



ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Overparameterized models

Đinh Viết Sang
Foundation Models Labs

BKAI

ONE LOVE. ONE FUTURE.

Contents

1. Modern Bias-Variance Trade-off
2. Loss landscape and implicit regularization
3. Scaling Laws
4. NTKs
5. Conclusions



ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

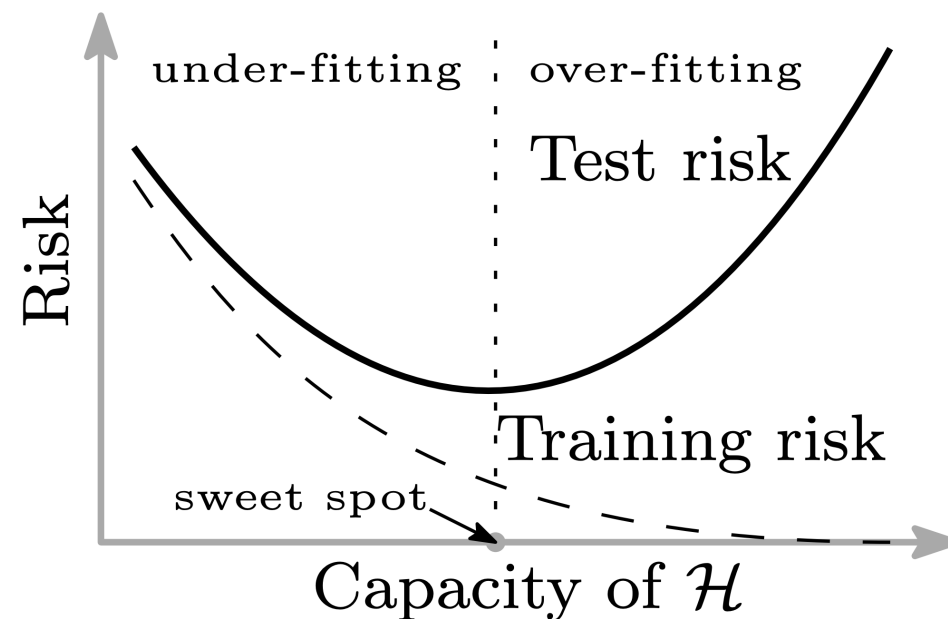
Modern Bias-Variance Trade-off

Classical Bias-variance decomposition

- Learned classifier $f(x)$ depends on dataset D , predict y
- Bias-variance decomposition for MSE:

$$\underbrace{\mathbb{E}_{\mathcal{D}}[(y - f(x))^2]}_{\text{MSE}} = \underbrace{(y - \mathbb{E}_{\mathcal{D}}[f(x)])^2}_{\text{Bias}^2} + \underbrace{\text{Var}_{\mathcal{D}}[f(x)]}_{\text{Variance}}$$

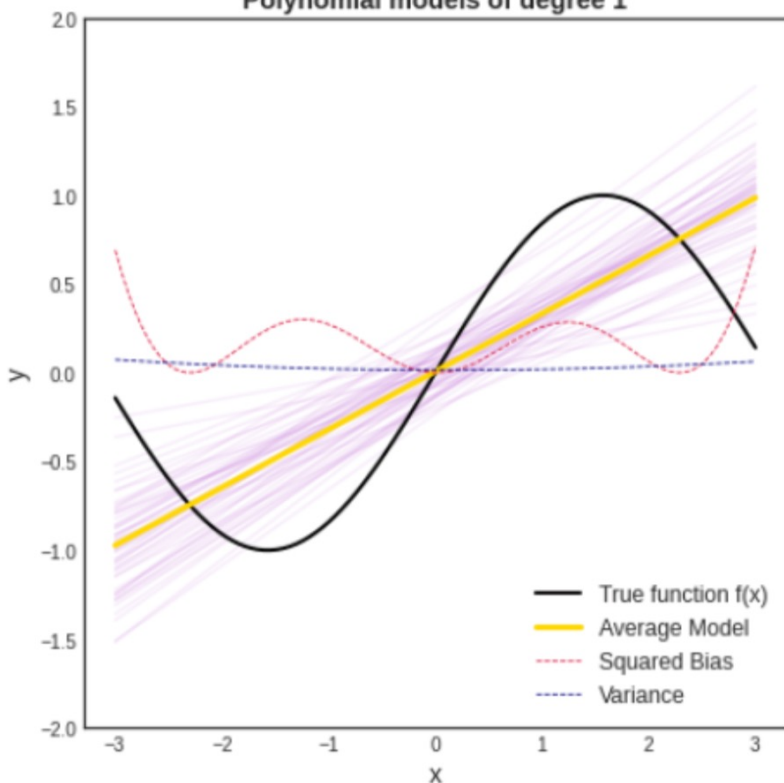
$$\underbrace{\mathcal{R}(f)}_{\text{expected risk}} - \underbrace{\mathcal{R}_{\text{emp}}(f)}_{\text{empirical risk}} < \underbrace{O^* \left(\sqrt{\frac{\text{cap}(\mathcal{H})}{n}} \right)}_{\text{capacity term}}$$



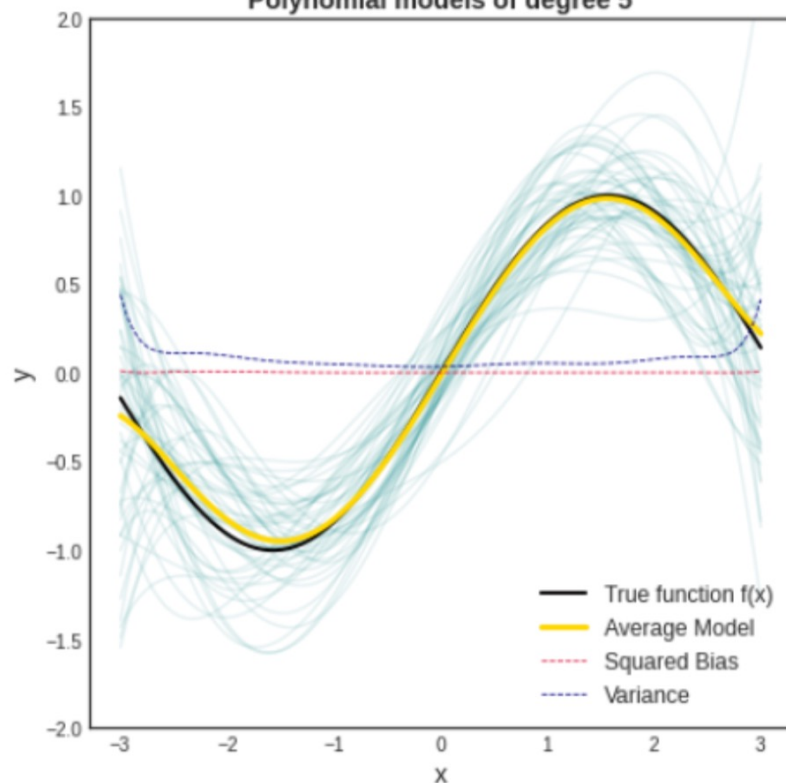
Under vs overfitting

Polynomial models of different degrees fit on random data

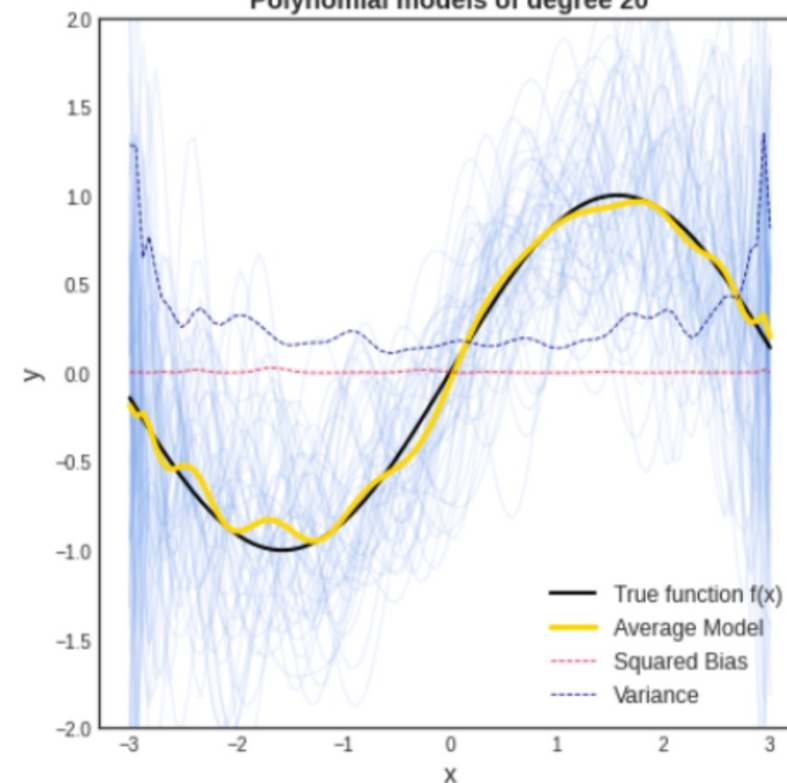
Polynomial models of degree 1



Polynomial models of degree 5

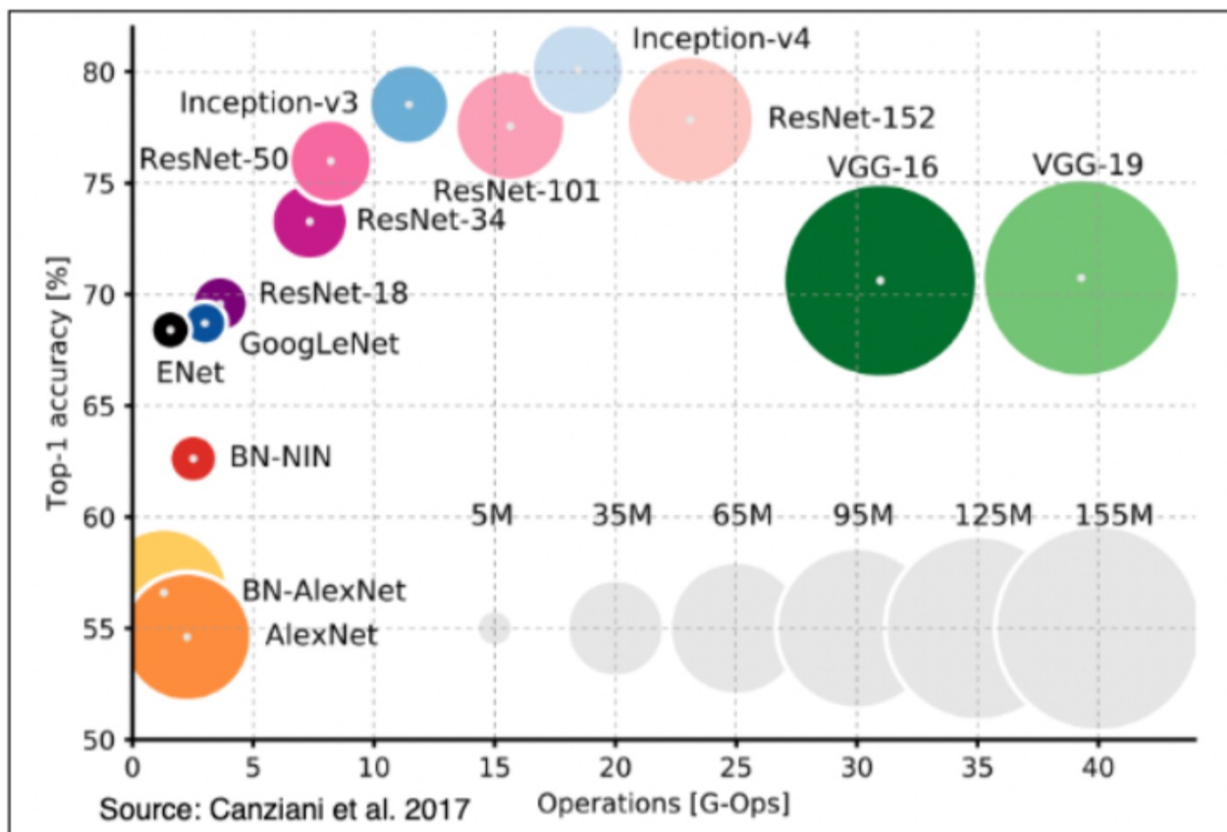


Polynomial models of degree 20



$$\underbrace{\mathbb{E}_{\mathcal{D}}[(y - f(x))^2]}_{\text{MSE}} = \underbrace{(y - \mathbb{E}_{\mathcal{D}}[f(x)])^2}_{\text{Bias}^2} + \underbrace{\text{Var}_{\mathcal{D}}[f(x)]}_{\text{Variance}}$$

Overparameterized models



GPT3, 2020
175 tỷ tham số

Switch Transformer, 2021
1600 tỷ tham số (1.6 trillion)

"The best way to solve the problem from practical standpoint is you **build a very big system** ... basically you want to **hit the zero training error**."



Ruslan Salakhutdinov

UPMC Professor, Machine Learning Department, [CMU](#)
Verified email at cs.cmu.edu - [Homepage](#)

[Machine Learning](#) [Artificial Intelligence](#) [Deep Learning](#)

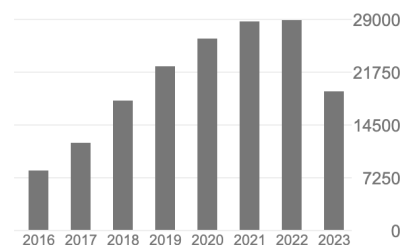


TITLE	CITED BY	YEAR
Dropout: a simple way to prevent neural networks from overfitting N Srivastava, G Hinton, A Krizhevsky, I Sutskever, R Salakhutdinov The journal of machine learning research 15 (1), 1929-1958	45773	2014
Reducing the dimensionality of data with neural networks GE Hinton, RR Salakhutdinov science 313 (5786), 504-507	21098	2006
Show, attend and tell: Neural image caption generation with visual attention K Xu, J Ba, R Kiros, K Cho, A Courville, R Salakhutdinov, RS Zemel, ... International Conference on Machine Learning (ICML) 2 (3), 5	11159	2015
Improving neural networks by preventing co-adaptation of feature detectors GE Hinton, N Srivastava, A Krizhevsky, I Sutskever, RR Salakhutdinov arXiv preprint arXiv:1207.0580	10471	2012
Xlnet: Generalized autoregressive pretraining for language understanding Z Yang, Z Dai, Y Yang, J Carbonell, RR Salakhutdinov, QV Le Advances in neural information processing systems 32	7674	2019
Probabilistic matrix factorization R Salakhutdinov, A Mnih Neural Information Processing Systems 21	5129 *	2007
Siamese neural networks for one-shot image recognition G Koch, R Zemel, R Salakhutdinov ICML deep learning workshop 2 (1)	4451	2015

Cited by

[VIEW ALL](#)

	All	Since 2018
Citations	180451	143682
h-index	112	105
i10-index	260	256



Public access

[VIEW ALL](#)

