# Loss landscape in Deep neural networks

*Dmitry P. Vetrov*
*Research professor at HSE, Lab lead in SAIC-Moscow*
*Head of BayesGroup*
*http://bayesgroup.ru*

# Preamble

Modern data scientists seem to be more like naturalists of 19th century rather than mathematicians

Experiment design and the scientific method are of great importance

Deep learning has a risk of becoming new quantum mechanics with its inglorious «shut up and calculate»

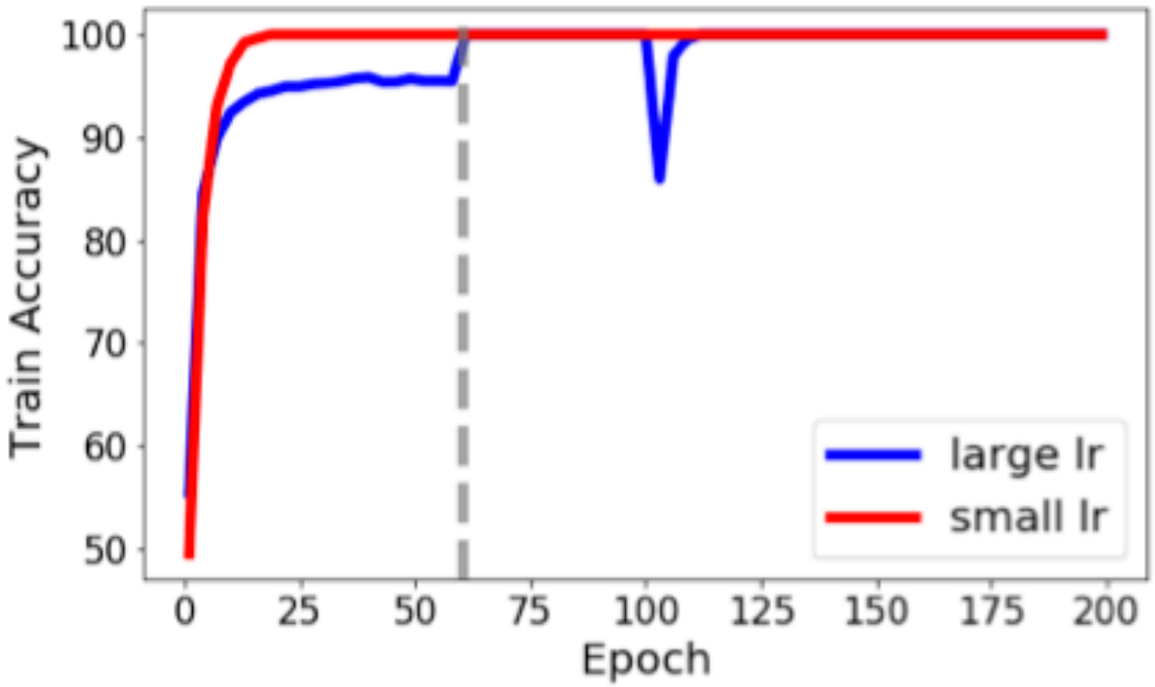
# Open questions in DL

- SLT predicts huge generalization gap in over-parameterized models but we observe the contrary
- DNNs can easily fit randomly labeled data
- There are minefields in loss landscape
- Global minima of the loss are connected
- Double descent still lacks an explanation
- How to predict the performance of large DNNs and of deep ensembles?

# Outline

- Learning rate and its influence to the width of loss minima
- Mode connectivity
- Statistical mechanics as a new language for generalization theory
- Double descent and its possible explanation
- Power laws in deep neural networks

# Different learning rates



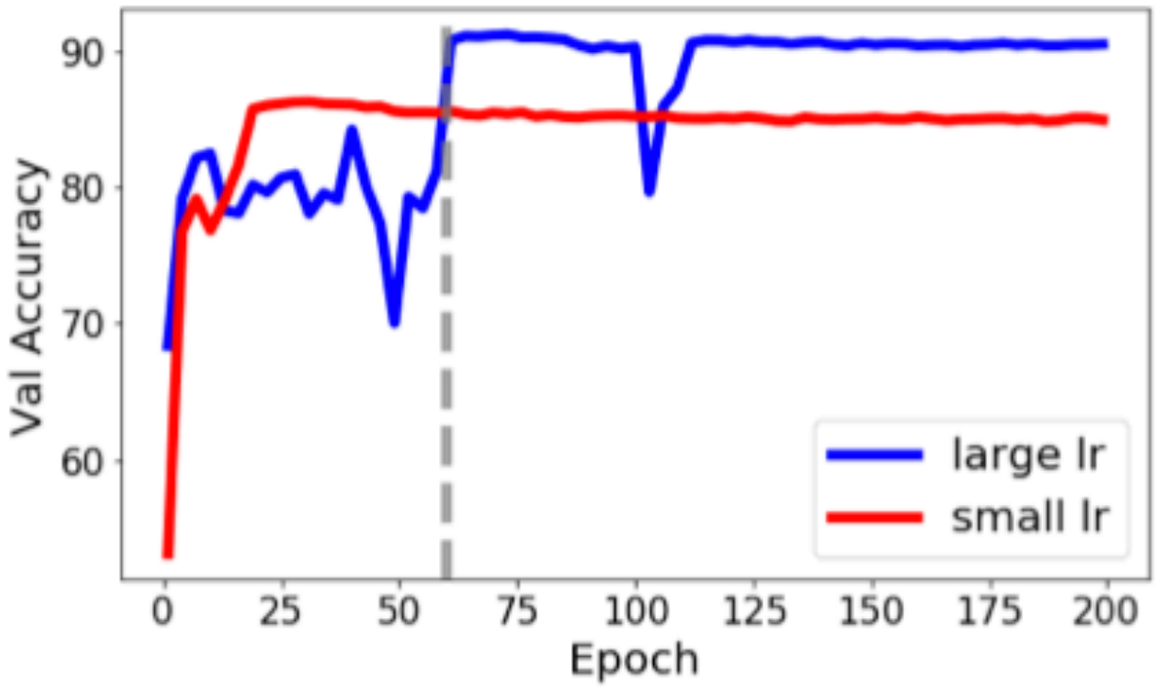




Figure 1: Accuracy on CIFAR-10 vs. epoch for WideResNet with standard hyperparameters (including weight decay). The gray line represents the annealing time. **Left:** Train. **Right:** Validation. Only after annealing does the large learning rate model visibly outperform the small learning rate in terms of generalization.

**Y. Li, C. Wei, T. Ma. Towards Explaining the Regularization Effect of Initial Large Learning Rate in Training Neural Networks. In NeurIPS 2019.**

# Two kinds of patterns

**Noisy patterns**

Easy-to-fit, Hard-to-generalize

- Noisy regularities
- Easy patterns

**Fixed patterns**

Hard-to-fit, Easy-to-generalize

- Low noise
- Complicated patterns

**Y. Li, C. Wei, T. Ma. Towards Explaining the Regularization Effect of Initial Large Learning Rate in Training Neural Networks. In NeurIPS 2019.**

# Two kinds of patterns

**Noisy patterns**
Easy-to-fit, Hard-to-generalize
- Noisy regularities
- Easy patterns

**Fixed patterns**
Hard-to-fit, Easy-to-generalize
- Low noise
- Complicated patterns

Main claim:
- smaller LR learns (memorizes) hard-to-fit patterns
- Larger LR learns easy-to-fit patterns
- Larger LR with annealing learns both

**Y. Li, C. Wei, T. Ma. Towards Explaining the Regularization Effect of Initial Large Learning Rate in Training Neural Networks. In NeurIPS 2019.**

# Toy experiment

**Group 1:** 20% examples with hard-to-generalize, easy-to-fit patterns

original image

**Group 2:** 20% examples with easy-to-generalize, hard-to-fit patterns

hard-to-fit patch indicating class

**Group 3:** 60% examples with both patterns

# Toy experiment

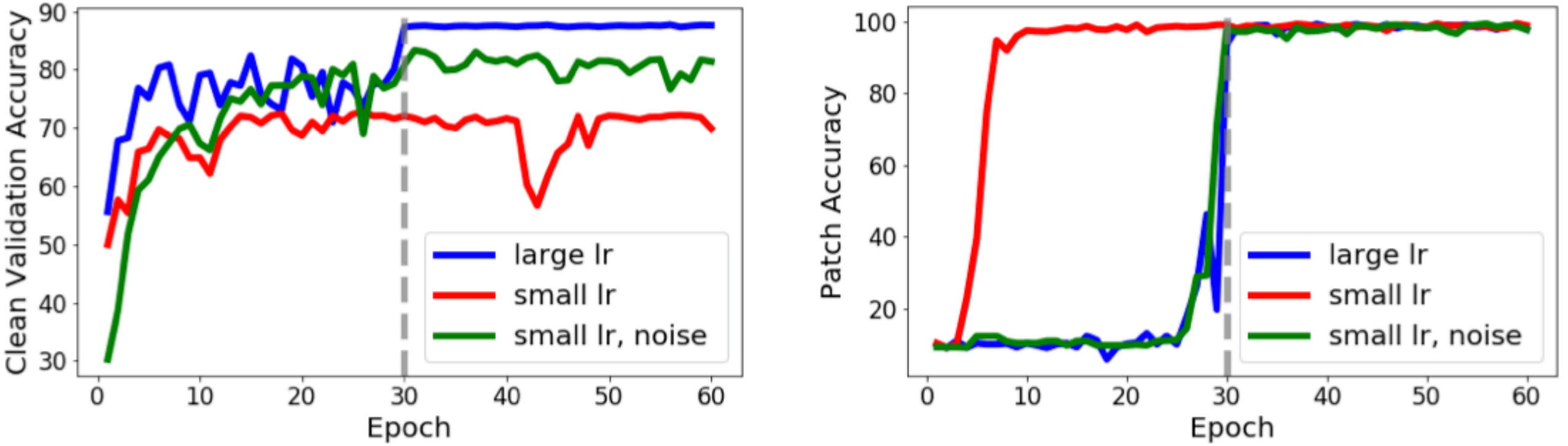
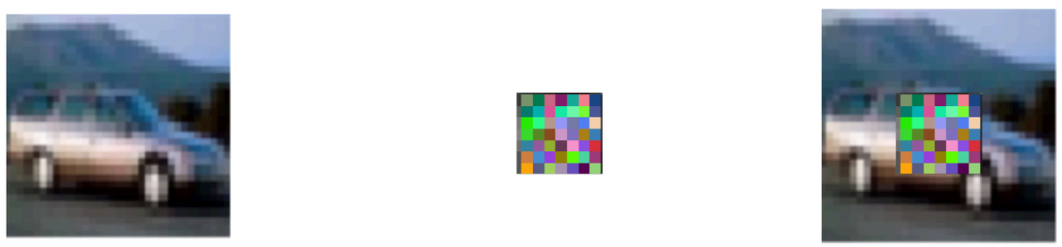



Figure 2: Accuracy vs. epoch for neural nets trained on patch-augmented CIFAR-10. The gray line indicates annealing of activation noise and learning rate. **Left:** Clean validation set. **Right:** Images containing only the patch.

