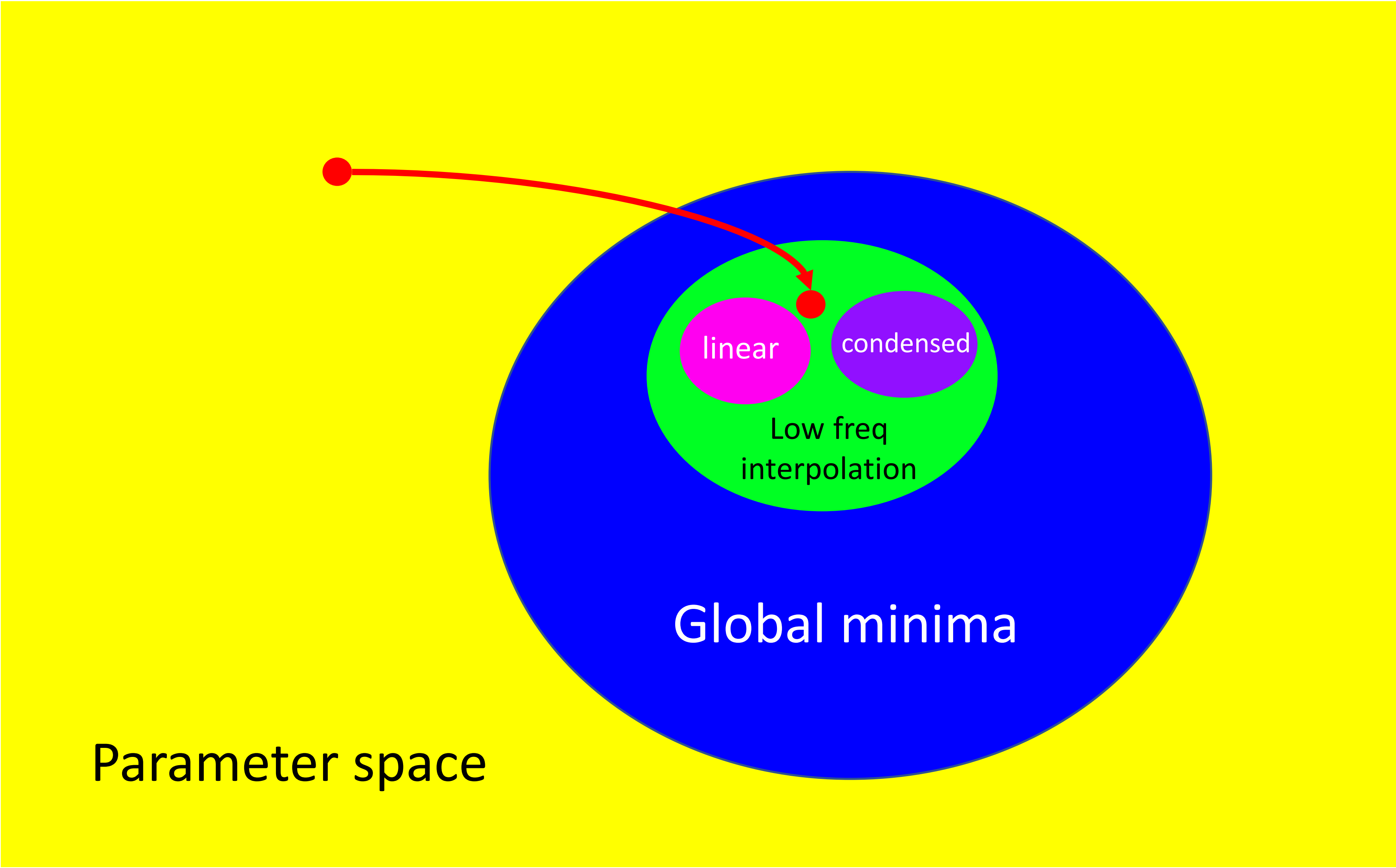
$$f_{\boldsymbol{\theta}}^{\alpha}(\boldsymbol{x}) = \frac{1}{\alpha}\sum_{k=1}^{m} a_k \sigma(\boldsymbol{w}_k^{\intercal}\boldsymbol{x})$$

