DEEP DOUBLE DESCENT: WHERE BIGGER MODELS AND MORE DATA HURT

Preetum Nakkiran* Harvard University

Gal Kaplun† Harvard University

Yamini Bansal† Harvard University

Tristan Yang Harvard University

Boaz Barak Harvard University

Ilya Sutskever OpenAI

2020 ICLR poster paper

presented by: Jinhee Kim

Conventional wisdoms

- **Model capacity**

  Larger models are “worse” vs. “better”

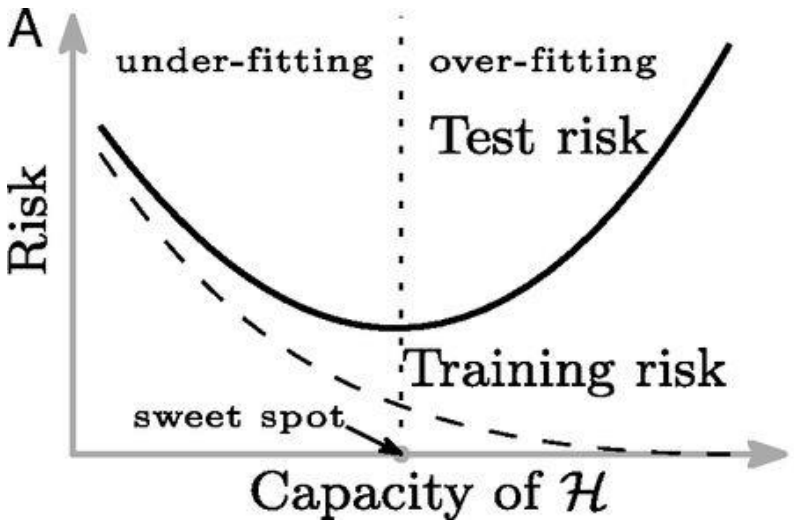
- **Training time**

  “early stopping” vs. “networks to zero training error”

- **Training data size**
  - One thing both classical statisticians and deep learning practitioners agree on: more data is always better.

Classical statistics

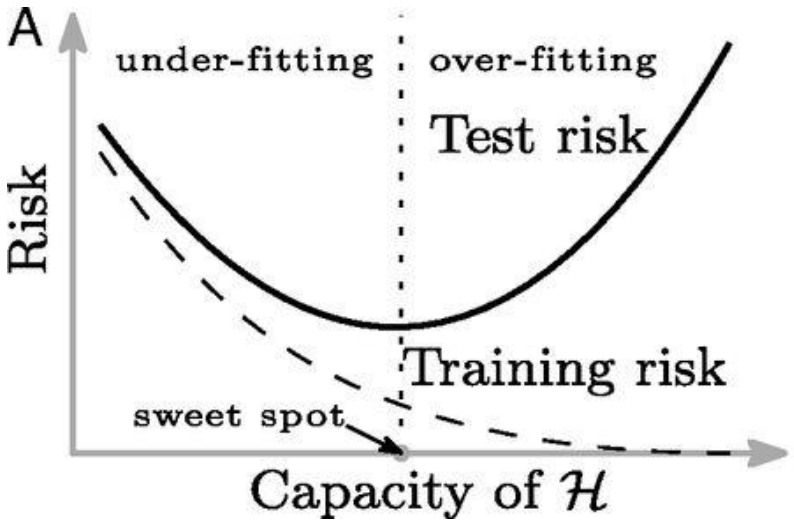
- **"Larger models are worse"** (once we pass a certain threshold)
  - Bias-variance trade-off

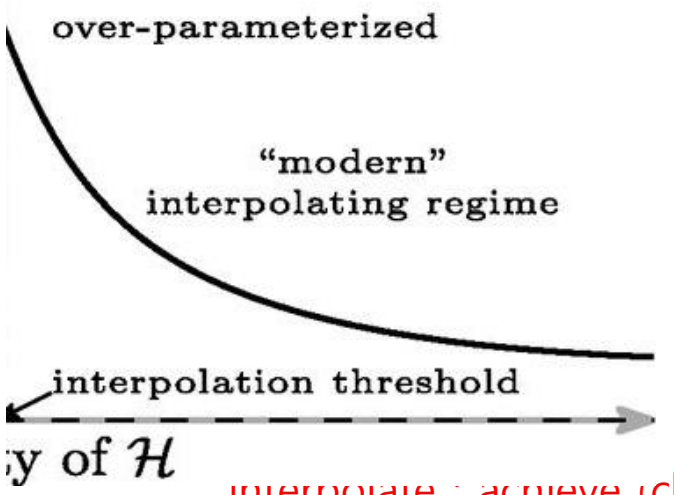


Belkin, Mikhail, et al. "Reconciling modern machine-learning practice and the classical bias–variance trade-off." *Proceedings of the National Academy of Sciences* 116.32 (2019): 15849-15854.

Modern neural networks

- **"Larger models are better"**
  - millions of parameters, more than enough to fit even random labels.



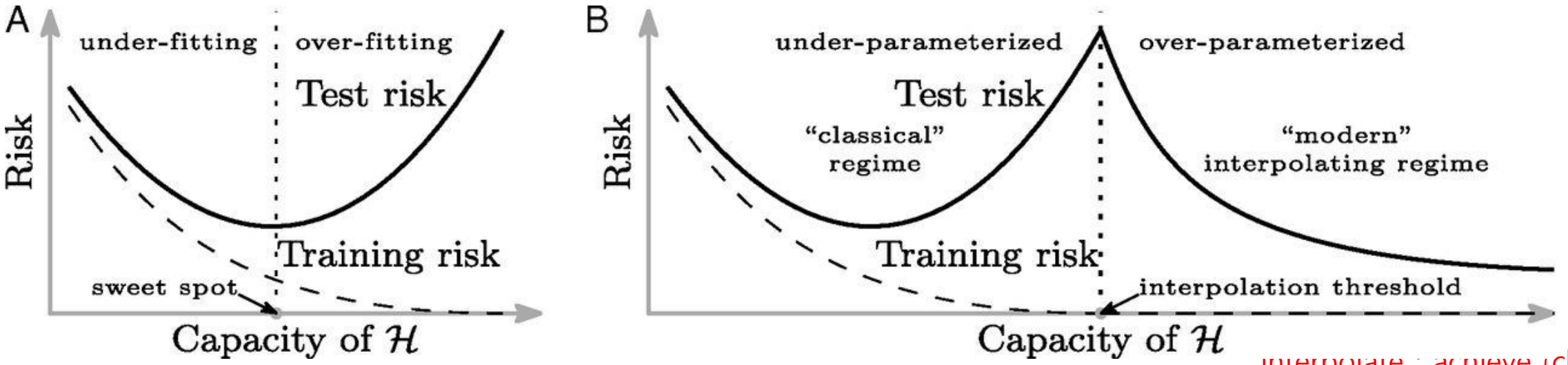


interpolate : achieve (close to) zero training error

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 phenomenon

- **"Larger models are not good in some regime"**
  - As we increase model size (or the #training epochs), performance first gets worse and then gets better.



interpolate : achieve (close to) zero training error

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.

Contribution

- This study **proposes much more general notion of "double descent"** that goes beyond varying the number of parameters.

