12. Is ERM enough?

CPSC 532D: Modern Statistical Learning Theory
November 2024
cs.ubc.ca/~dsuth/532D/24w1/

- A good thing for deep learning: universal approximation

- Two major challenges for using ERM to explain deep learning:
  - We don’t do ERM, because it’s NP-hard to compute
  - Uniform convergence ERM bounds might not be enough for generalization

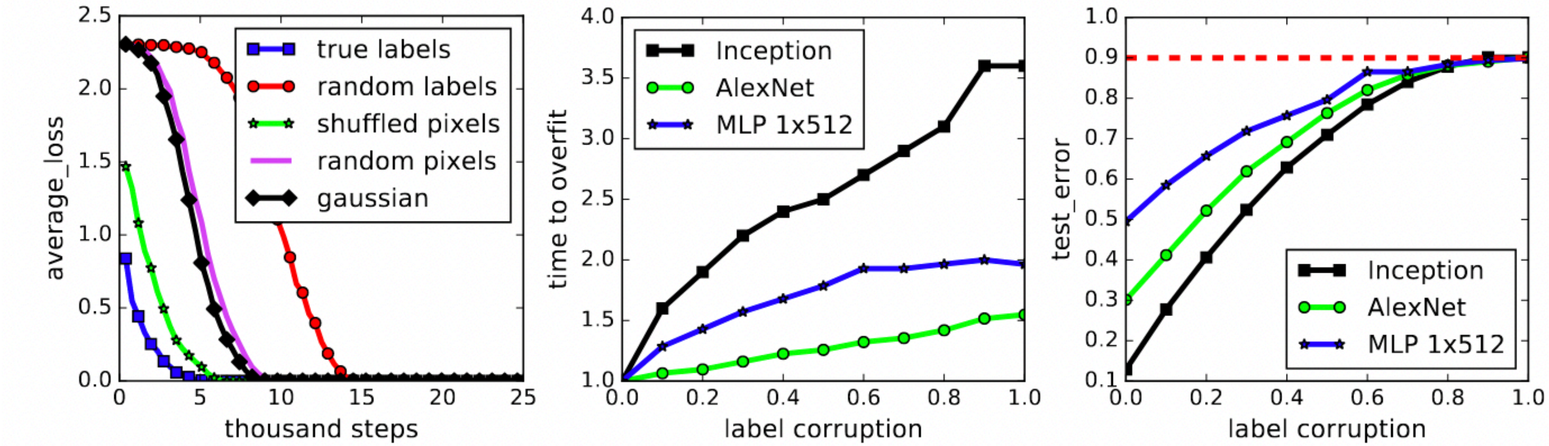
Deep learning: VC dimension

- For ReLU (or general piecewise-linear) nets with P params and depth D:
  - VCdim = $\mathcal{O}(PD \log P)$, $\Omega\left(PD \log \frac{P}{D}\right)$, so nearly tight [BHLM19]
  - $P = \prod_{k=1}^{D} d_{\ell-1} d_{\ell}$ for fully-connected networks
- For piecewise-constant, e.g. threshold functions, VCdim = $\Theta(P \log P)$
- For piecewise-polynomial, $\mathcal{O}(PD^2 + PD \log P)$, $\mathcal{O}(PU)$ with U units
- For sigmoids/similar, $\mathcal{O}(P^2U^2)$ and $\Omega(P^2)$
  - Theorem 8.13/8.14 of Anthony & Bartlett (1999) textbook - UBC access

Problems with parameter counting

- We use networks with a **lot** of parameters
  - ResNet-50 has ~25 million parameters and depth 50: VCdim > 1 billion

- We can train our networks to get zero error even for **random labels**
  - Even AlexNet can shatter CIFAR-10, *almost* shatter ImageNet
  - Neyshabur et al. (2015), Zhang et al. (2017)



(a) learning curves (b) convergence slowdown (c) generalization error growth

Figure 1: Fitting random labels and random pixels on CIFAR10. (a) shows the training loss of various experiment settings decaying with the training steps. (b) shows the relative convergence time with different label corruption ratio. (c) shows the test error (also the generalization error since training error is 0) under different label corruptions.

Problems with parameter counting

- We use networks with a **lot** of parameters
  - ResNet-50 has ~25 million parameters and depth 50: VCdim > 1 billion

- We can train our networks to get zero error even for **random labels**
  - Even AlexNet can shatter CIFAR-10, *almost* shatter ImageNet
  - Neyshabur et al. (2015), Zhang et al. (2017)
  - But these architectures do generalize well – VC of arch. can't explain that

- Making hidden layers wider can often improve generalization, but worsens parameter counting-based bounds
- Remember that $\mathscr{H}_k$ has infinite VCdim for universal kernels, but we can still learn with *small-norm* predictors

Rademacher complexity

- Solution to assignment 3, question 3 showed:
  - for depth-D nets whose weights W_k have all rows $\|(W_k)_{(i,:)}\|_1 \leq B$ and no intercept,
  - if they use componentwise M-Lipschitz activations with $\sigma(0) = 0$,
  - if $\|X\|_\infty \leq C$ a.s., the Rademacher complexity $\leq \frac{1}{\sqrt{m}} C(2MB)^D \sqrt{2 \log d}$
- A similar but fancier proof (see here) when $M = 1$, $\|X\|_2 \leq C$, and all $\|W_k\|_F \leq B$,
  - the Rademacher complexity $\leq \frac{1}{\sqrt{m}} B^D C(1 + \sqrt{2D \log 2})$
- Another way via covering numbers gives a bound based on $\prod_k \|W_k\|$
  - Product of spectral norms upper bounds the Lipschitz constant of the net

Problem with norm-based bounds

- These kinds of bounds tend to be “vacuous” (e.g. prove 0-1 error is less than 17) for realistic problems

Fantastic Generalization Measures and Where to Find Them

Yiding Jiang*, Behnam Neyshabur*, Hossein Mobahi
Dilip Krishnan, Samy Bengio
Google
{ydjiang,neyshabur,hmobahi,dilipkay,bengio}@google.com