**Group 1:** 20% examples with hard-to-generalize, easy-to-fit patterns

**Group 2:** 20% examples with easy-to-generalize, hard-to-fit patterns

**Group 3:** 60% examples with both patterns



original image

hard-to-fit patch indicating class

# Discussion

## Small LR

First: Memorizes fixed patterns

Second: Learns noisy patterns using less data

## Large LR

Learns noisy patterns using full data

Unable to memorize complicated fixed patterns

## Large LR + Annealing

First: Learns flexible noisy patterns using full data

Second: Memorizes fixed patterns using less data

# Discussion

## Small LR

First: Memorizes fixed patterns
Second: Learns noisy patterns using less data

## Large LR

Learns noisy patterns using full data
Unable to memorize complicated fixed patterns

## Large LR + Annealing

First: Learns flexible noisy patterns using full data
Second: Memorizes fixed patterns using less data

Both noisy and fixed patterns are present in real data
Larger LR corresponds to wider local optima (see Khan 2019)

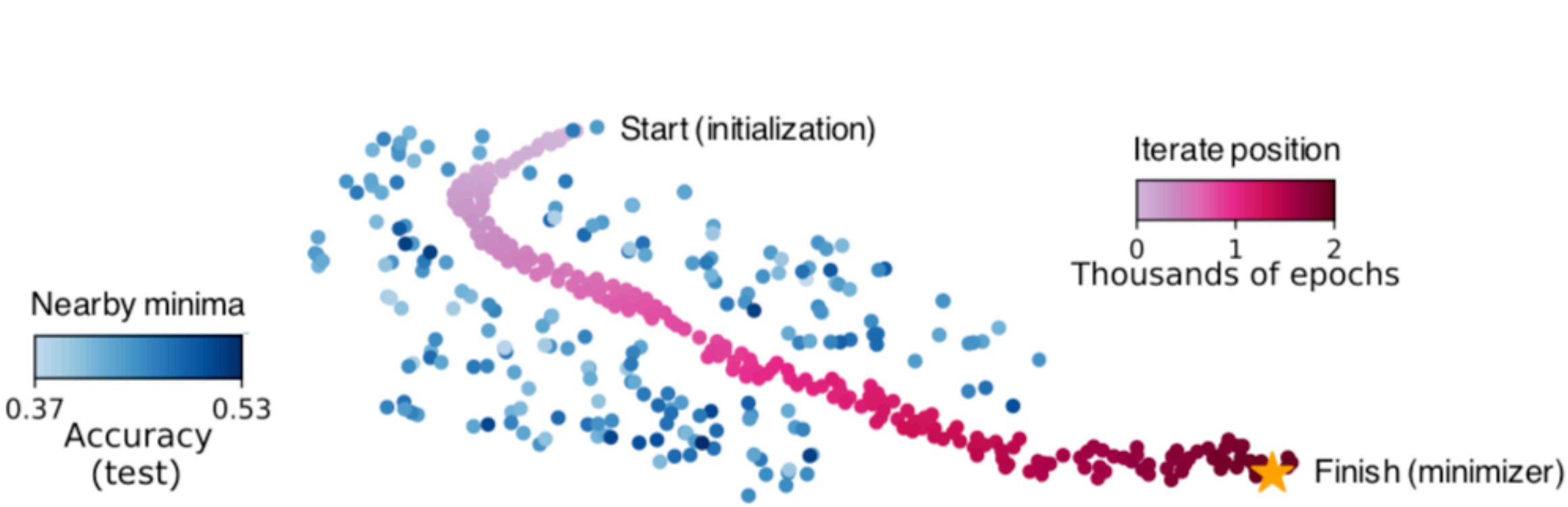
**E. Khan. Deep learning with Bayesian principles. NeurIPS 2019 tutorial**

# Minefields in loss landscape

It appears that there are lots of poor global minima with almost zero train loss and arbitrarily poor validation loss

Moreover many of them lie in close vicinity to optimization path

SGD does not «see» them unless we start directly looking for poor minima




**Huang et al. Understanding Generalization through Visualizations. https://arxiv.org/abs/1906.03291**

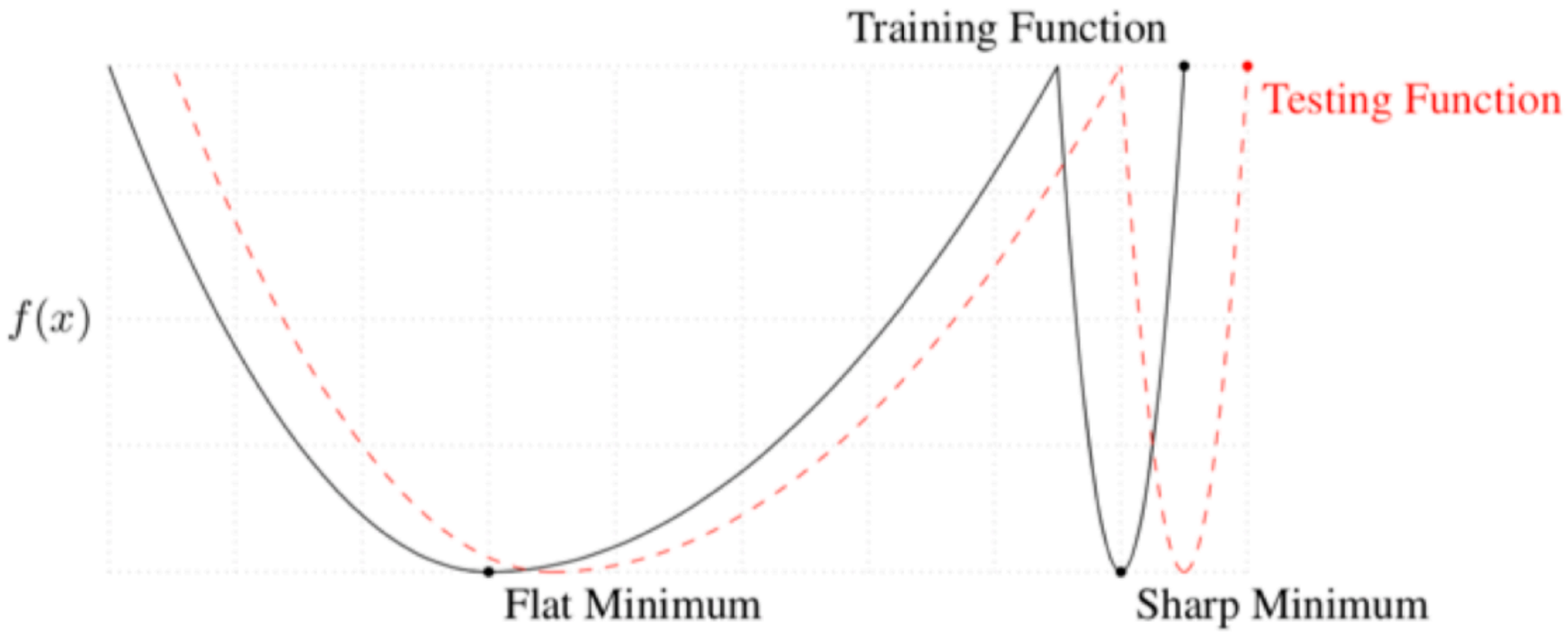
# Implicit regularization

Modern DNNs are trained with stochastic optimization techniques

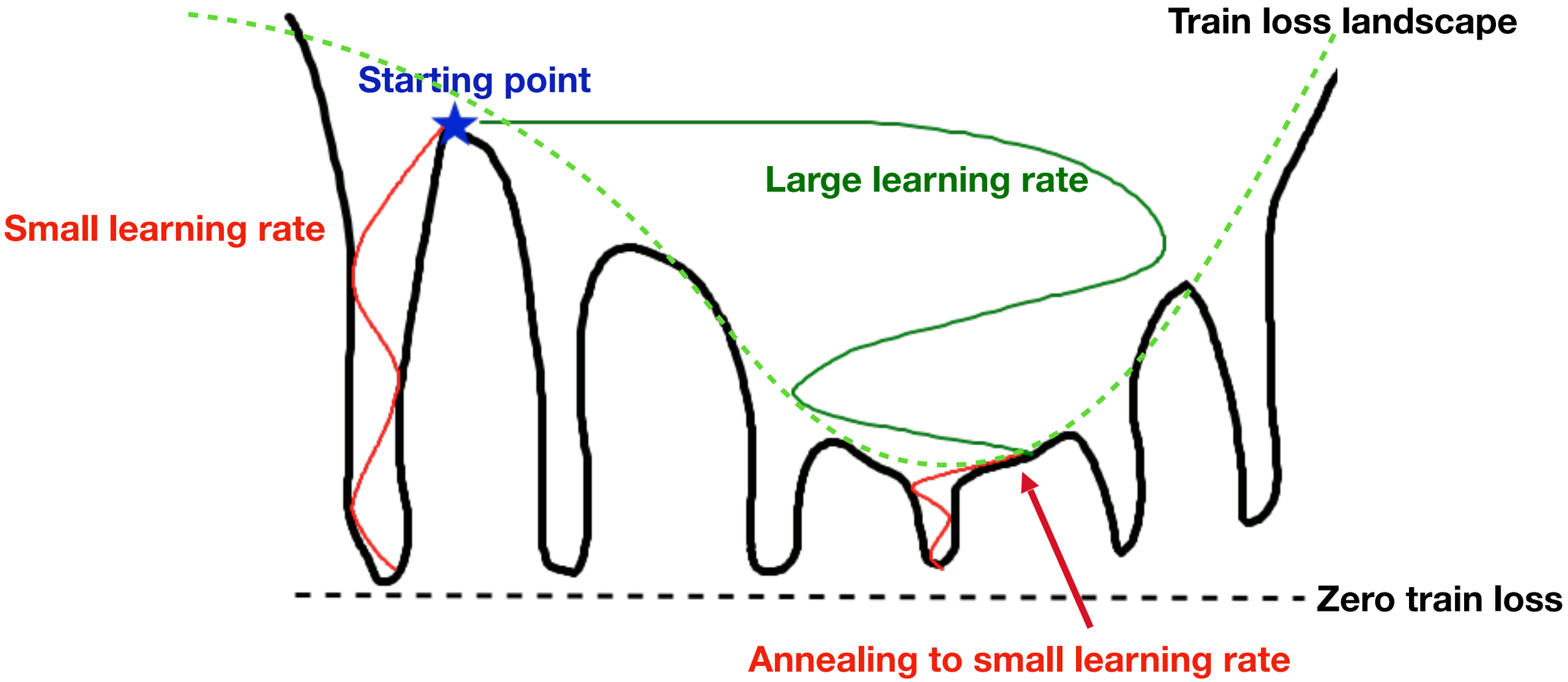
The noise in stochastic gradient prevents us from seeing small details of the loss