1 article	59 articles
not available	available

Based on funding mandates

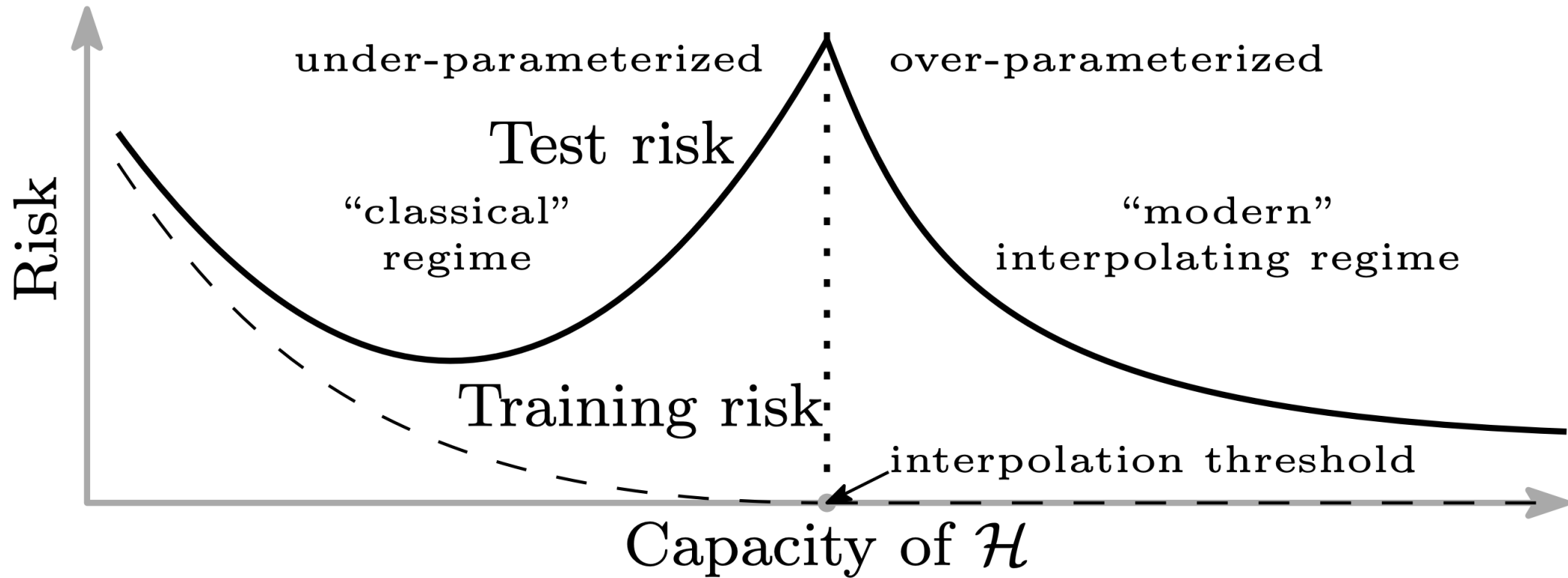
Co-authors

[VIEW ALL](#)



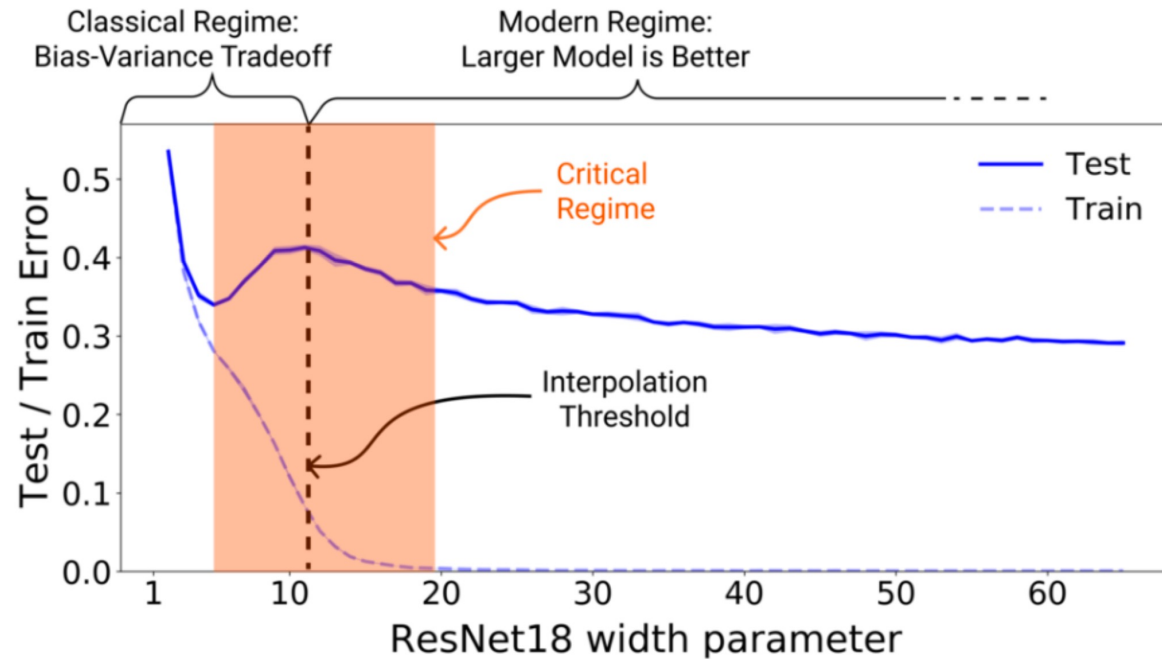
Double descent

- Contradict modern practice: **bigger models generalize better**, rather than overfitting

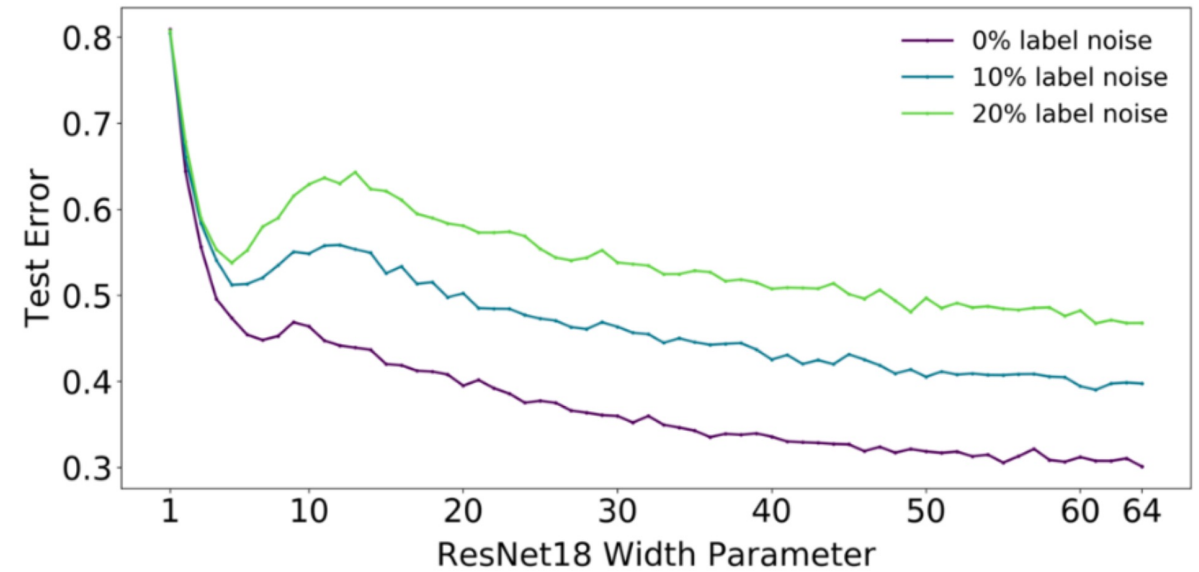


Belkin et al., 2018

Double descent



CIFAR 10



CIFAR 100

Double descent

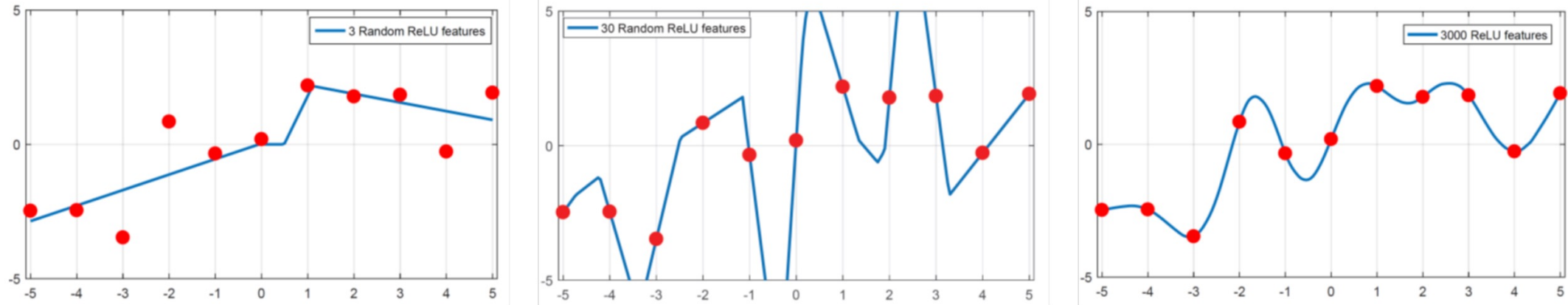
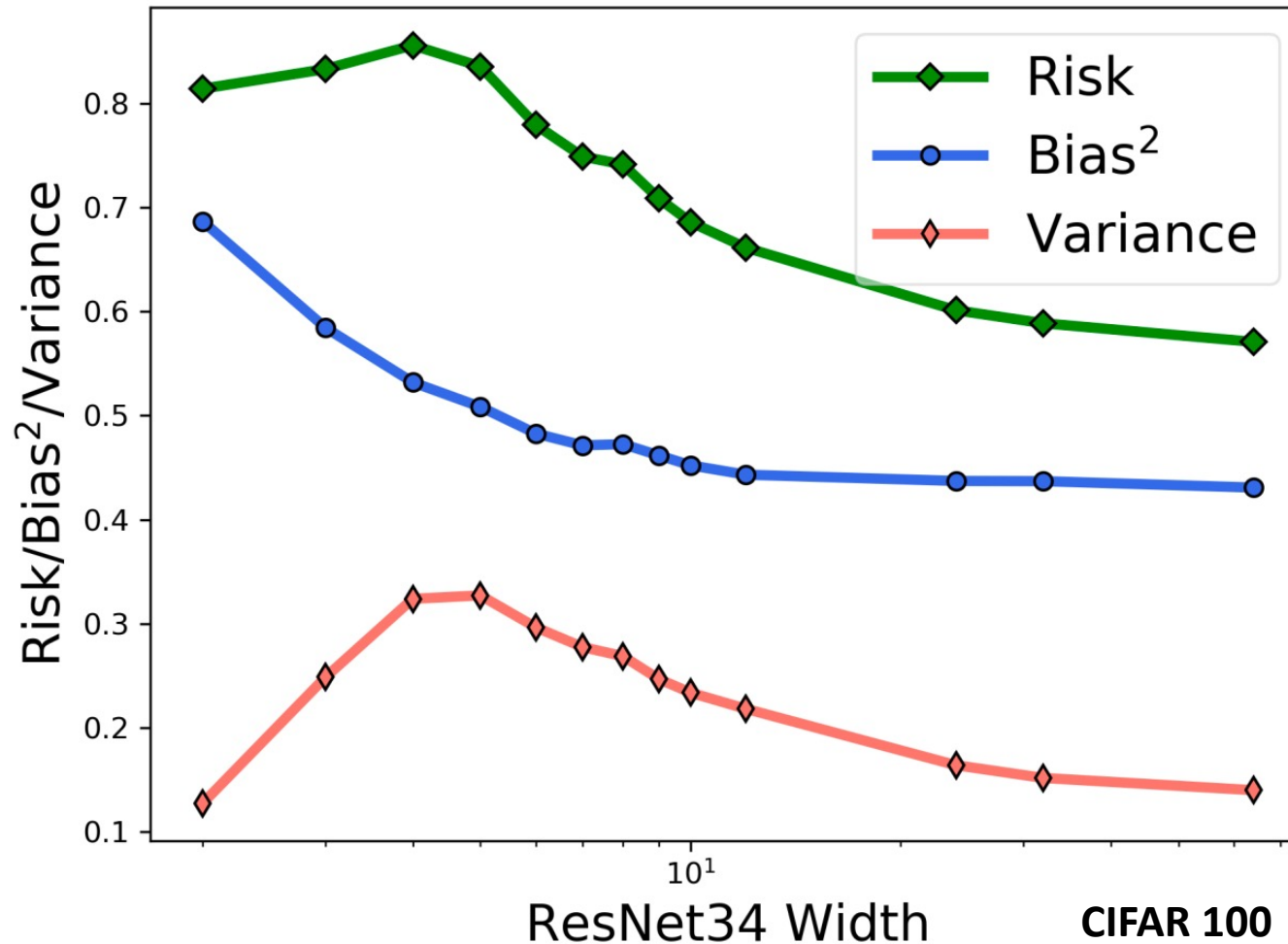


Figure 6: Illustration of double descent for Random ReLU networks in one dimension. Left: Classical under-parameterized regime (3 parameters). Middle: Standard over-fitting, slightly above the interpolation threshold (30 parameters). Right: “Modern” heavily over-parameterized regime (3000 parameters).

Double descent

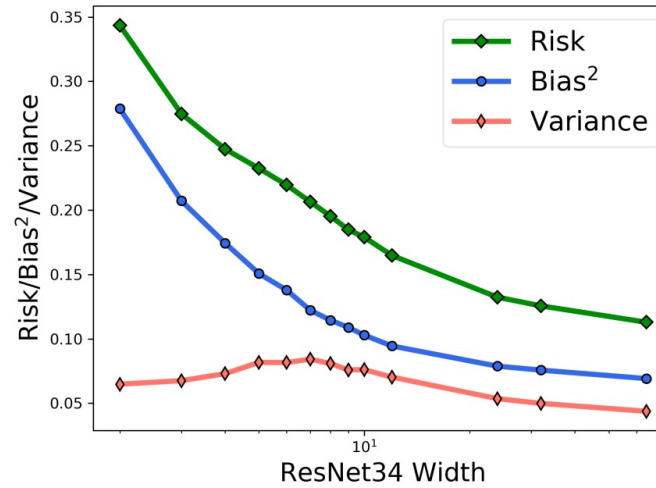
- Phenomenon: **monotonic** bias + **unimodal** variance