- It proposes a new complexity measure, **effective model complexity (EMC) of a training procedure,** as the maximum number of samples on which it can achieve close to zero training error.
  - The EMC depends not just on the data distribution and the architecture of the classifier but also on the training procedure—and in particular increasing training time will increase the EMC.
- **Double descent occurs as a function of the EMC**

Effective model complexity (EMC)

a *training procedure* $\mathcal{T}$ to be any procedure that takes as input a set $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ of labeled training samples and outputs a classifier $\mathcal{T}(S)$ mapping data to labels.

effective model complexity of $\mathcal{T}$ (w.r.t. distribution $\mathcal{D}$) to be the maximum number of samples n on which $\mathcal{T}$ achieves on average ≈ 0 *training error*.

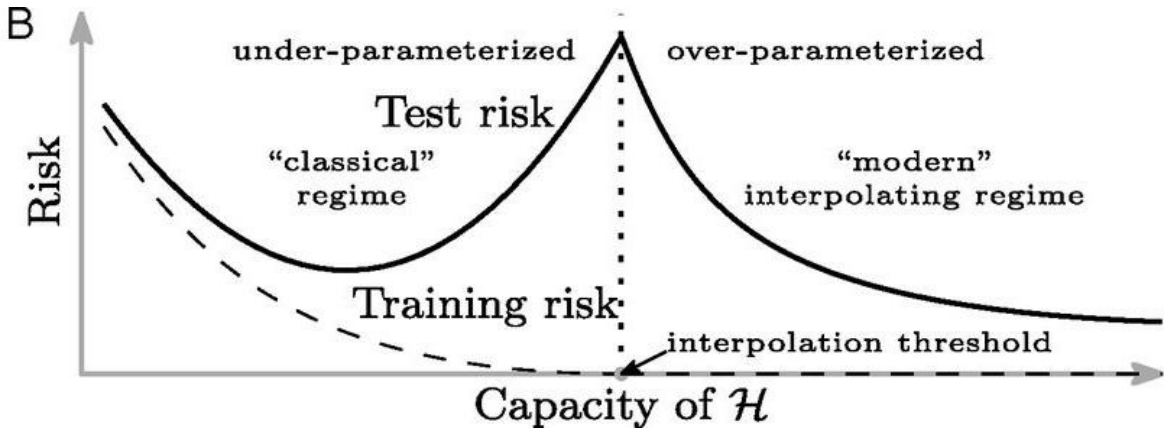
Definition 1 (Effective Model Complexity) *The* Effective Model Complexity *(EMC) of a training procedure $\mathcal{T}$, with respect to distribution $\mathcal{D}$ and parameter $\epsilon > 0$, is defined as:*

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

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

currently heuristically use $\epsilon = 0.1$

Effective model complexity (EMC)



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.*

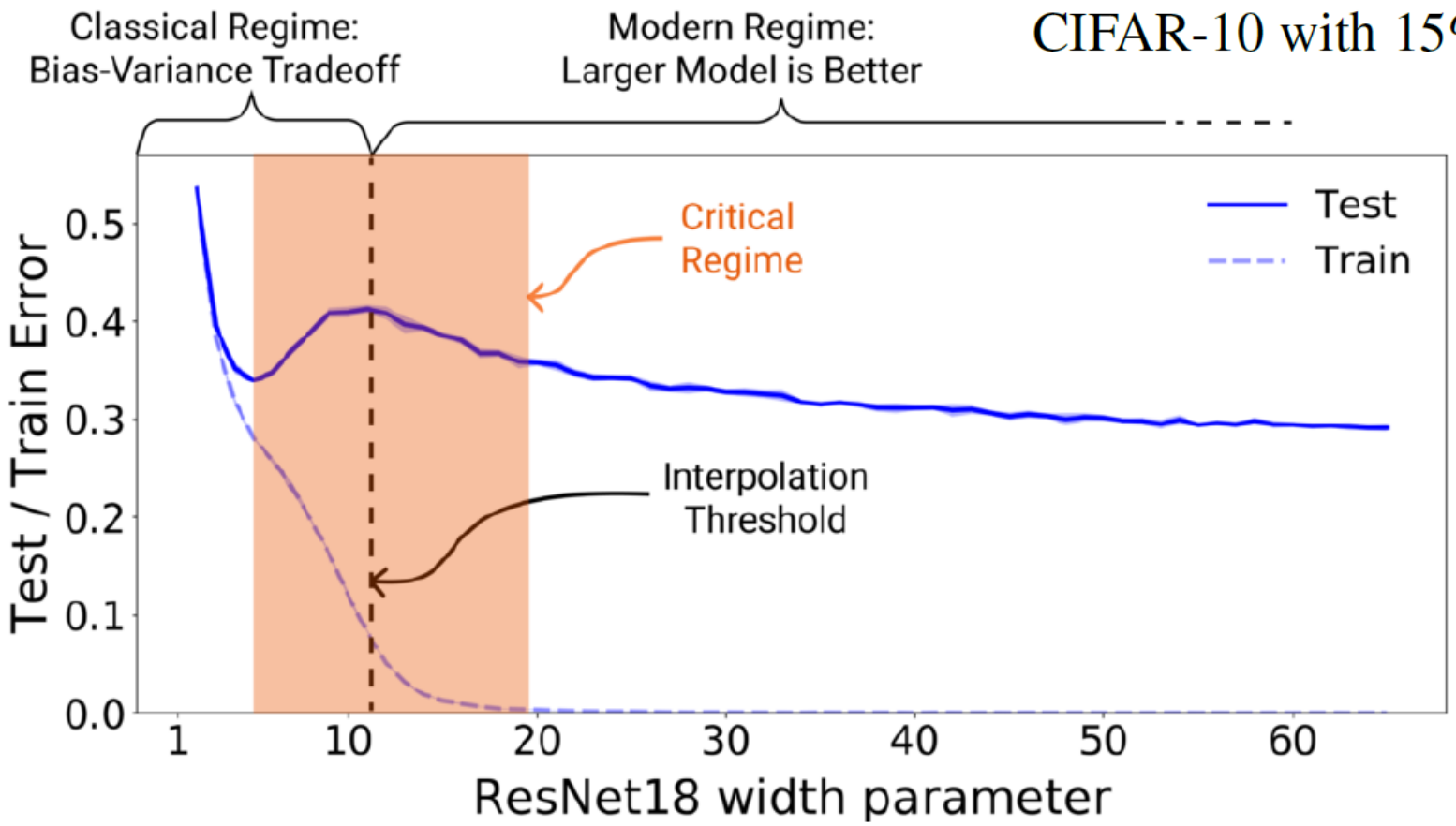
Double descent occurs as a function of the EMC

- **Model-wise double descent**
  - There is a regime where bigger models are worse.
- **Epoch-wise double descent**
  - Training longer can correct overfitting.
- **Sample-wise non-monotonicity**
  - There is a regime where more samples hurts.

+ **Remarks on label noise**
  - All forms of double descent are most strongly observed in settings with label noise in the train set.