We simply cannot find narrow minima

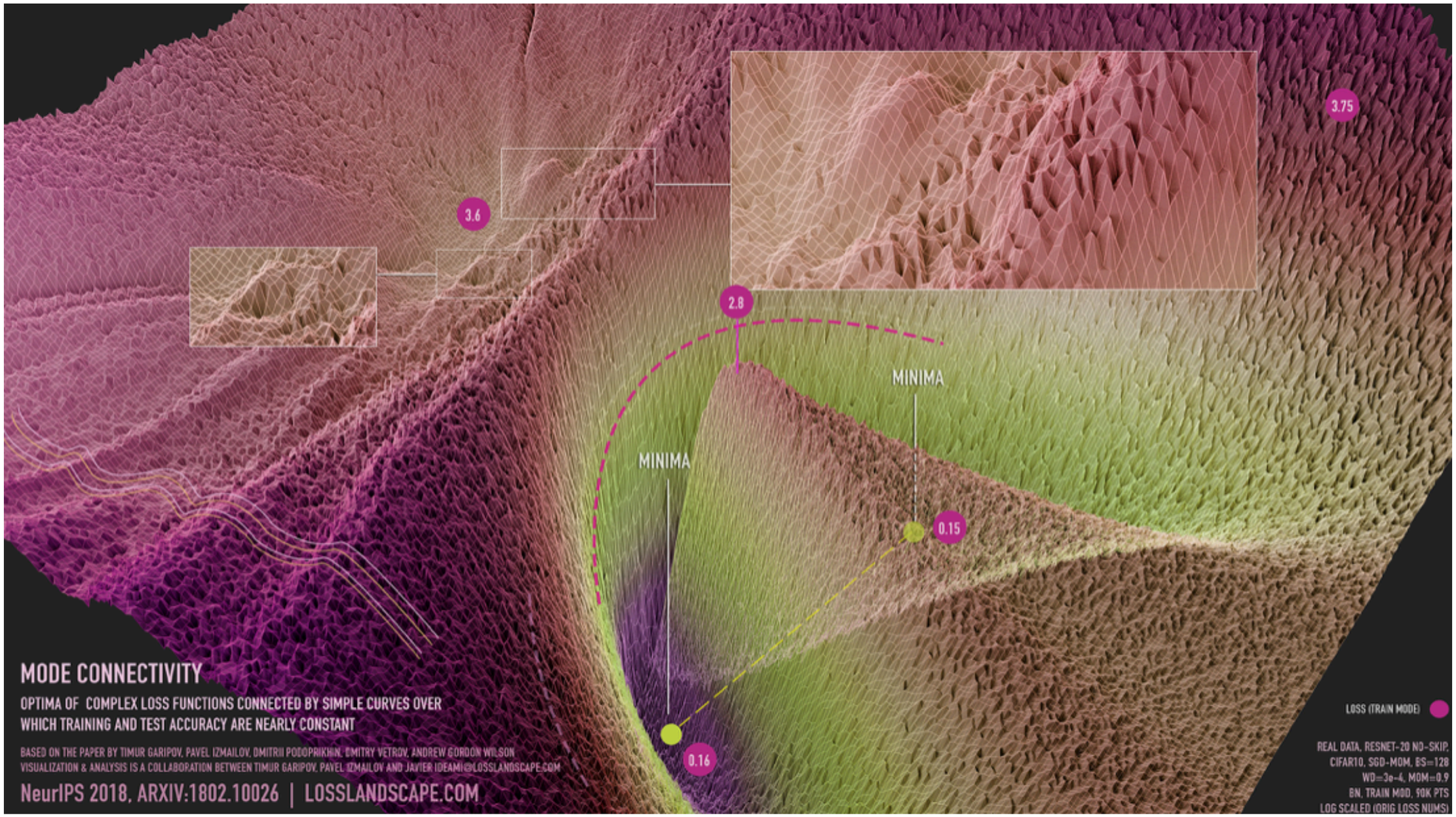
There is a common (yet unproven) belief in community that wide minima have better generalization



# Hypothesis


Train loss landscape
Starting point
Large learning rate
Small learning rate
Zero train loss
Annealing to small learning rate

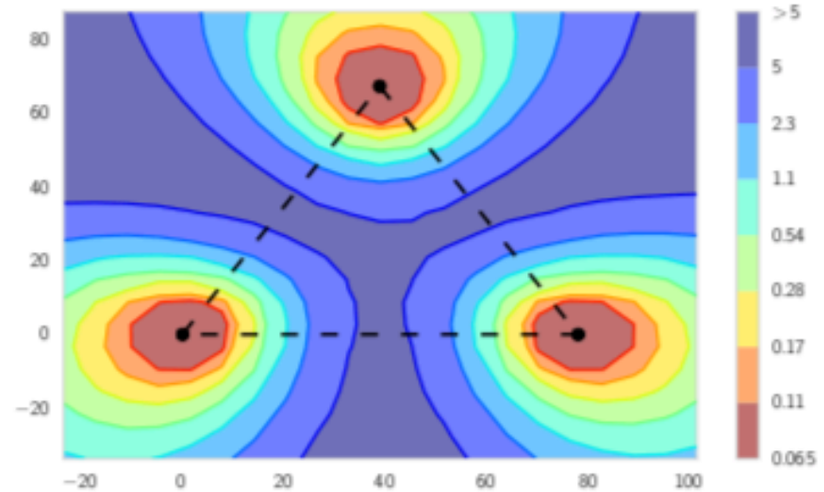
# 2-dimensional slice

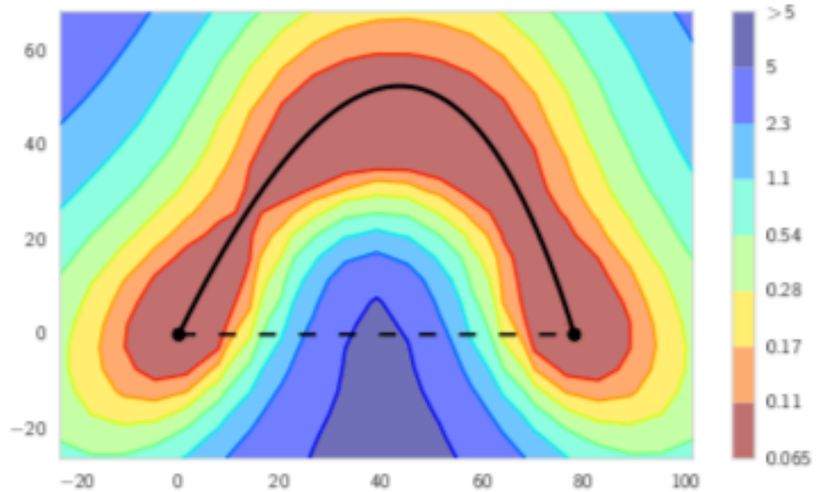


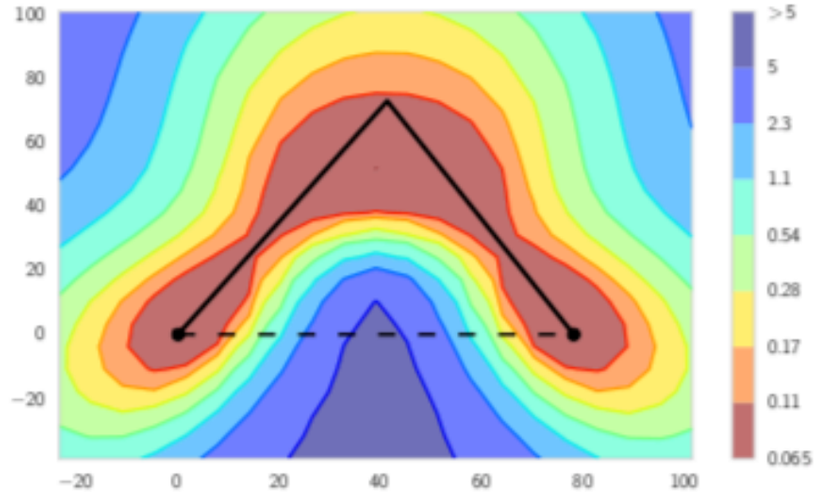

https://www.youtube.com/watch?v=dqX2LBcp5Hs

**T. Garipov, P. Izmailov, D. Podoprikhin, D. Vetrov, A. Wilson. Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs. In NeurIPS 2018**

# Discussion



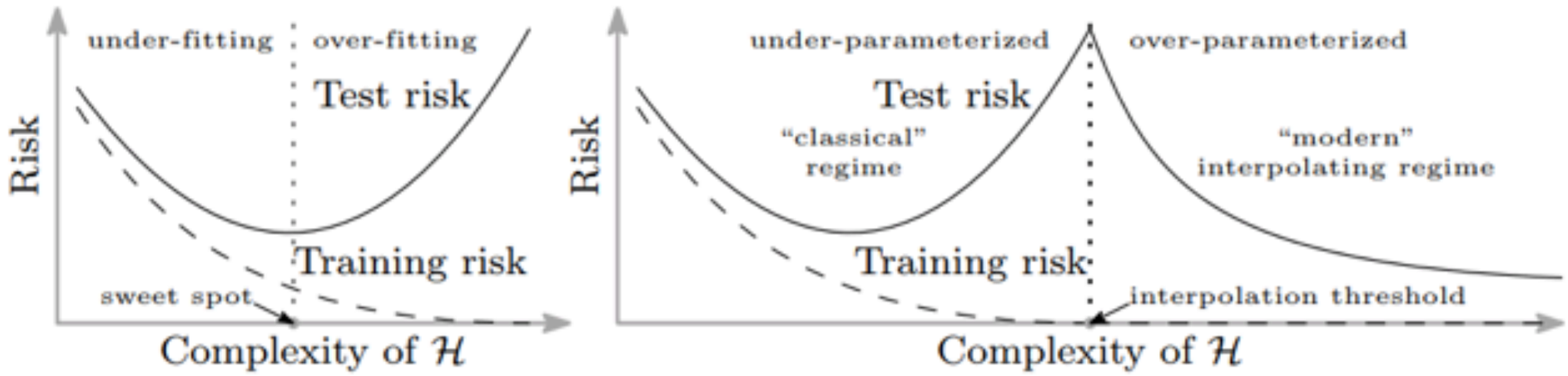




- Mode connectivity is observed even for minima obtained by random permutations of neurons

- Probably the points of global minima of the loss form multi-dimensional manifold

- After reaching zero train loss a drift within this manifold is possible

# Double descent




(a) U-shaped "bias-variance" risk curve

(b) "double descent" risk curve

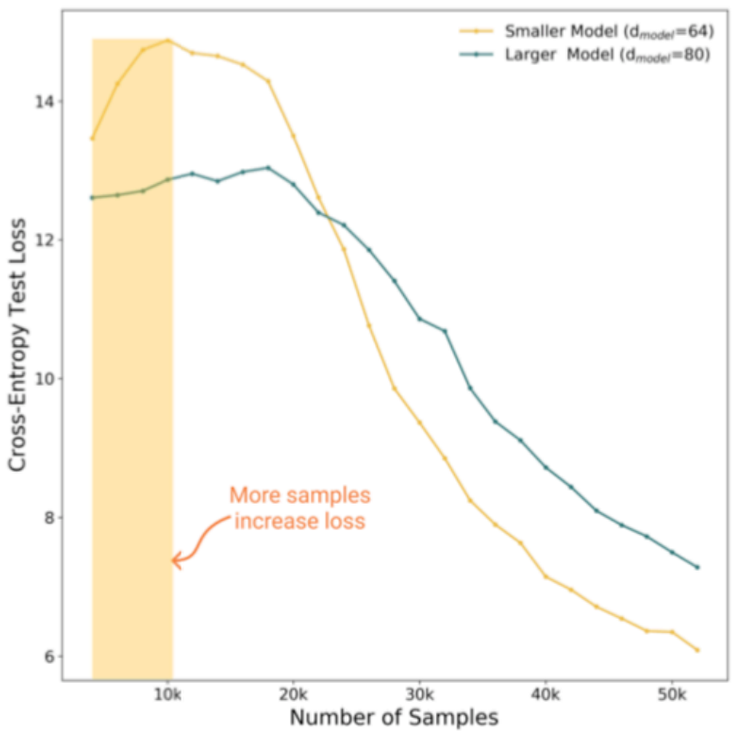
Figure 1: Curves for training risk (dashed line) and test risk (solid line). (a) The classical *U-shaped risk curve* arising from the bias-variance trade-off. (b) The *double descent risk curve*, which incorporates the U-shaped risk curve (i.e., the "classical" regime) together with the observed behavior from using high complexity function classes (i.e., the "modern" interpolating regime), separated by the interpolation threshold. The predictors to the right of the interpolation threshold have zero training risk.