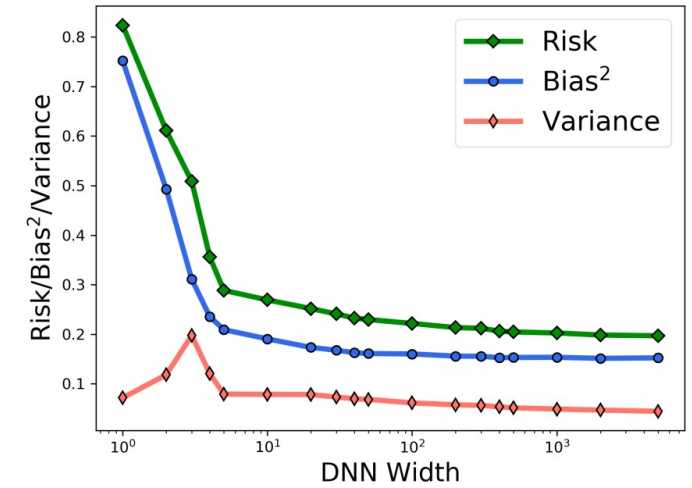
Double descent

- Phenomenon:
monotonic bias +
unimodal variance

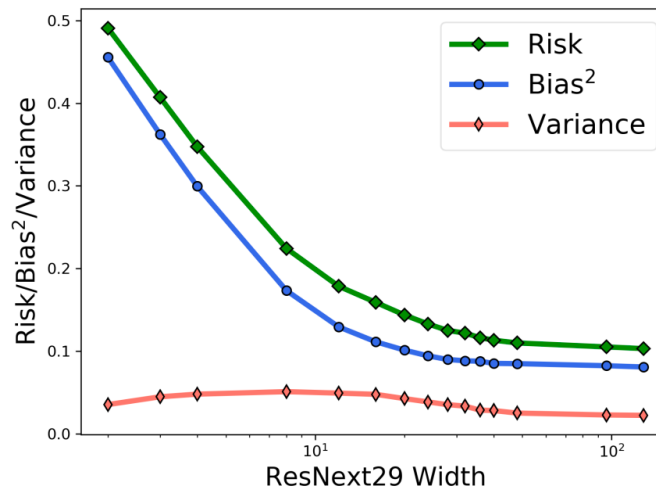
CIFAR-10



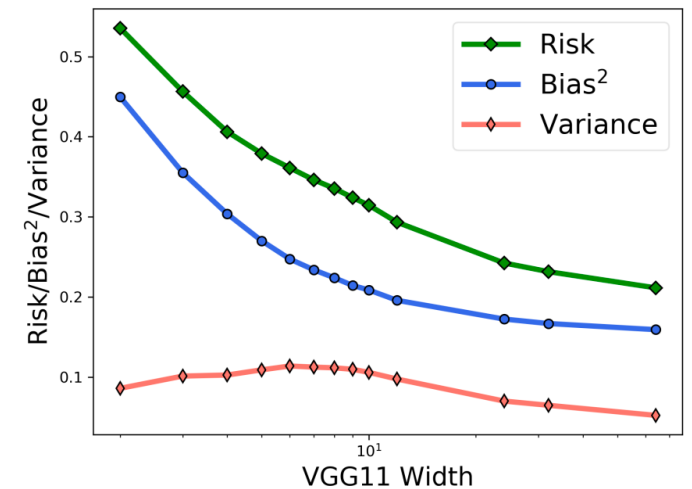
Fashion-MNIST



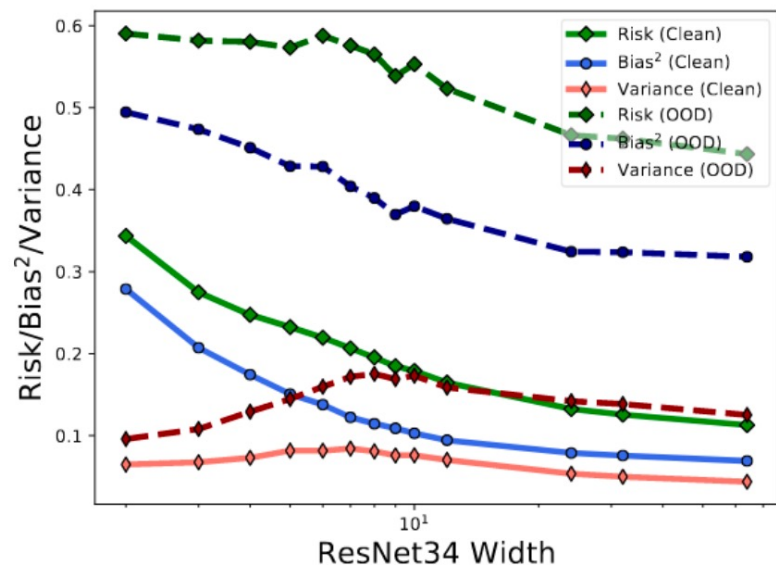
ResNext29



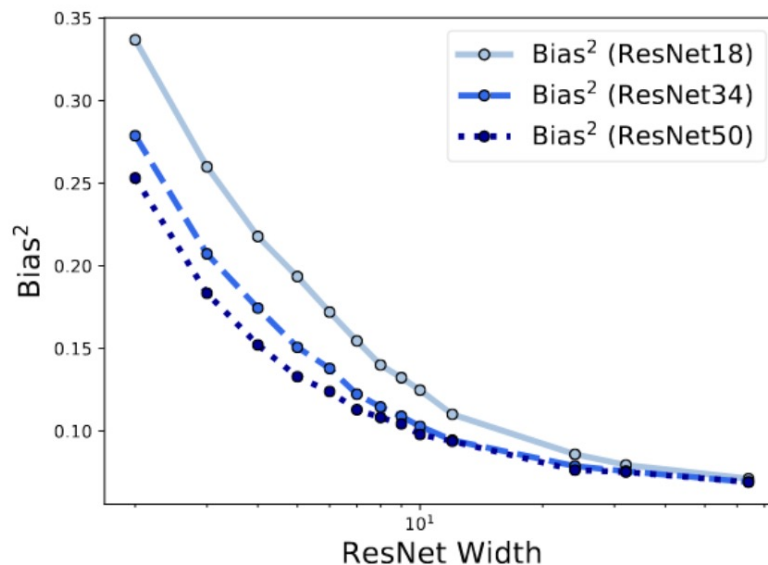
VGG11



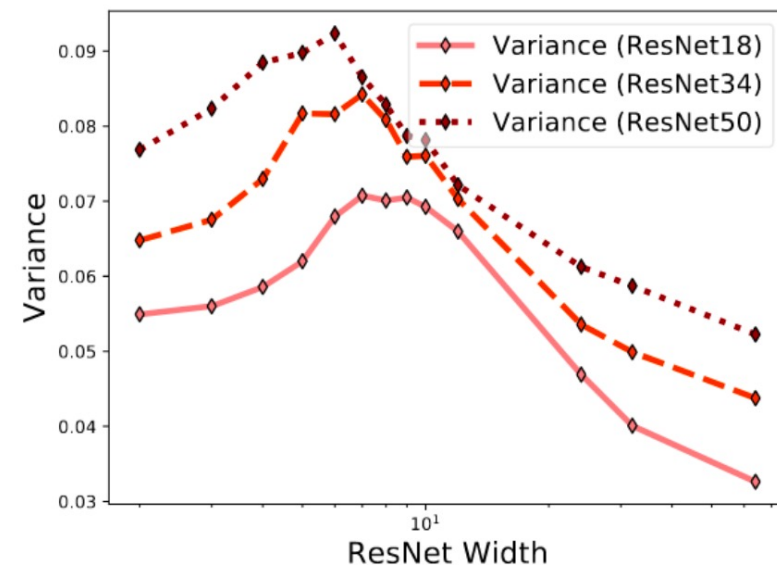
Effect of depth on bias-variance



(a) OOD Example

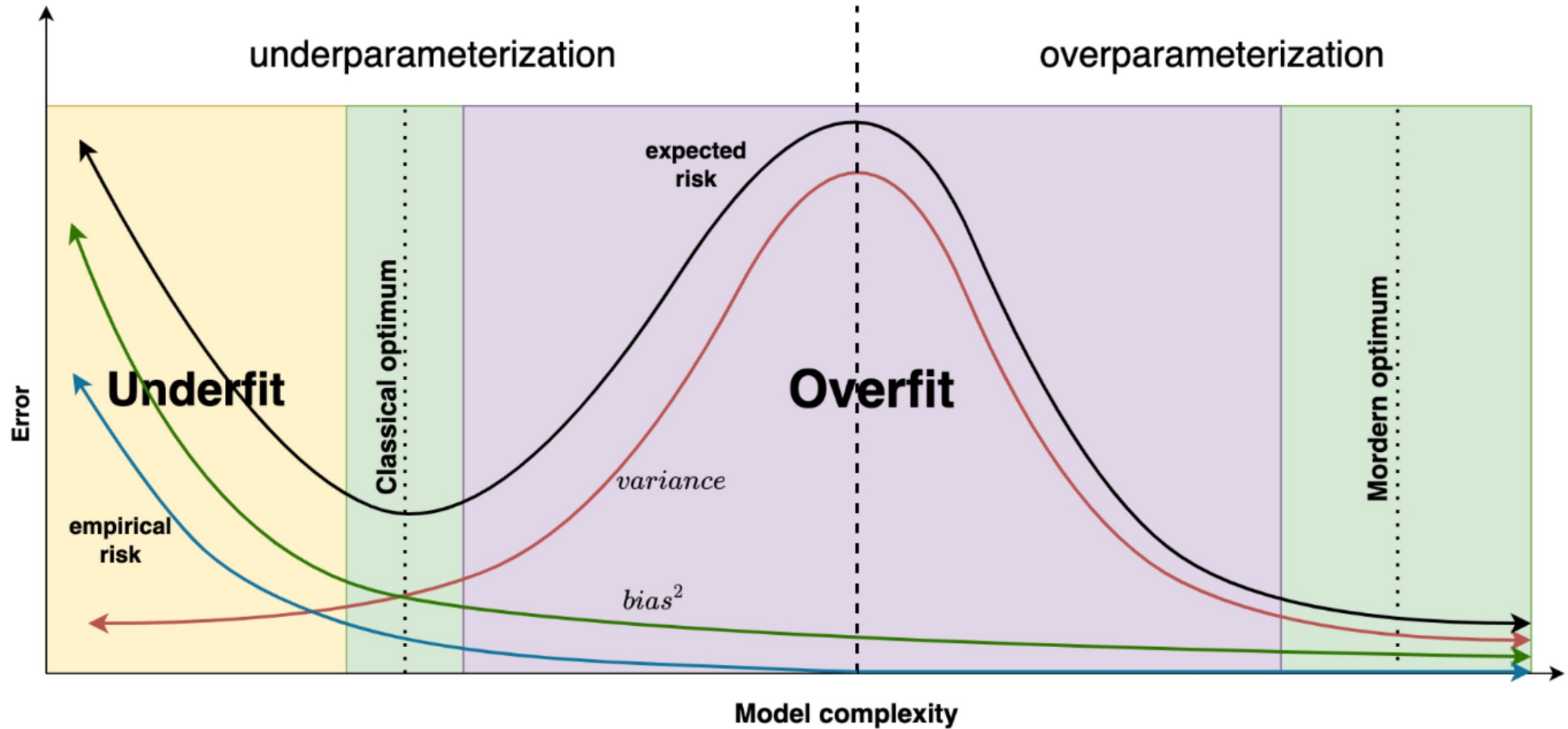


(b) Bias of model with different depth



(c) Variance of model with different depth

Modern Bias-Variance Trade-off?



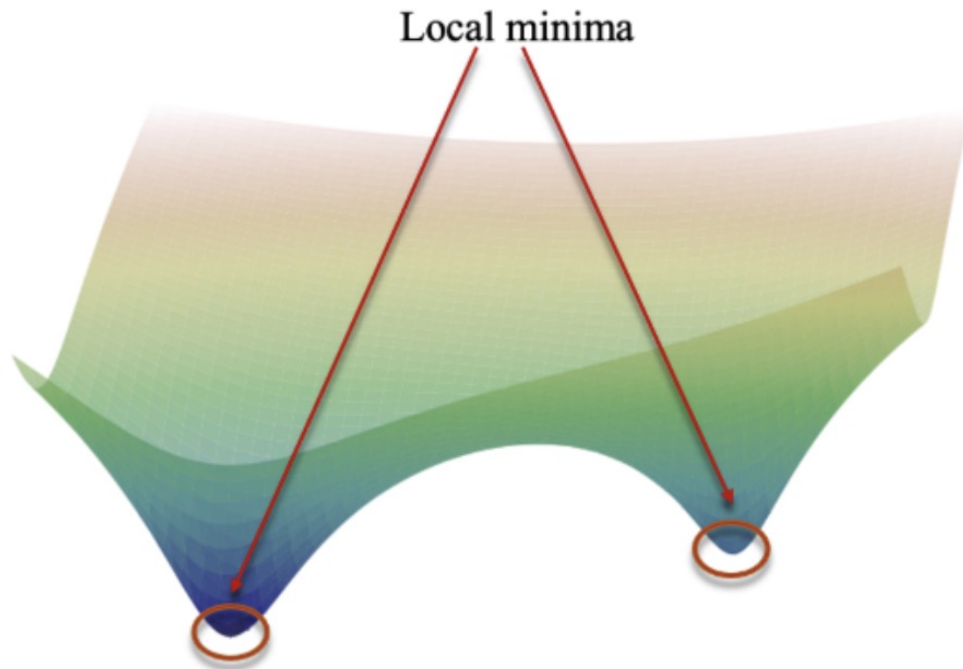


ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

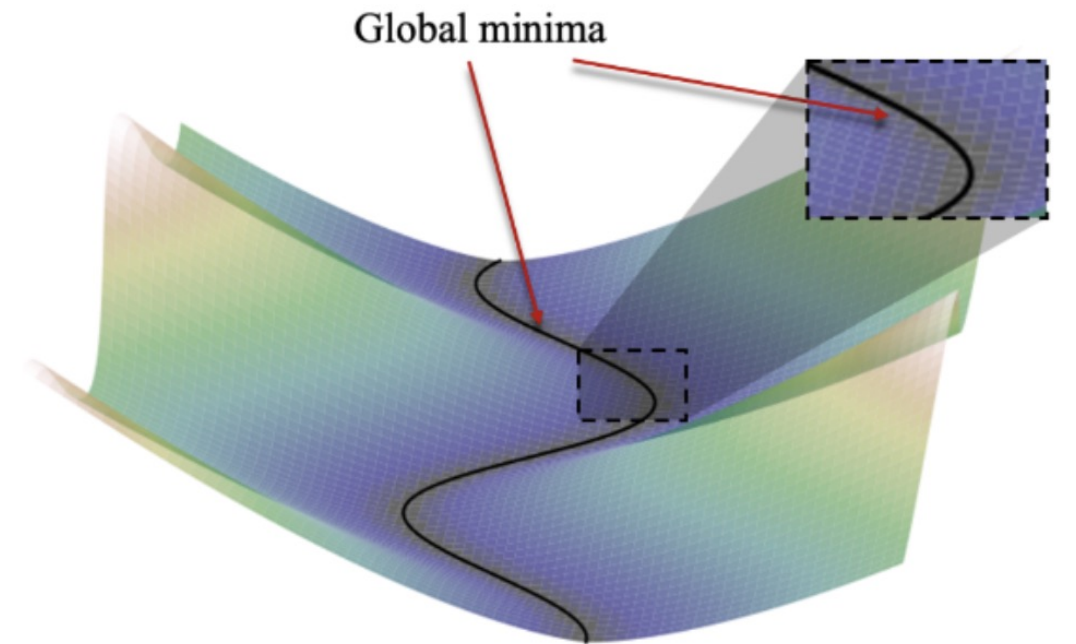
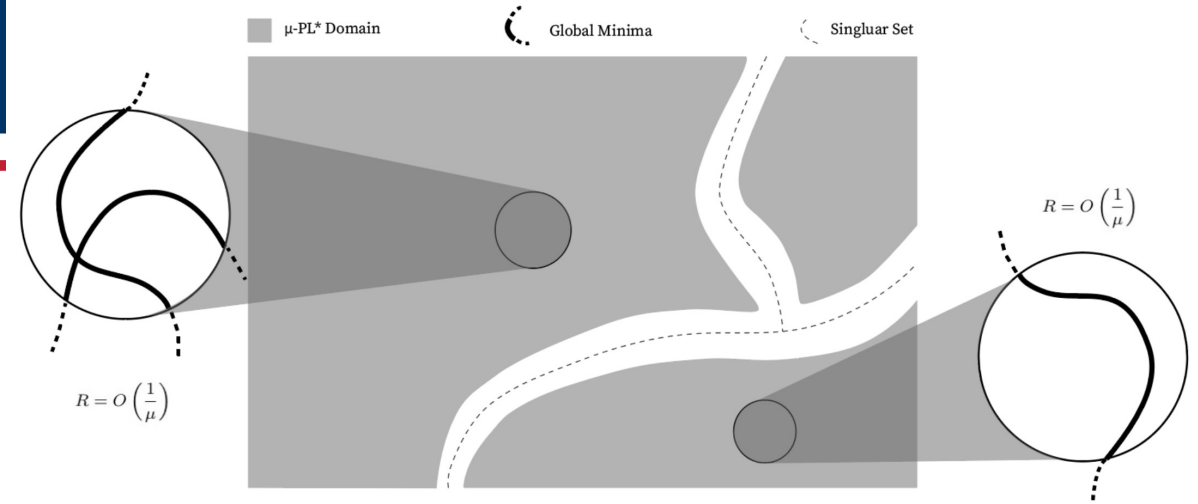
Loss landscape and implicit regularization