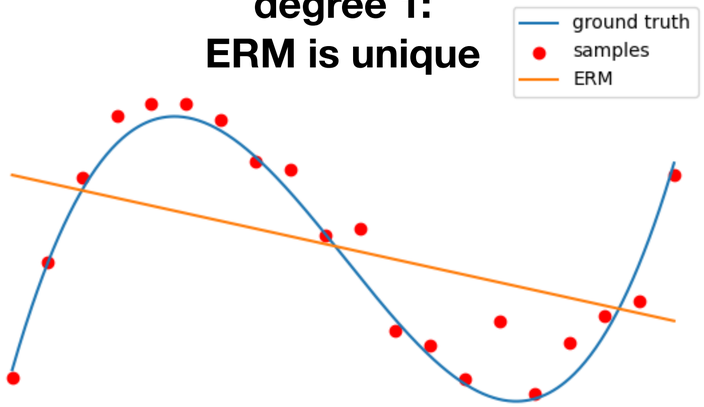
In this study, we trained more than 10,000 models over two image classification datasets, namely, CIFAR-10 (Krizhevsky et al., 2014) and Street View House Numbers (SVHN) Netzer et al. (2011). In

2. Many norm-based measures not only perform poorly, but *negatively* correlate with generalization specifically when the optimization procedure injects some stochasticity. In particular, the generalization bound based on the product of spectral norms of the layers (similar to that of Bartlett et al. (2017)) has very strong negative correlation with generalization.

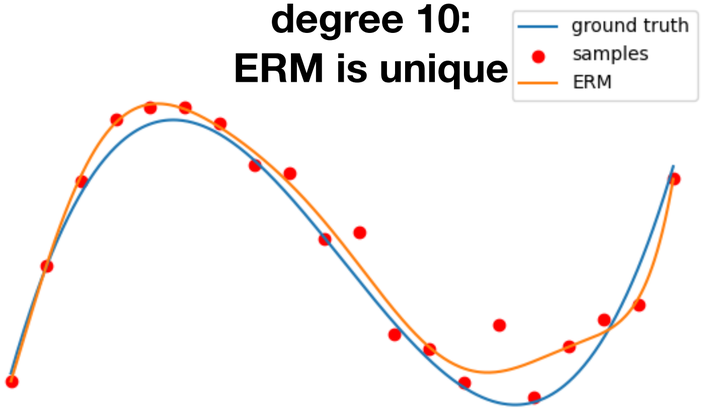
Some ERMs actually might not generalize

(m = 20, y = cubic(x) + noise)

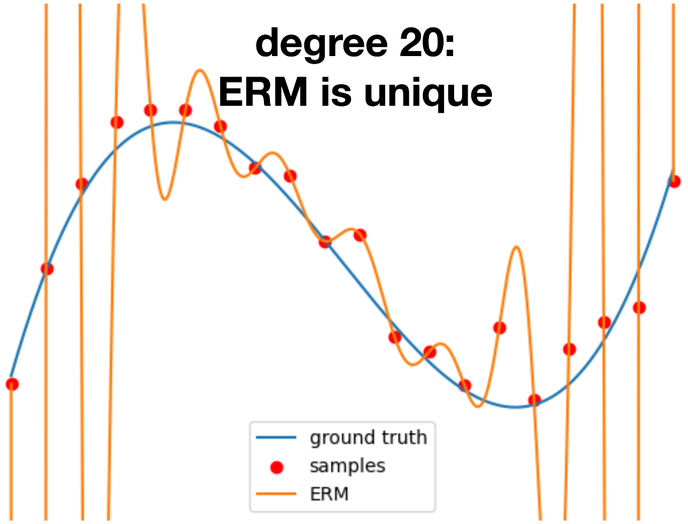
degree 1:
ERM is unique



degree 10:
ERM is unique

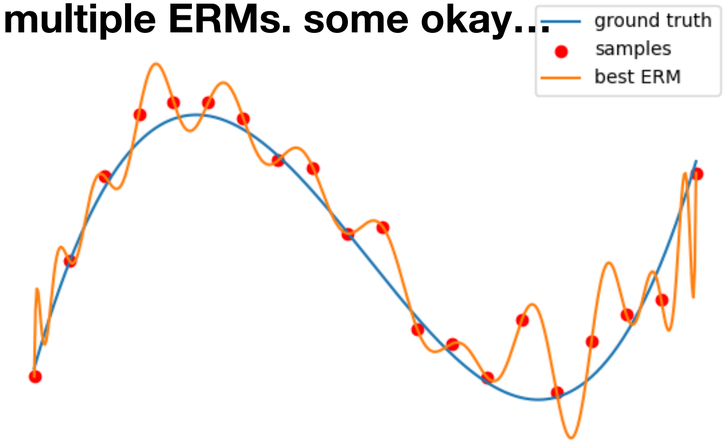


degree 20:
ERM is unique

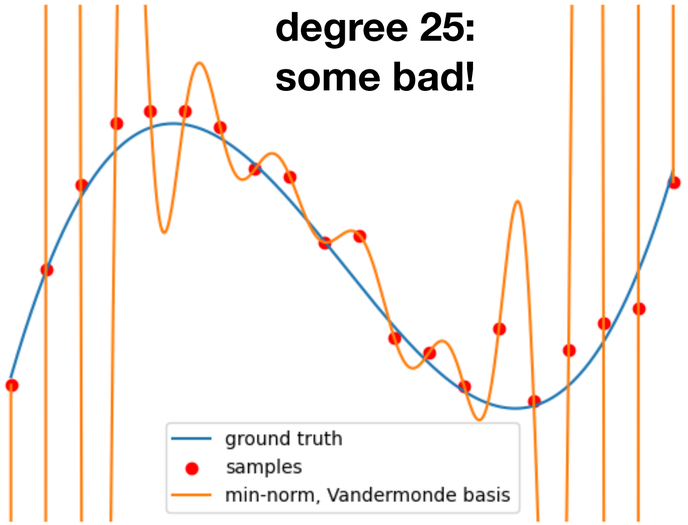


should be degree 19!!!

degree 25:
multiple ERMs. some okay…



degree 25:
some bad!