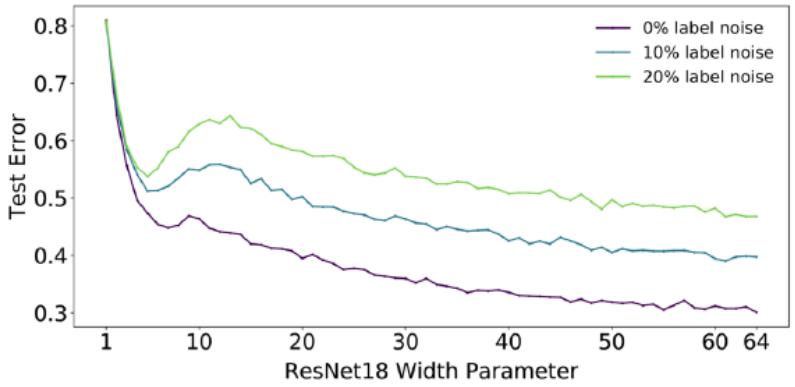
Model-wise double descent

- There is a regime where bigger models are worse

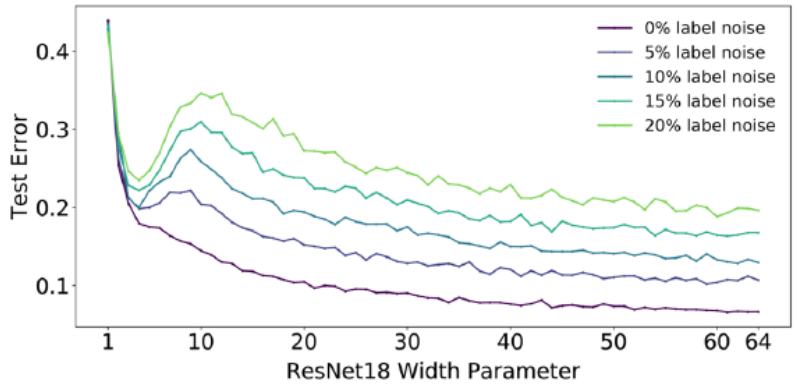


The intuition is that for model-sizes at the interpolation threshold, there is effectively only one model that fits the train data and this interpolating model is very sensitive to noise in the train set and/or model mis-specification.

Model-wise double descent



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

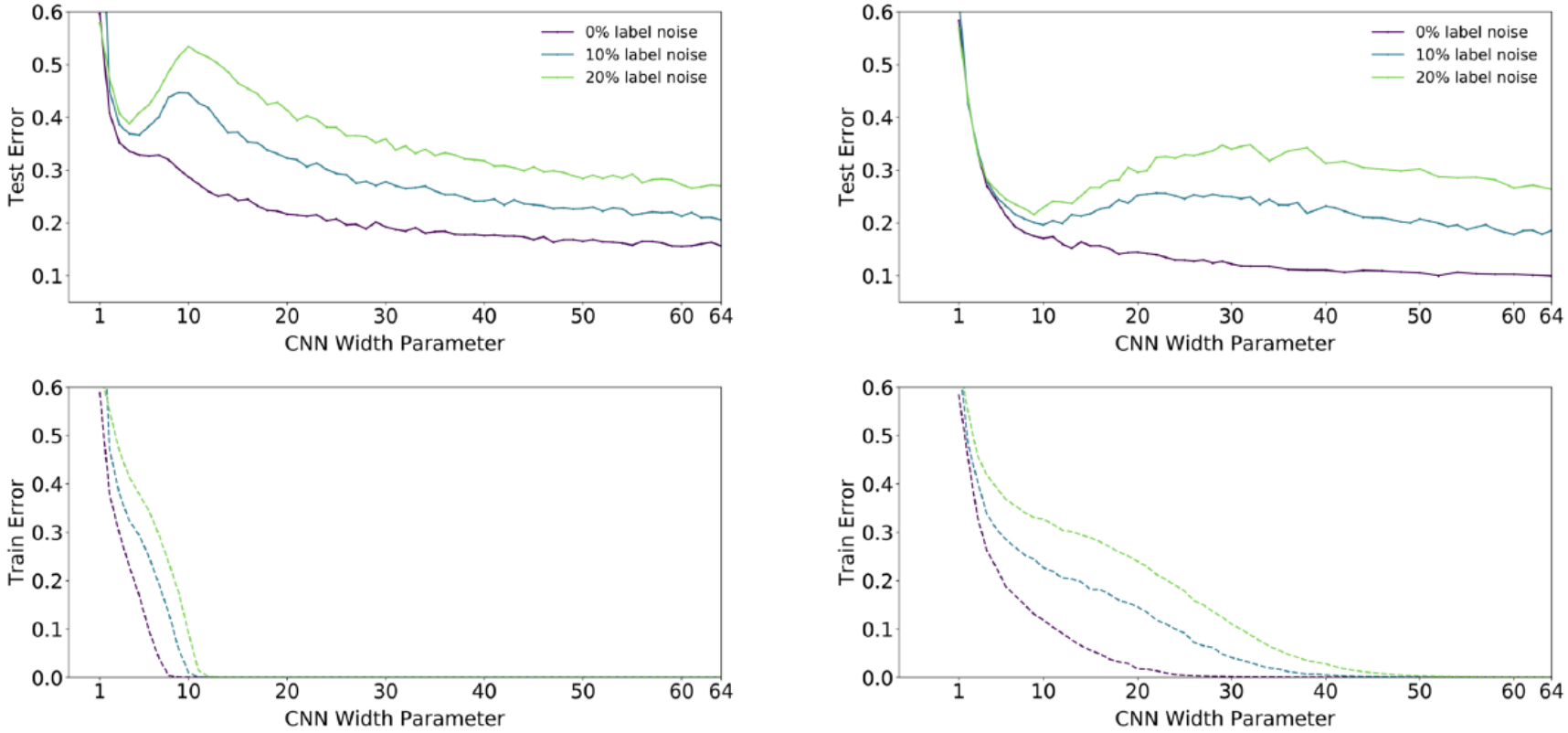


(b) **CIFAR-10.** There is a "plateau" in test error around the interpolation point with no label noise, which develops into a peak for added label noise.

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.

*The width of the critical interval depends on both the distribution and the training procedure in ways not yet completely understood.

Model-wise double descent



(a) Without data augmentation.

(b) With data augmentation.

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

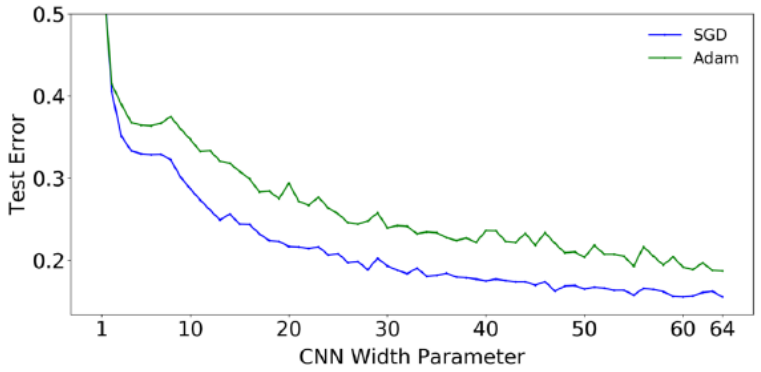


Figure 6: **SGD vs. Adam.** 5-Layer CNNs on CIFAR-10 with no label noise, and no data augmentation. Optimized using SGD for 500K gradient steps, and Adam for 4K epochs.