Loss landscape

- All minima are global



(a) Loss landscape of under-parameterized models



(b) Loss landscape of over-parameterized models

θ HIGH DIMENSIONAL SPACE 1 BILLION DIMENSIONS

LOSS LANDSCAPE
VISUALIZATION

1

DIMENSIONALITY
REDUCTION
TECHNIQUE

2

MINIMA IN
HIGH DIMENSIONAL
SPACE

SHORTCUTS IN
HIGH DIM. SPACE

RANGE IN
WEIGHT SPACE

θ

$-1, -1$
 α, β

$\alpha, \beta, f(\alpha, \beta)$

SMOOTH
MORPHOLOGY

LOSS: 2.54
GRADIENT
DESCENT

LOSS: 1.81

INITIALIZATION
SGD

LOSS: 0.02

MINIMA

ROUGH
MORPHOLOGY

MINIMA

$\alpha, \beta, f(\alpha, \beta)$

$-1, 0$
 α, β

η

MODE CONNECTIVITY
arXiv: 1802.10026

MINIMA

MINIMA

LOSS: 0.01

LOSS: 0.02

$\alpha, \beta, f(\alpha, \beta)$

$\alpha, \beta, f(\alpha, \beta)$

LOSS: 0.02

MINIMA

3

θ 2D SLICE

RAND ORTHO DIRECTIONS
arXiv:1712.09913

LOW DIMENSIONALITY INTERPRETATION

4

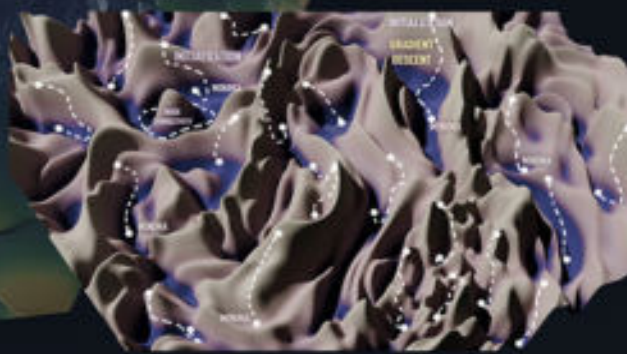
3 DIMENSIONS

$$f(\alpha, \beta) = L(\theta + \alpha\delta + \beta\eta)$$

MINIMA

5

THE BLESSING OF DIMENSIONALITY



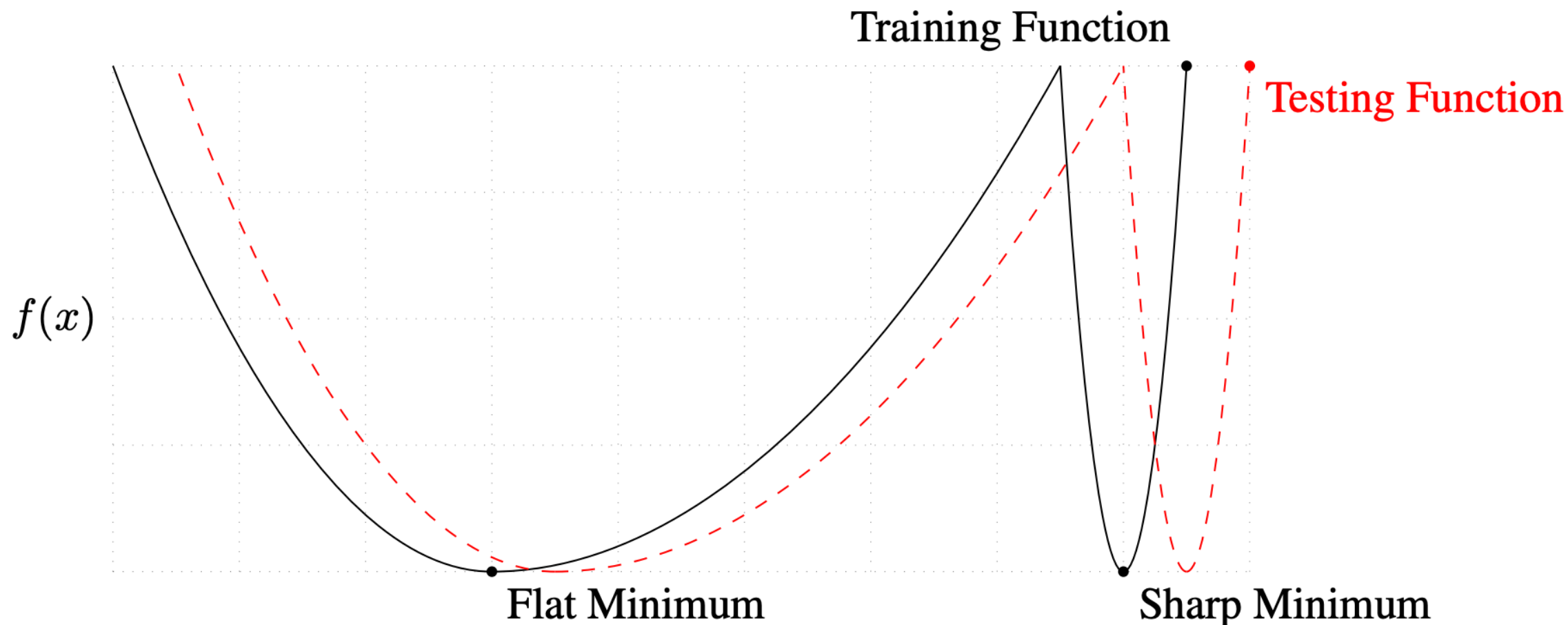
FINDING A MINIMA BECOMES A "LOCAL" CHALLENGE

Under vs overparameterization

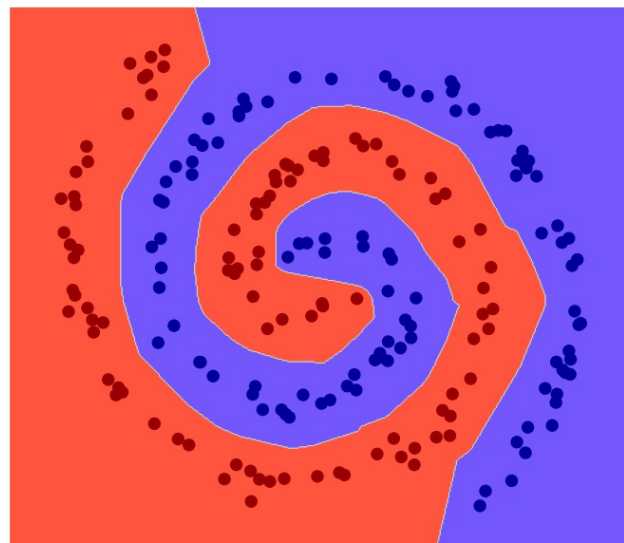
Underparameterization	Overparameterization
Non-zero train error	Interpolation mode
Isolated minima	Manifolds of global minima
Locally convex	Non-convex
Exists optimal complexity (sweet spot)	More complex is better

Sharp vs flat minima

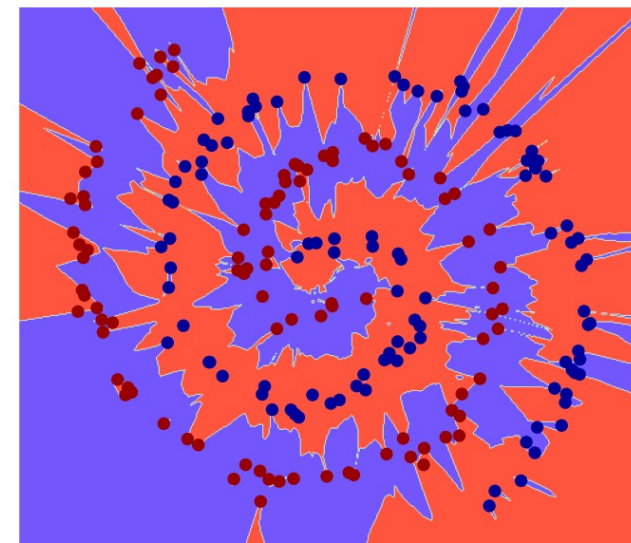
- A common (yet unproven) belief that wide minima have better generalization



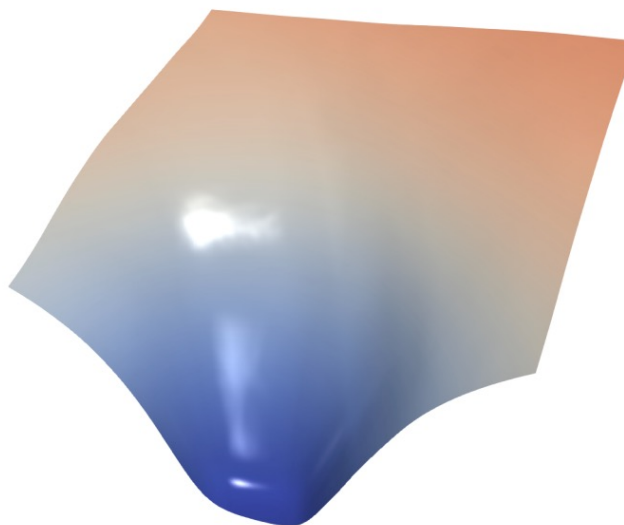
Sharp vs flat minima



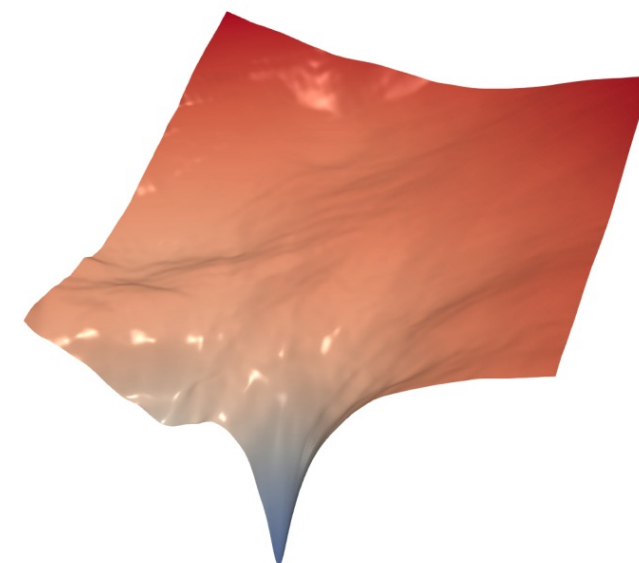
(a) 100% train, 100% test



(b) 100% train, 7% test



(c) Minimizer of network in (a) above



(d) Minimizer of network in (b) above

Implicit regularization

- SGD does not "see" bad minima

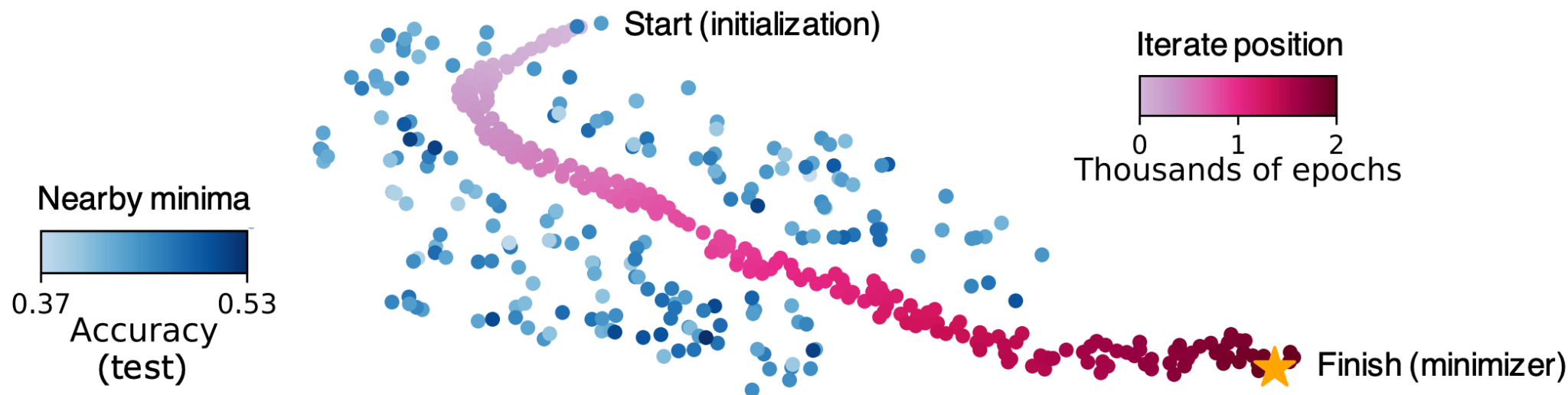
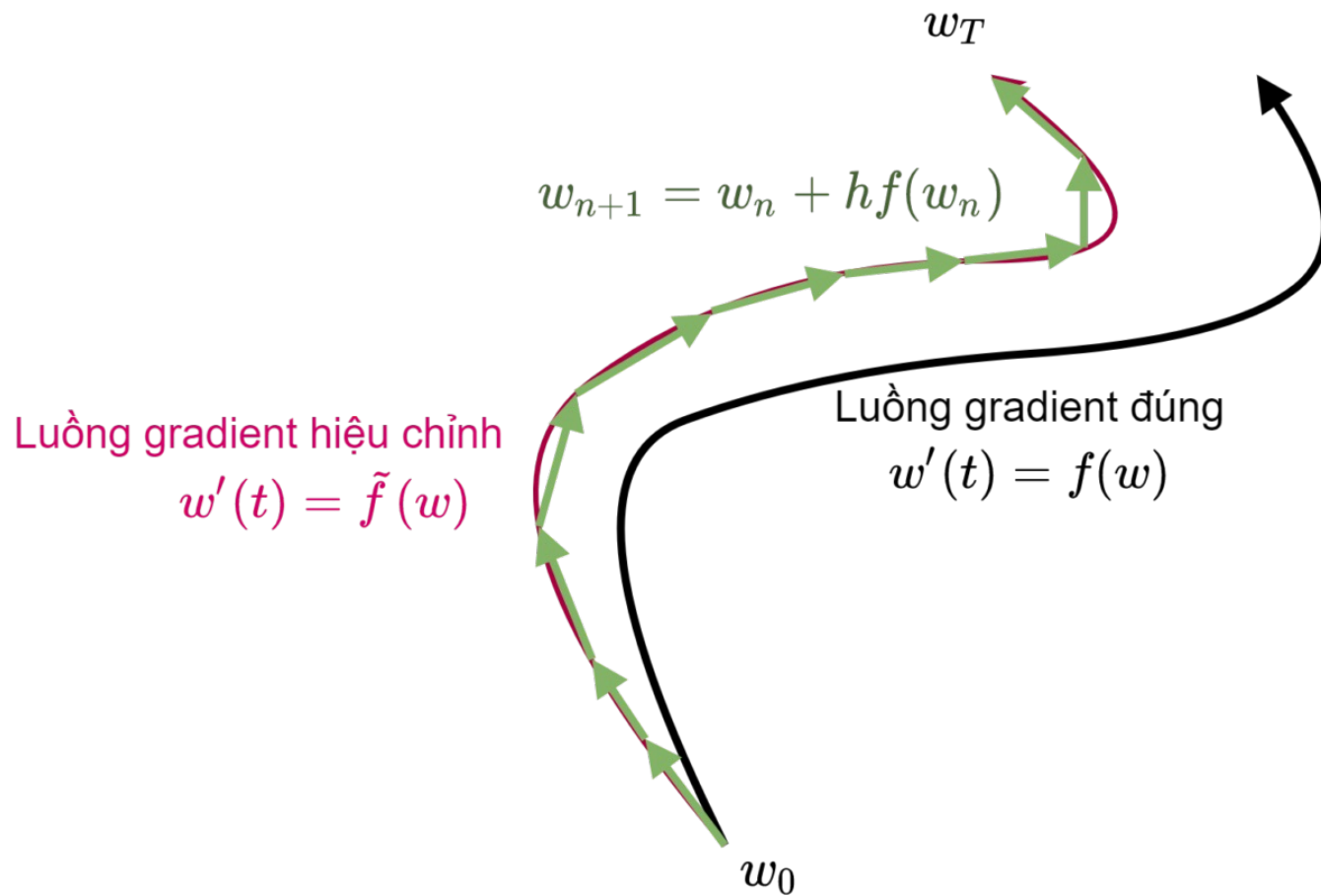


Figure 1: A minefield of bad minima: we train a neural net classifier and plot the iterates of SGD after each tenth epoch (red dots). We also plot locations of nearby “bad” minima with poor generalization (blue dots). We visualize these using t-SNE embedding. All blue dots achieve near perfect train accuracy, but with test accuracy below 53% (random chance is 50%). The final iterate of SGD (yellow star) also achieves perfect train accuracy, but with 98.5% test accuracy. Miraculously, SGD avoids the bad minima, and lands at a minimum with excellent generalization. See Section 3 for experimental details.