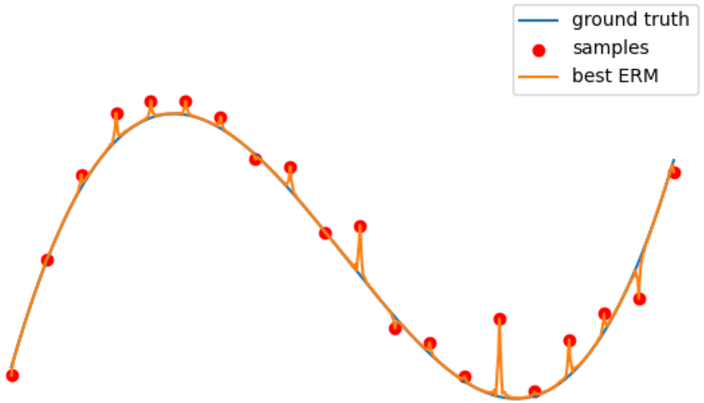


"best" = $\min_{h:L_S(h)=0} L_{\mathscr{D}}(h)$ for $\mathscr{D}_x = \text{Uniform}([-1,1])$

inspired by Nakkiran et al. blog post's companion notebook

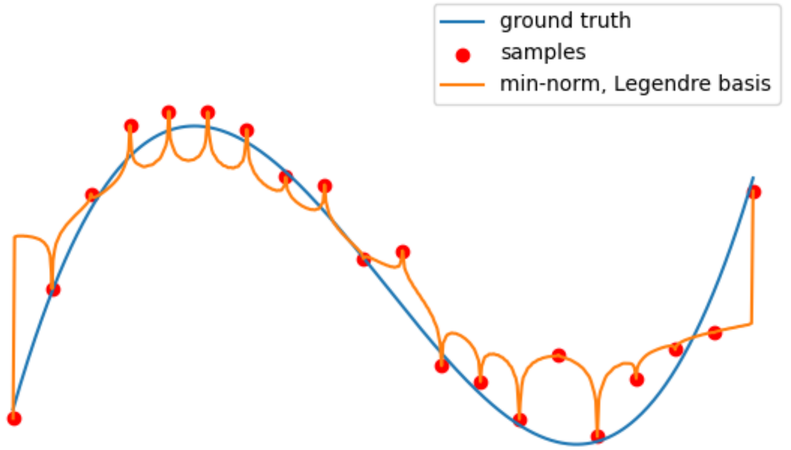
Some algorithms generalize, some don't

degree 1,000:



$$\min_{L_S(h)=0} L_{\mathscr{D}}(h)$$

$L_S(h) = 0, L_{\mathscr{D}}(h)$ almost Bayes error

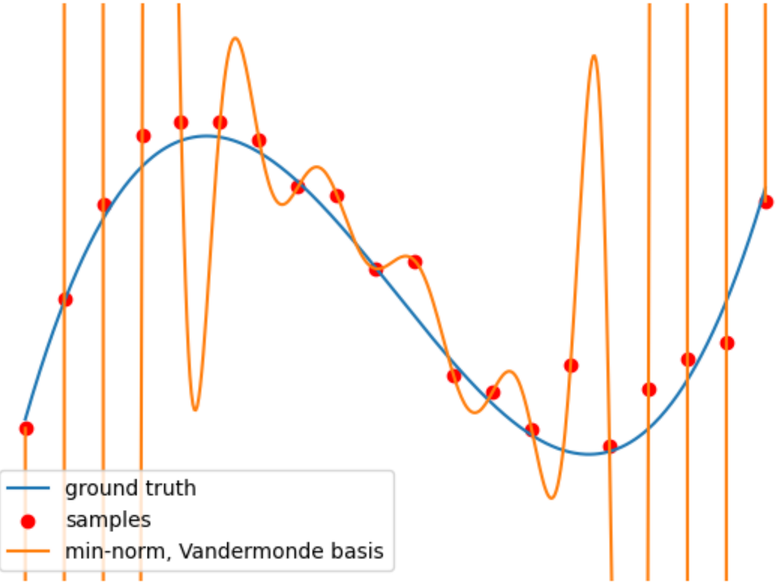


$$\min_{L_S(h)=0} \|h\|, \text{ using Legendre basis}$$

linear combination of *orthogonal* polynomials

$L_S(h) = 0, L_{\mathscr{D}}(h)$ okay-ish

$$\|h\|_{Legendre} \approx 2$$
$$\|h\|_{Vandermonde} \approx 2{,}000{,}000{,}000{,}000$$