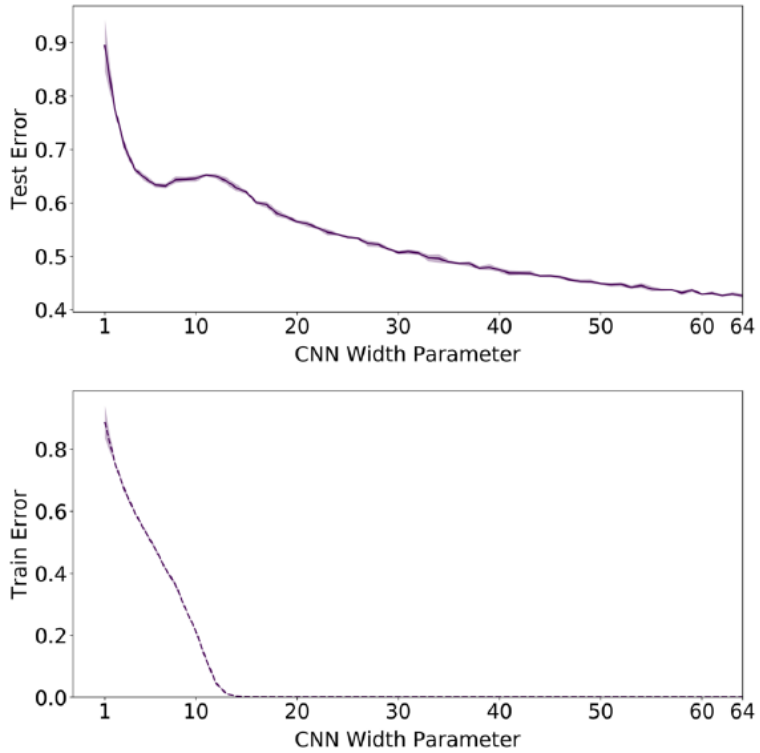


Figure 7: **Noiseless settings.** 5-layer CNNs on CIFAR-100 with no label noise; note the peak in test error. Trained with SGD and no data augmentation. See Figure 20 for the early-stopping behavior of these models.

Model-wise double descent

Transformer

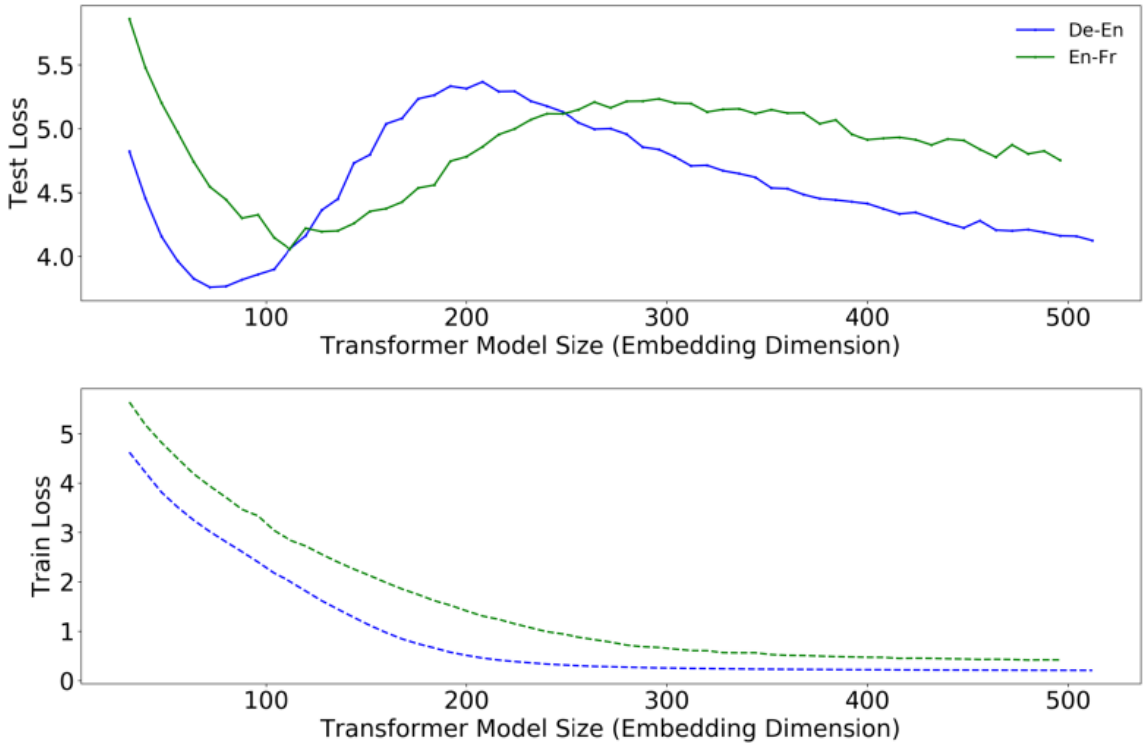


Figure 8: **Transformers on language translation tasks:** Multi-head-attention encoder-decoder Transformer model trained for 80k gradient steps with labeled smoothed cross-entropy loss on IWSLT'14 German-to-English (160K sentences) and WMT'14 English-to-French (subsampled to 200K sentences) dataset. Test loss is measured as per-token perplexity.

Epoch-wise double descent

- Training longer can correct overfitting.

