

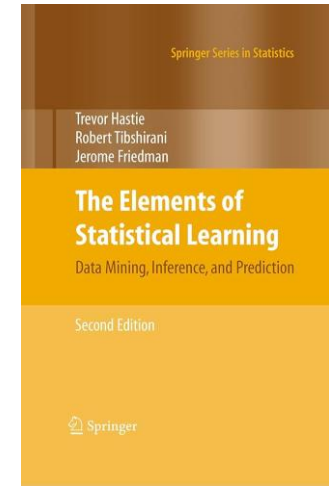
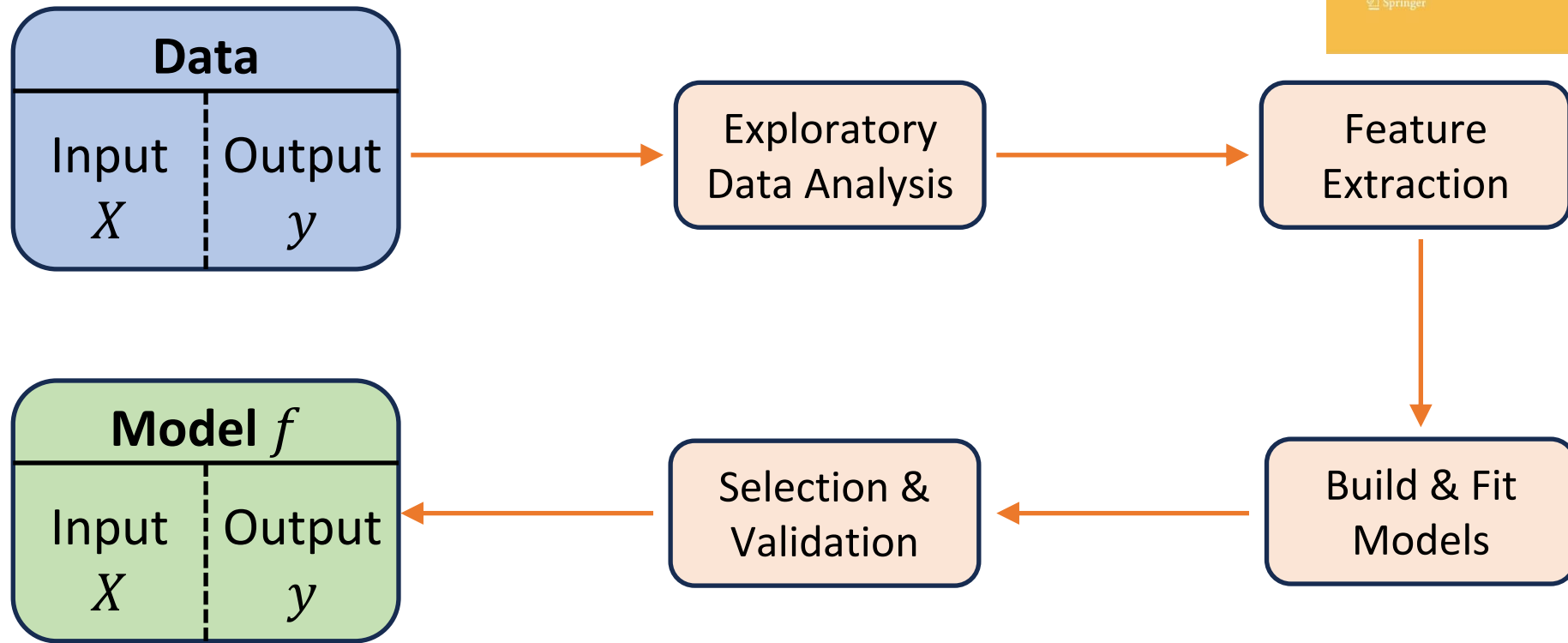
Double Descent: Occam's Razor's Edge?

LAI Reading Group

3 December 2025

Rui-Yang Zhang

Classical Statistical Learning



Cengage



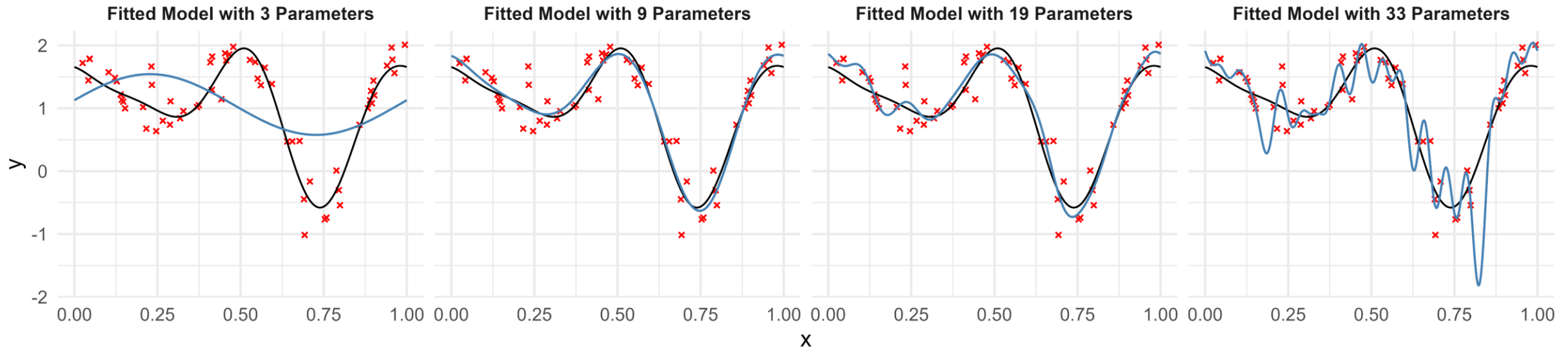
Statistical Inference

Second Edition

George Casella
Roger L. Berger

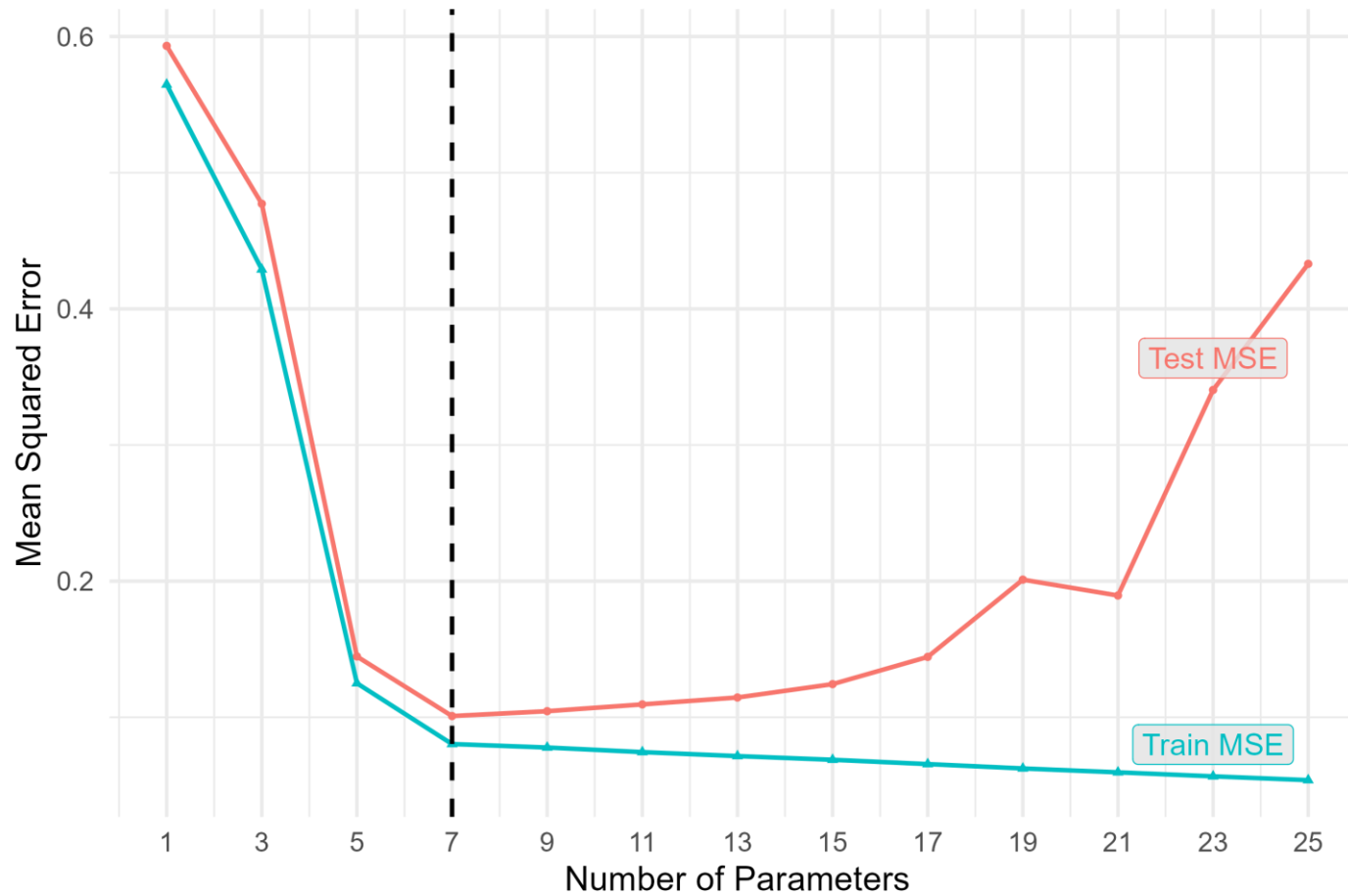
DUXBURY ADVANCED SERIES

Bias-Variance Trade-off

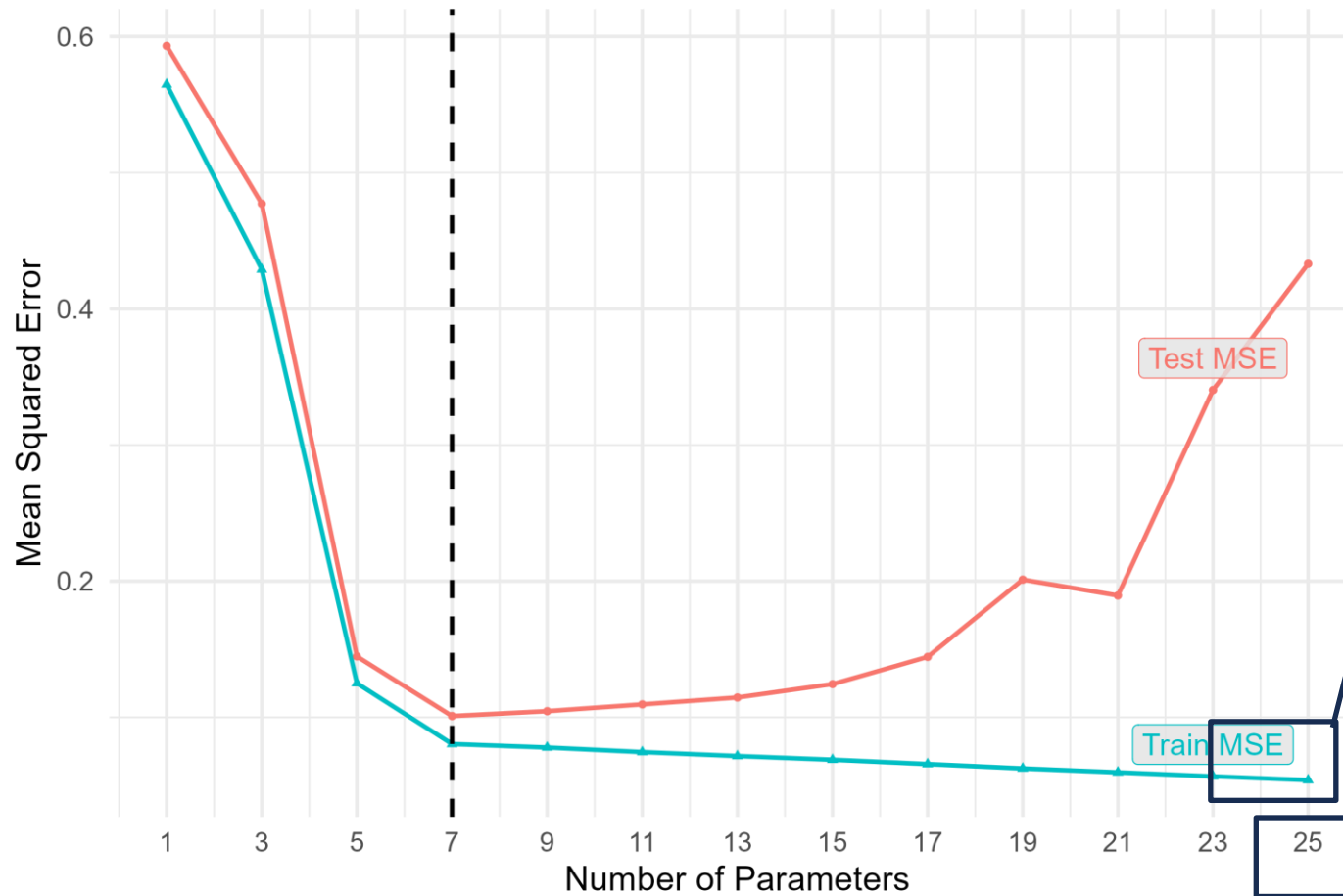


Fourier-basis model regressions with a varying number of bases.