$$\min_{L_S(h)=0} \|h\|, \text{ using Vandermonde form}$$

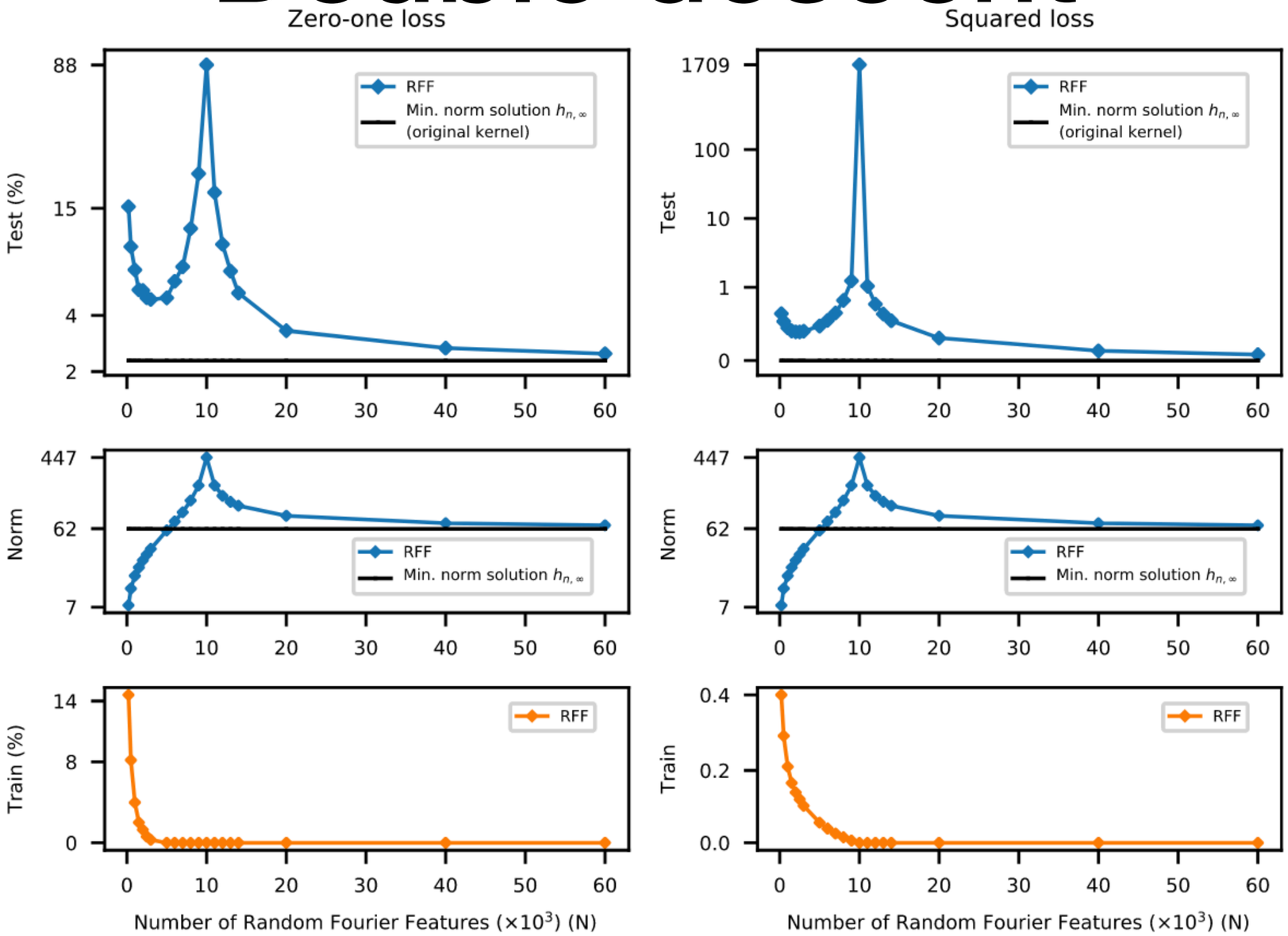
$$h(x) = w_0 + w_1 x + w_2 x^2 + \ldots + w_d x^d$$

$L_S(h) = 0, L_{\mathscr{D}}(h)$ awful

$$\|h\|_{Legendre} \approx 700{,}000$$
$$\|h\|_{Vandermonde} \approx 2{,}000{,}000$$

Double descent

Classical regime (left of peak): unique ERM



Interpolating regime (right of peak): many possible interpolators

which one we get depends on the algorithm

Fig. 2. Double-descent risk curve for the RFF model on MNIST. Shown are test risks (log scale), coefficient ℓ_2 norms (log scale), and training risks of the RFF model predictors $h_{n,N}$ learned on a subset of MNIST ($n = 10^4$, 10 classes). The interpolation threshold is achieved at $N = 10^4$.

Belkin/Hsu/Ma/Mandal, PNAS 2019

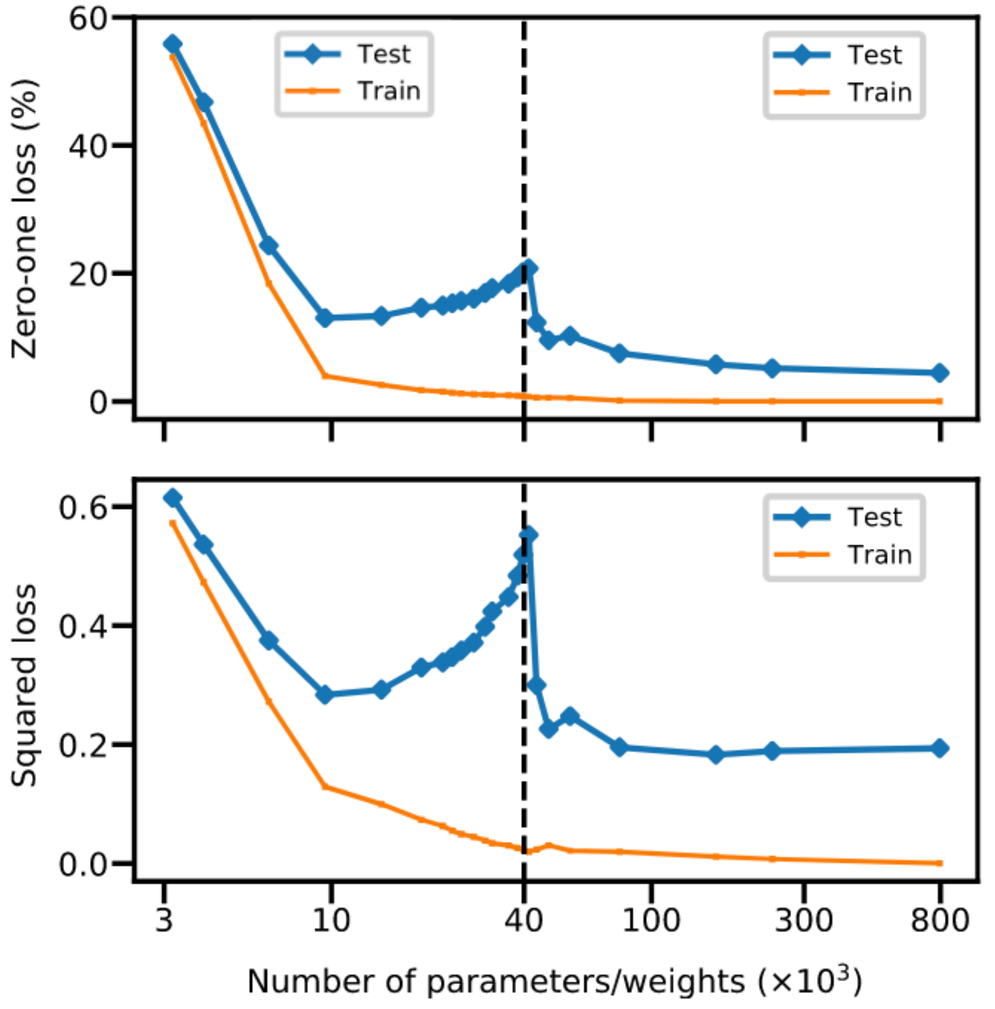


Fig. 3. Double-descent risk curve for a fully connected neural network on MNIST. Shown are training and test risks of a network with a single layer of H hidden units, learned on a subset of MNIST ($n = 4 \cdot 10^3$, $d = 784$, $K = 10$ classes). The number of parameters is $(d + 1) \cdot H + (H + 1) \cdot K$. The interpolation threshold (black dashed line) is observed at $n \cdot K$.