**M. Belkin, D. Hsu, S. Ma, S. Mandal. Reconciling modern machine learning practice and the bias-variance trade-off. Proceedings of the National Academy of Sciences 116 (32), 15849-15854, 2019**

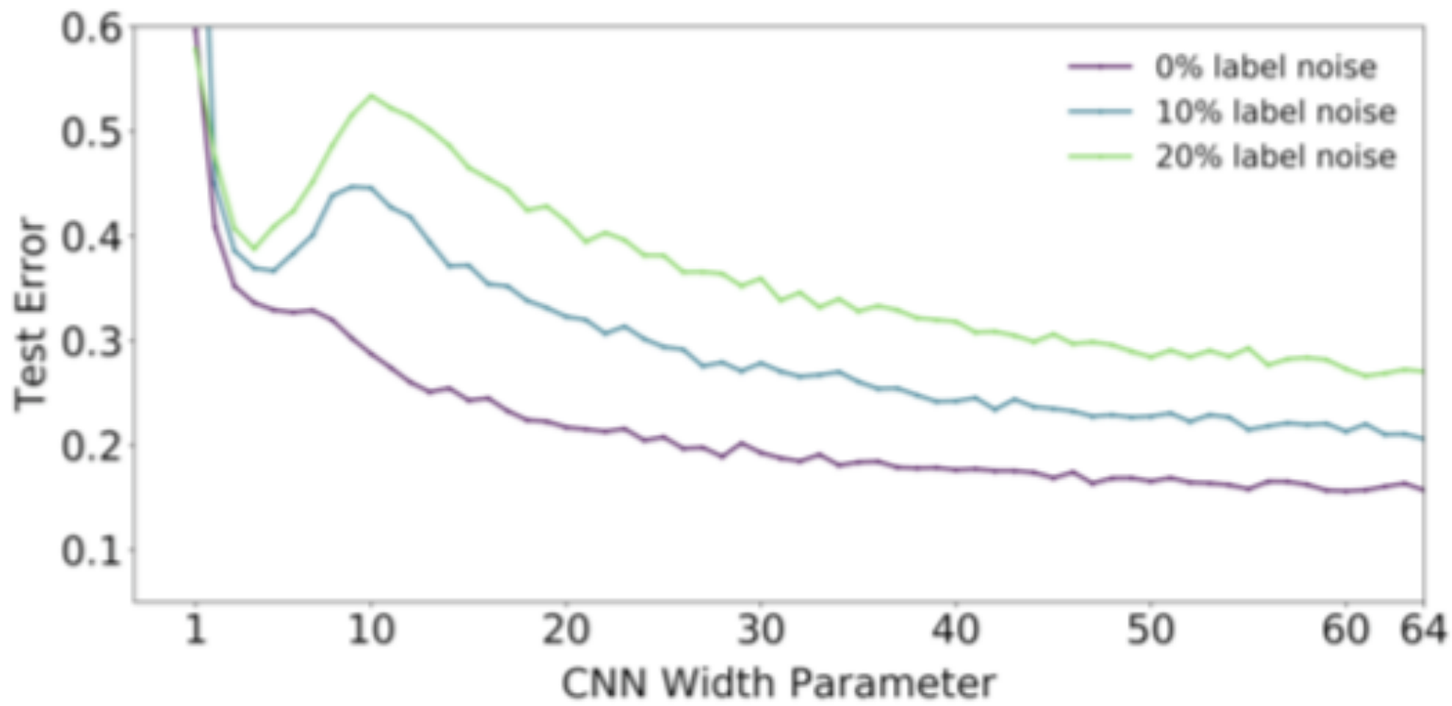
# Deep double descent

Arrival of more training data may worsen the performance for a given model!

The effect is especially observable when there label noise is present in training data




(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.