Bias-Variance Trade-off



Bias-Variance Trade-off



Issue: Near-zero training loss.
Overfitting!

Issue: Too many parameters.
Over-parameterisation!

Model Information Criteria

Akaike Information Criterion (AIC)

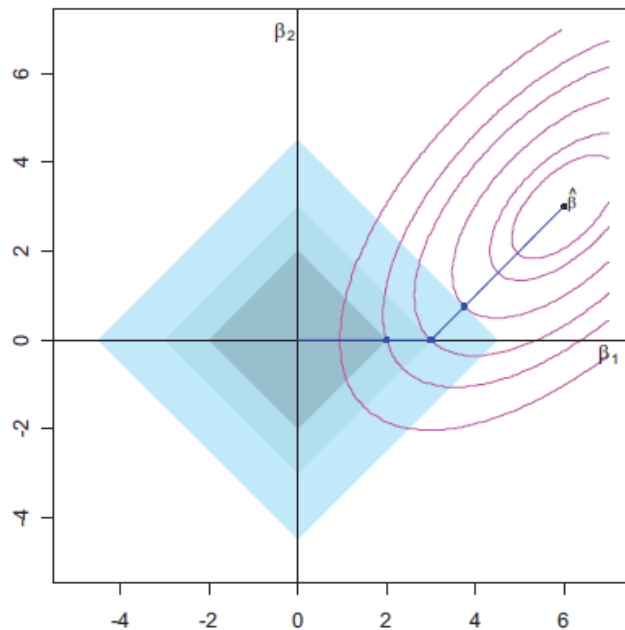
$$\text{AIC} = 2k - 2\log(L)$$

Bayesian Information Criterion (BIC)

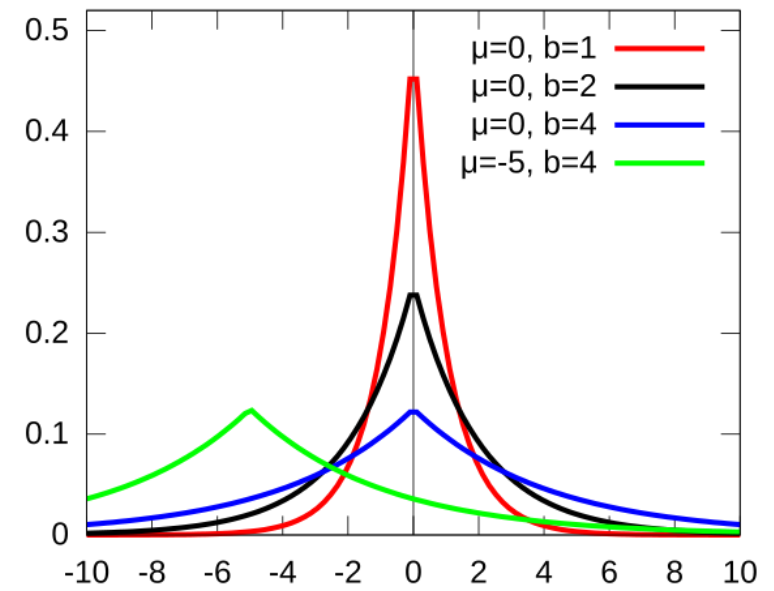
$$\text{BIC} = k\log(n) - 2\log(L)$$

Sparsity Regularisation

LASSO



Laplace Prior

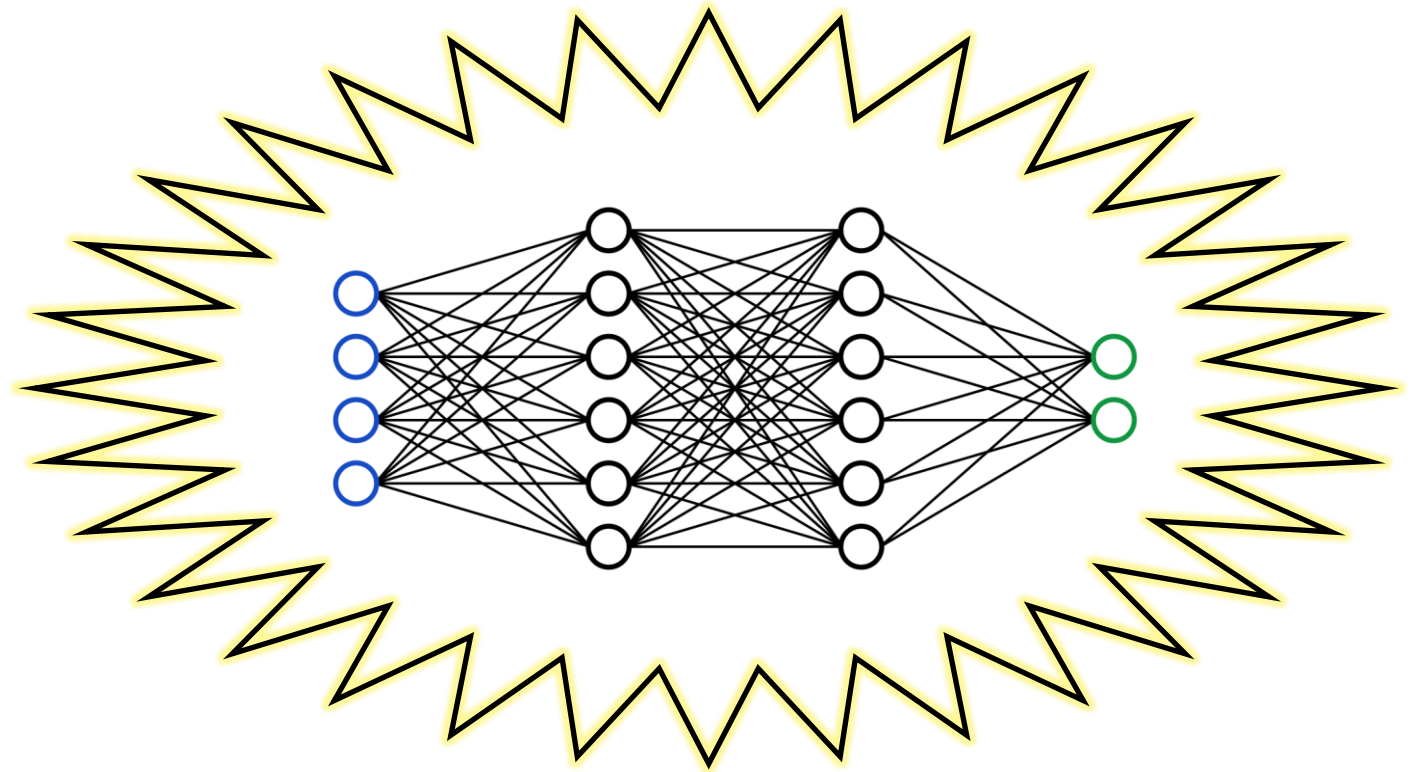
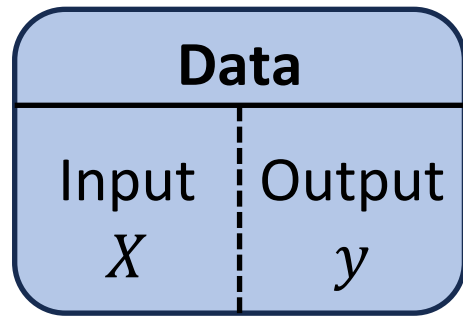


Occam's Razor

*“It is futile to do with more things
that which can be done with fewer.”*



“Modern” Statistical Learning

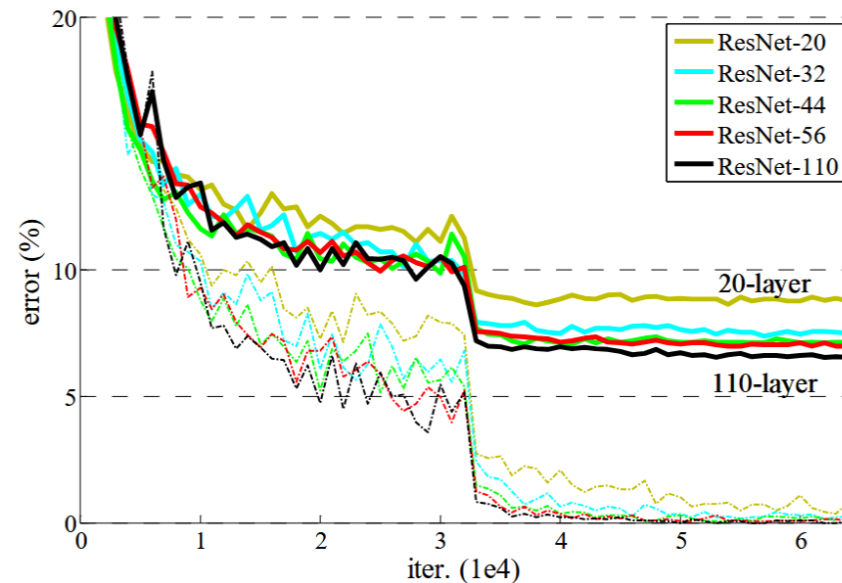


“Modern” Statistical Learning

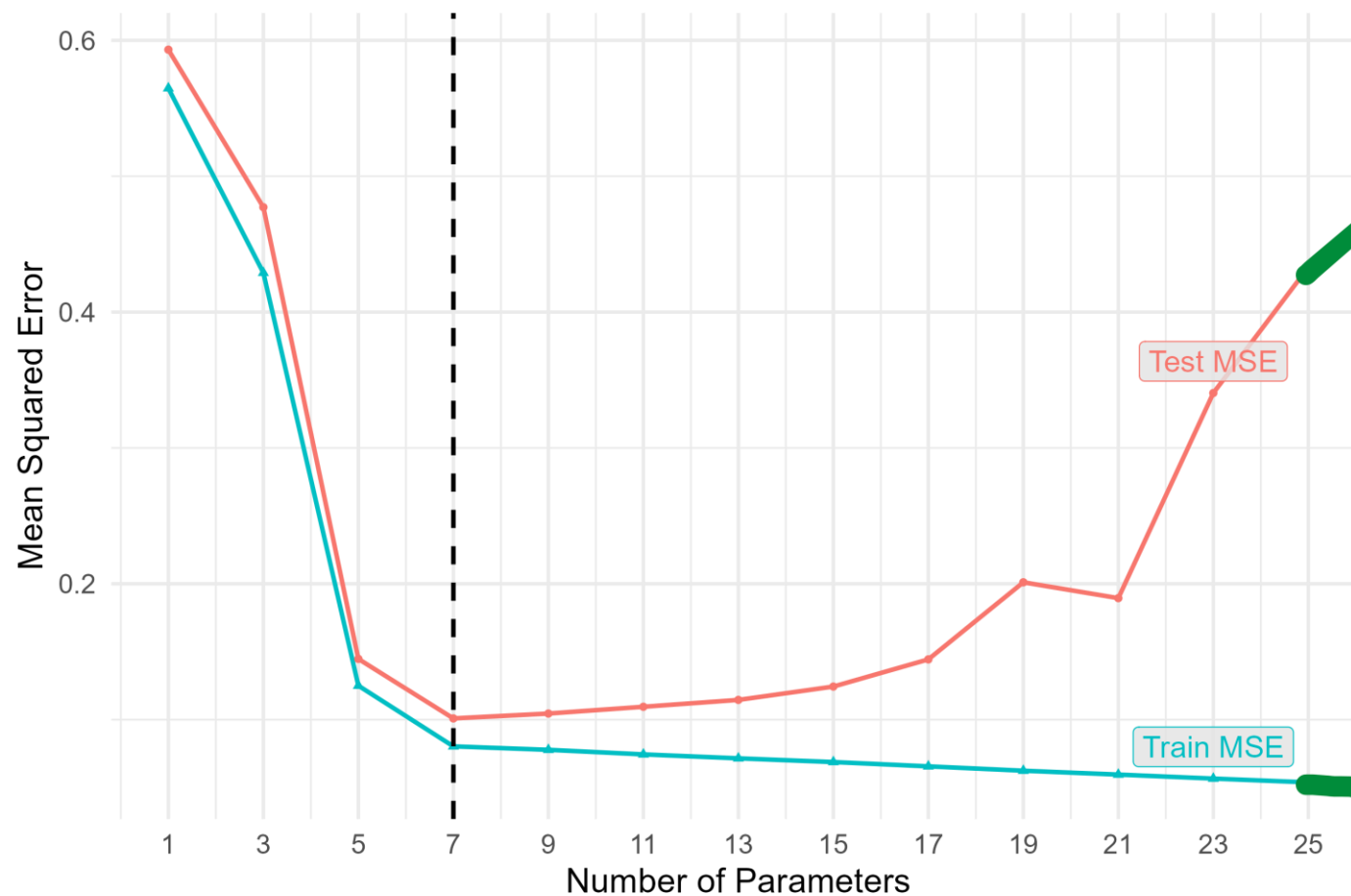
- Huge parameter number
 - ~10M for images (e.g. AlexNet has 60M, ResNet-50 has 25M)
 - ~100B for texts (e.g. GPT-3 has 175B, Llama 3.3 has 70B)
- Near-zero training loss

Over-parameterisation!