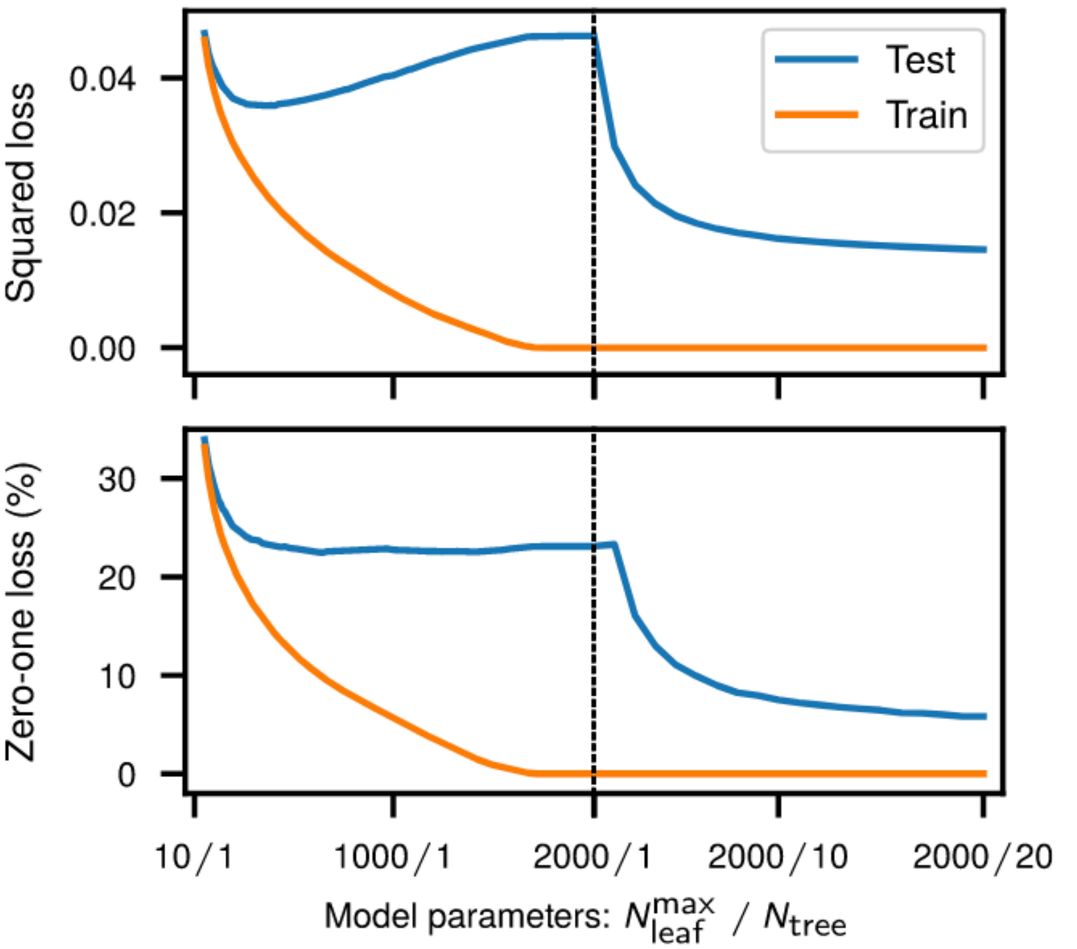
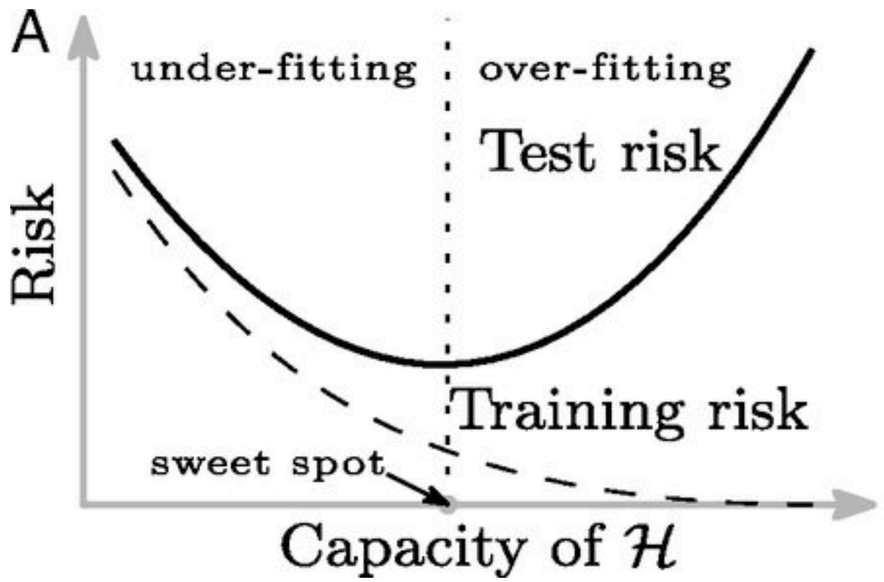
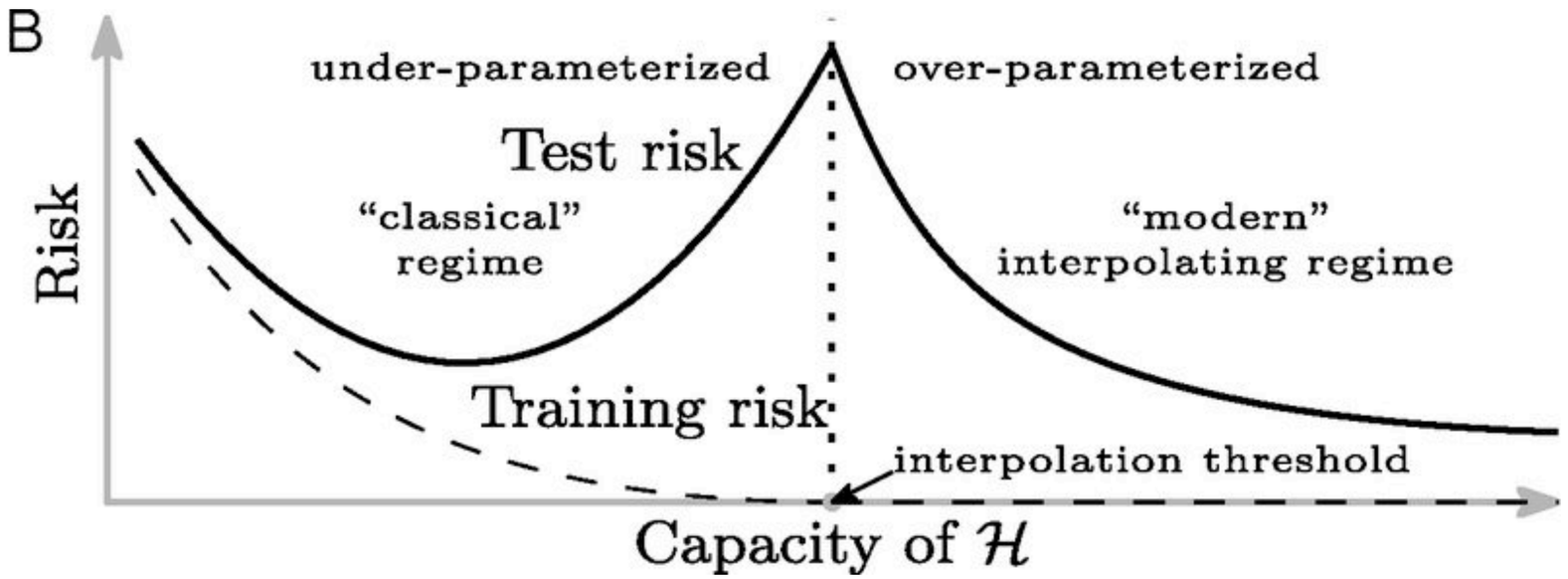