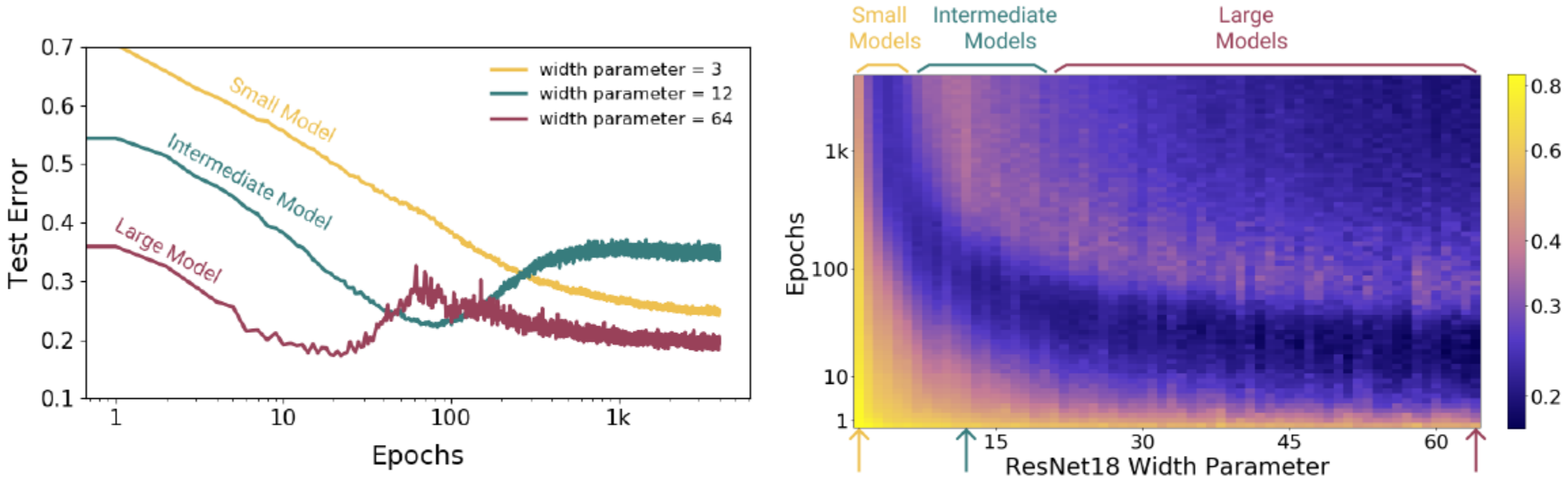
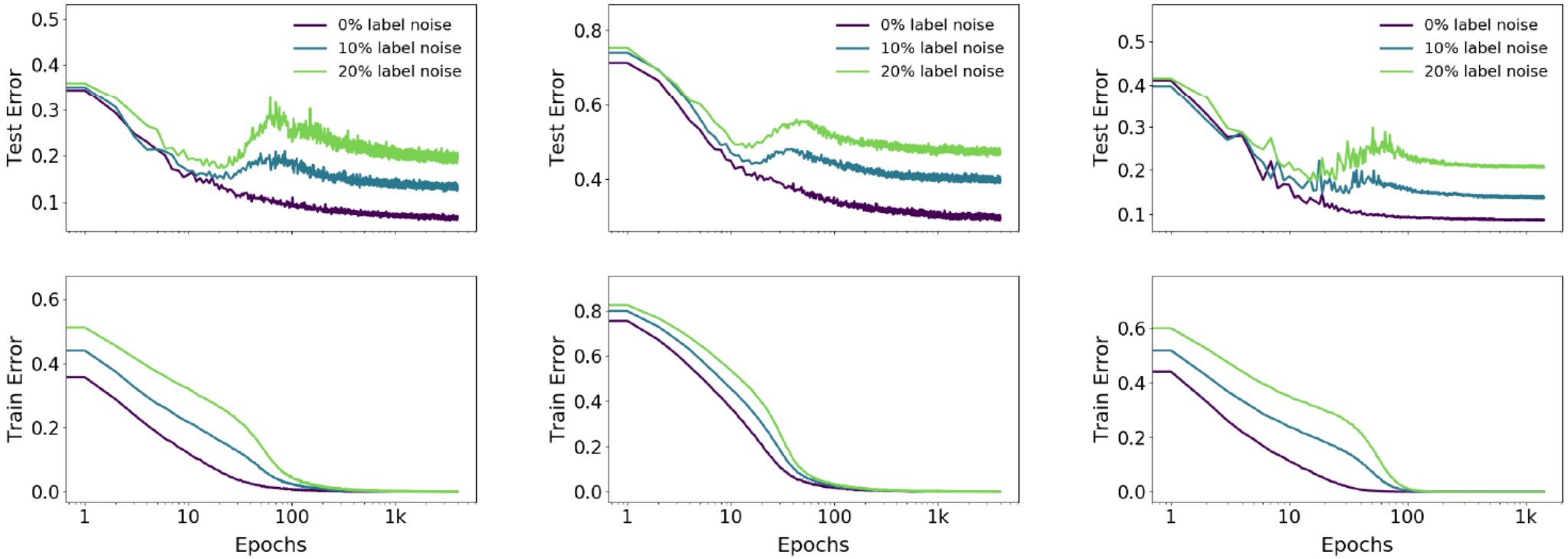


Figure 9: **Left:** Training dynamics for models in three regimes. Models are ResNet18s on CIFAR10 with 20% label noise, trained using Adam with learning rate 0.0001, and data augmentation. **Right:** Test error over (Model size × Epochs). Three slices of this plot are shown on the left.

Epoch-wise double descent

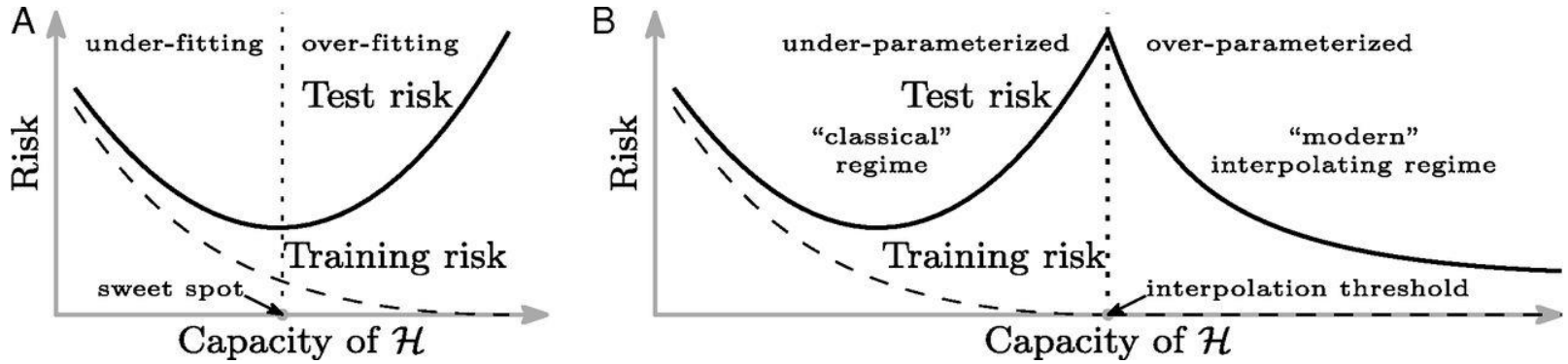


(a) ResNet18 on CIFAR10. (b) ResNet18 on CIFAR100. (c) 5-layer CNN on CIFAR 10.

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.