Overfitting!



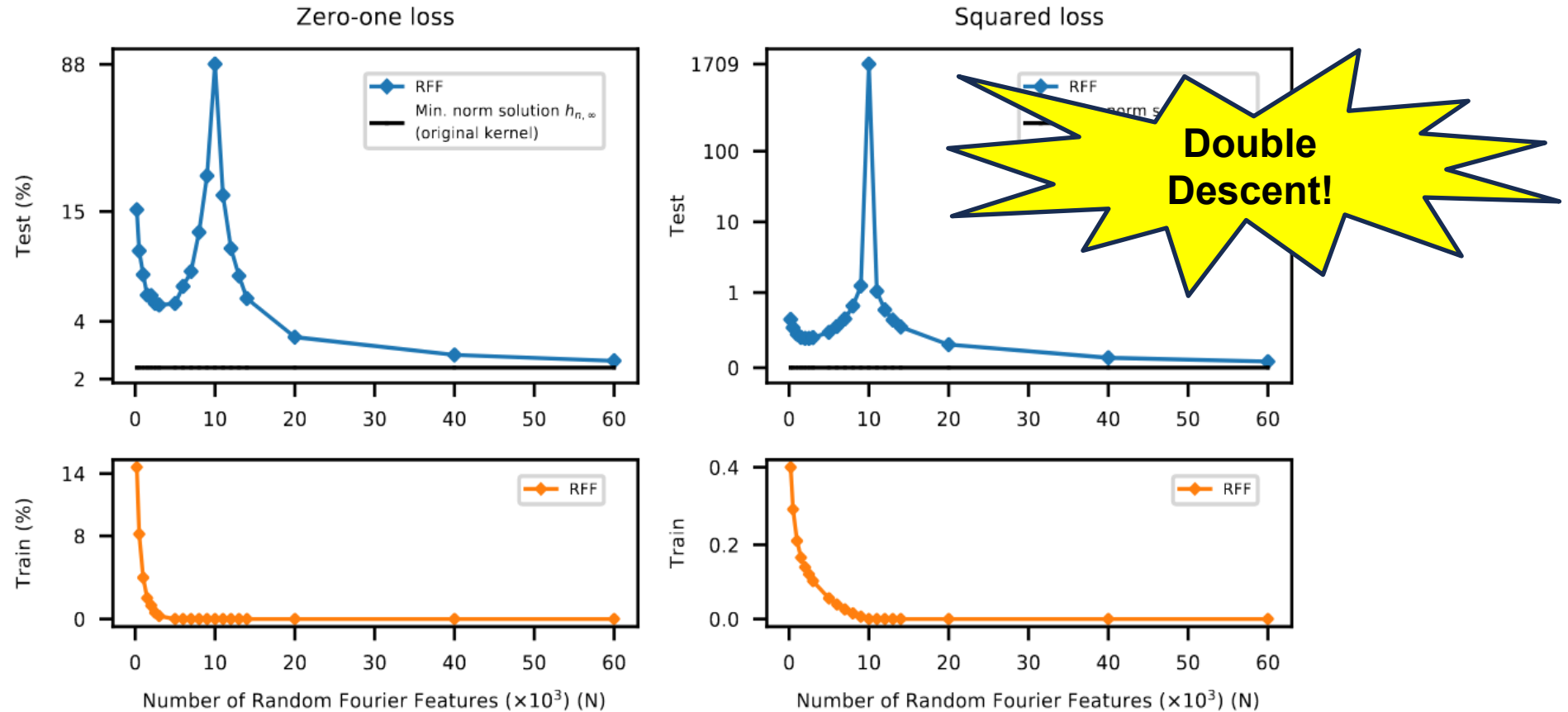
Bias-Variance Trade-off is Wrong?



Overfitting

Over-parameterisation

Bias-Variance Trade-off is Wrong?



Belkin, Mikhail, et al. "Reconciling modern machine-learning practice and the classical bias–variance trade-off." *Proceedings of the National Academy of Sciences* 116.32 (2019): 15849-15854.

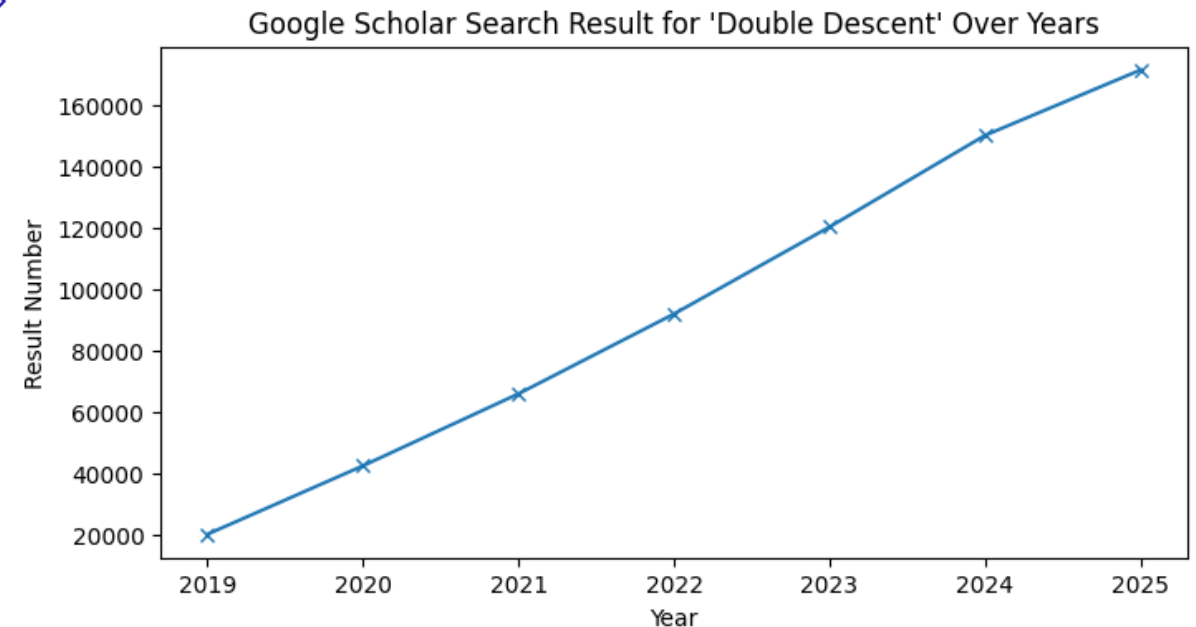
Double Descent

Reconciling modern machine-learning practice and the classical bias–variance trade-off

[M Belkin](#), [D Hsu](#), [S Ma](#), [S Mandal](#) - ... of the National Academy of Sciences, 2019 - pnas.org

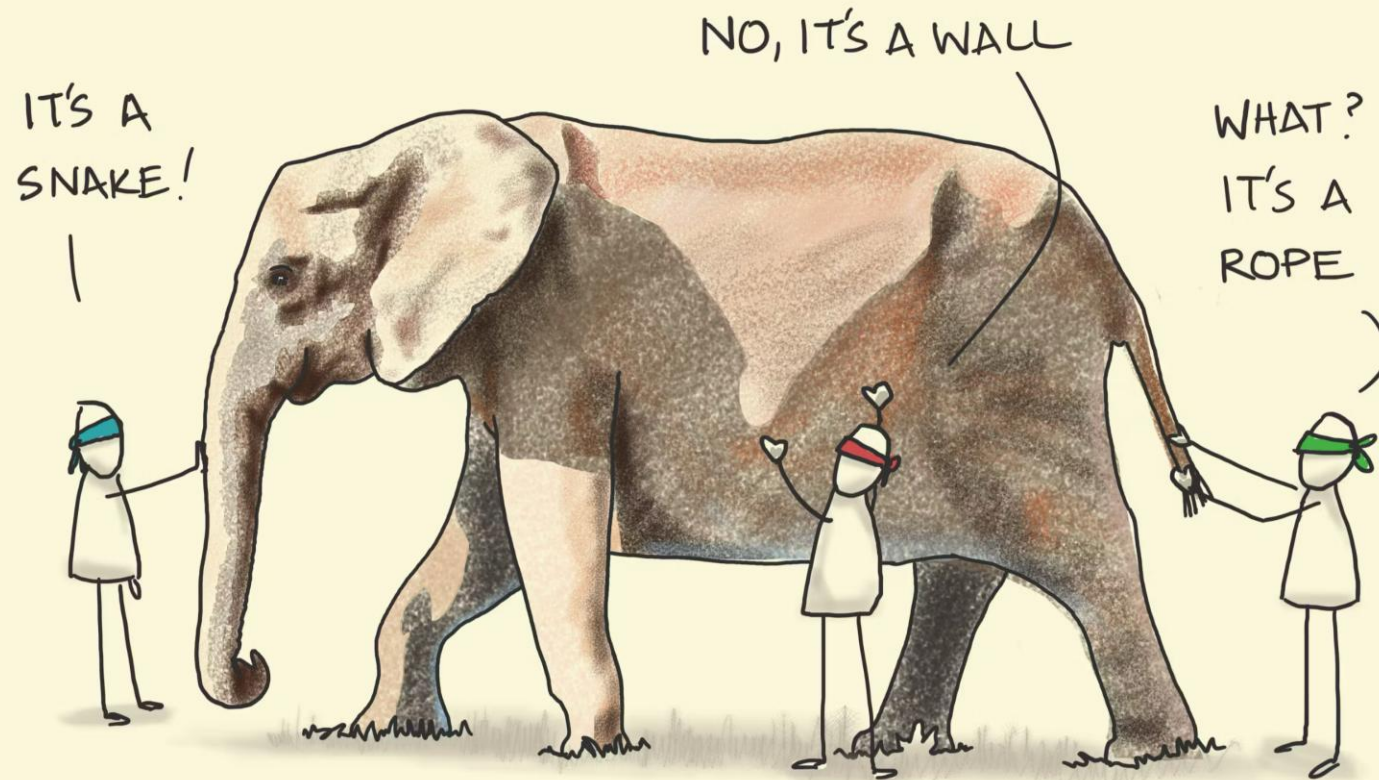
... of **machine learning** and their relevance to practitioners. In this paper, we **reconcile** the classical understanding and the **modern** ... the structure of **machine-learning** models delineates the ...

☆ Save  Cite Cited by 2615 Related articles All 13 versions 



THE BLIND AND THE ELEPHANT

OUR OWN EXPERIENCE IS RARELY THE WHOLE TRUTH



sketchplanations

Benign Overfitting in Linear Regression

[Submitted on 26 Jun 2019 (v1), last revised 29 Jan 2020 (this version, v3)]

Benign Overfitting in Linear Regression


Peter L. Bartlett, Philip M. Long, Gábor Lugosi, Alexander Tsigler

The phenomenon of benign overfitting is one of the key mysteries uncovered by deep learning methodology: deep neural networks seem to predict well, even with a perfect fit to noisy training data. Motivated by this phenomenon, we consider when a perfect fit to training data in linear regression is compatible with accurate prediction. We give a characterization of linear regression problems for which the minimum norm interpolating prediction rule has near-optimal prediction accuracy. The characterization is in terms of two notions of the effective rank of the data covariance. It shows that overparameterization is essential for benign overfitting in this setting: the number of directions in parameter space that are unimportant for prediction must significantly exceed the sample size. By studying examples of data covariance properties that this characterization shows are required for benign overfitting, we find an important role for finite-dimensional data: the accuracy of the minimum norm interpolating prediction rule approaches the best possible accuracy for a much narrower range of properties of the data distribution when the data lies in an infinite dimensional space versus when the data lies in a finite dimensional space whose dimension grows faster than the sample size.