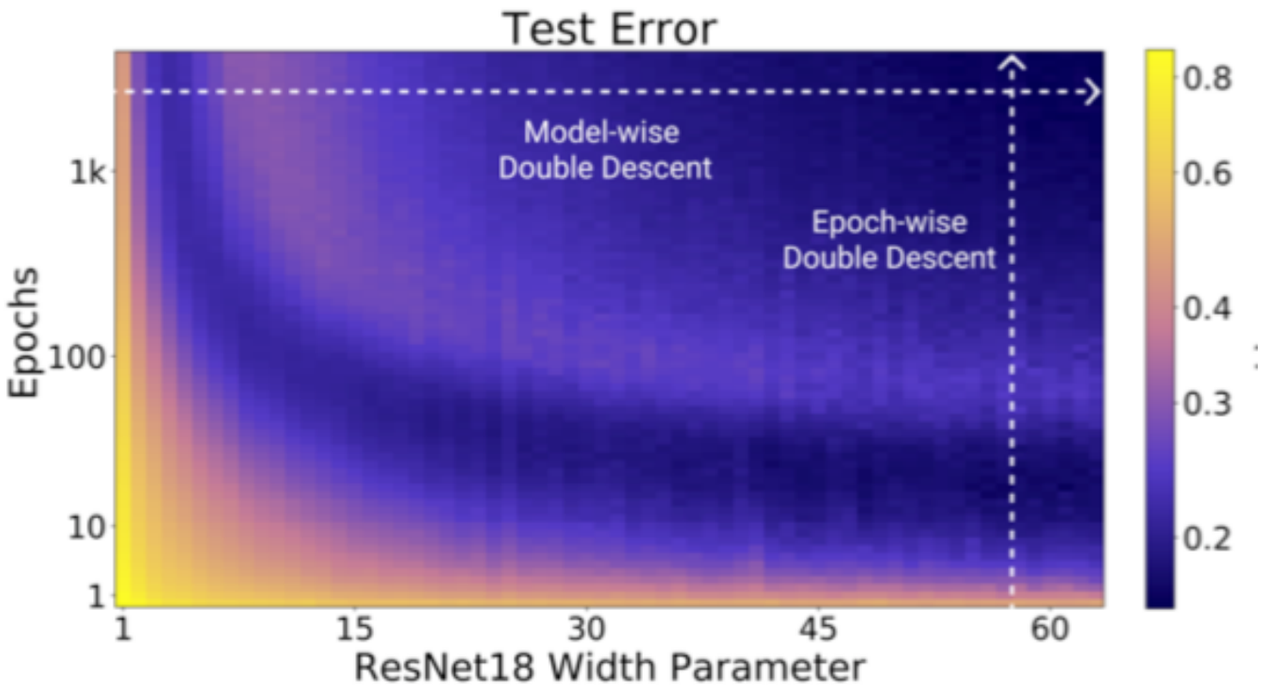


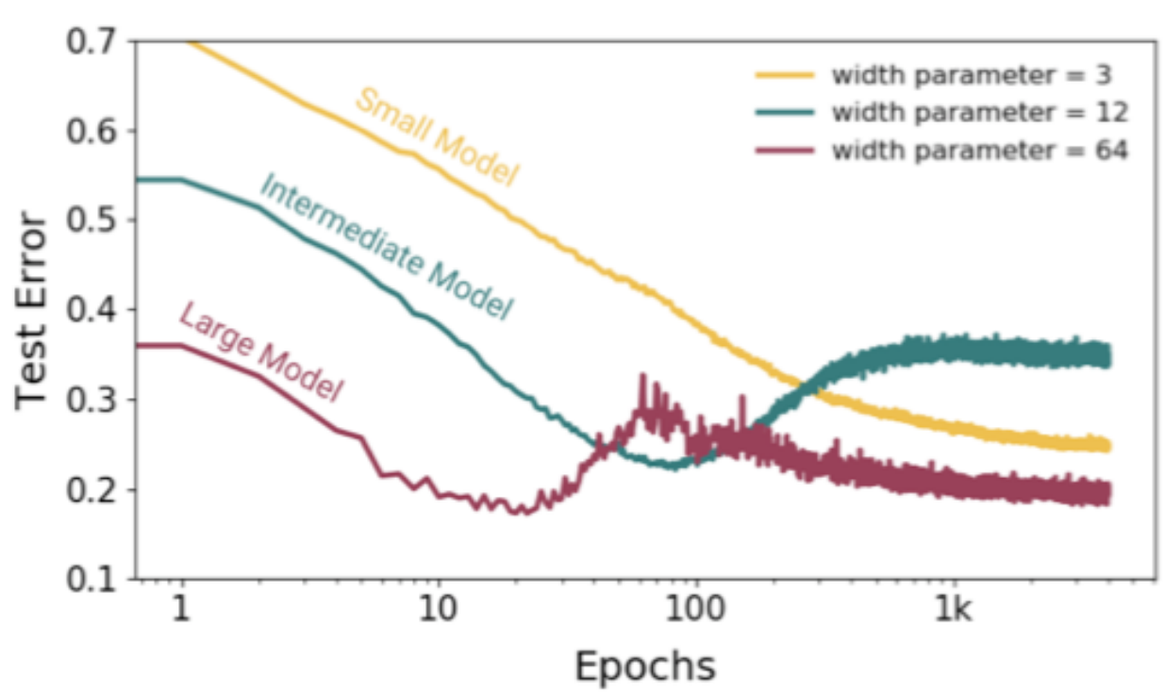

**Nakkiran et al. Deep Double Descent: Where Bigger Models and More Data Hurt. In ICLR 2020**

# Deep double descent

Double descent is also reported w.r.t. training epochs

The peak of test loss/error happens when the model reaches zero training error



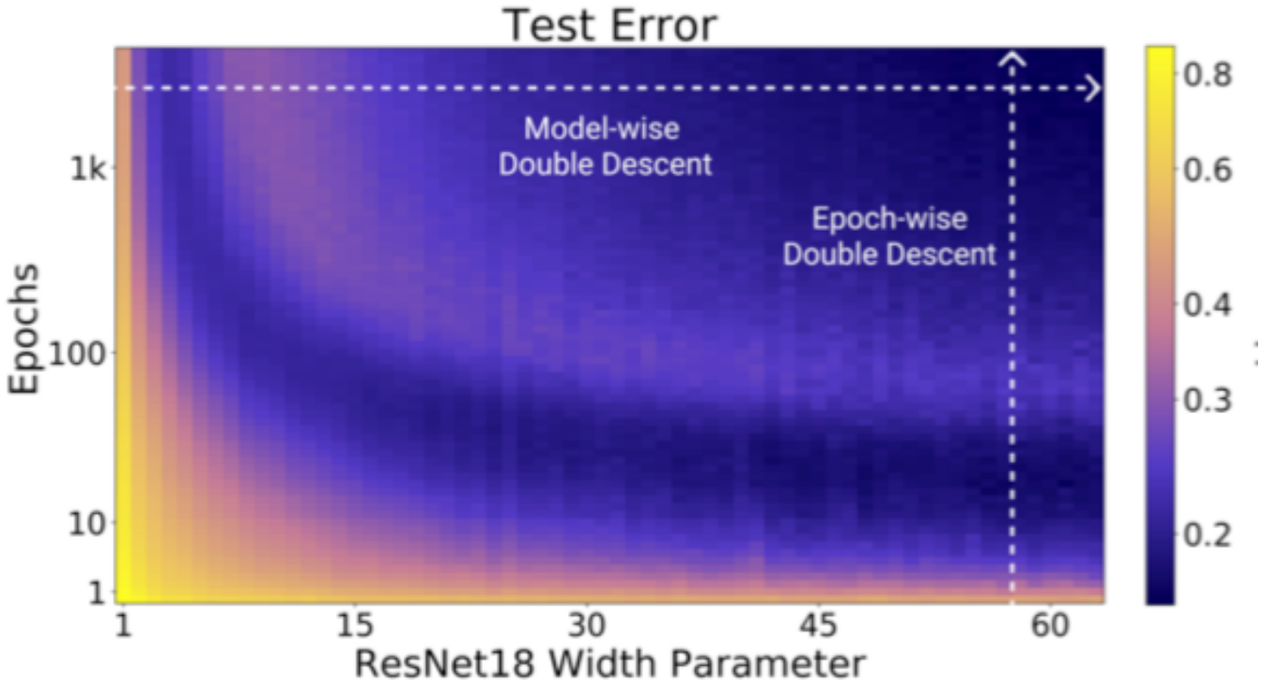



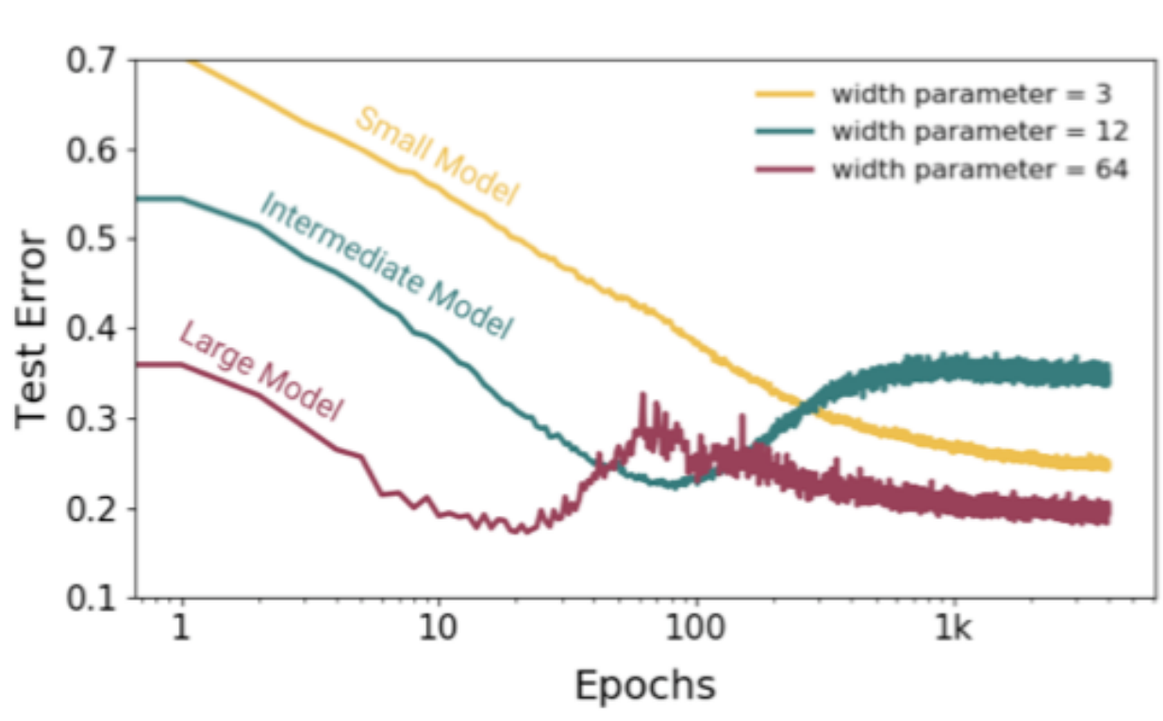

**Nakkiran et al. Deep Double Descent: Where Bigger Models and More Data Hurt. In ICLR 2020**

# Deep double descent

Double descent is also reported w.r.t. training epochs

The peak of test loss/error happens when the model reaches zero training error




Hypothesis: DNN first memorizes training data by finding relatively narrow minimum and then drifts to wider minima thus improving generalization (a kind of memory consolidation phase)

# Explaining DD

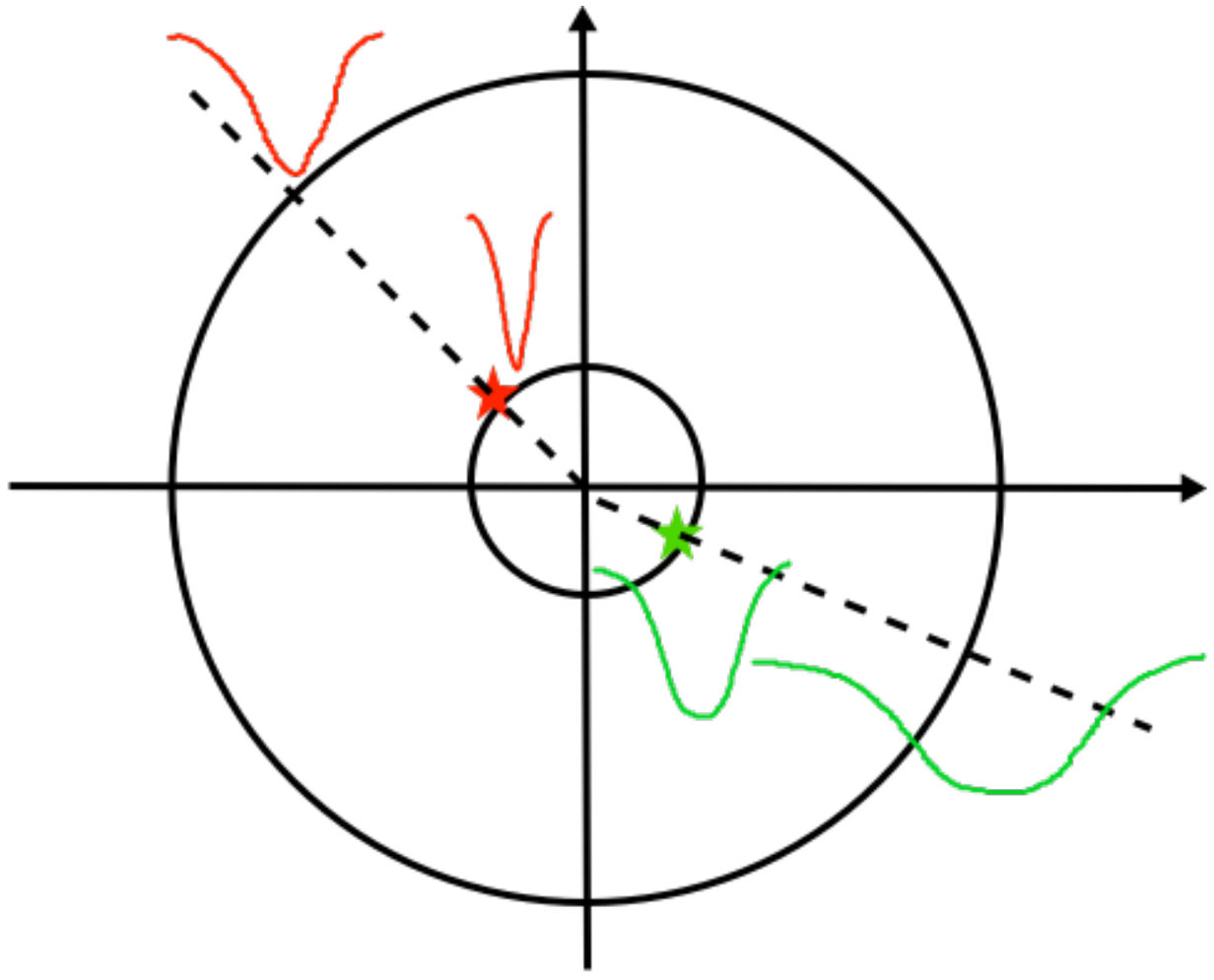
Hypothesis: DNN first memorizes training data by finding relatively narrow minimum and then drifts wider minima thus improving generalization (a kind of memory consolidation phase)

But we cannot find narrow minima using SGD!.. Or can we?