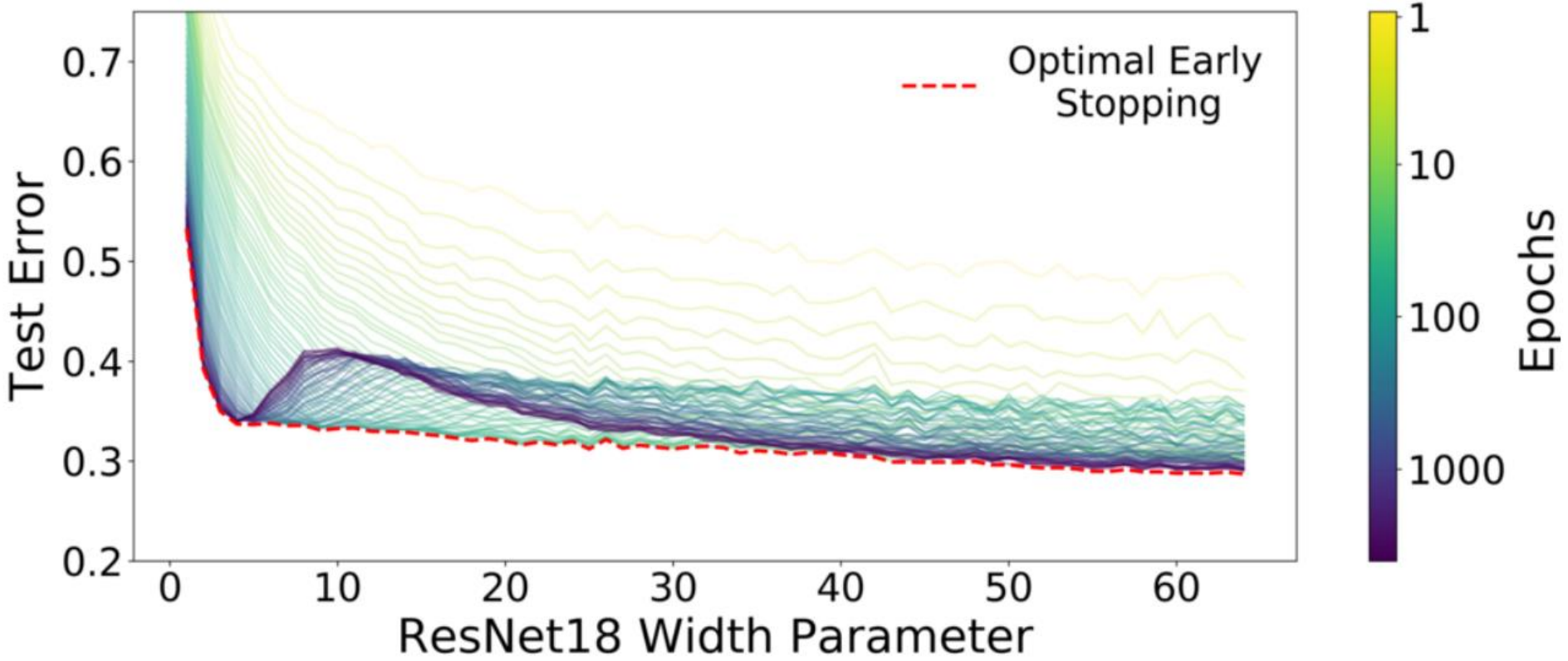
Conventional wisdom

- Training is split into two phases:
  - (1) In the first phase, the network learns a function with a small generalization gap
  - (2) In the second phase, the network starts to over-fit the data leading to an increase in test error
  - (3) But this is not the complete picture—in some regimes, the test error decreases again and may achieve a lower value at the end of training as compared to the first minimum.



Double descent occurs as a function of the EMC

- As a corollary, early stopping only helps in the relatively narrow parameter regime of critically parameterized models



Double descent occurs as a function of the EMC

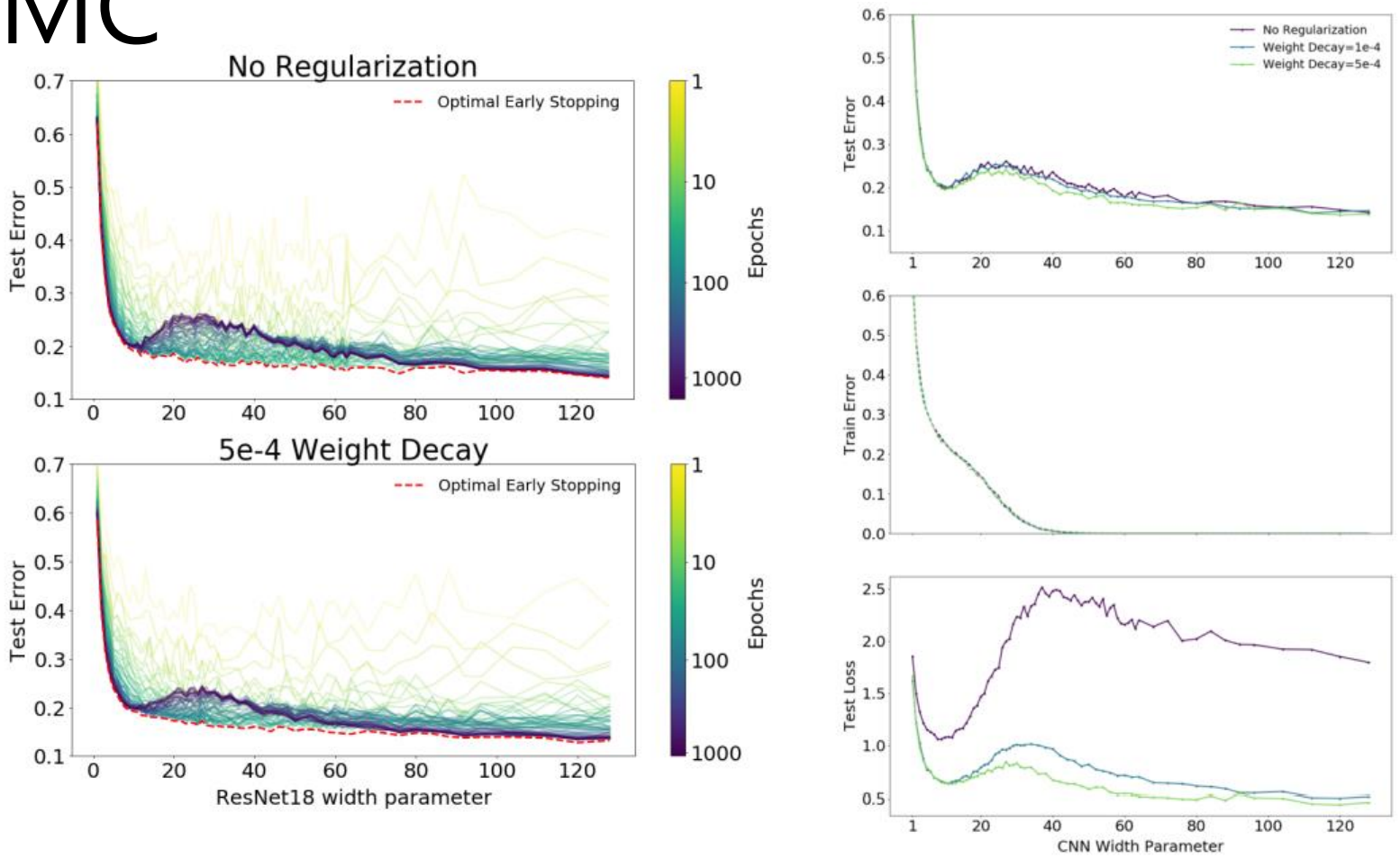


Figure 21: **Left:** Test error dynamics with weight decay of 5e-4 (bottom left) and without weight decay (top left). **Right:** Test and train error and *test loss* for models with varying amounts of weight decay. All models are 5-Layer CNNs on CIFAR-10 with 10% label noise, trained with data-augmentation and SGD for 500K steps.