Subjects: **Machine Learning (stat.ML)**; Machine Learning (cs.LG); Statistics Theory (math.ST)

Cite as: [arXiv:1906.11300](https://arxiv.org/abs/1906.11300) [stat.ML]
(or [arXiv:1906.11300v3](https://arxiv.org/abs/1906.11300v3) [stat.ML] for this version)

<https://doi.org/10.48550/arXiv.1906.11300> 

Related DOI: <https://doi.org/10.1073/pnas.1907378117> 

Benign Overfitting in Linear Regression

Definition 1 (Linear regression). *A linear regression problem in a separable Hilbert space \mathbb{H} is defined by a random covariate vector $x \in \mathbb{H}$ and outcome $y \in \mathbb{R}$. We define*

1. *the covariance operator $\Sigma = \mathbb{E}[xx^\top]$, and*
2. *the optimal parameter vector $\theta^* \in \mathbb{H}$, satisfying $\mathbb{E}(y - x^\top \theta^*)^2 = \min_{\theta} \mathbb{E}(y - x^\top \theta)^2$.*

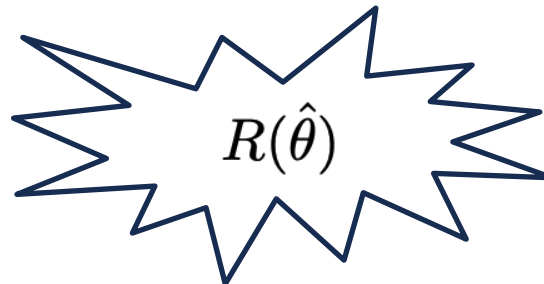
Given a training sample $(x_1, y_1), \dots, (x_n, y_n)$ of n i.i.d. pairs with the same distribution as (x, y) , an estimator returns a parameter estimate $\theta \in \mathbb{H}$. The excess risk of the estimator is defined as

$$R(\theta) := \mathbb{E}_{x,y} \left[\left(y - x^\top \theta \right)^2 - \left(y - x^\top \theta^* \right)^2 \right],$$

Benign Overfitting in Linear Regression

$$\begin{aligned}\hat{\theta} &= \arg \min_{\theta} \left\{ \|\theta\|^2 : X^\top X \theta = X^\top \mathbf{y} \right\} \\ &= \left(X^\top X \right)^\dagger X^\top \mathbf{y} \\ &= X^\top \left(X X^\top \right)^\dagger \mathbf{y},\end{aligned}$$

$(X^\top X)^\dagger$ denotes the pseudoinverse of the bounded linear operator $X^\top X$



Benign Overfitting in Linear Regression

Roughly speaking, we have:

$$c_1 \left[\frac{k^*}{n} + \frac{n}{R_{k^*}(\Sigma)} \right] \leq R(\hat{\theta}) \leq c_2 \left[\frac{k^*}{n} + \frac{n}{R_{k^*}(\Sigma)} \right] + c_3 \left[\frac{r_0(\Sigma)}{n} \right]$$

Definition 3 (Effective Ranks). For the covariance operator Σ , define $\lambda_i = \mu_i(\Sigma)$ for $i = 1, 2, \dots$. If $\sum_{i=1}^{\infty} \lambda_i < \infty$ and $\lambda_{k+1} > 0$ for $k \geq 0$, define

$$r_k(\Sigma) = \frac{\sum_{i>k} \lambda_i}{\lambda_{k+1}},$$

$$R_k(\Sigma) = \frac{(\sum_{i>k} \lambda_i)^2}{\sum_{i>k} \lambda_i^2}.$$

Benign Overfitting in Linear Regression

Roughly speaking, we have:

$$c_1 \left[\frac{k^*}{n} + \frac{n}{R_{k^*}(\Sigma)} \right] \leq R(\hat{\theta}) \leq c_2 \left[\frac{k^*}{n} + \frac{n}{R_{k^*}(\Sigma)} \right] + c_3 \left[\frac{r_0(\Sigma)}{n} \right]$$

For benign overfitting, we want $\frac{k^*}{n}, \frac{n}{R_{k^*}(\Sigma)}, \frac{r_0(\Sigma)}{n} \rightarrow 0$

Benign Overfitting in Linear Regression

If

$$\mu_k(\Sigma_n) = \begin{cases} \gamma_k + \epsilon_n & \text{if } k \leq p_n, \\ 0 & \text{otherwise,} \end{cases}$$

and $\gamma_k = \Theta(\exp(-k/\tau))$, then Σ_n is benign iff $p_n = \omega(n)$ and $ne^{-o(n)} = \epsilon_n p_n = o(n)$. Furthermore, for $p_n = \Omega(n)$ and $\epsilon_n p_n = ne^{-o(n)}$,

$$R(\hat{\theta}) = O\left(\frac{\epsilon_n p_n + 1}{n} + \frac{\ln(n/(\epsilon_n p_n))}{n} + \max\left\{\frac{1}{n}, \frac{n}{p_n}\right\}\right).$$

covariance operator decay just slowly enough for their sum to remain finite. Part 2 shows that the situation is very different if the data has finite dimension and a small amount of isotropic noise is added to the covariates. In that case, even if the eigenvalues of the original covariance operator (before the addition of isotropic noise) decay very rapidly, benign overfitting occurs iff both the dimension is large compared to the sample size, and the isotropic component of the covariance is sufficiently small—but not exponentially small—compared to the sample size.

Benign Overfitting in Linear Regression

- How do covariance eigenvalue properties translate to data properties?
- Is data noise essential in practice?
- How well do these results extrapolate beyond linear regression?

Empirical Observations and Doubts

- Train till interpolation (zero train loss) is not always improving performance ... sometimes early stopping in training is desirable.
- The optimizer choice seems to make a difference – implicit regularization.
- More data is usually better, but not always.

Deep Double Descent

[Submitted on 4 Dec 2019]

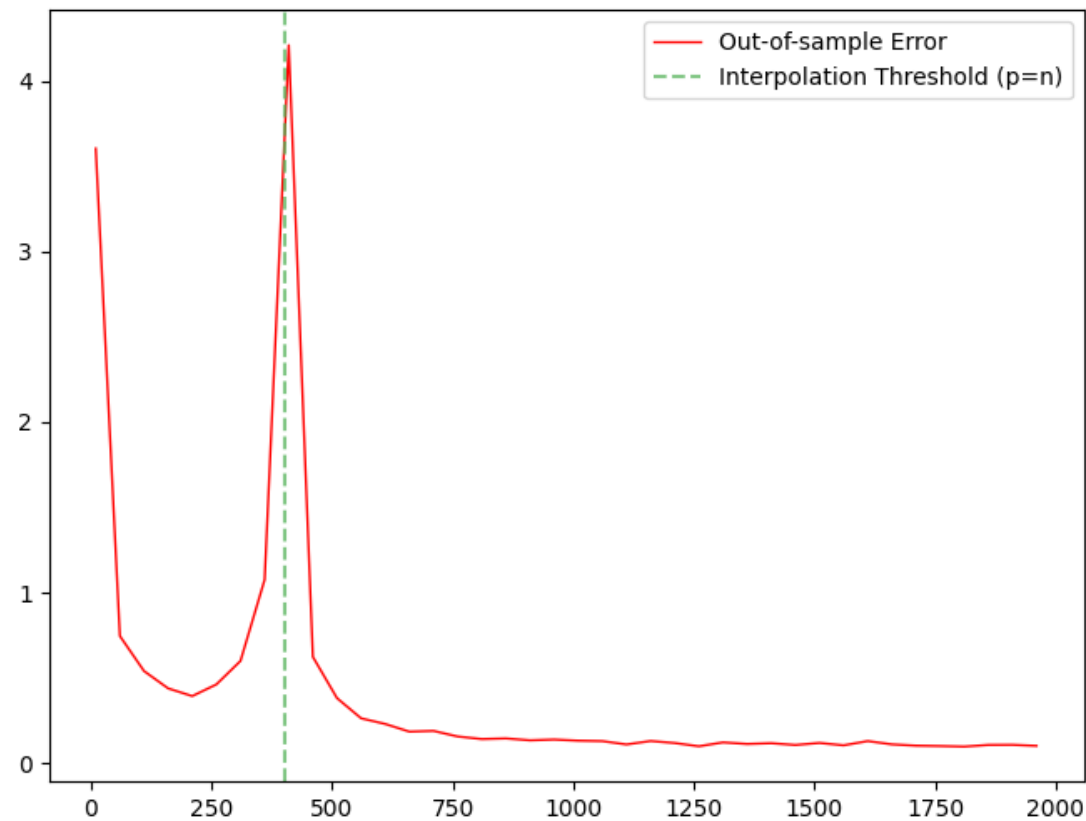
Deep Double Descent: Where Bigger Models and More Data Hurt

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, Ilya Sutskever

We show that a variety of modern deep learning tasks exhibit a "double-descent" phenomenon where, as we increase model size, performance first gets worse and then gets better. Moreover, we show that double descent occurs not just as a function of model size, but also as a function of the number of training epochs. We unify the above phenomena by defining a new complexity measure we call the effective model complexity and conjecture a generalized double descent with respect to this measure. Furthermore, our notion of model complexity allows us to identify certain regimes where increasing (even quadrupling) the number of train samples actually hurts test performance.

Deep Double Descent

Test Error



**Effective Model
Complexity**

Deep Double Descent

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:*

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

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

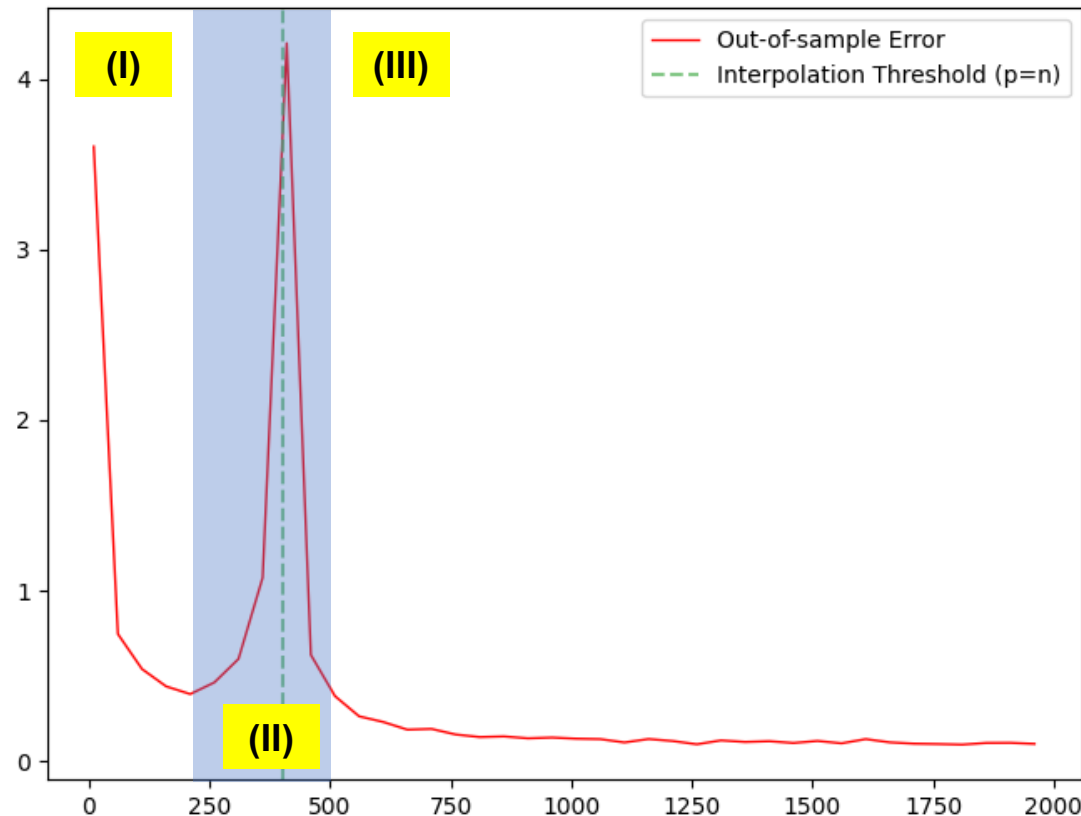
Training procedure = Parameter Number + Optimizer + Training Time + ...

e.g. linear model with L2 loss has tiny EMC (≈ 2)

One can change \mathcal{T} in many ways ...