# Scale invariance of DNNs

DNNs with batch-normalization are scale-invariant w.r.t. to their weights

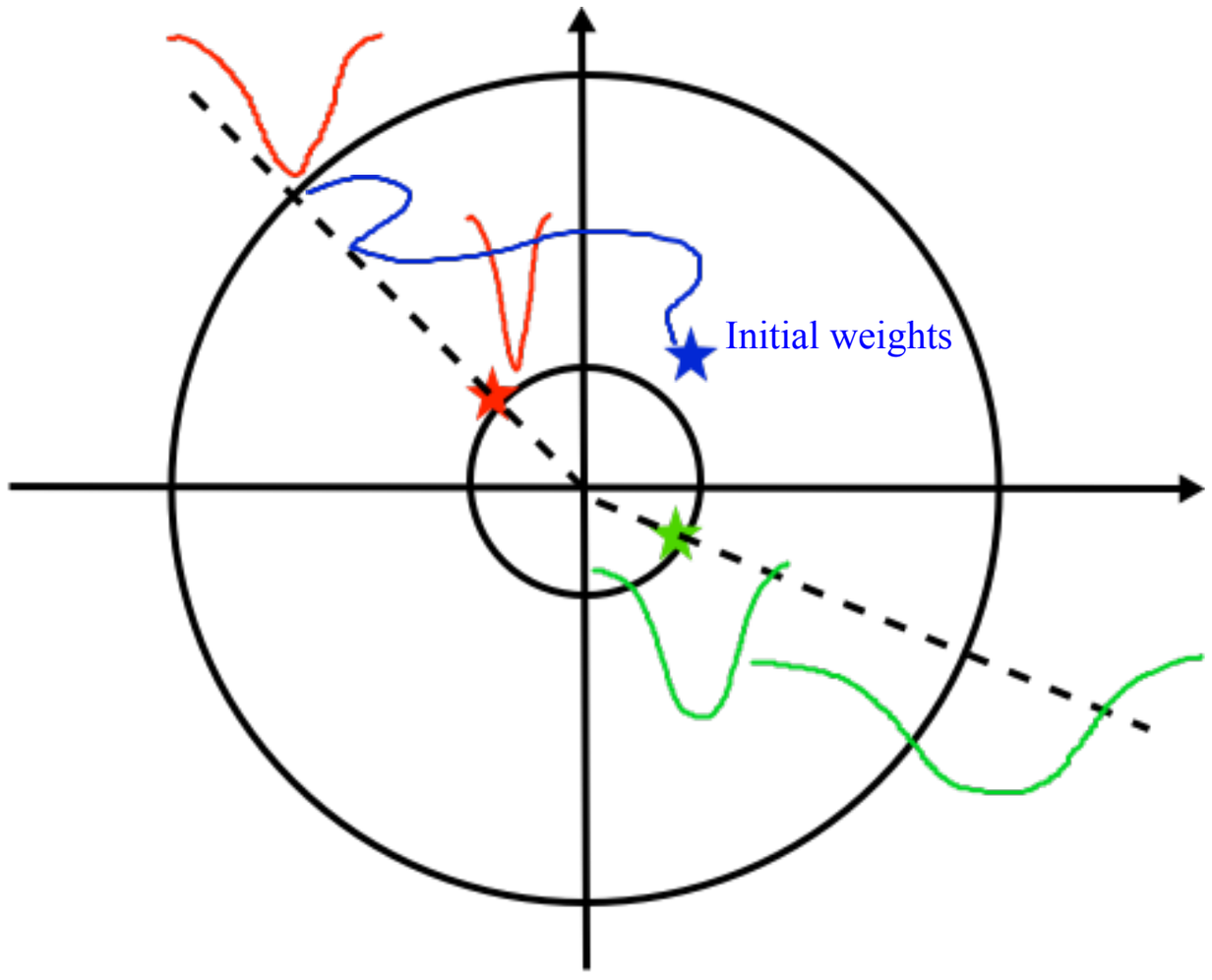
But the width of local minima is not!



# Scale invariance of DNNs

DNNs with batch-normalization are scale-invariant w.r.t. to their weights

But the width of local minima is not!

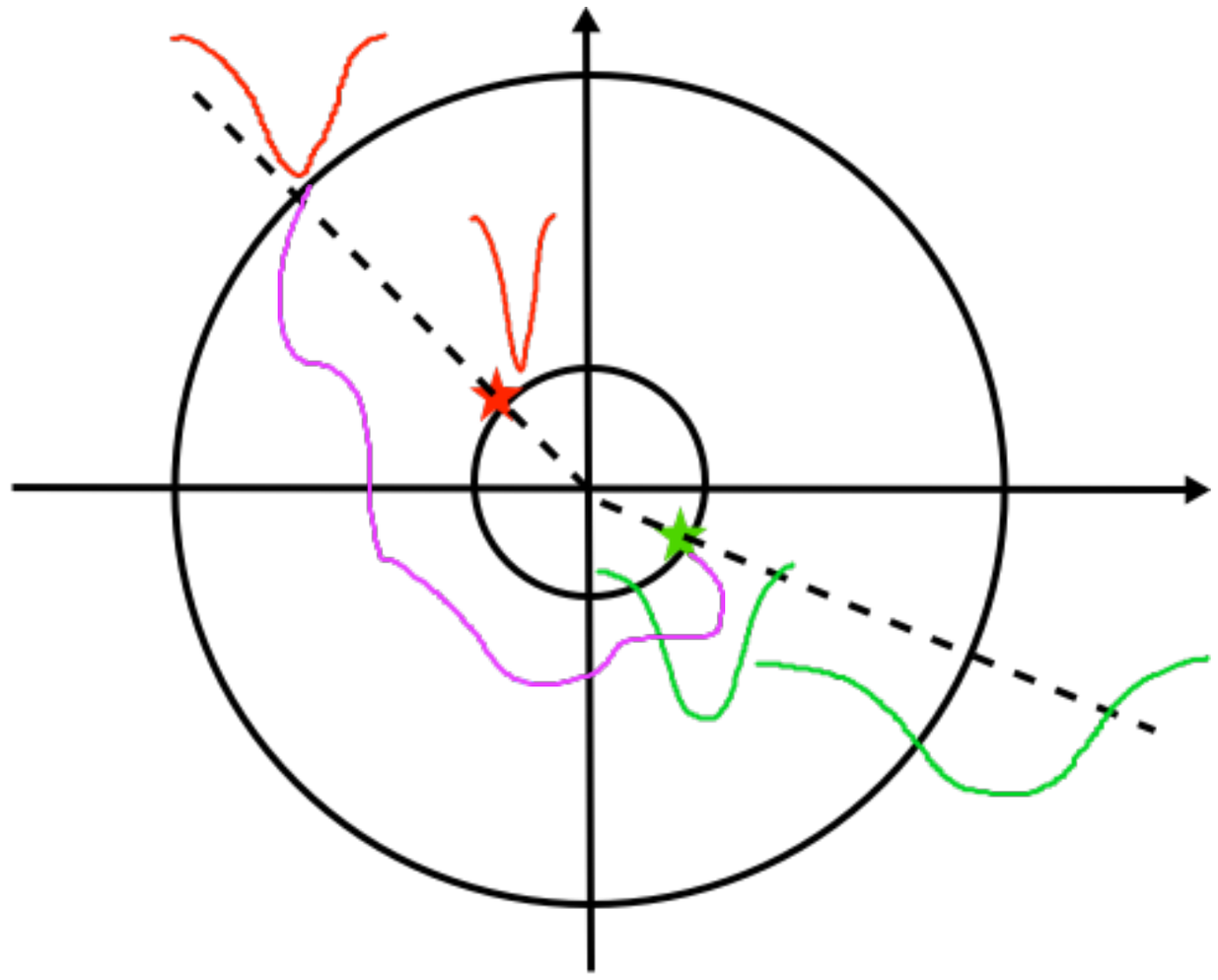



Phase 1: Memorization. Train loss descreases to almost zero. Weight norm increases. The effective width of minimum is small. Test loss peaks

# Scale invariance of DNNs

DNNs with batch-normalization are scale-invariant w.r.t. to their weights

But the width of local minima is not!



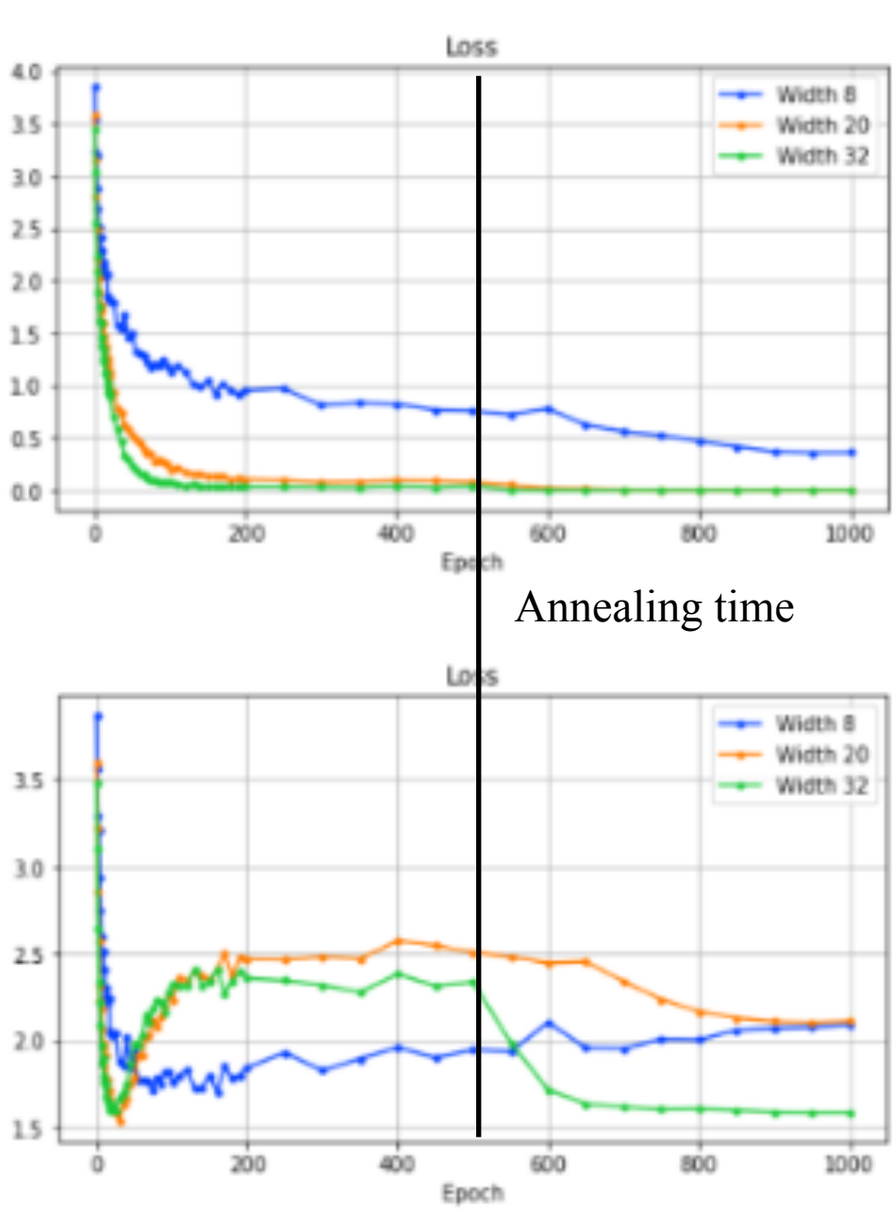
Phase 2: Consolidation. Weight norm drops. Effective width of minimum increases. Test loss improves