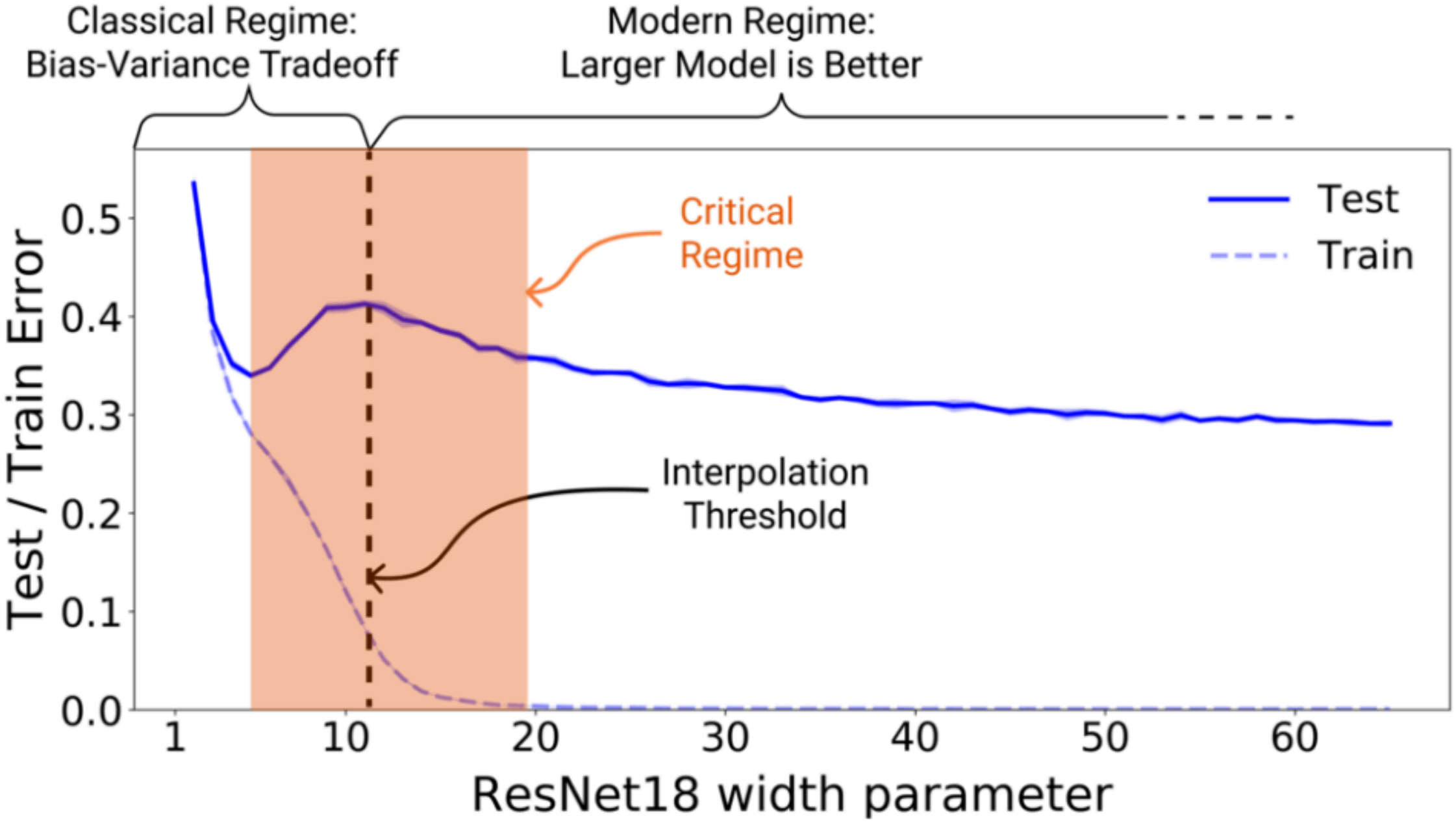
Fig. 4. Double-descent risk curve for random forests on MNIST. The double-descent risk curve is observed for random forests with increasing model complexity trained on a subset of MNIST ($n = 10^4$, 10 classes). Its complexity is controlled by the number of trees N_{tree} and the maximum number of leaves allowed for each tree $N_{\text{leaf}}^{\max}$.