Deep Double Descent

Test Error



(I) Under-Parametrised

Improve EMC always reduces test error

(II) Critically Parametrised

Improve EMC increases / reduces test error

(III) Over Parametrised

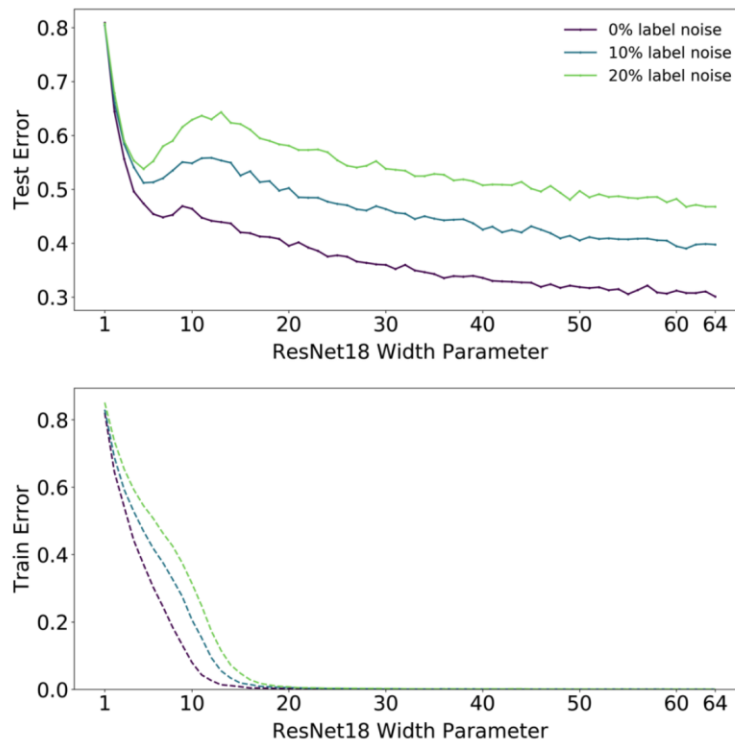
Improve EMC always reduces test error

Effective Model
Complexity

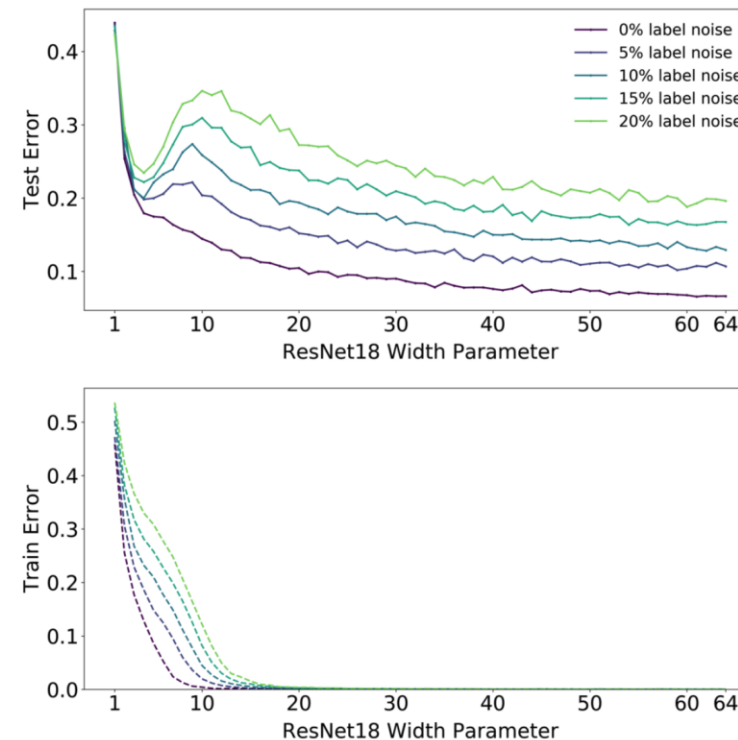
Deep Double Descent

- Changing Model Size
- Changing Training Epochs
- Changing Sample Number

Model-Wise Double Descent



(a) **CIFAR-100.** There is a peak in test error even with no label noise.



(b) **CIFAR-10.** There is a “plateau” in test error around the interpolation point with no label noise, which develops into a peak for added label noise.

Figure 4: **Model-wise double descent for ResNet18s.** Trained on CIFAR-100 and CIFAR-10, with varying label noise. Optimized using Adam with LR 0.0001 for 4K epochs, and data-augmentation.

Model-Wise Double Descent

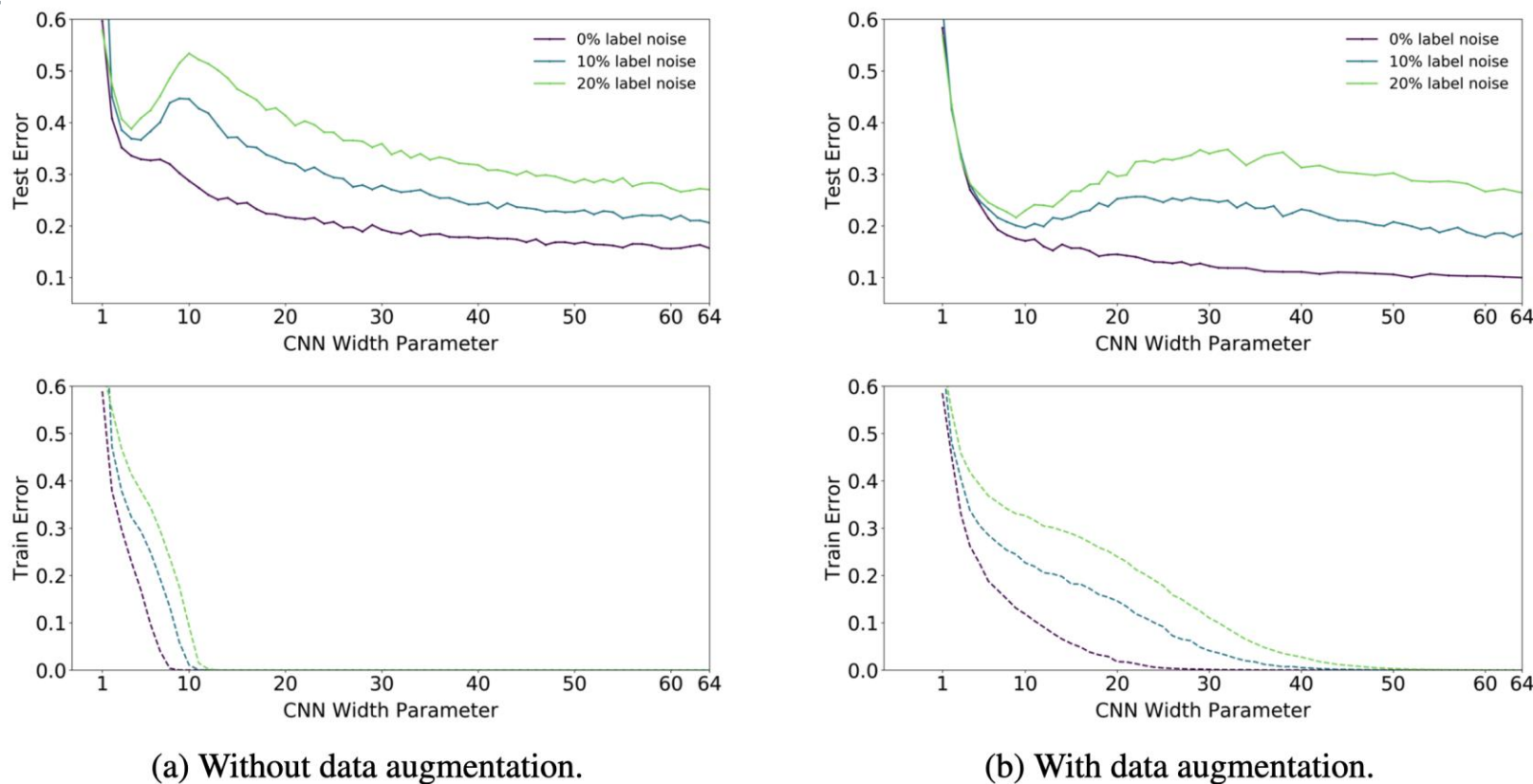


Figure 5: **Effect of Data Augmentation.** 5-layer CNNs on CIFAR10, with and without data-augmentation. Data-augmentation shifts the interpolation threshold to the right, shifting the test error peak accordingly. Optimized using SGD for 500K steps. See Figure 27 for larger models.

Model-Wise Double Descent

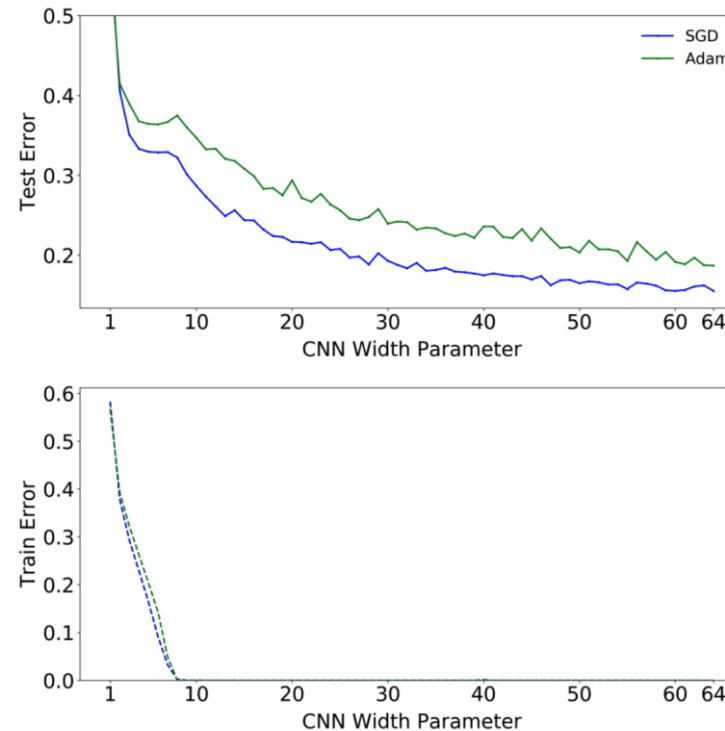


Figure 6: **SGD vs. Adam.** 5-Layer CNNs on CIFAR-10 with no label noise, and no data augmentation. Optimized using SGD for 500K gradient steps, and Adam for 4K epochs.

Model-Wise Double Descent

E.2.3 EARLY STOPPING DOES NOT EXHIBIT DOUBLE DESCENT

Language models

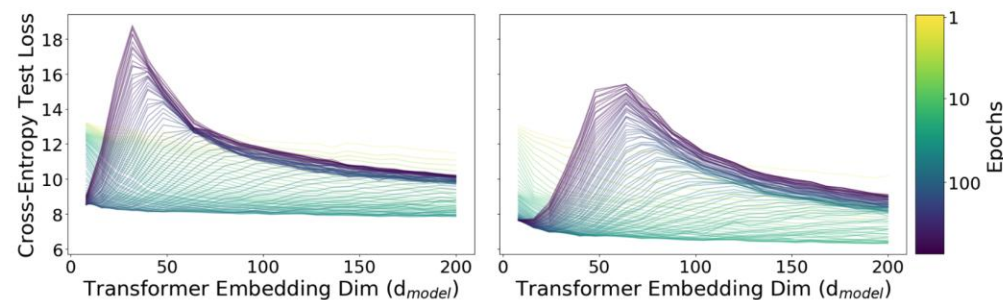


Figure 23: Model-wise test error dynamics for a subsampled IWSLT'14 dataset. Left: 4k samples, Right: 18k samples. Note that with optimal early-stopping, more samples is always better.