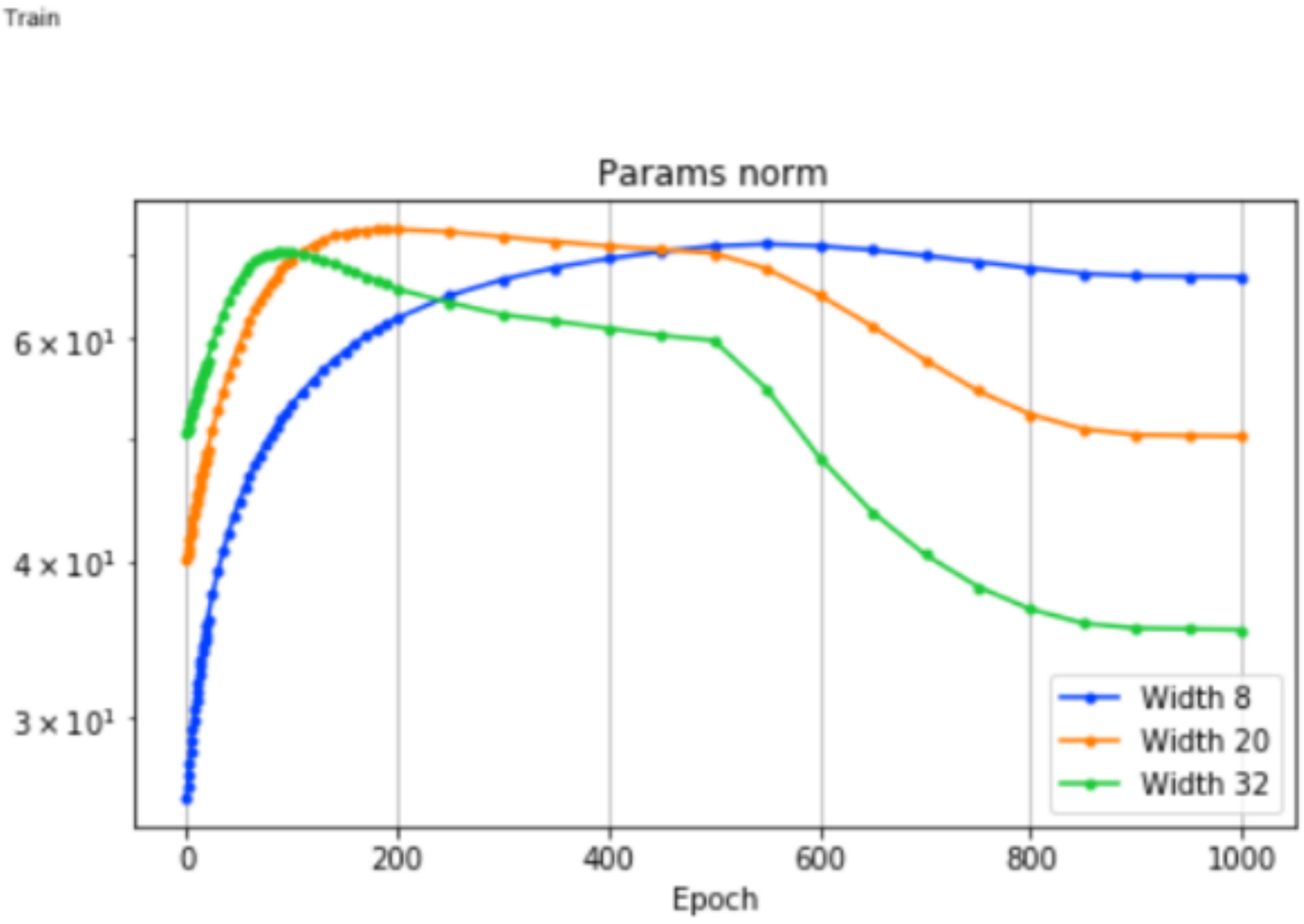
# Experiment 1

CIFAR100 data, 0% label noise, ResNet architecture, LR annealing after 500 epochs

Train

Test



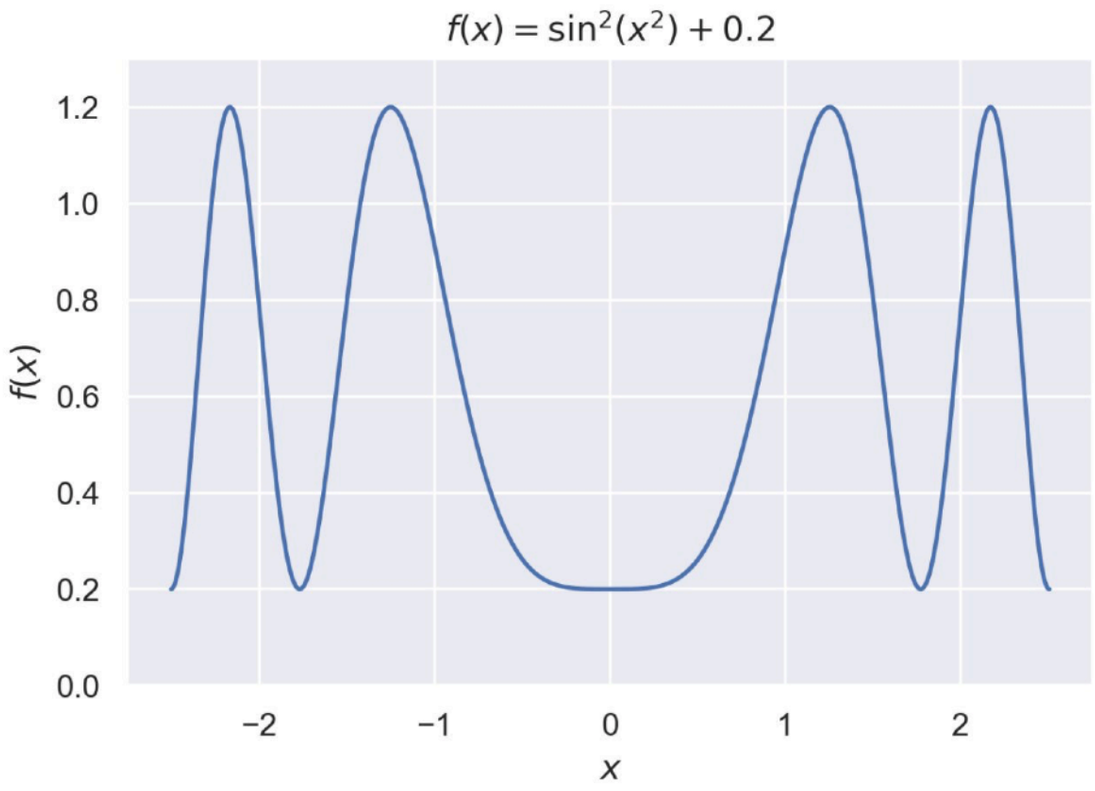



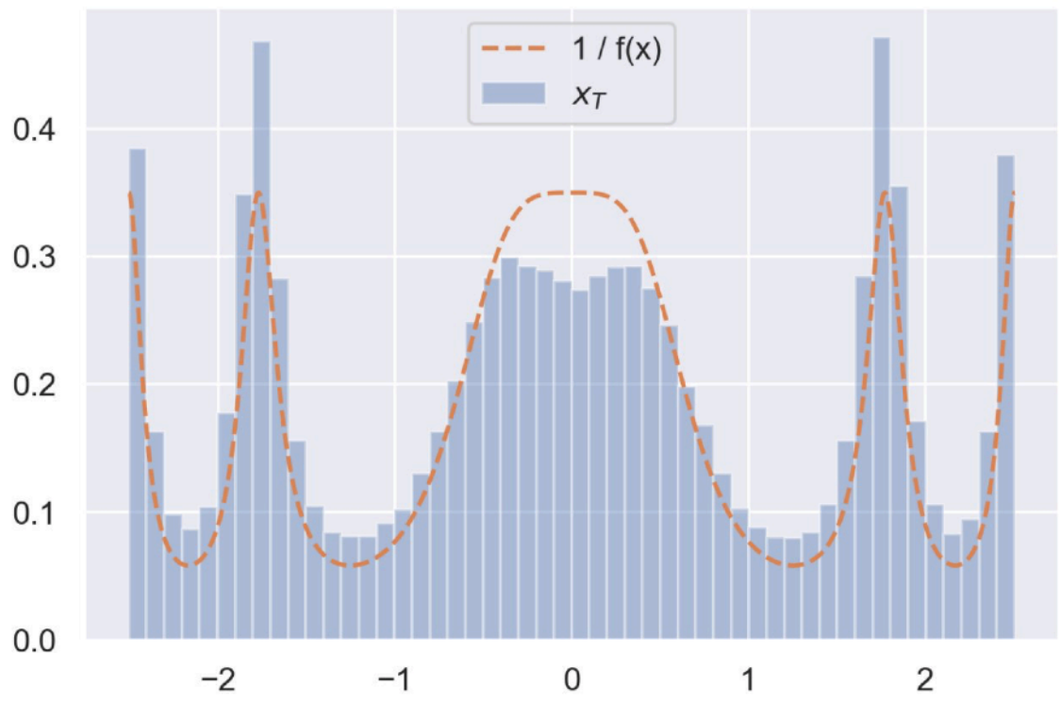
# Experiment 2

Consider drift (random walk) along the manifold of (almost-)zero train loss

The higher the curvature the larger is variance of zero-mean stochastic gradient

We can derive the stationary distribution of random walk with changing variance



# Generalization gap

- Statistical learning theory provides us upper bounds for the gap between train and test loss

- The result is derived assuming that we are able to find global minimum of train loss

- But are we really solving an optimization problem when training DNNs?

# Langevin dynamics

Assume that we want to sample from distribution *p(x)*. We may solve the following stochastic differential equation (SDE):

$$dx = \frac{1}{2}\frac{d\log p(x)}{dx}dt + dW$$

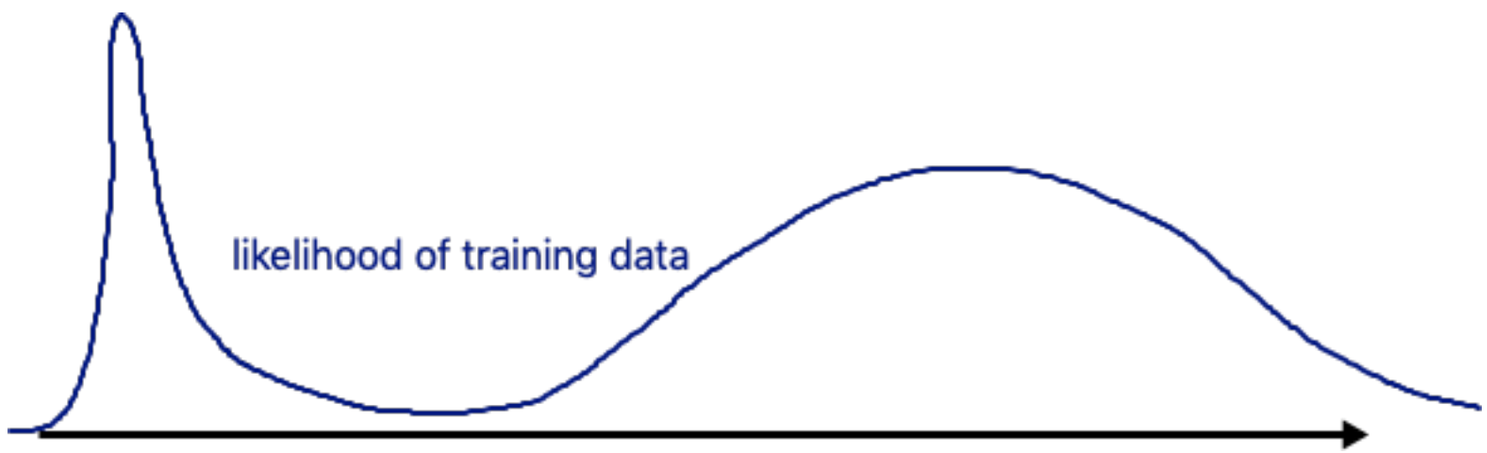
Here *W(t)* is standard Wiener process. After some warm-up each point from the trajectory is a sample from *p(x)*