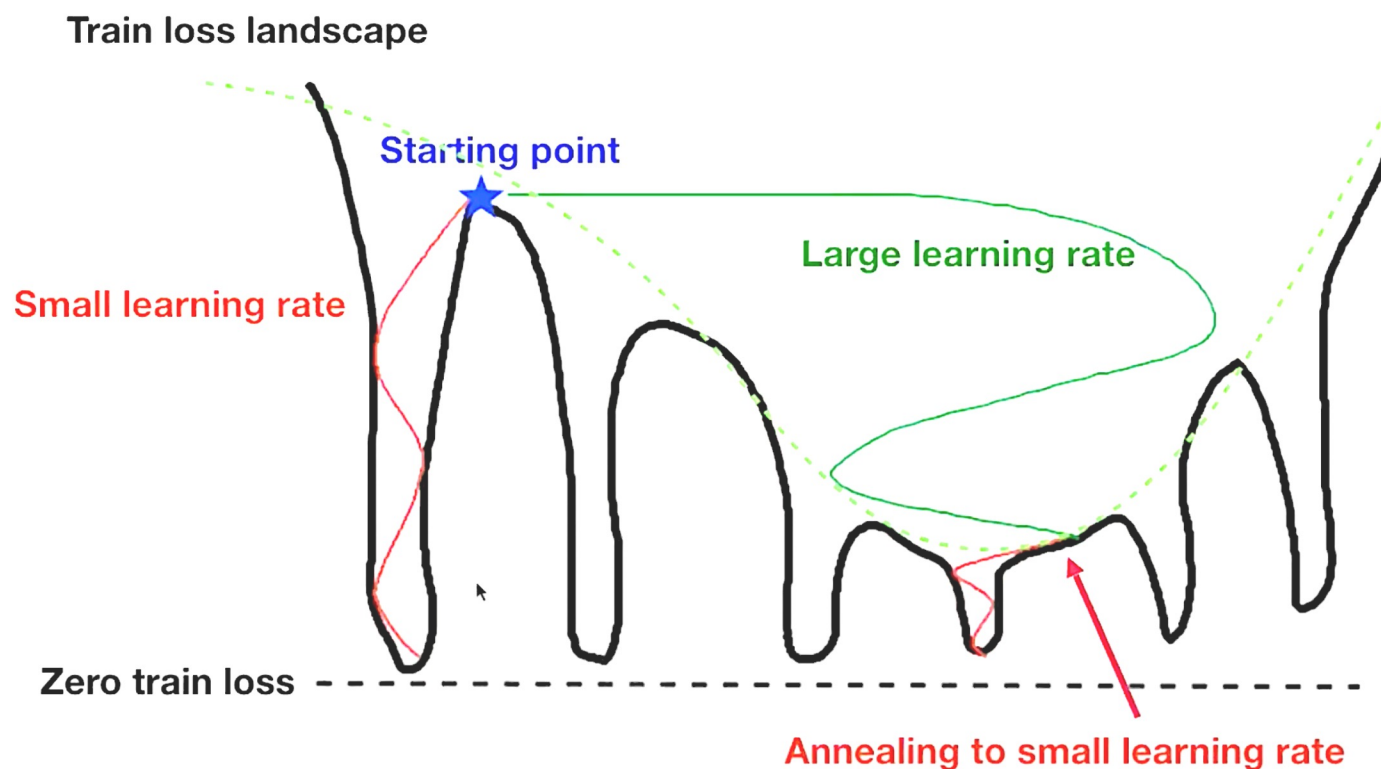
Implicit regularization



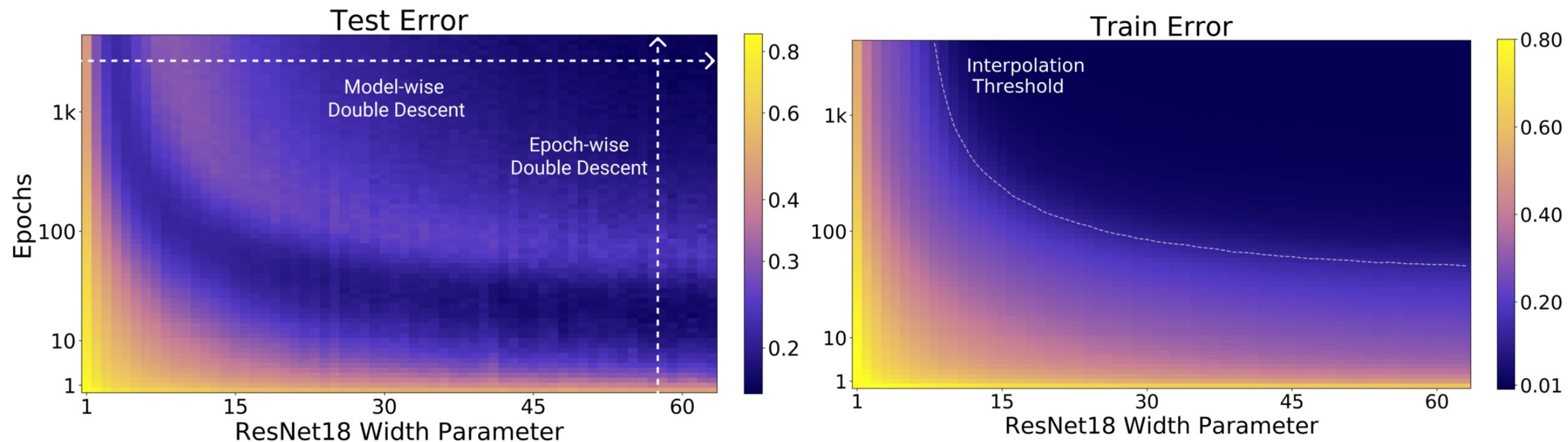
$$\mathbb{E}[\tilde{L}_{SGD}(w)] = L(w) + \frac{h}{4} \|\nabla L(w)\|^2 + \frac{N-B}{N(N-1)} \frac{h}{4B} \sum_{k=1}^m \|\nabla L_i(w) - \nabla L(w)\|^2$$

Implicit regularization

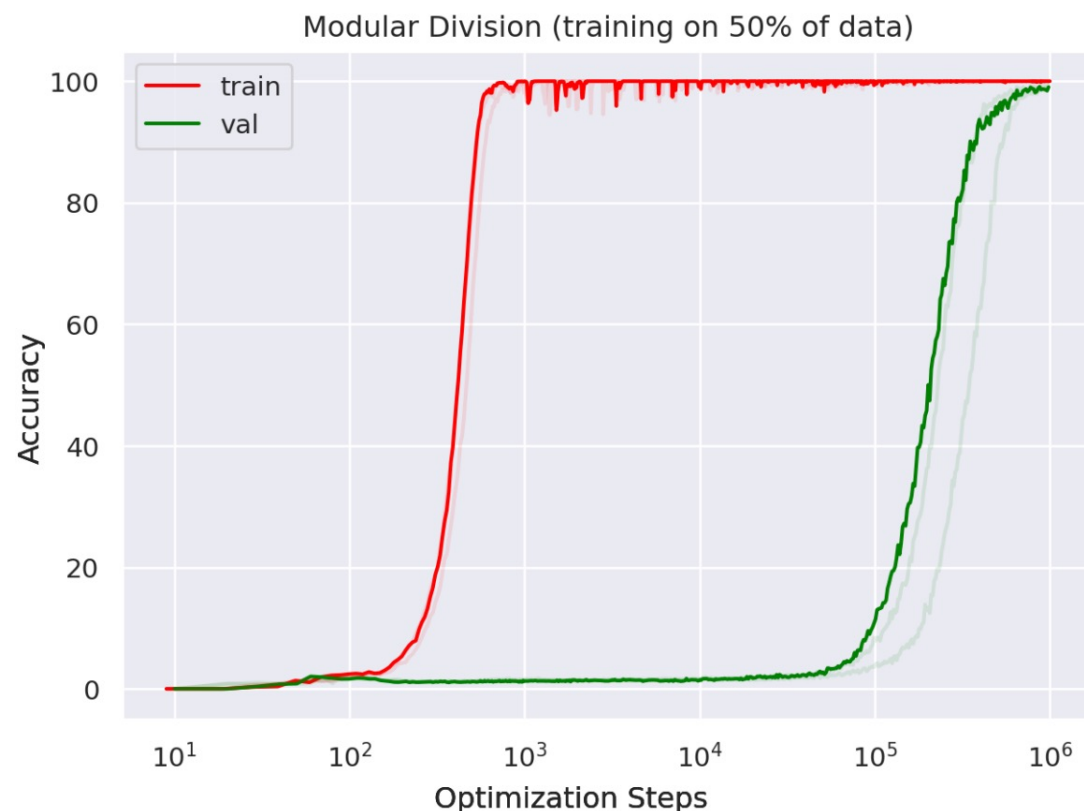
- **Hypothesis:** The noise in SGD (small batch size, large LR, implicit regularization) **prevent us from seeing small details** of the loss → find **flatter global minima** which tend to give more Lipschitz models and better generalization.



Epoch-wise Double descent

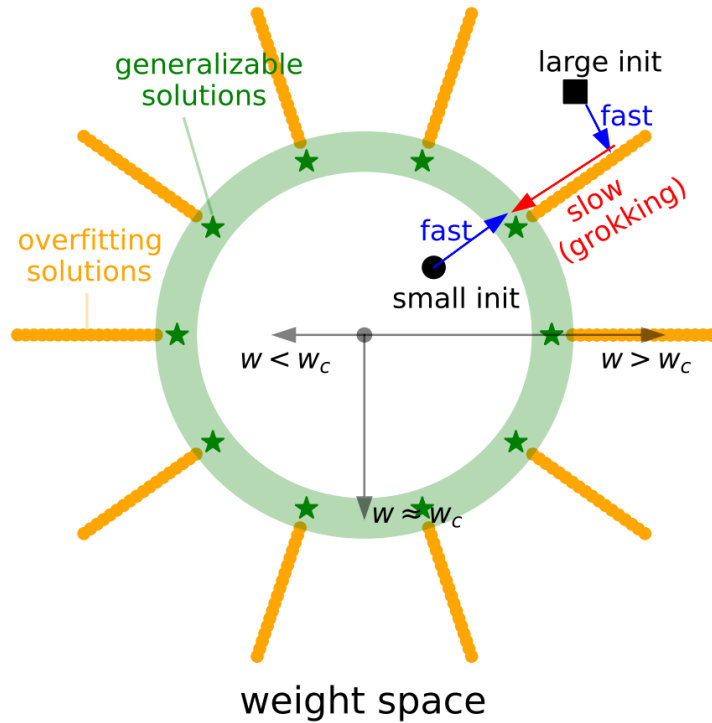


- Grokking: when a neural network suddenly learns a pattern in the dataset and jumps from random chance generalization to perfect generalization very suddenly.

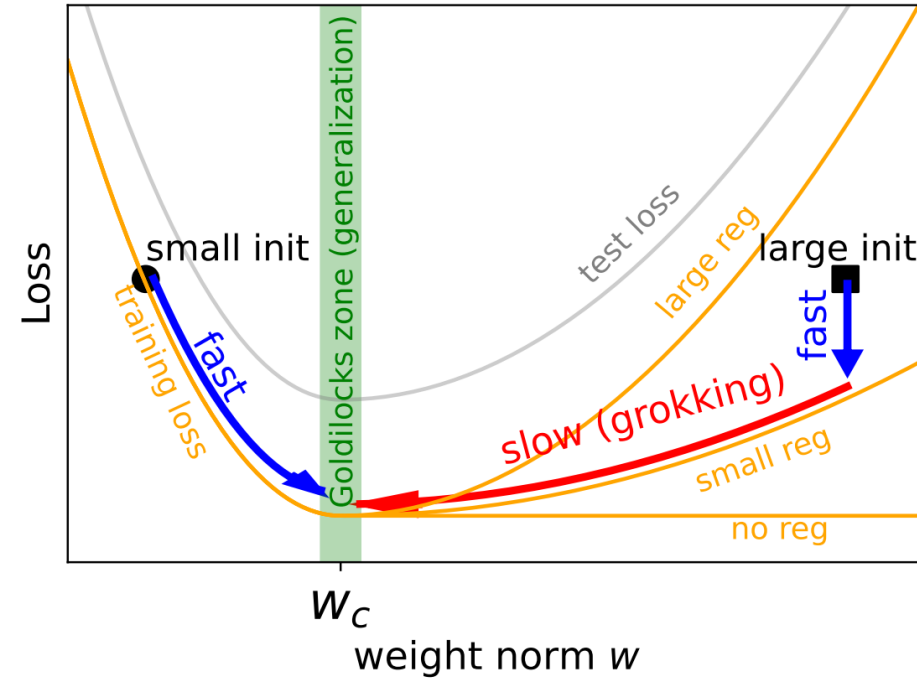


Epoch-wise Double descent

- **Epoch-wise DD = generalization + memorization + consolidation**
 1. At first, model learns simple useful features and generalizes on normal examples - test error decreases.
 2. Then it starts memorizing noise examples - test error increases.
 3. Finally, network consolidates: removes redundancy, **slowly drift to a wider minima** (flat regions), improves generalization - test error decreases again.



(a)



(b)

Figure 1: (a) w : L_2 norm of model weights. Generalizing solutions (green stars) are concentrated around a sphere in the weight space where $w \approx w_c$ (green). Overfitting solutions (orange) populate the $w \gtrsim w_c$ region. (b) The training loss (orange) and test loss (gray) have the shape of L and U, respectively. Their mismatch in the $w > w_c$ region leads to fast-slow dynamics, resulting in grokking.