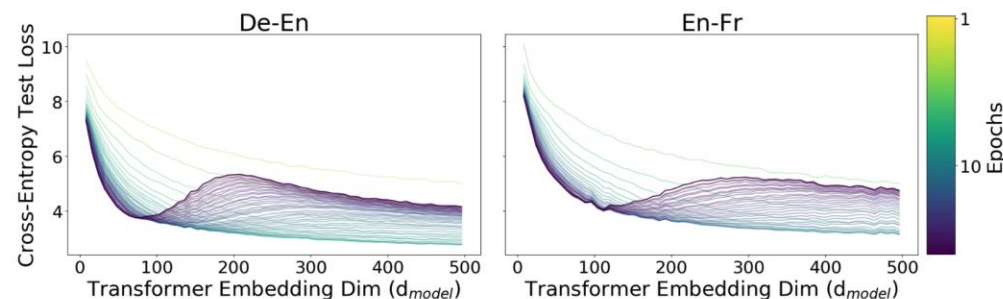


Figure 24: Model-wise test error dynamics for a IWSLT'14 de-en and subsampled WMT'14 en-fr datasets. **Left:** IWSLT'14, **Right:** subsampled (200k samples) WMT'14. Note that with optimal early-stopping, the test error is much lower for this task.

Model-Wise Double Descent

- All modifications which increase the interpolation threshold (such as adding label noise, using data augmentation) also correspondingly shift the peak in test error towards larger models.
- For model-sizes at the interpolation threshold, there is effectively only one model that fits the train data and this interpolating model is very sensitive to noise in the train set and/or model mis-specification
- For over-parameterized models, there are many interpolating models that fit the train set, and SGD is able to find one that “memorizes” (or “absorbs”) the noise while still performing well on the distribution.

Epoch-Wise Double Descent

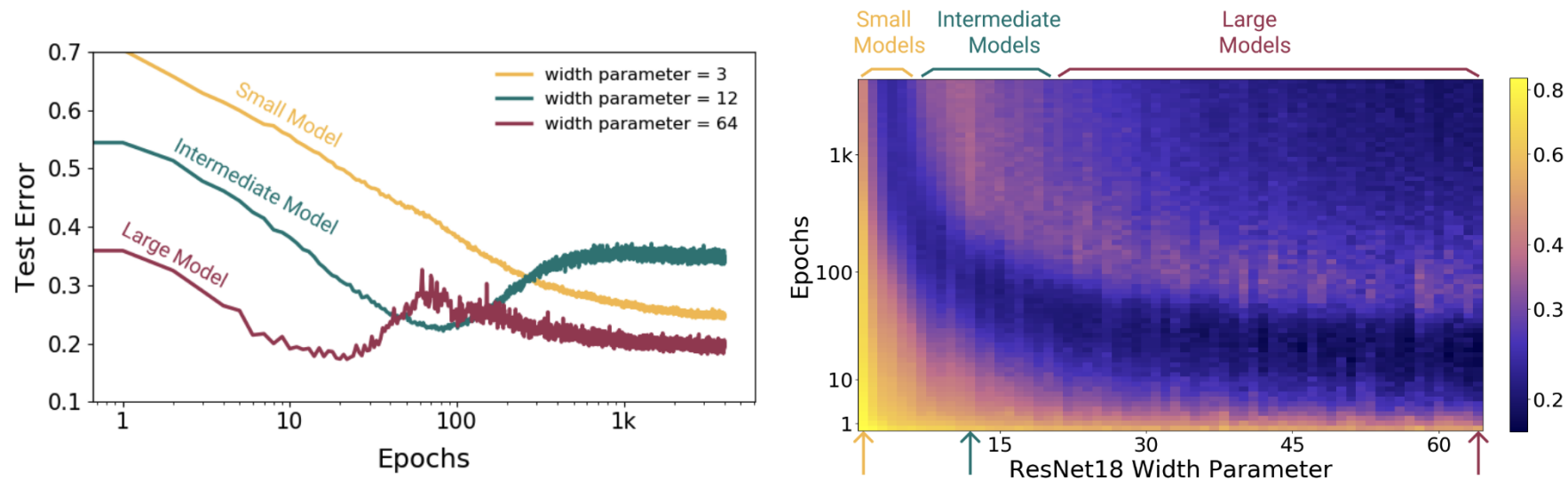
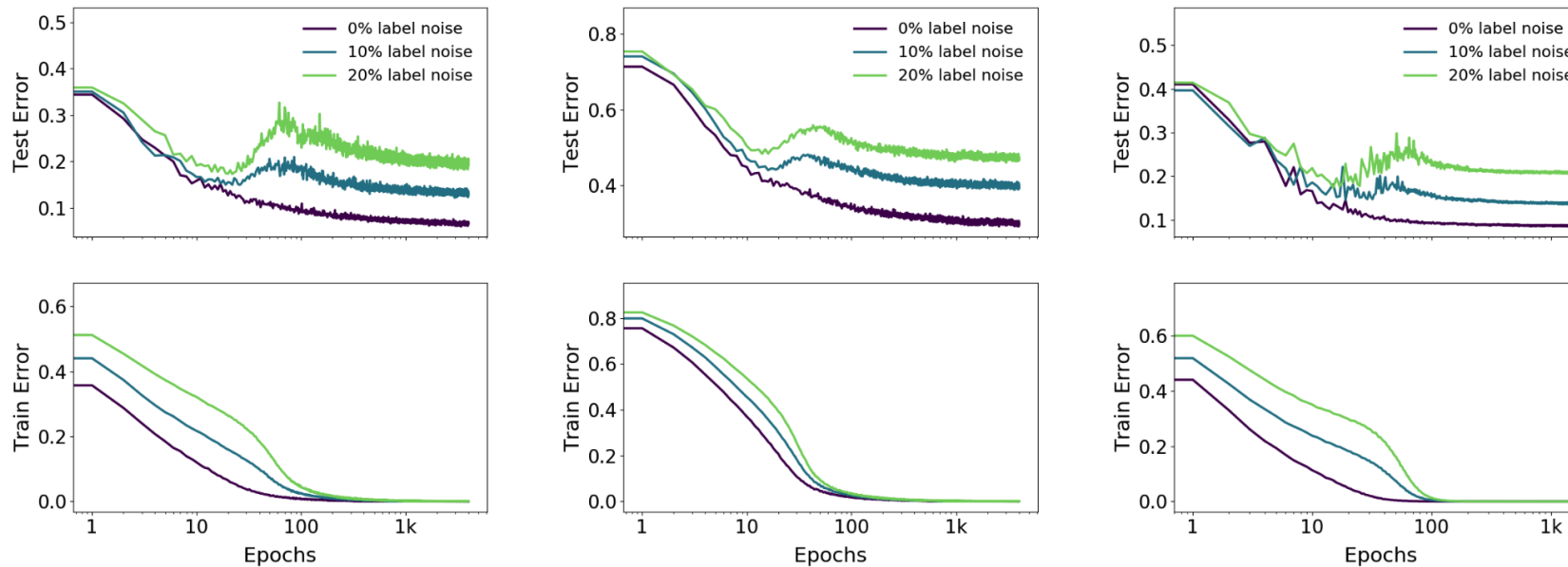


Figure 9: **Left:** Training dynamics for models in three regimes. Models are ResNet18s on CIFAR10 with 20% label noise, trained using Adam with learning rate 0.0001, and data augmentation. **Right:** Test error over (Model size \times Epochs). Three slices of this plot are shown on the left.

Epoch-Wise Double Descent



(a) ResNet18 on CIFAR10.

(b) ResNet18 on CIFAR100.

(c) 5-layer CNN on CIFAR 10.

Figure 10: **Epoch-wise double descent** for ResNet18 and CNN (width=128). ResNets trained using Adam with learning rate 0.0001, and CNNs trained with SGD with inverse-squareroot learning rate.

Epoch-Wise Double Descent

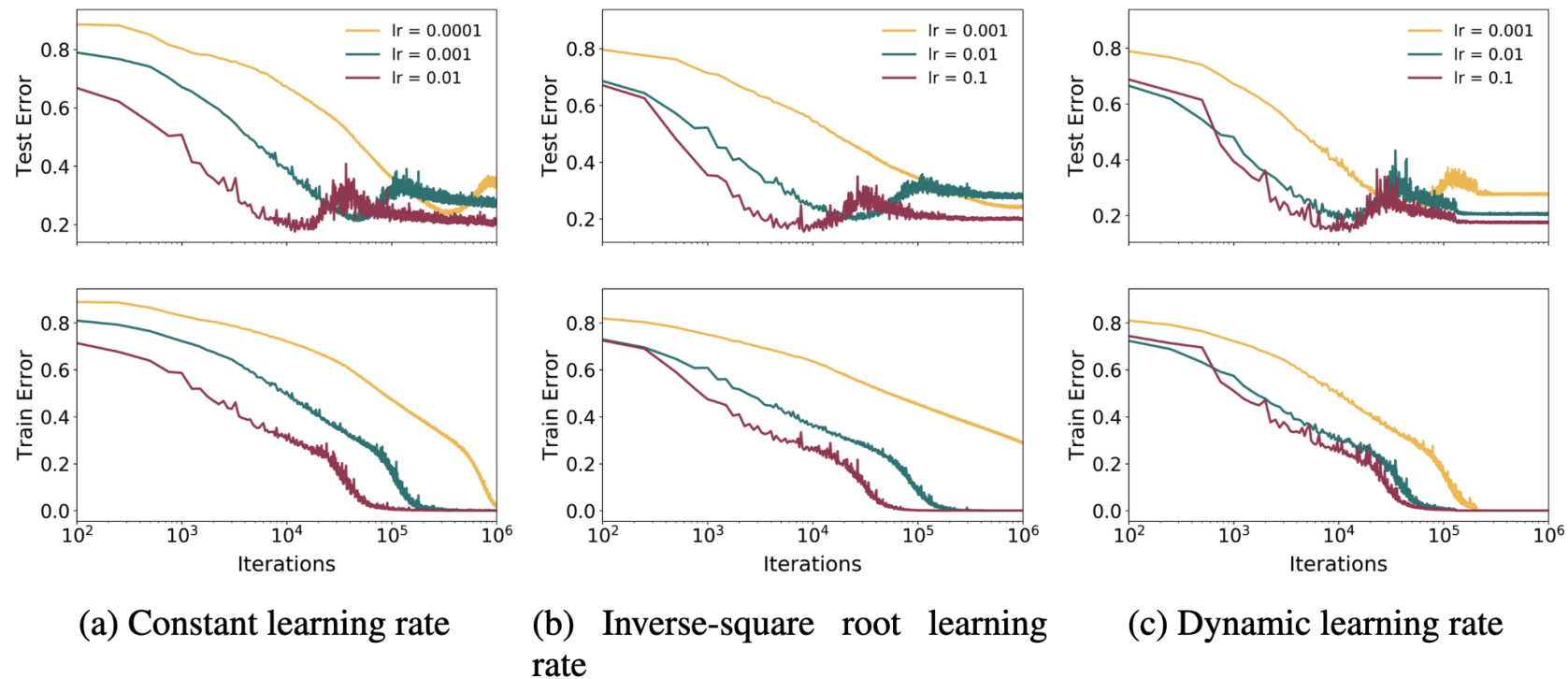
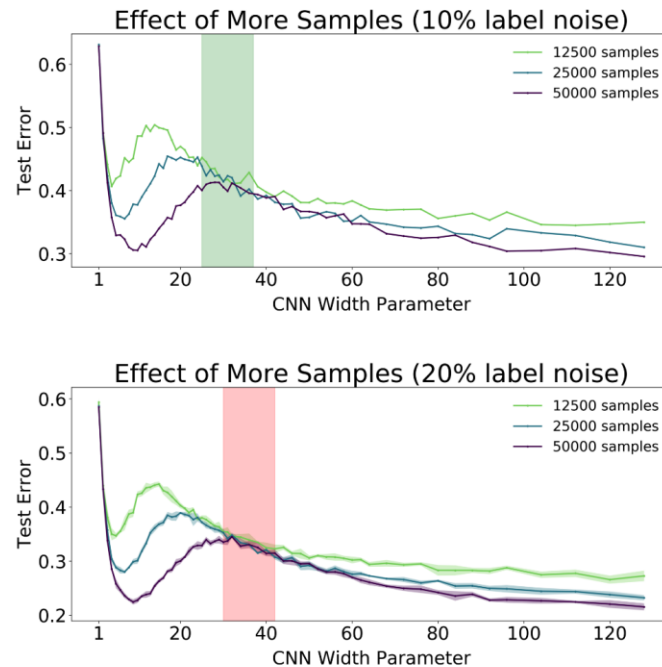


Figure 17: **Epoch-wise double descent** for ResNet18 trained with SGD and multiple learning rate schedules