Double descent occurs as a function of the EMC

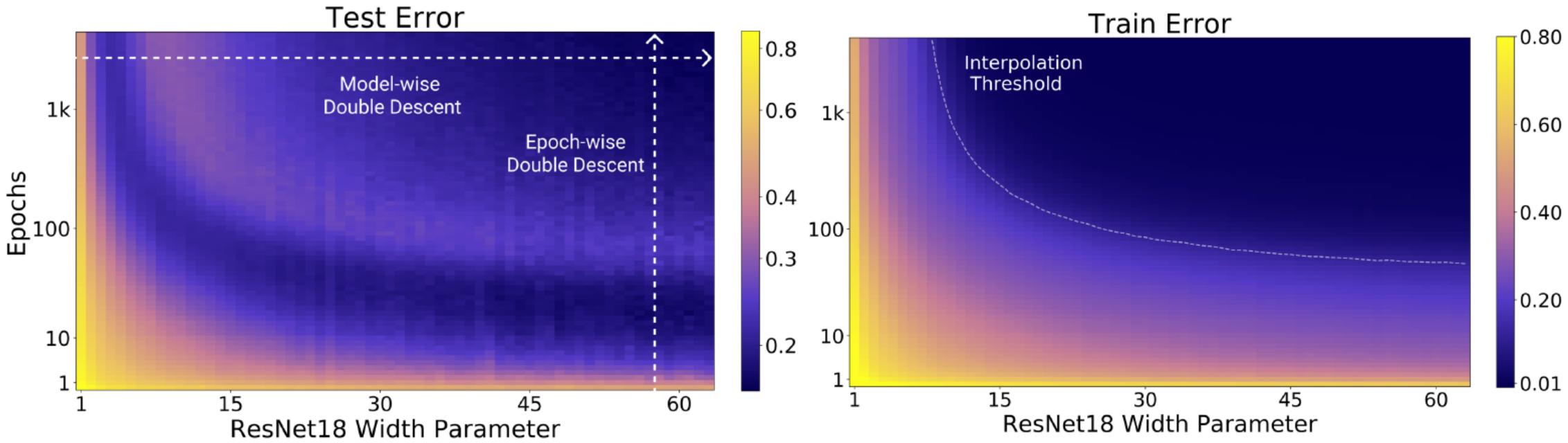


Figure 2: **Left:** Test error as a function of model size and train epochs. The horizontal line corresponds to model-wise double descent–varying model size while training for as long as possible. The vertical line corresponds to epoch-wise double descent, with test error undergoing double-descent as train time increases. **Right** Train error of the corresponding models. All models are Resnet18s trained on CIFAR-10 with 15% label noise, data-augmentation, and Adam for up to 4K epochs.

Sample-wise non-monotonicity

- There is a regime where more samples hurts

Transformer

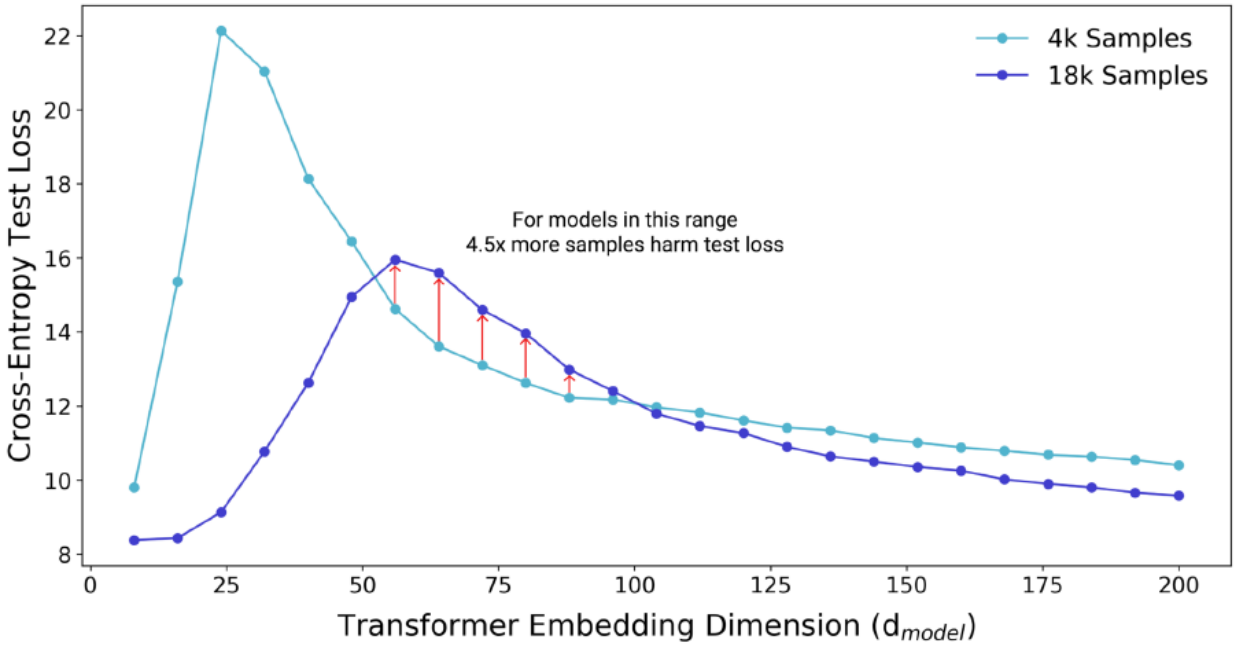
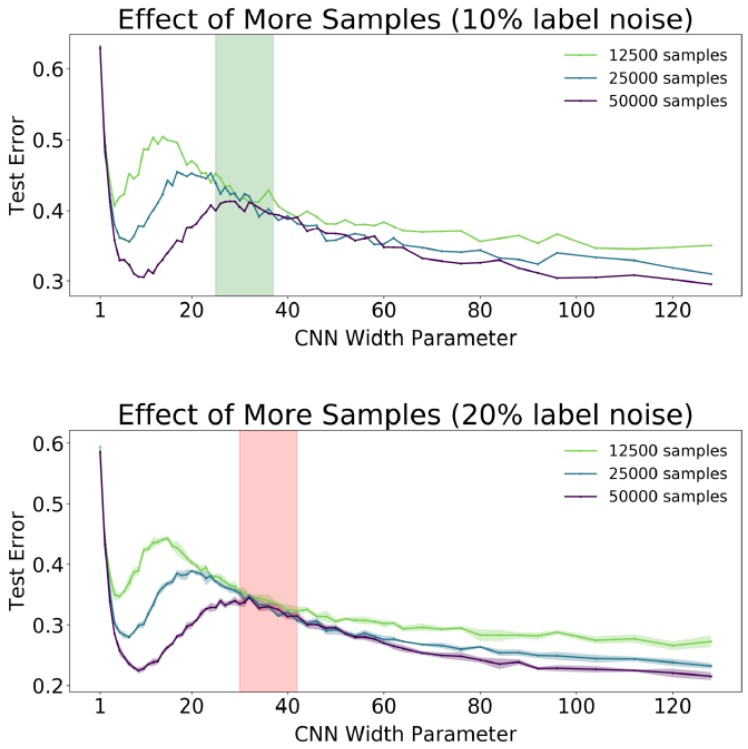
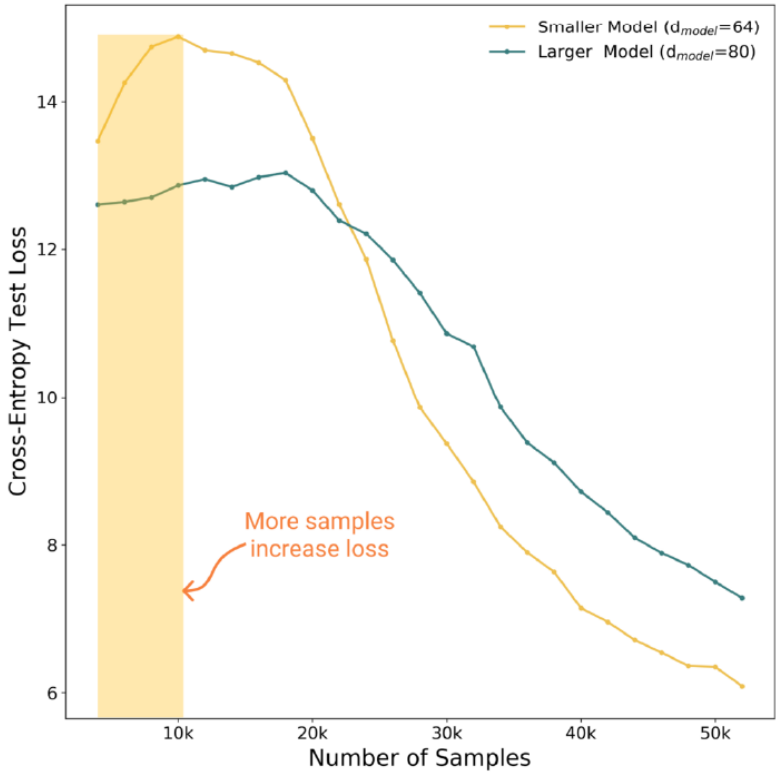