Belkin/Hsu/Ma/Mandal, PNAS 2019

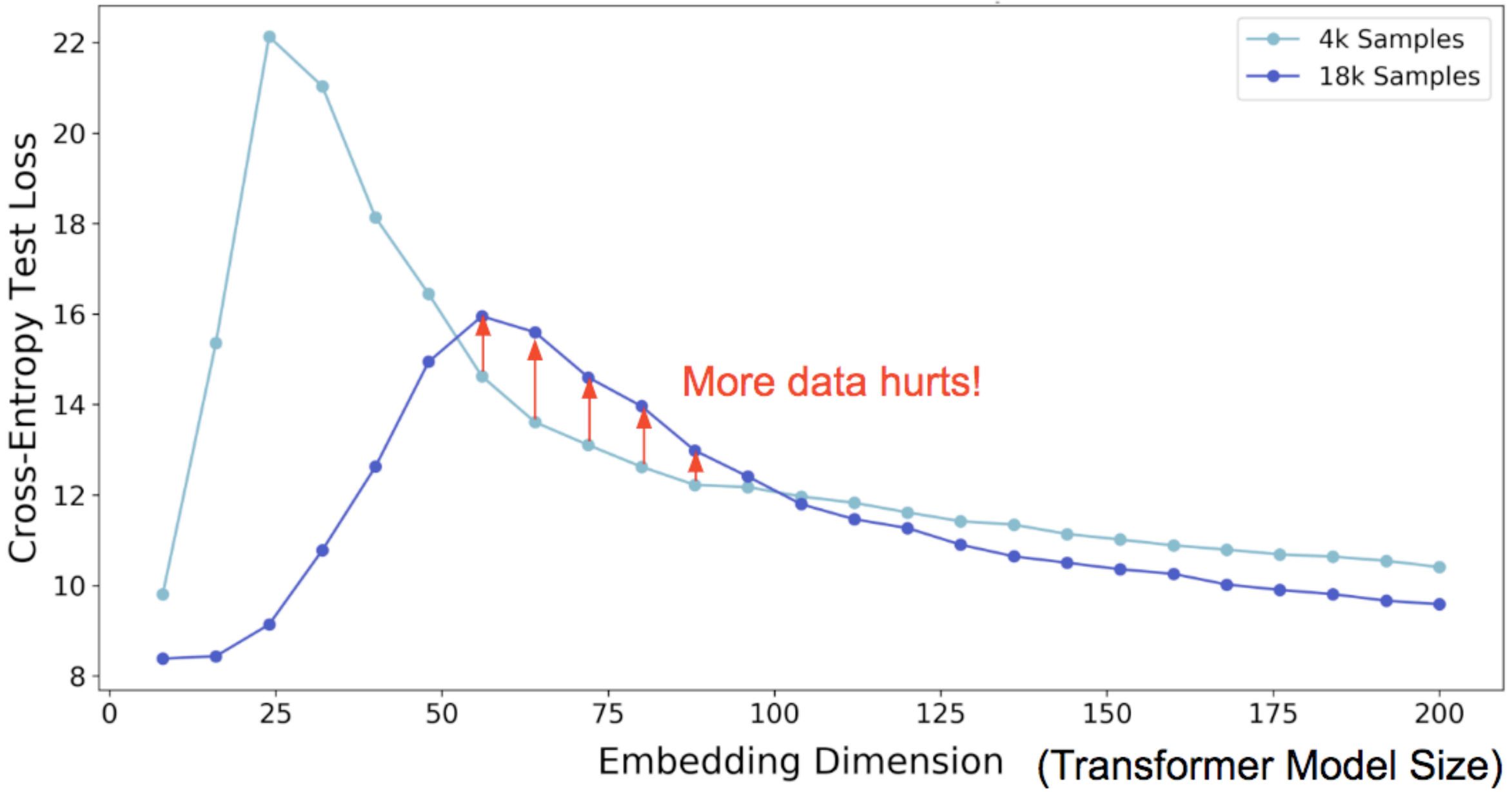




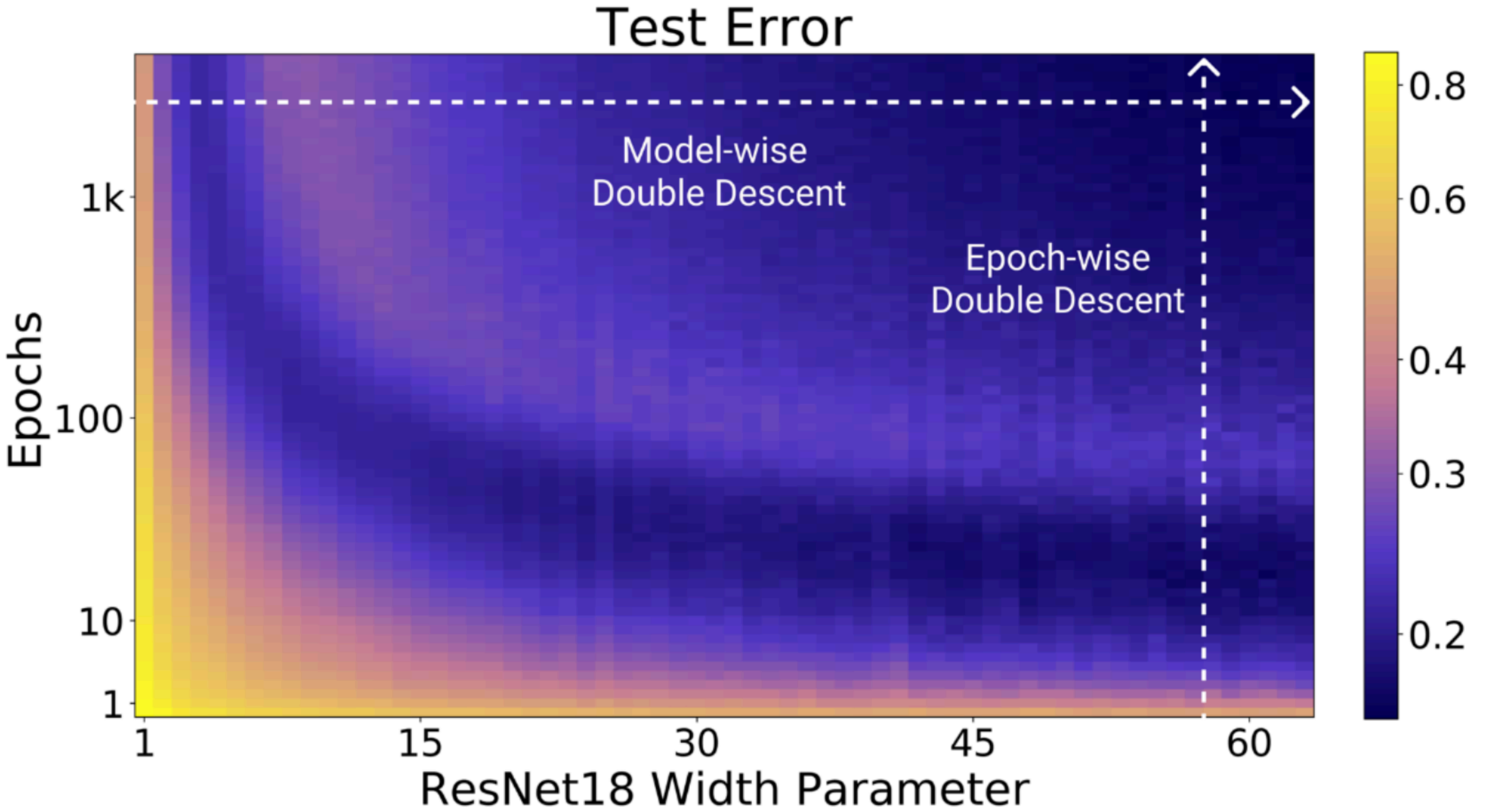
Belkin/Hsu/Ma/Mandal, PNAS 2019



Nakkiran et al. ICLR-20



Nakkiran et al. ICLR-20



Nakkiran et al. ICLR-20

Definition 1 (Effective Model Complexity) *The* Effective Model Complexity *(EMC) of a training procedure $\mathcal{T}$, with respect to distribution $\mathcal{D}$ and parameter $\epsilon > 0$, is defined as:*

$$\mathrm{EMC}_{\mathcal{D},\epsilon}(\mathcal{T}) := \max\{n \mid \mathbb{E}_{S\sim\mathcal{D}^n}[\mathrm{Error}_S(\mathcal{T}(S))] \leq \epsilon\}$$

where $\mathrm{Error}_S(M)$ is the mean error of model M on train samples S.

Our main hypothesis can be informally stated as follows:

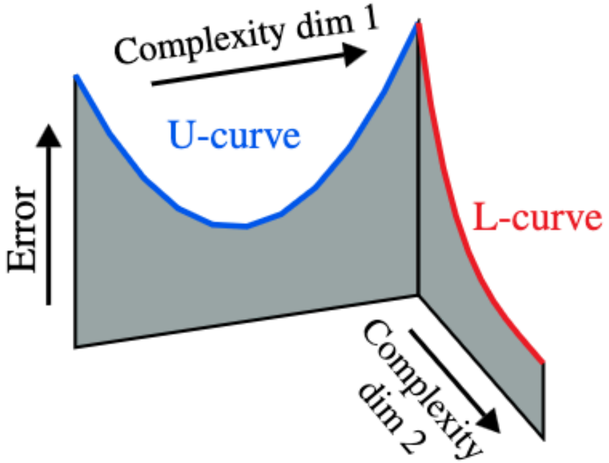
Hypothesis 1 (Generalized Double Descent hypothesis, informal) *For any natural data distribution $\mathcal{D}$, neural-network-based training procedure $\mathcal{T}$, and small $\epsilon > 0$, if we consider the task of predicting labels based on n samples from $\mathcal{D}$ then:*

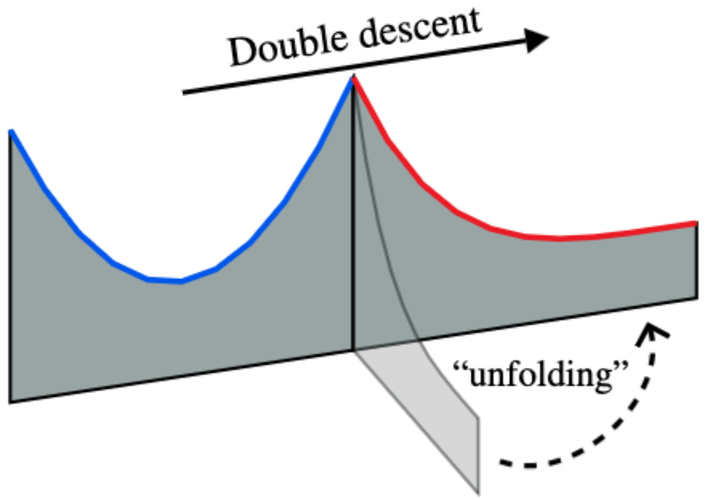
Under-paremeterized regime. *If $\mathrm{EMC}_{\mathcal{D},\epsilon}(\mathcal{T})$ is sufficiently smaller than n, any perturbation of $\mathcal{T}$ that increases its effective complexity will decrease the test error.*

Over-parameterized regime. *If $\mathrm{EMC}_{\mathcal{D},\epsilon}(\mathcal{T})$ is sufficiently larger than n, any perturbation of $\mathcal{T}$ that increases its effective complexity will decrease the test error.*

Critically parameterized regime. *If $\mathrm{EMC}_{\mathcal{D},\epsilon}(\mathcal{T}) \approx n$, then a perturbation of $\mathcal{T}$ that increases its effective complexity might decrease* **or increase** *the test error.*

Nakkiran et al. ICLR-20





- Claim: double descent isn't "really" about interpolation
  - For trees, gradient boosting: previous experiments start ensembling after the model interpolates (so you can keep adding parameters)
  - For linear regression: more subtle, but can view it that way too
  - Red regime actually *decreases* (one notion of) "effective" parameters

- This paper (October 2023) doesn't try to explain the neural setting

A U-turn on Double Descent: Rethinking Parameter Counting in Statistical Learning

Alicia Curth*
University of Cambridge
amc253@cam.ac.uk

Alan Jeffares*
University of Cambridge
aj659@cam.ac.uk

Mihaela van der Schaar
University of Cambridge
mv472@cam.ac.uk