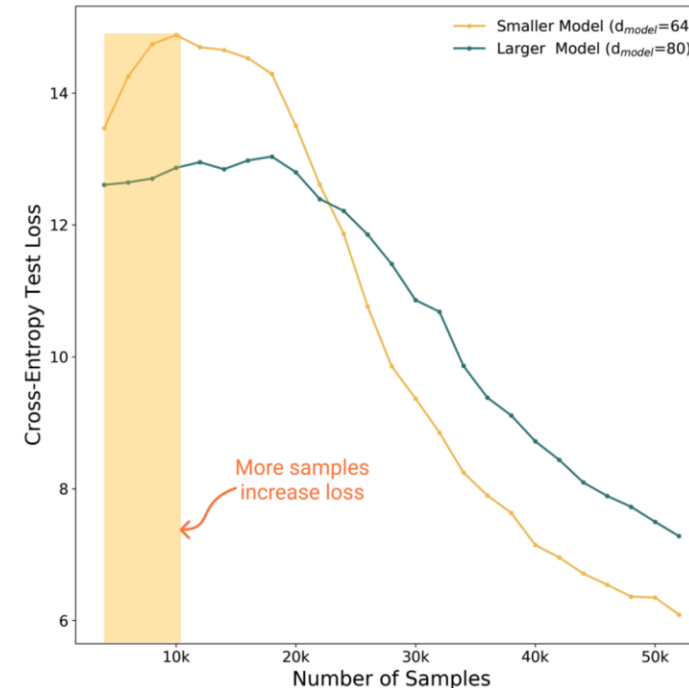
Epoch-Wise Double Descent

- Sufficiently large models can undergo a “double descent” behavior where test error first decreases then increases near the interpolation threshold, and then decreases again
- In contrast, for “medium sized” models, for which training to completion will only barely reach ≈ 0 error, the test error as a function of training time will follow a classical U-like curve where it is better to stop early.
- Models that are too small to reach the approximation threshold will remain in the “under parameterized” regime where increasing train time monotonically decreases test error.
- Further, this phenomenon is robust across optimizer variations and learning rate schedules

Sample-Wise Non-Monotonicity



(a) Model-wise double descent for 5-layer CNNs on CIFAR-10, for varying dataset sizes. **Top:** There is a range of model sizes (shaded green) where training on $2\times$ more samples does not improve test error. **Bottom:** There is a range of model sizes (shaded red) where training on $4\times$ more samples does not improve test error.



(b) **Sample-wise non-monotonicity.** Test loss (per-word perplexity) as a function of number of train samples, for two transformer models trained to completion on IWSLT'14. For both model sizes, there is a regime where more samples hurt performance. Compare to Figure 3, of model-wise double-descent in the identical setting.

Sample-Wise Non-Monotonicity

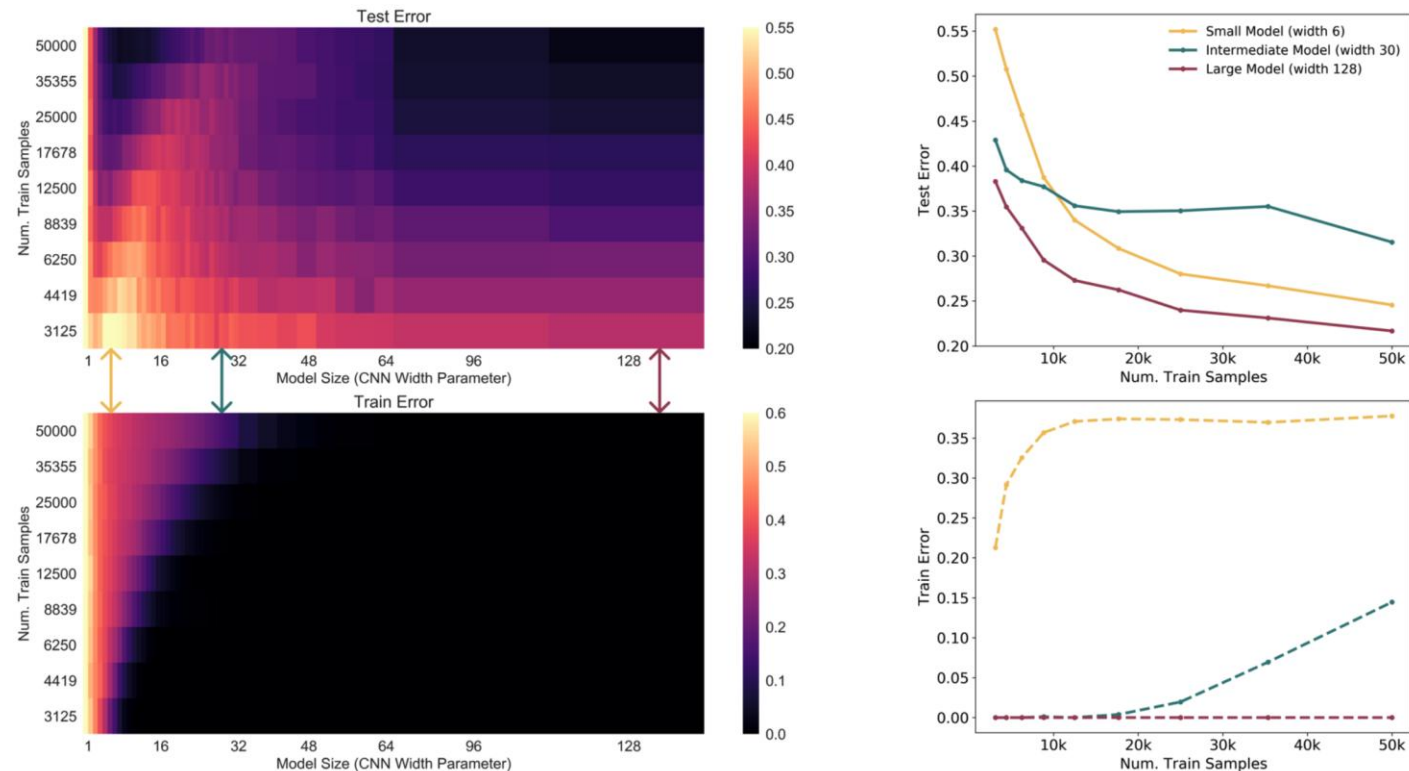


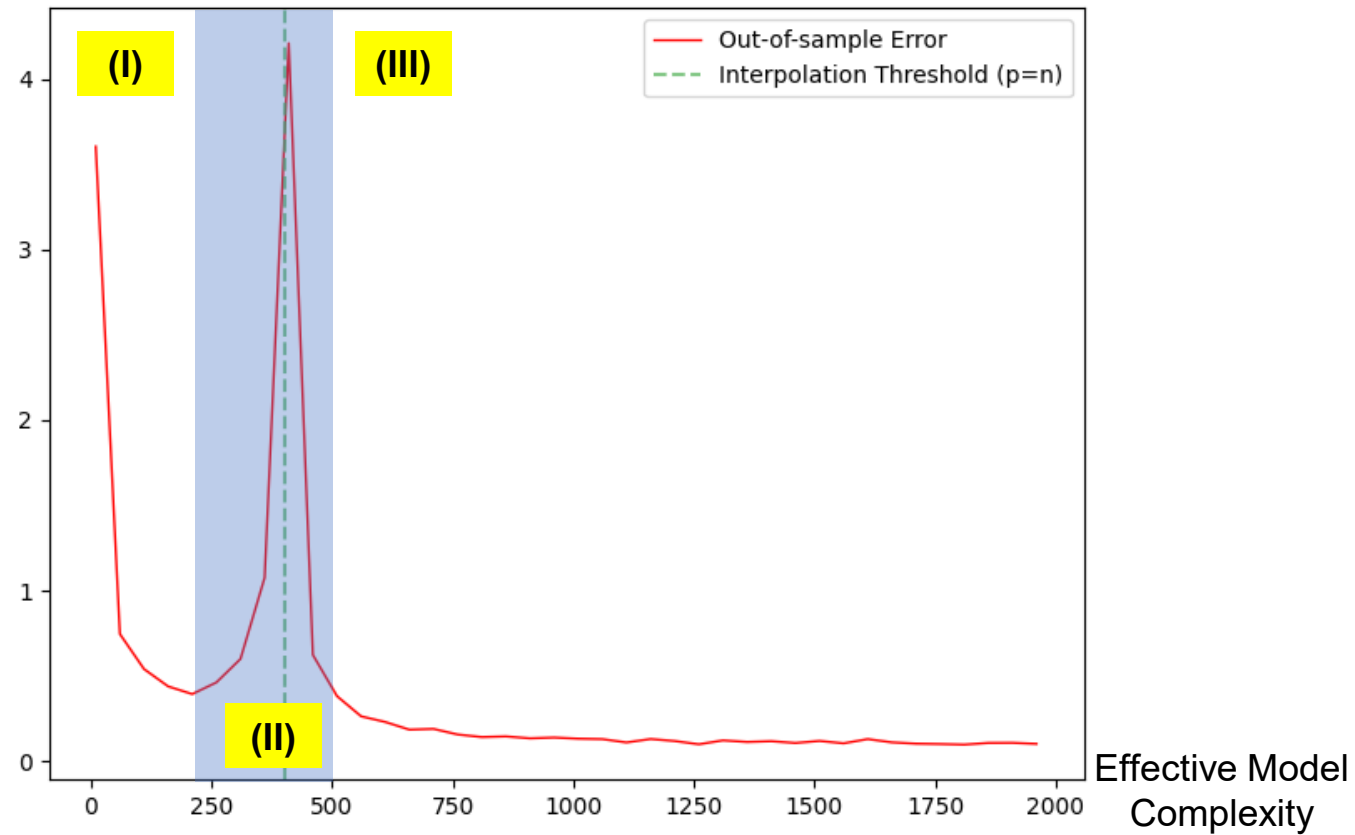
Figure 12: **Left:** Test Error as a function of model size and number of train samples, for 5-layer CNNs on CIFAR-10 + 20% noise. Note the ridge of high test error again lies along the interpolation threshold. **Right:** Three slices of the left plot, showing the effect of more data for models of different sizes. Note that, when training to completion, more data helps for small and large models, but does not help for near-critically-parameterized models (green).

Sample-Wise Non-Monotonicity

- By increasing n , the same training procedure T can switch from being effectively over-parameterized to effectively under-parameterized.
- On the one hand, (as expected) increasing the number of samples shrinks the area under the curve.
- On the other hand, increasing the number of samples also has the effect of “shifting the curve to the right” and increasing the model complexity at which test error peaks.
- There is a range of model sizes where the effects “cancel out”—and having more train samples does not help test performance when training to completion. Outside the critically-parameterized regime, for sufficiently under- or overparameterized models, having more samples helps.