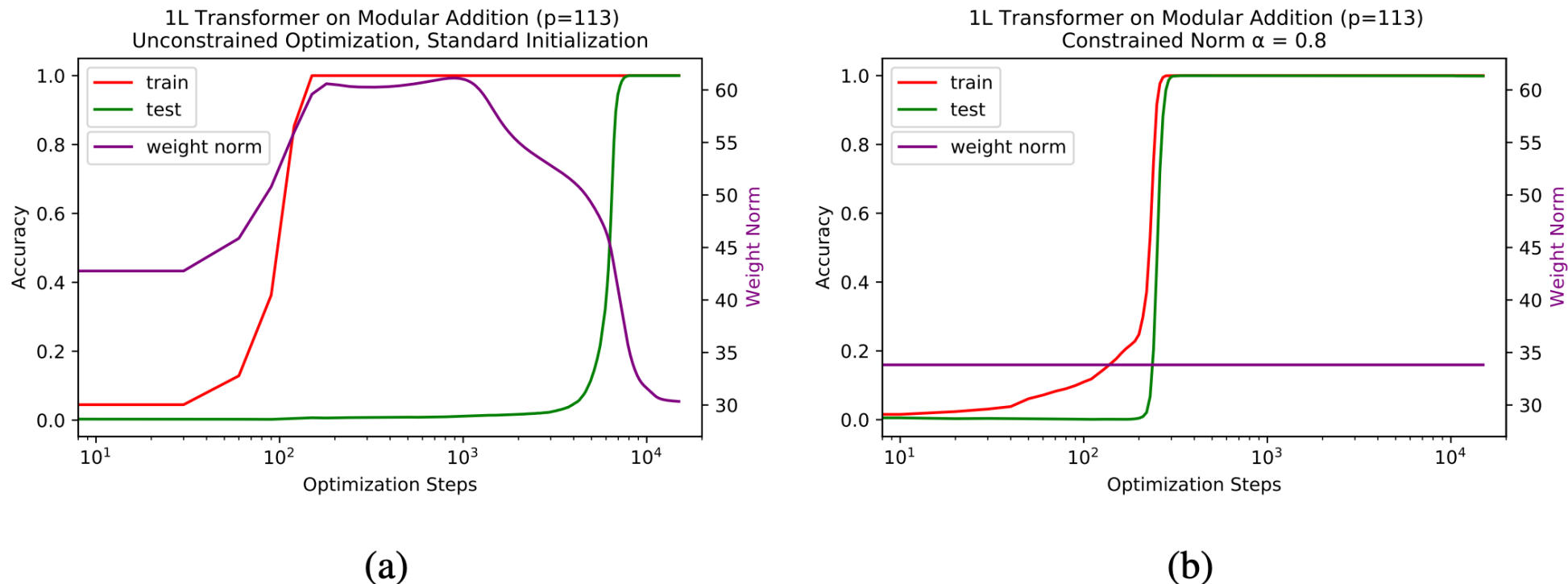


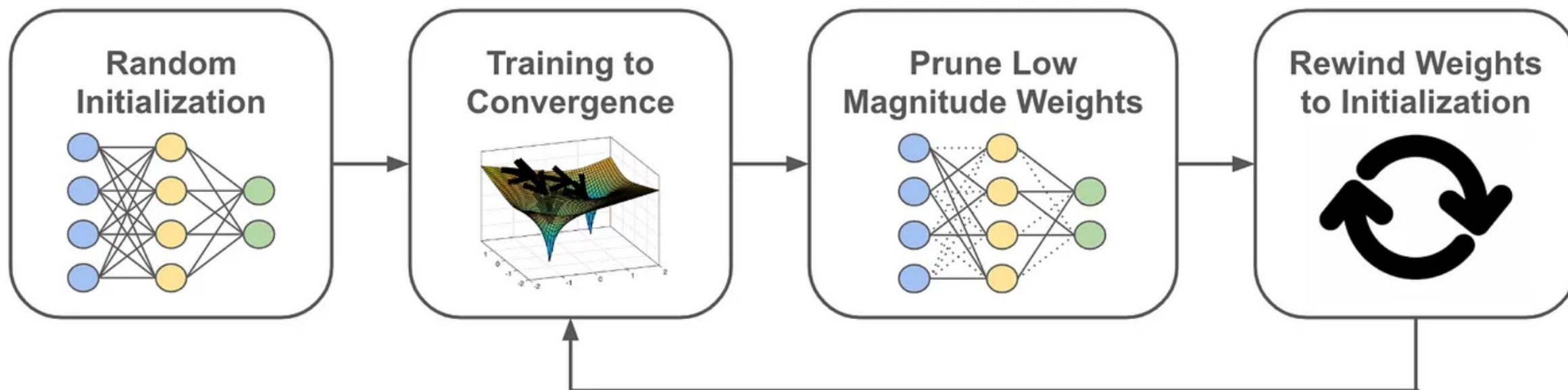
Figure 7: Training 1L transformer on modular addition ($p = 113$). (a) Weight norm, train accuracy, and test accuracy over time, initialized and trained normally. Weight norm first increases, and is highest during the period of overfitting, but then drops to become lower than initial weight norm when the model generalizes. (b) Constrained optimization at constant weight norm ($\alpha = 0.8$) largely eliminates grokking, with test and train accuracy improving concurrently.

Regularization

- In overparameterized models, regularization plays **another role**
- Among continuum solutions with zero train error, it select **the one with minimal norm**
- Type of regularization:
 - weight decay
 - implicit regularization
 - Normalization that induces scale-invariance and encourages convergence to wider minima

Lottery Ticket Hypothesis

- A randomly-initialized, dense neural network contains a subnetwork that is initialized such that—when trained in isolation—it can match the test accuracy of the original network after training for at most the same number of iterations



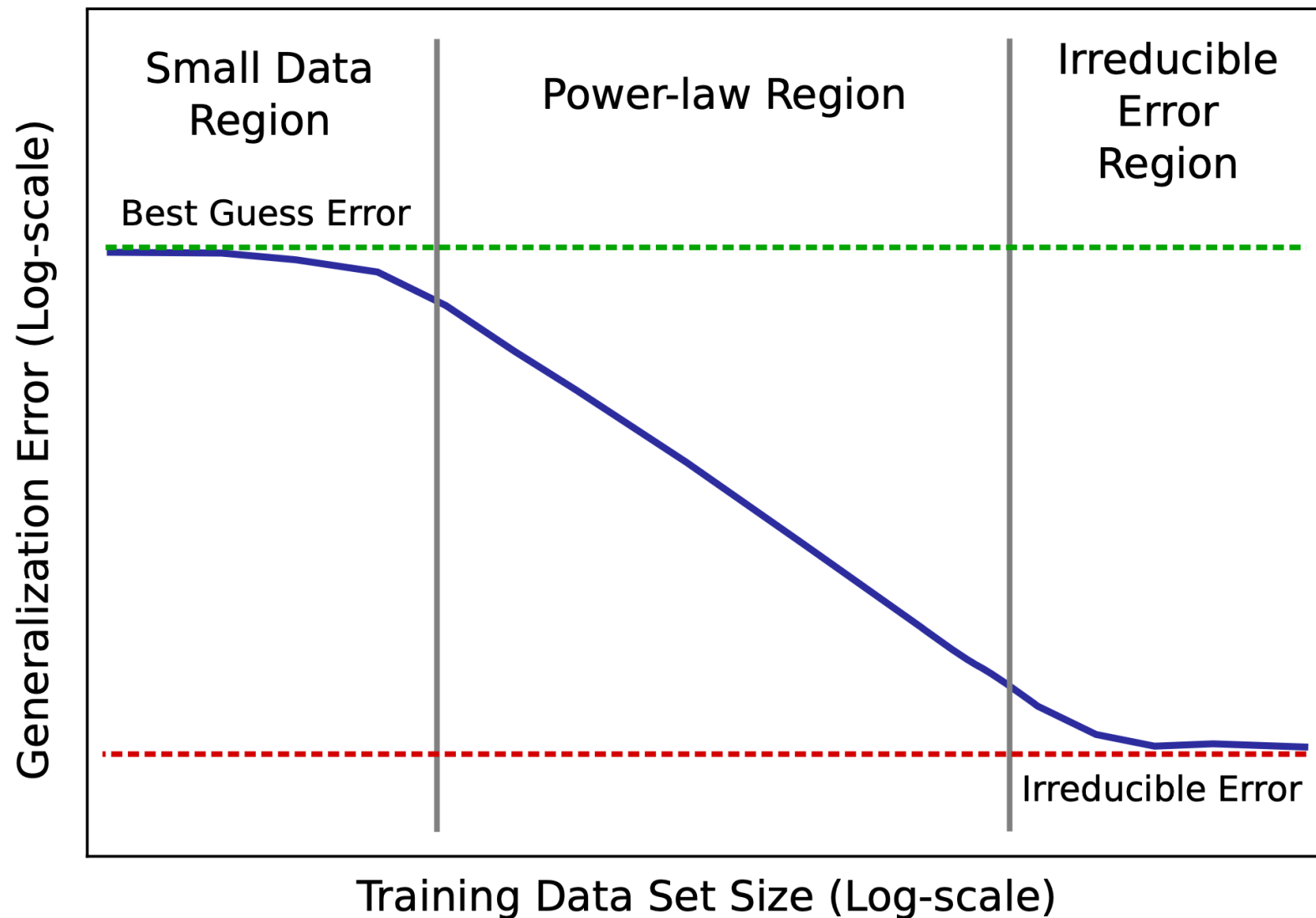
Demo: <https://www.youtube.com/watch?v=kplJTDXkOKY>



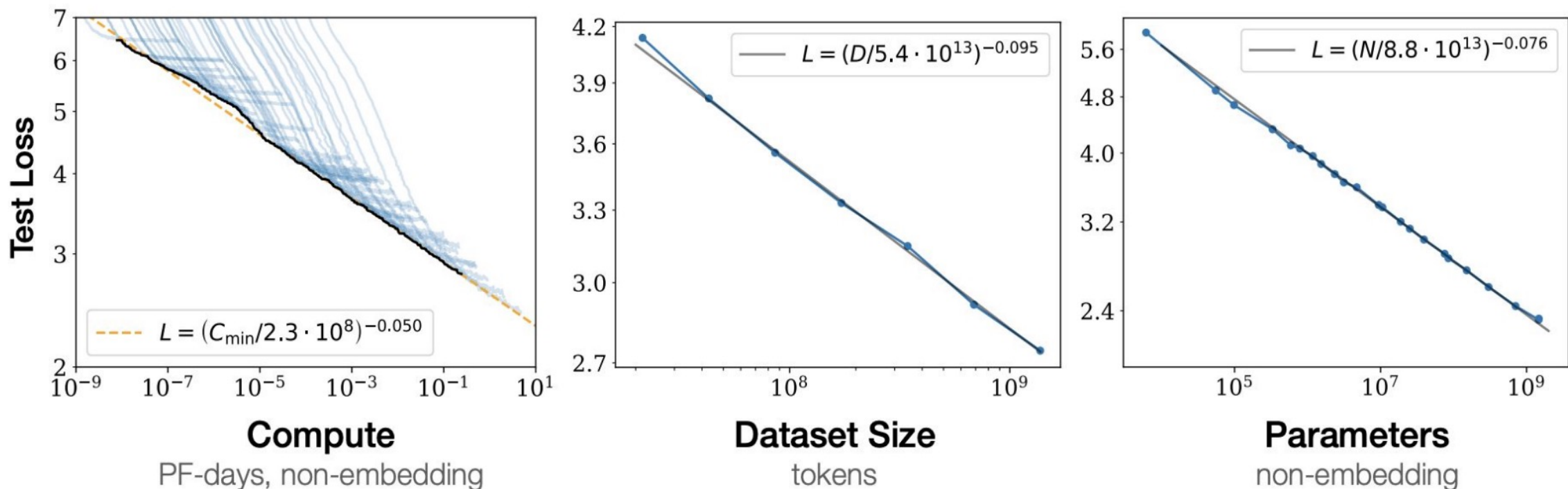
ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Scaling laws

Scaling laws



Scaling laws



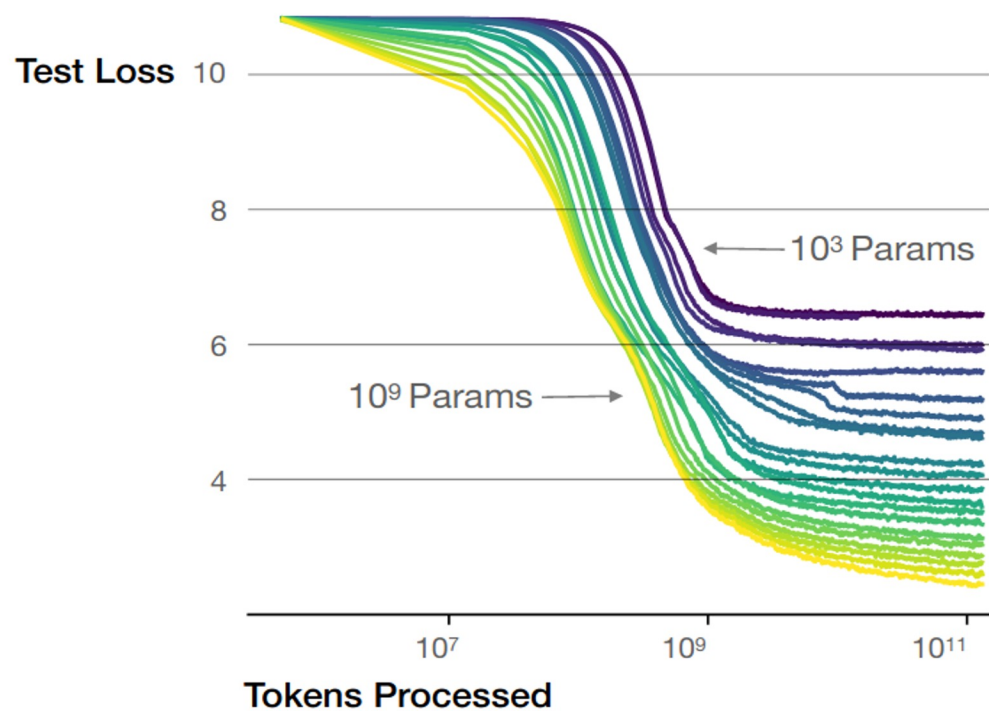
Joint data-model scaling laws

$$\hat{L}(N, D) \triangleq E + \frac{A}{N^\alpha} + \frac{B}{D^\beta}.$$

Approach	Coeff. a where $N_{opt} \propto C^a$	Coeff. b where $D_{opt} \propto C^b$
1. Minimum over training curves	0.50 (0.488, 0.502)	0.50 (0.501, 0.512)
2. IsoFLOP profiles	0.49 (0.462, 0.534)	0.51 (0.483, 0.529)
3. Parametric modelling of the loss	0.46 (0.454, 0.455)	0.54 (0.542, 0.543)
Kaplan <i>et al.</i> (2020) [23]	0.73	0.27