# Impact of initialization on generalization

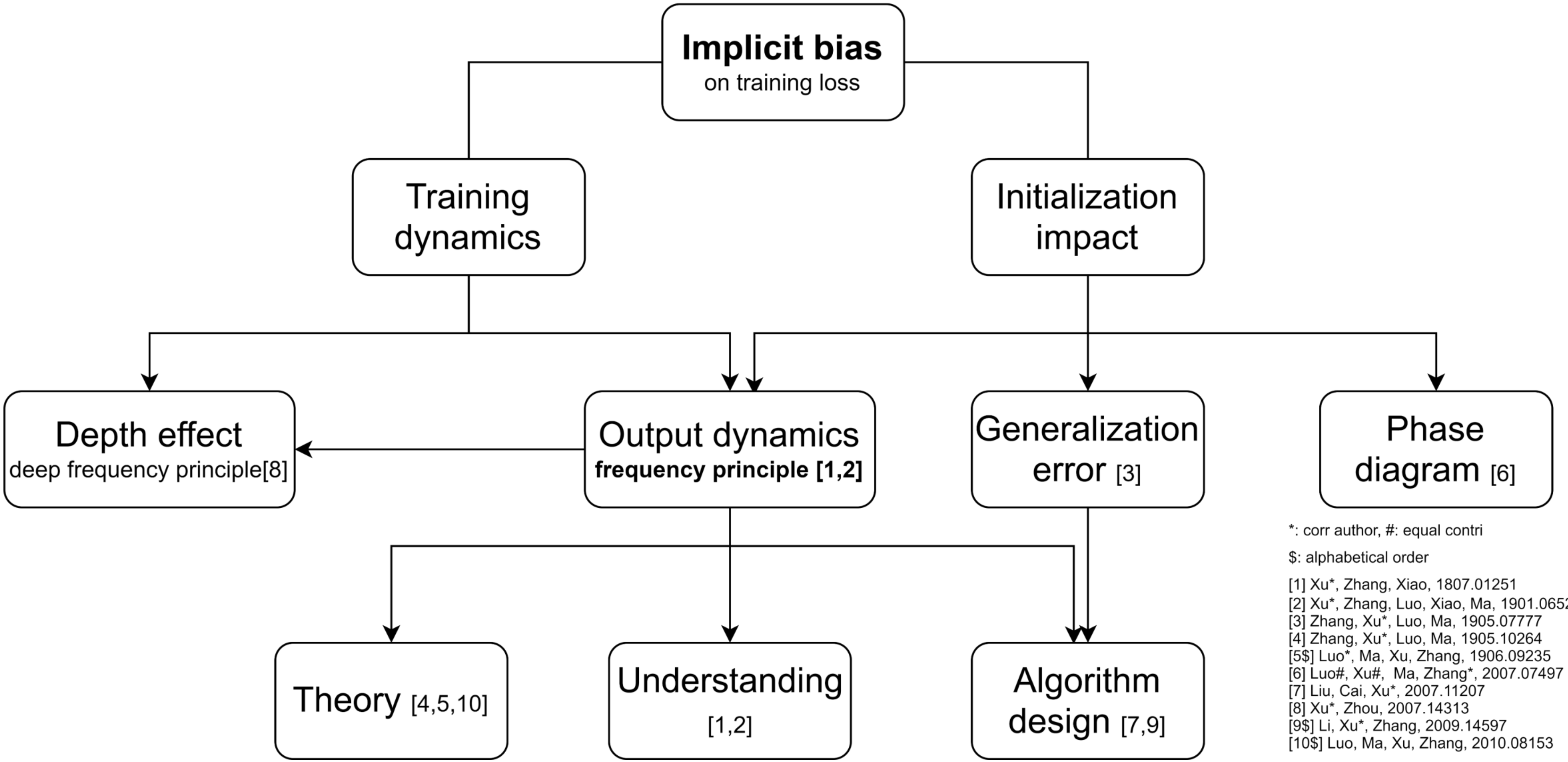


# Revisit picture of deep learning (regression)

loss
initialization
dynamics



# A research picture on studying deep neural networks

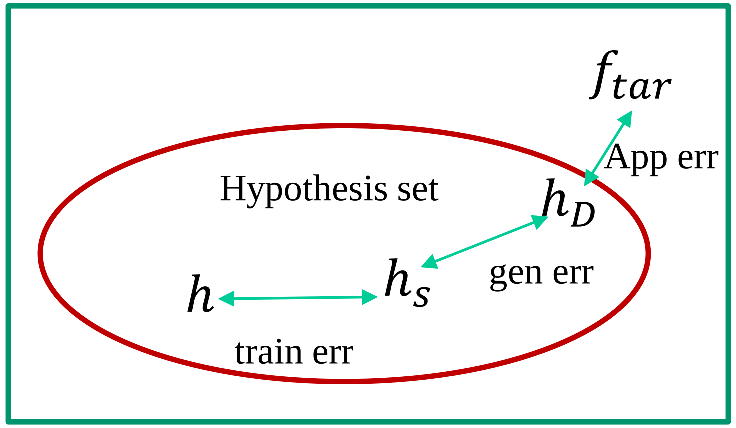



*: corr author, #: equal contri

$: alphabetical order

[1] Xu*, Zhang, Xiao, 1807.01251
[2] Xu*, Zhang, Luo, Xiao, Ma, 1901.06523
[3] Zhang, Xu*, Luo, Ma, 1905.07777
[4] Zhang, Xu*, Luo, Ma, 1905.10264
[5$] Luo*, Ma, Xu, Zhang, 1906.09235
[6] Luo#, Xu#, Ma, Zhang*, 2007.07497
[7] Liu, Cai, Xu*, 2007.11207
[8] Xu*, Zhou, 2007.14313
[9$] Li, Xu*, Zhang, 2009.14597
[10$] Luo, Ma, Xu, Zhang, 2010.08153

# Important problems in this field




- Error analysis
  - Approximation error
  - Generalization error
  - Training error
- Multiple layers: what is the advantage of multiple layers?
- High-dimensional problems (overcome curse of dimensionality)
- Huge number of parameters: why algorithms can find good solutions in large para space?

Credit to the discussion with Prof. Weinan E

# More specific problems

- Phase diagram for multiple layer NN
  - The mechanism of the condensation
  - Implicit bias in different regimes
  - Generalization
- Loss landscape: properties of minima
- Implicit bias of network structures
- The characteristics of real data

# Acknowledge

上海交通大学 SHANGHAI JIAO TONG UNIVERSITY | 自然科学研究院 Institute of Natural Sciences

Homepage: https://ins.sjtu.edu.cn/people/xuzhiqin/


**Joint work with**

Yaoyu Zhang, Tao Luo, Zheng Ma, Lei Zhang, Hanxu Zhou, Xi'an Li (SJTU), .
Yanyang Xiao (Shenzhen IAT), Wei Cai (SMU)
Ziqi Liu (CSRC), Jiwei Zhang (WHU), Jihong Wang(CSRC).

**Acknowledge**

Weinan E (Princeton), David W. McLaughlin (NYU CIMS), Dan Hu (SJTU)

**In memorial of**

David Cai (NYU CIMS, SJTU INS)