Deep Double Descent

Test Error

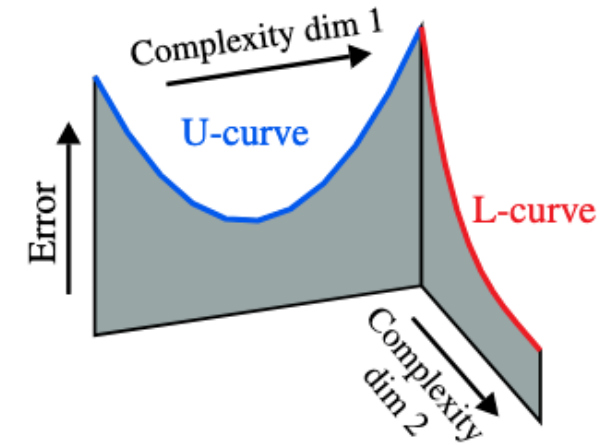
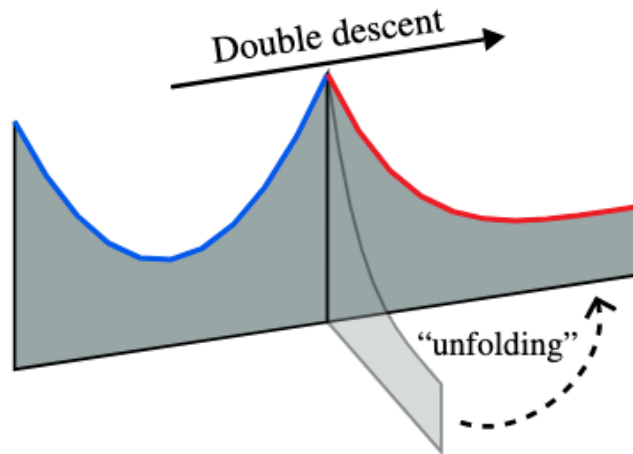


Occam's Razor

*“It is futile to do with more things
that which can be done with fewer.”*



A U-turn on Double Descent



A U-turn on Double Descent

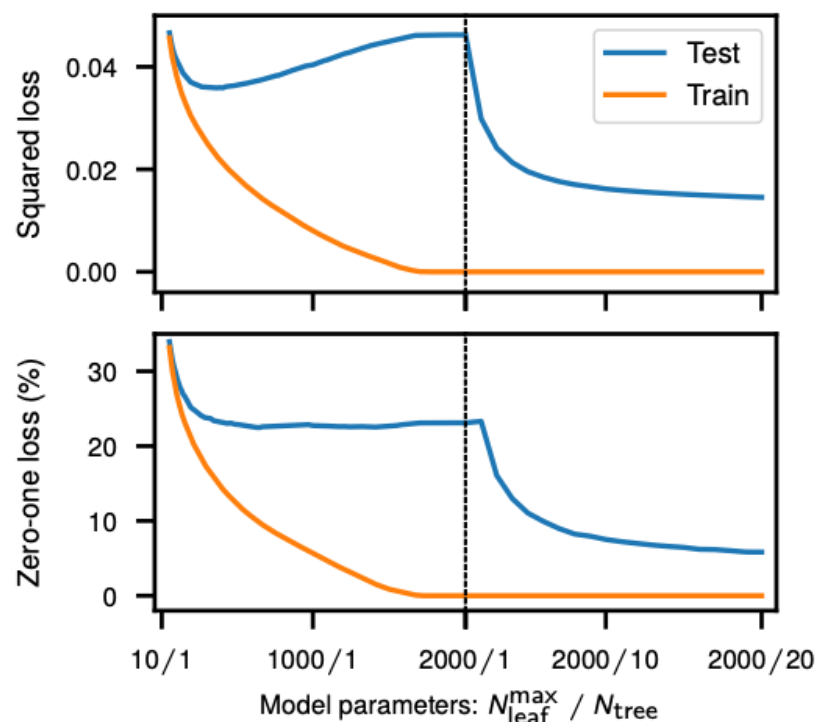
[Submitted on 29 Oct 2023]

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

Alicia Curth, Alan Jeffares, Mihaela van der Schaar

Conventional statistical wisdom established a well-understood relationship between model complexity and prediction error, typically presented as a U-shaped curve reflecting a transition between under- and overfitting regimes. However, motivated by the success of overparametrized neural networks, recent influential work has suggested this theory to be generally incomplete, introducing an additional regime that exhibits a second descent in test error as the parameter count p grows past sample size n – a phenomenon dubbed double descent. While most attention has naturally been given to the deep-learning setting, double descent was shown to emerge more generally across non-neural models: known cases include linear regression, trees, and boosting. In this work, we take a closer look at evidence surrounding these more classical statistical machine learning methods and challenge the claim that observed cases of double descent truly extend the limits of a traditional U-shaped complexity-generalization curve therein. We show that once careful consideration is given to what is being plotted on the x-axes of their double descent plots, it becomes apparent that there are implicitly multiple complexity axes along which the parameter count grows. We demonstrate that the second descent appears exactly (and only) when and where the transition between these underlying axes occurs, and that its location is thus not inherently tied to the interpolation threshold $p=n$. We then gain further insight by adopting a classical nonparametric statistics perspective. We interpret the investigated methods as smoothers and propose a generalized measure for the effective number of parameters they use on unseen examples, using which we find that their apparent double descent curves indeed fold back into more traditional convex shapes – providing a resolution to tensions between double descent and statistical intuition.

A U-turn on Double Descent



examples. It is a classical observation that the U-shaped bias-variance trade-off curve manifests in many problems when the class capacity is considered this way [21]. (The interpolation threshold may be reached with fewer than n leaves in many cases, but n is clearly an upper bound.) To further enlarge the function class, we consider ensembles (averages) of several interpolating trees.¹

When expanding beyond n , no longer in the same model class!

Figure 5: **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 $N_{\text{leaf}}^{\text{max}}$.

A U-turn on Double Descent

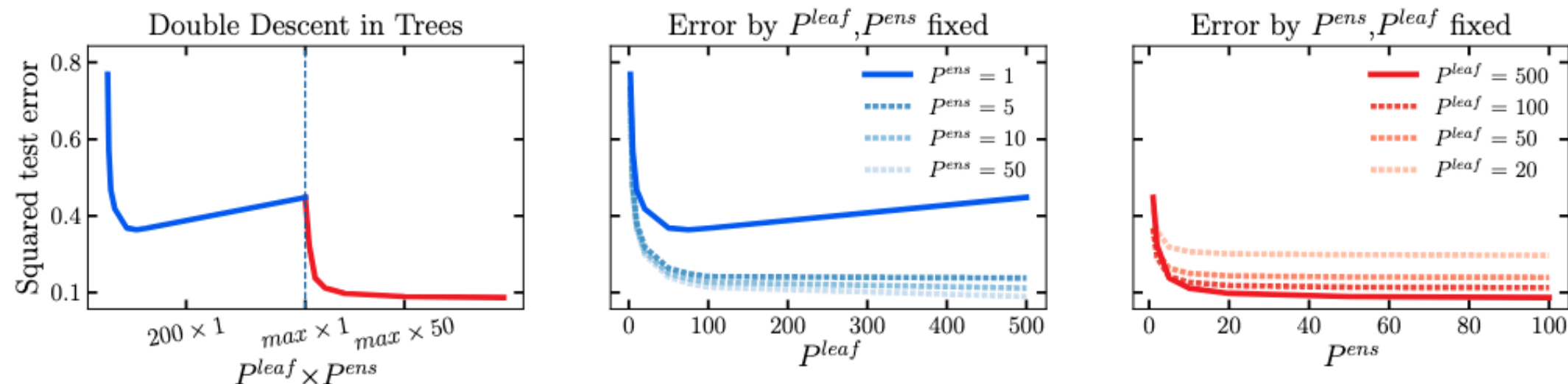


Figure 2: **Decomposing double descent for trees.** Reproducing [BHMM19]’s tree experiment (left). Test error by P^{leaf} for fixed P^{ens} (center). Test error by P^{ens} for fixed P^{leaf} (right).

A U-turn on Double Descent

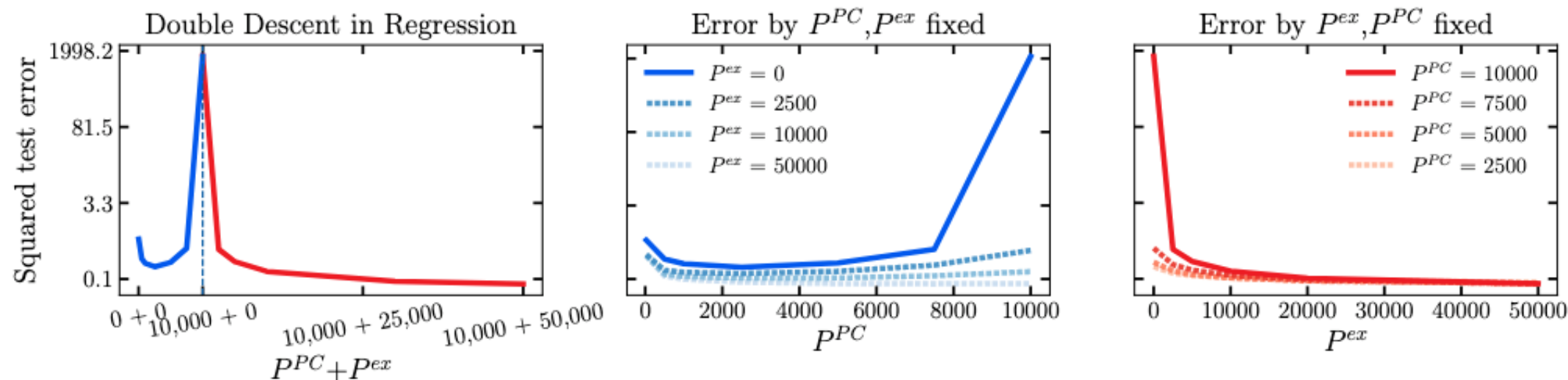


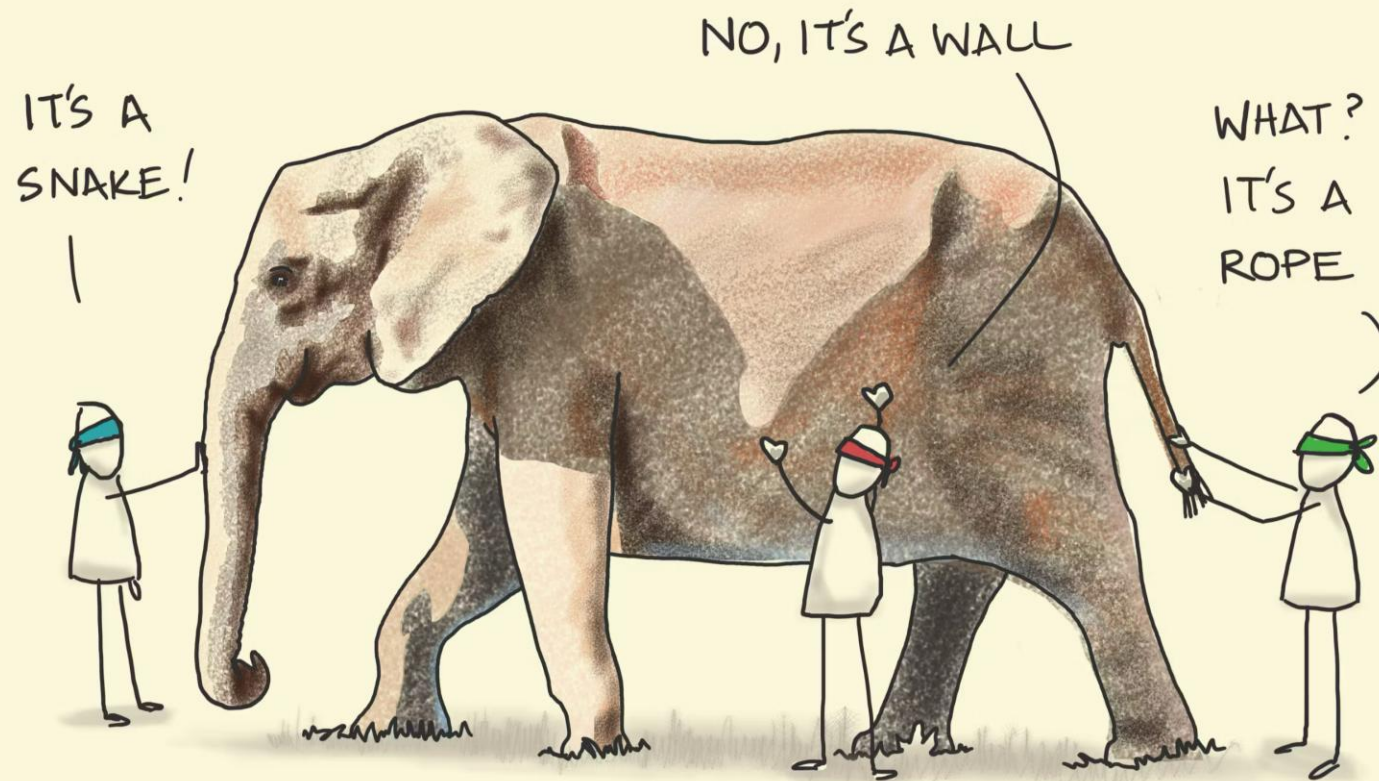
Figure 5: **Decomposing double descent for RFF Regression.** Double descent reproduced from [BHMM19] (left) can be decomposed into the standard U-curve of ordinary linear regression with P^{PC} features (center) and decreasing error achieved by a *fixed capacity* model with basis improving in P^{ex} (right).

A U-turn on Double Descent

- We demonstrated that existing experimental evidence for double descent in trees, boosting and linear regression does not contradict the traditional notion of a U-shaped complexity generalization curve.
- To the contrary, there are actually two independent underlying complexity axes that each exhibit a standard convex shape, and that the observed double descent phenomenon is a direct consequence of transitioning between these two distinct mechanisms of increasing the total number of model parameters.
- In the case of deep learning, ... it may indeed be instructive to investigate whether there also exist multiple implicitly entangled complexity axes in neural networks, and whether this may help to explain double descent in that setting.

THE BLIND AND THE ELEPHANT

OUR OWN EXPERIENCE IS RARELY THE WHOLE TRUTH



sketchplanations