Larger models require **fewer samples** to reach the same performance



The optimal model size grows smoothly with the loss target and compute budget

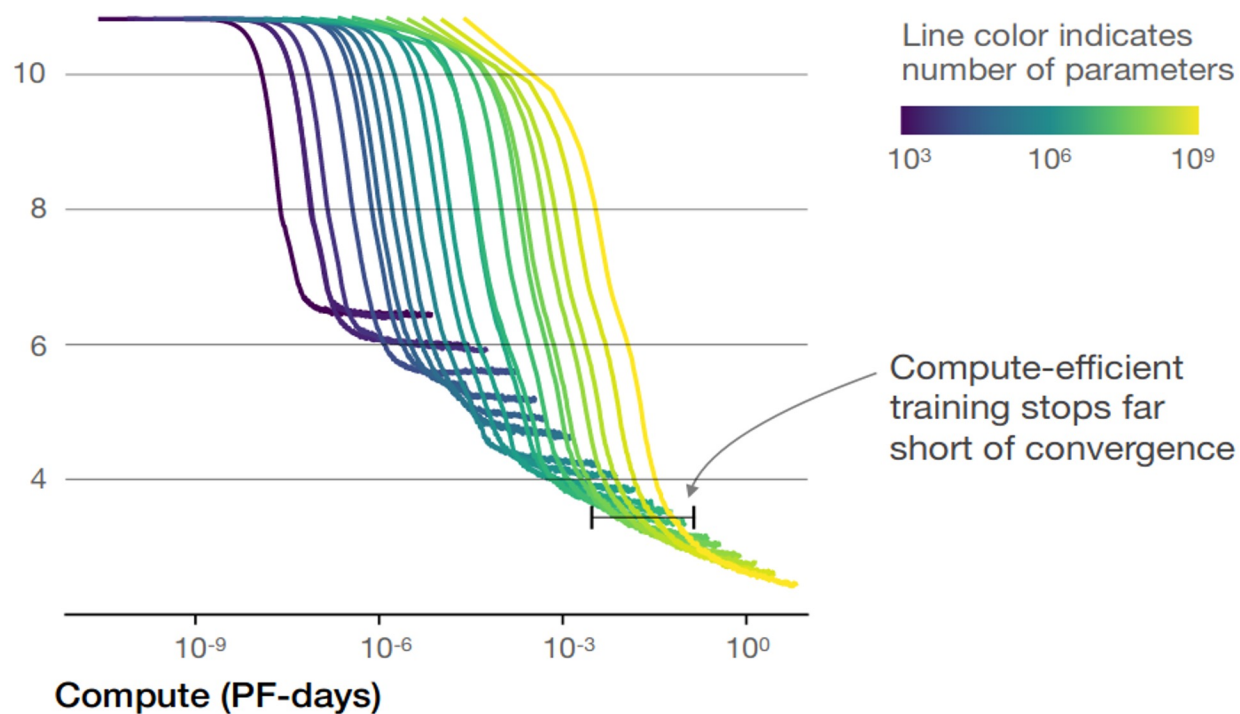
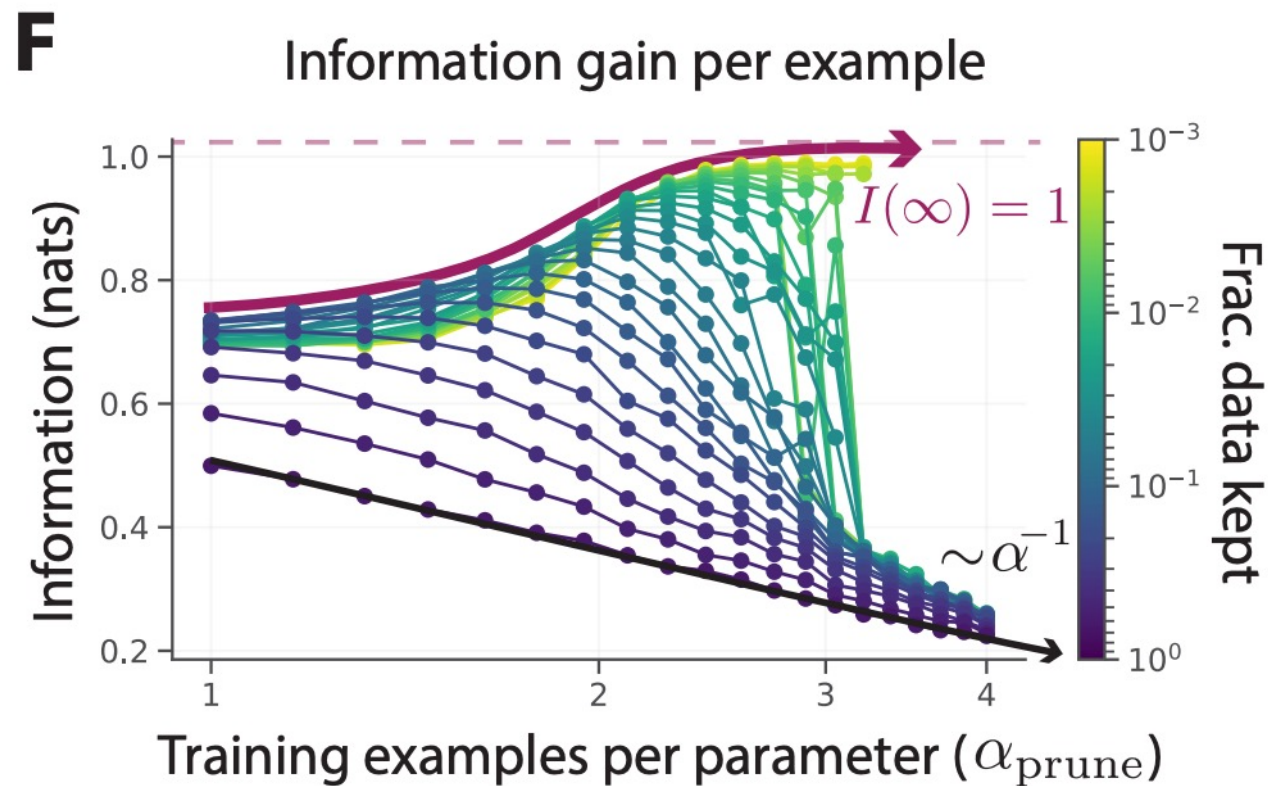
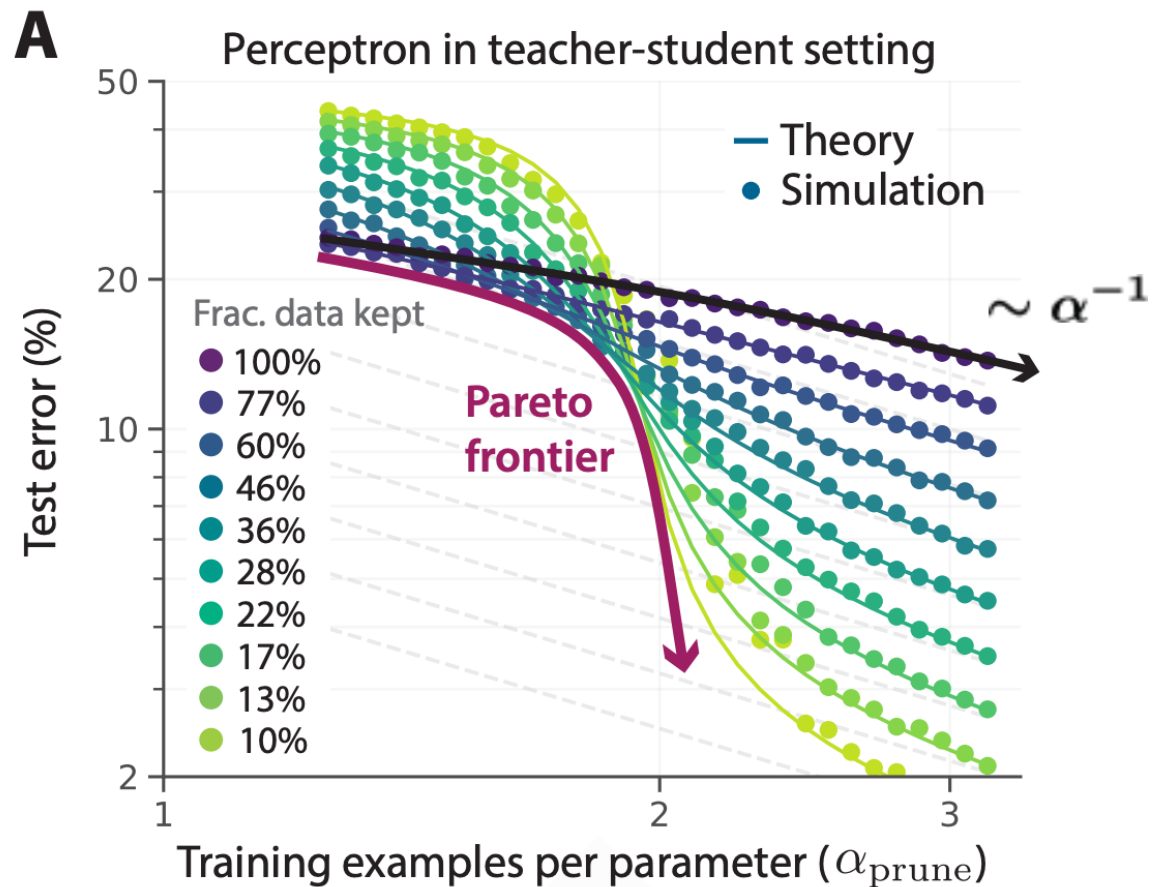


Figure 2 We show a series of language model training runs, with models ranging in size from 10^3 to 10^9 parameters (excluding embeddings).

Beat power laws by data pruning



Scaling on other tasks

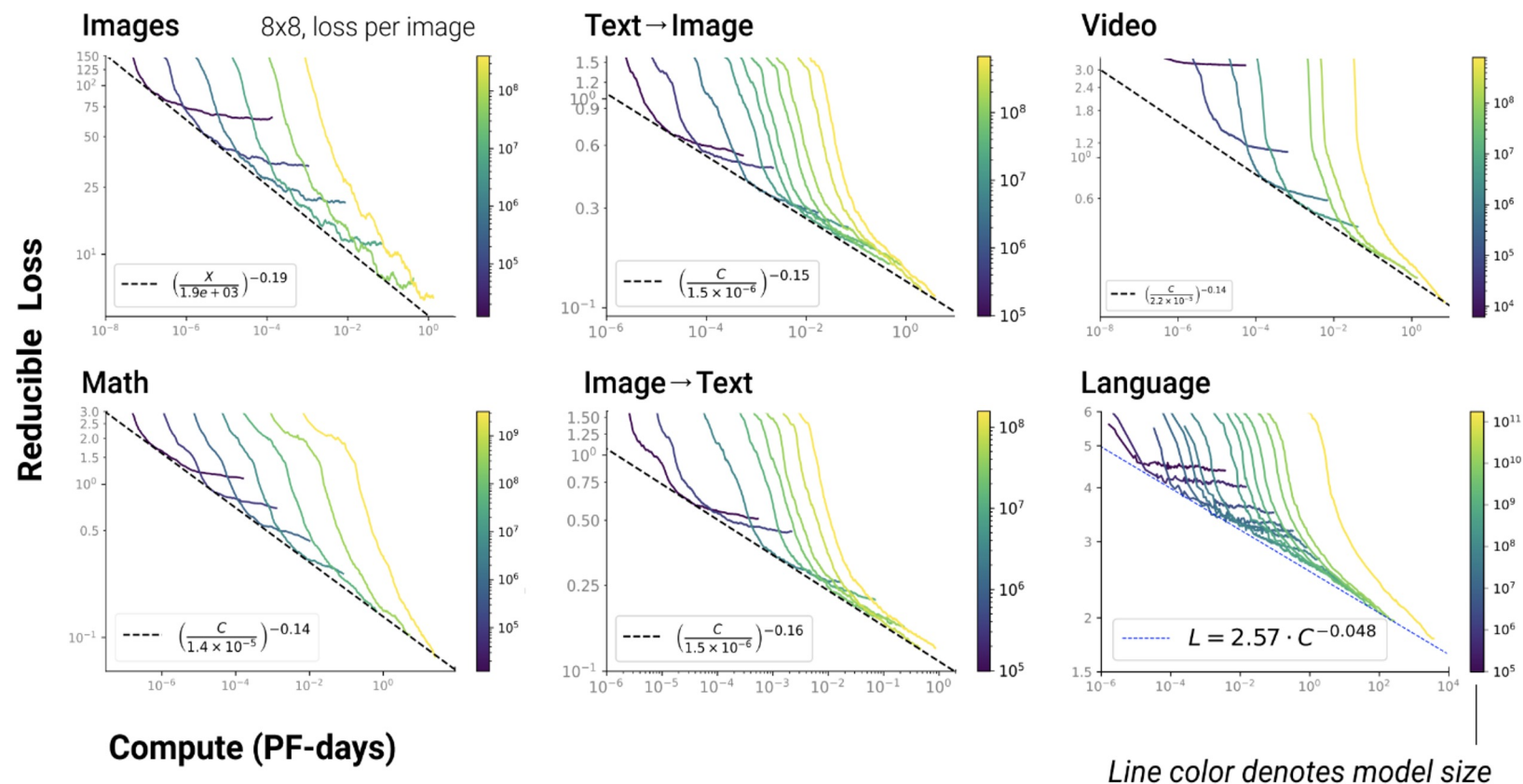


Figure 1 Smooth scaling of reducible loss across domains— We show power-law scaling laws for the reducible loss $L - L_\infty$ as a function of compute, where the irreducible loss L_∞ is a fitted domain-dependent constant. Under plausible assumptions concerning the infinite data and compute limits, the irreducible loss estimates the entropy of the underlying data distribution, while the reducible loss approximates the KL divergence between the data and model distributions. In the case of language we use results from [BMR⁺20], and only show the full loss L .

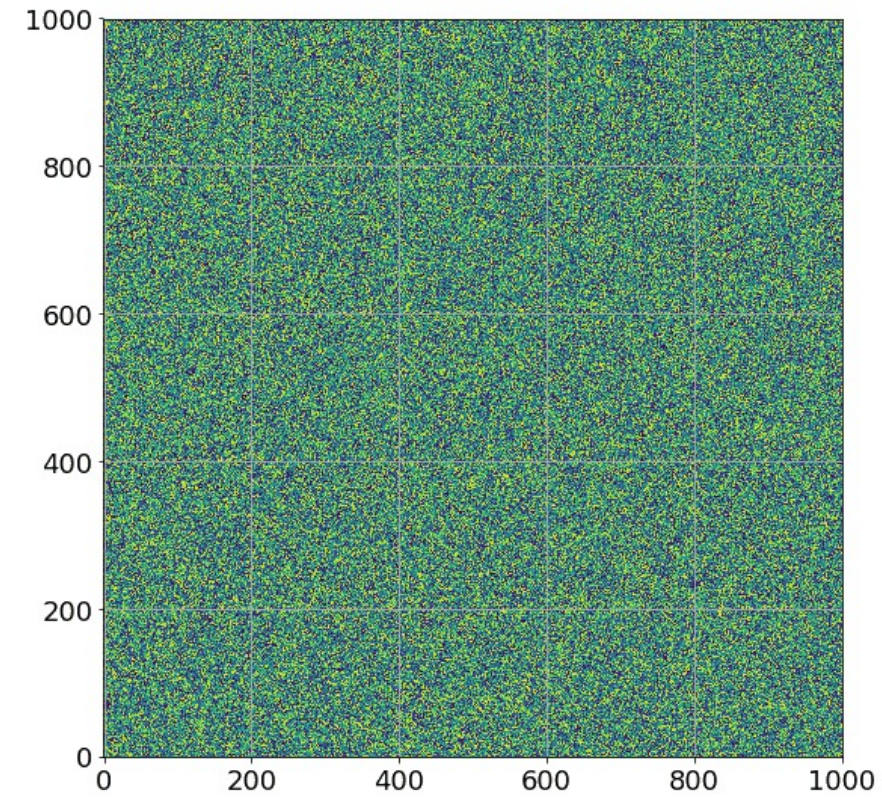
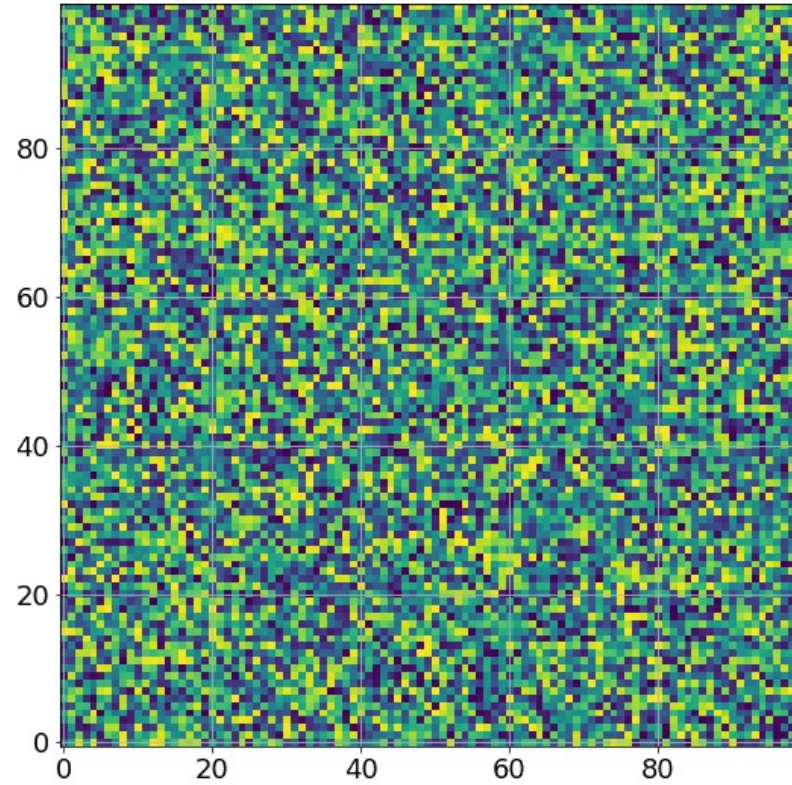
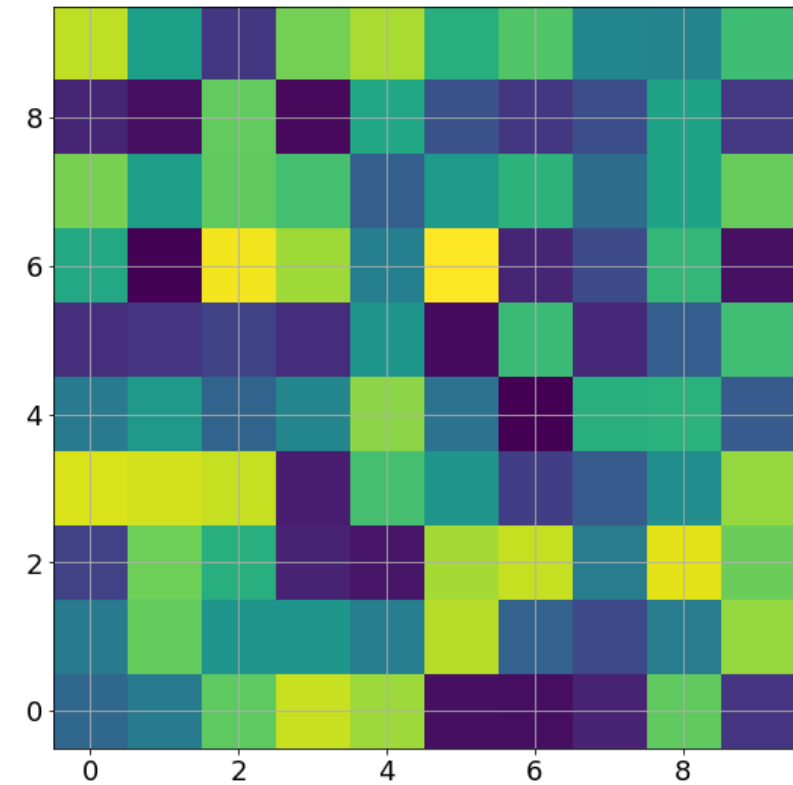
Domain	$L(N)$ (model size)	$L(C)$ (compute)	$N_{\text{opt}}(C)$
Language	$\left(\frac{N}{1.47 \times 10^{14}}\right)^{-0.070}$	$\left(\frac{C}{3.47 \times 10^8}\right)^{-0.048}$	$\left(\frac{C}{3.3 \times 10^{-13}}\right)^{0.73}$
Image 8x8	$3.12 + \left(\frac{N}{8.0 \times 10^1}\right)^{-0.24}$	$3.13 + \left(\frac{C}{1.8 \times 10^{-8}}\right)^{-0.19}$	$\left(\frac{C}{5.3 \times 10^{-14}}\right)^{0.64}$
Image 16x16	$2.64 + \left(\frac{N}{2.8 \times 10^2}\right)^{-0.22}$	$2.64 + \left(\frac{C}{1.6 \times 10^{-8}}\right)^{-0.16}$	$\left(\frac{C}{4.8 \times 10^{-12}}\right)^{0.75}$
Image 32x32	$2.20 + \left(\frac{N}{6.3 \times 10^1}\right)^{-0.13}$	$2.21 + \left(\frac{C}{3.6 \times 10^{-9}}\right)^{-0.1}$	$\left(\frac{C}{1.6 \times 10^{-13}}\right)^{0.65}$
Image VQ 16x16	$3.99 + \left(\frac{N}{2.7 \times 10^4}\right)^{-0.13}$	$4.09 + \left(\frac{C}{6.1 \times 10^{-7}}\right)^{-0.11}$	$\left(\frac{C}{6.2 \times 10^{-14}}\right)^{0.64}$
Image VQ 32x32	$3.07 + \left(\frac{N}{1.9 \times 10^4}\right)^{-0.14}$	$3.17 + \left(\frac{C}{2.6 \times 10^{-6}}\right)^{-0.12}$	$\left(\frac{C}{9.4 \times 10^{-13}}\right)^{0.7}$
Text-to-Im (Text)	$\left(\frac{N}{5.6 \times 10^8}\right)^{-0.037}$	(combined text/image loss)	$\left(\frac{C}{9.4 \times 10^{-13}}\right)^{0.7}$
Text-to-Im (Image)	$2.0 + \left(\frac{N}{5.1 \times 10^3}\right)^{-0.16}$	$1.93 + \left(\frac{C}{1.5 \times 10^{-6}}\right)^{-0.15}$	
Im-to-Text (Text)	$\left(\frac{N}{7.0 \times 10^8}\right)^{-0.039}$	(combined text/image loss)	$\left(\frac{C}{3.3 \times 10^{-12}}\right)^{0.72}$
Im-to-Text (Image)	$2.0 + \left(\frac{N}{5.5 \times 10^3}\right)^{-0.15}$	$1.97 + \left(\frac{C}{1.5 \times 10^{-6}}\right)^{-0.16}$	
Video VQ 16x16x16	$1.01 + \left(\frac{N}{3.7 \times 10^4}\right)^{-0.24}$	$0.95 + \left(\frac{C}{2.2 \times 10^{-5}}\right)^{-0.14}$	$\left(\frac{C}{1.13 \times 10^{-12}}\right)^{0.71}$
Math (Extrapolate)	$0.28 + \left(\frac{N}{1.1 \times 10^4}\right)^{-0.16}$	$0.14 + \left(\frac{C}{1.4 \times 10^{-5}}\right)^{-0.17}$	$\left(\frac{C}{2.3 \times 10^{-12}}\right)^{0.69}$



ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

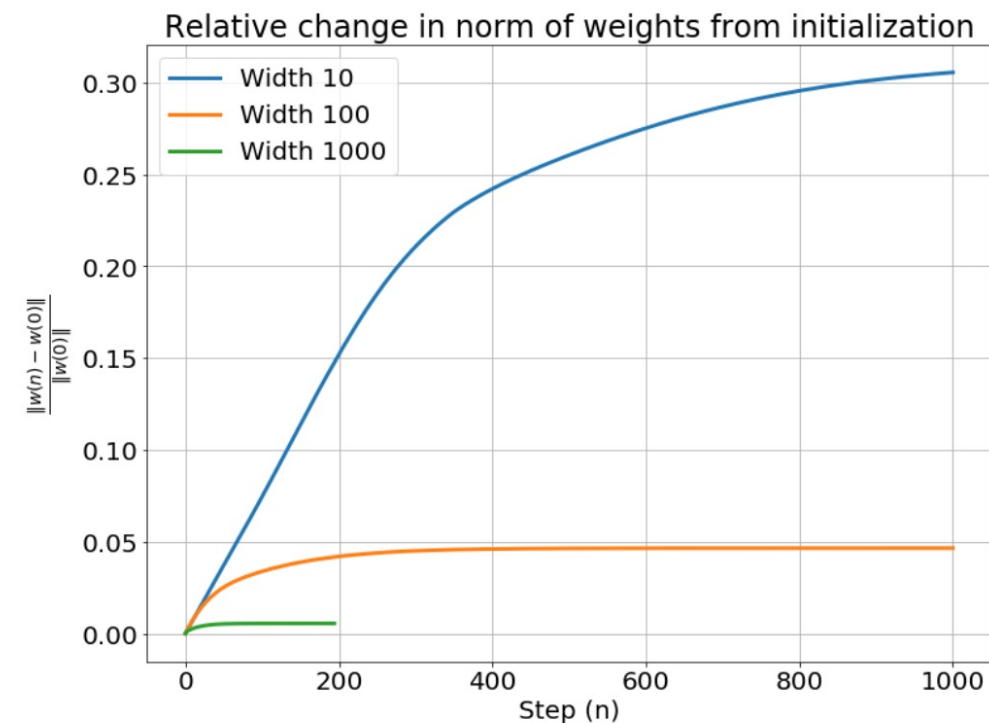
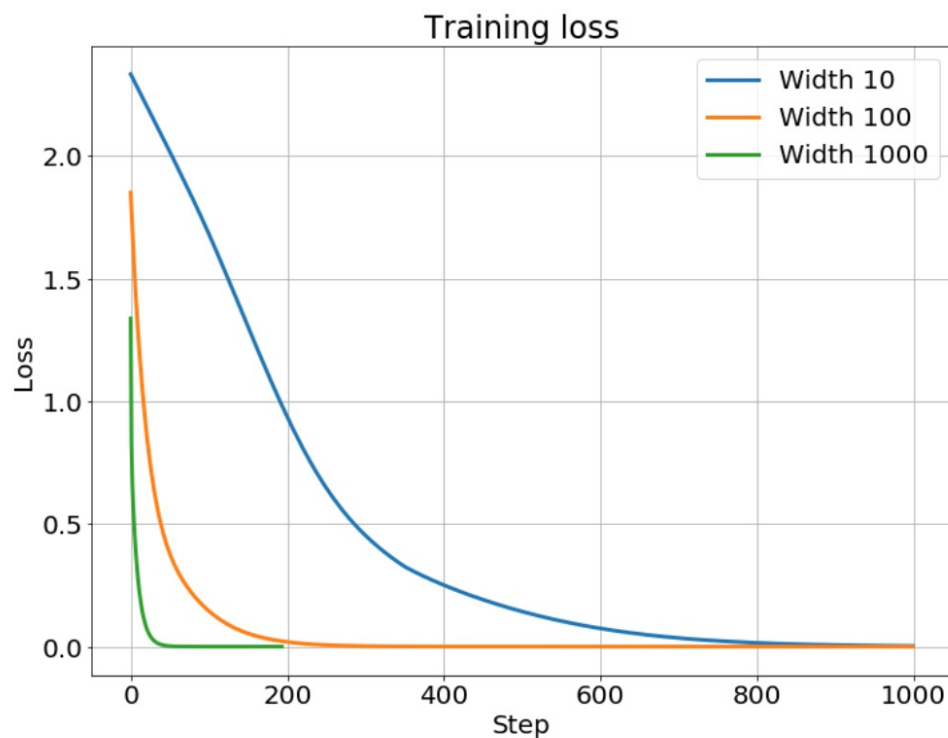
Neural Tangent Kernels

Lazy learning



Lazy learning

$$\frac{\|\mathbf{w}(n) - \mathbf{w}_0\|_2}{\|\mathbf{w}_0\|_2}$$



Linearization

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^N \ell(f(\mathbf{x}^{(i)}; \theta), y^{(i)}) \quad \nabla_{\theta} \mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^N \underbrace{\nabla_{\theta} f(\mathbf{x}^{(i)}; \theta)}_{\text{size } P \times n_L} \underbrace{\nabla_f \ell(f, y^{(i)})}_{\text{size } n_L \times 1}$$

$$\frac{d\theta}{dt} = -\nabla_{\theta} \mathcal{L}(\theta) = -\frac{1}{N} \sum_{i=1}^N \nabla_{\theta} f(\mathbf{x}^{(i)}; \theta) \nabla_f \ell(f, y^{(i)})$$

$$\frac{df(\mathbf{x}; \theta)}{dt} = \frac{df(\mathbf{x}; \theta)}{d\theta} \frac{d\theta}{dt} = -\frac{1}{N} \sum_{i=1}^N \underbrace{\nabla_{\theta} f(\mathbf{x}; \theta)^{\top} \nabla_{\theta} f(\mathbf{x}^{(i)}; \theta)}_{\text{Neural tangent kernel}} \nabla_f \ell(f, y^{(i)})$$

$$K(\mathbf{x}, \mathbf{x}'; \theta) = \nabla_{\theta} f(\mathbf{x}; \theta)^{\top} \nabla_{\theta} f(\mathbf{x}'; \theta)$$

$$K(\mathbf{x}, \mathbf{x}'; \theta) = \nabla_{\theta} f(\mathbf{x}; \theta)^{\top} \nabla_{\theta} f(\mathbf{x}'; \theta)$$

When $n_1, \dots, n_L \rightarrow \infty$ (network with infinite width), the NTK converges to be:

- (1) deterministic at initialization, meaning that the kernel is irrelevant to the initialization values and only determined by the model architecture; and
- (2) stays constant during training.

$$\kappa(\theta) = \frac{\Delta(\nabla_{\theta} f)}{\|\nabla_{\theta} f(\theta(0))\|} = \|\hat{y} - f(\theta(0))\| \frac{\nabla_{\theta}^2 f(\theta(0))}{\|\nabla_{\theta} f(\theta(0))\|^2} \rightarrow 0$$

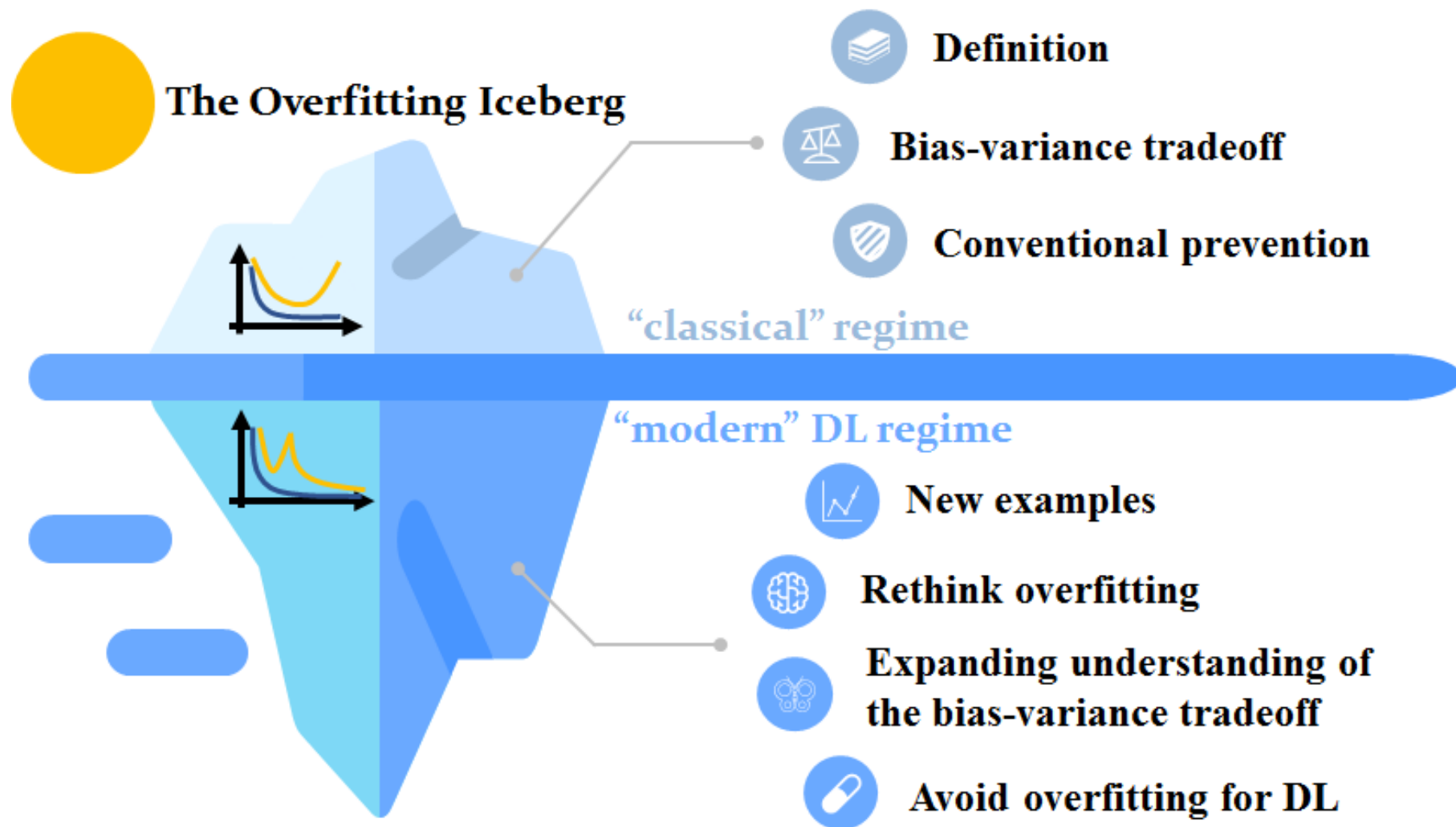
$$f(\mathbf{w}, \mathbf{x}) = f(\mathbf{w}_0, \mathbf{x}) + \langle \mathbf{w} - \mathbf{w}_0, \phi_{\mathbf{w}_0}(\mathbf{x}) \rangle + \frac{1}{2} \langle \mathbf{w} - \mathbf{w}_0, H(\xi)(\mathbf{w} - \mathbf{w}_0) \rangle$$

$$\sup_{\mathbf{w} \in \mathcal{B}} f(\mathbf{w}, \mathbf{x}) - f(\mathbf{w}_0, \mathbf{x}) - \langle \mathbf{w} - \mathbf{w}_0, \phi_{\mathbf{w}_0}(\mathbf{x}) \rangle \leq \frac{R^2}{2} \sup_{\xi \in \mathcal{B}} \|H(\xi)\|$$

- For a general (feed-forward) neural network with L hidden layers and a linear output layer

$$\sup_{\xi \in \mathcal{B}} \|H(\xi)\| \leq O^* \left(\frac{1}{\sqrt{m}} \right), \text{ where } m = \min_{l=1, \dots, L} (d_l)$$

Conclusions



A graphic on the left side of the slide. It features a dark blue background with a large, stylized circular pattern of red dots. The dots are arranged in concentric, slightly irregular rings, creating a sense of depth and movement. In the center of this pattern, the word "HUST" is written in a bold, white, sans-serif font.

HUST

THANK YOU !