Figure 3: Test loss (per-token perplexity) as a function of Transformer model size (embedding dimension d_{model}) on language translation (IWSLT'14 German-to-English). The curve for 18k samples is generally lower than the one for 4k samples, but also shifted to the right, since fitting 18k samples requires a larger model. Thus, for some models, the performance for 18k samples is *worse* than for 4k samples.

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× more samples does not improve test error. **Bottom:** There is a range of model sizes (shaded red) where training on 4× 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.

1) Increasing the number of samples shrinks the area under the curve.
2) 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.

Sample-wise non-monotonicity

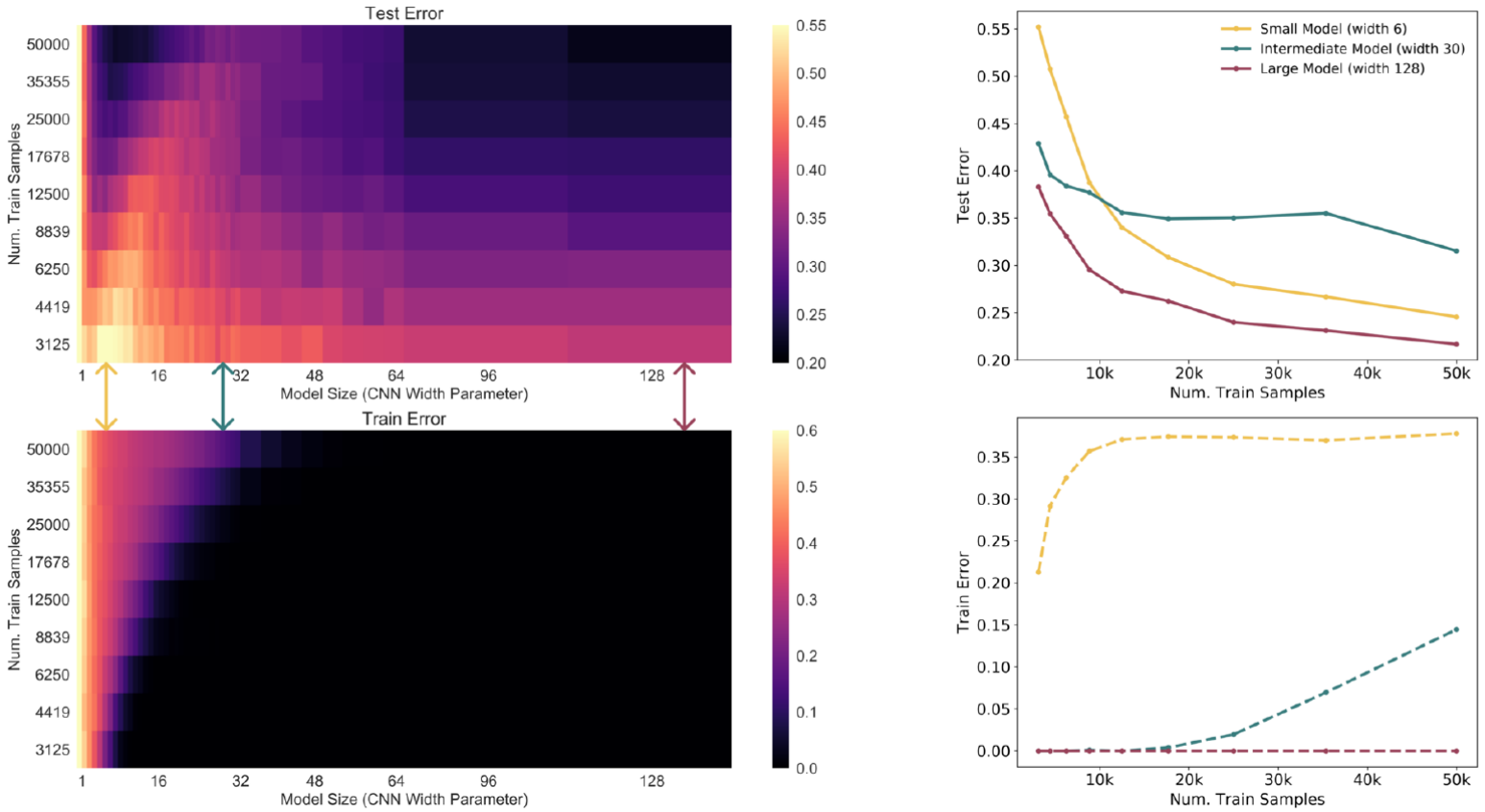


Figure 12: **Left:** Test Error as a function of model size and number of train samples, for 5-layer CNNs on CIFAR-10 + 20% noise. Note the ridge of high test error again lies along the interpolation threshold. **Right:** Three slices of the left plot, showing the effect of more data for models of different 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).

A SUMMARY TABLE OF EXPERIMENTAL RESULTS

					Double-Descent		
Dataset	Architecture	Opt.	Aug.	% Noise	Model	Epoch	Figure(s)
CIFAR 10	CNN	SGD	✓	0	✗	✗	5, 27
			✓	10	✓	✓	5, 27, 6
			✓	20	✓	✓	5, 27
				0	✗	✗	5, 25
				10	✓	✓	5
				20	✓	✓	5
		SGD + w.d.	✓	20	✓	✓	21
		Adam		0	✓	–	25
	ResNet	Adam	✓	0	✗	✗	4, 10
			✓	5	✓	–	4
			✓	10	✓	✓	4, 10
			✓	15	✓	✓	4, 2
			✓	20	✓	✓	4, 9, 10
		Various	✓	20	–	✓	16, 17, 18
(subsampled)	CNN	SGD	✓	10	✓	–	11a
		SGD	✓	20	✓	–	11a, 12
(adversarial)	ResNet	SGD		0	Robust err.	–	26
CIFAR 100	ResNet	Adam	✓	0	✓	✗	4, 19, 10
			✓	10	✓	✓	4, 10
			✓	20	✓	✓	4, 10
	CNN	SGD		0	✓	✗	20
IWSLT '14 de-en	Transformer	Adam		0	✓	✗	8, 24
(subsampled)	Transformer	Adam		0	✓	✗	11b, 23
WMT '14 en-fr	Transformer	Adam		0	✓	✗	8, 24