But when we do SGD we almost solve this SDE numerically

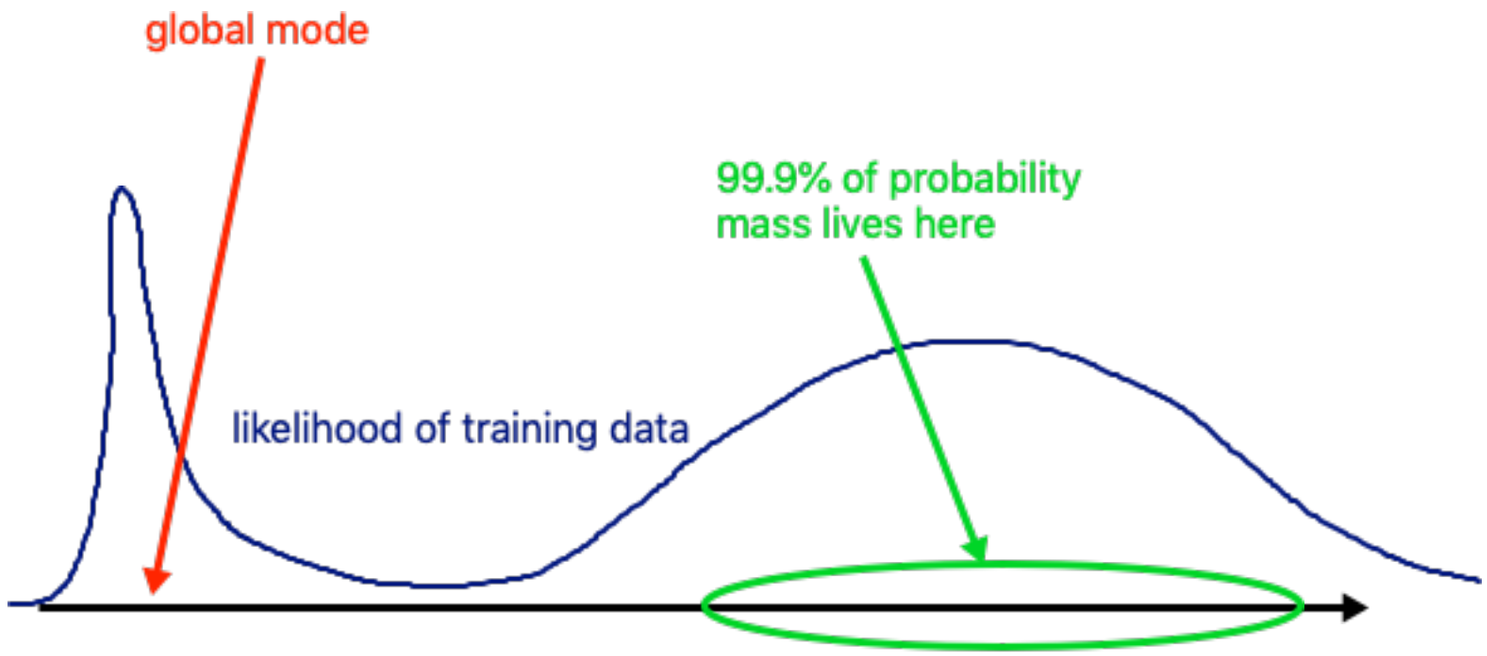
$$w(t+1) = w(t) + \eta_t \frac{d}{dw}\log p(D_{train}\,|\,w) + \epsilon, \quad \text{where} \quad \epsilon \sim \mathcal{N}(\epsilon\,|\,0,\eta_t\Sigma_t)$$

This means that SGD returns samples from some implicit distribution induced by training likelihood rather than its mode (i.e. global minimum of train loss)

# Statistical mechanics



# Statistical mechanics




SGD-like methods return the typical samples induced by training likelihood rather than the global mode which obeys the results from SLT and may have arbitrary poor generalization

We need a new mathematical framework to study the generalization gap of typical samples from a distribution (statistical mechanics could be adopted?)

It will probably state that the increase of model's size decreases the gap

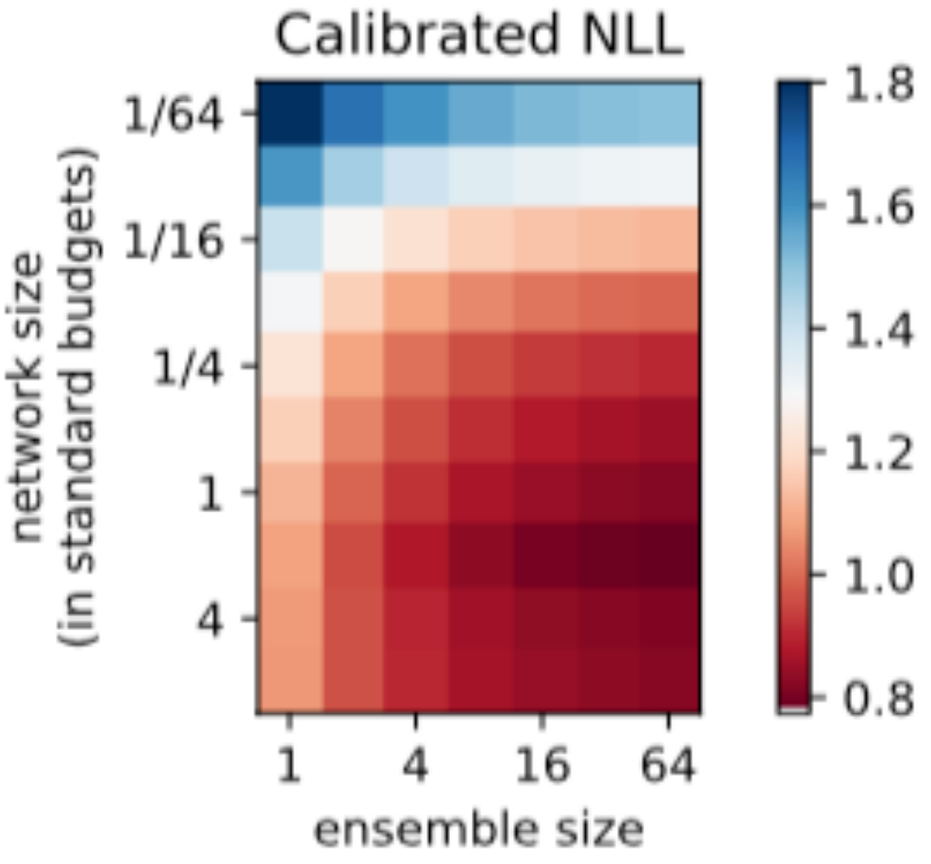
We will have to describe the properties of a distribution depending on learning rate, batch size, etc.

# Increasing the width and ensemble size

It is widely known that ensembles of DNNs have better accuracy

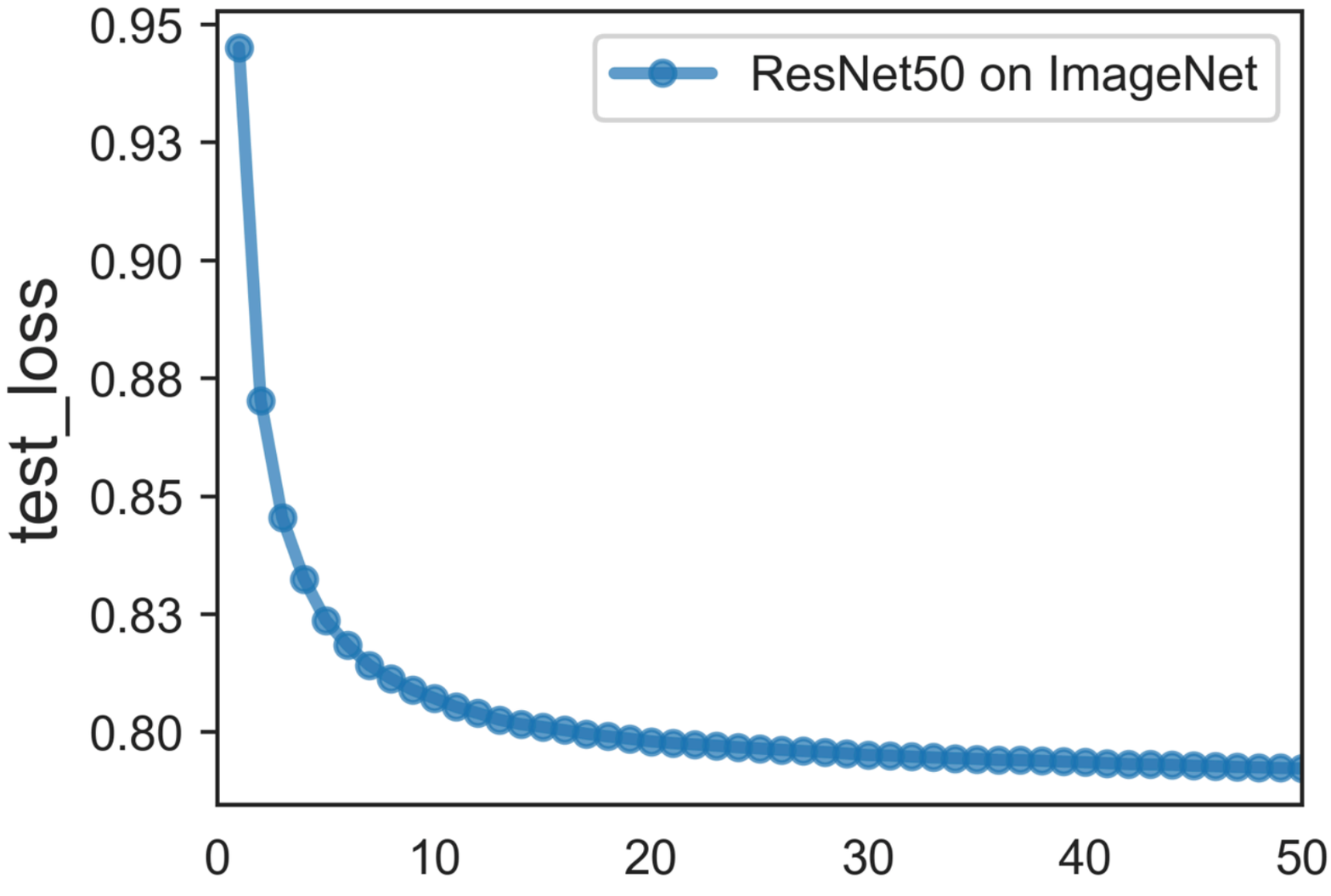
Empirically we also observe that wider DNNs work better

In both cases we see a clear saturation effect



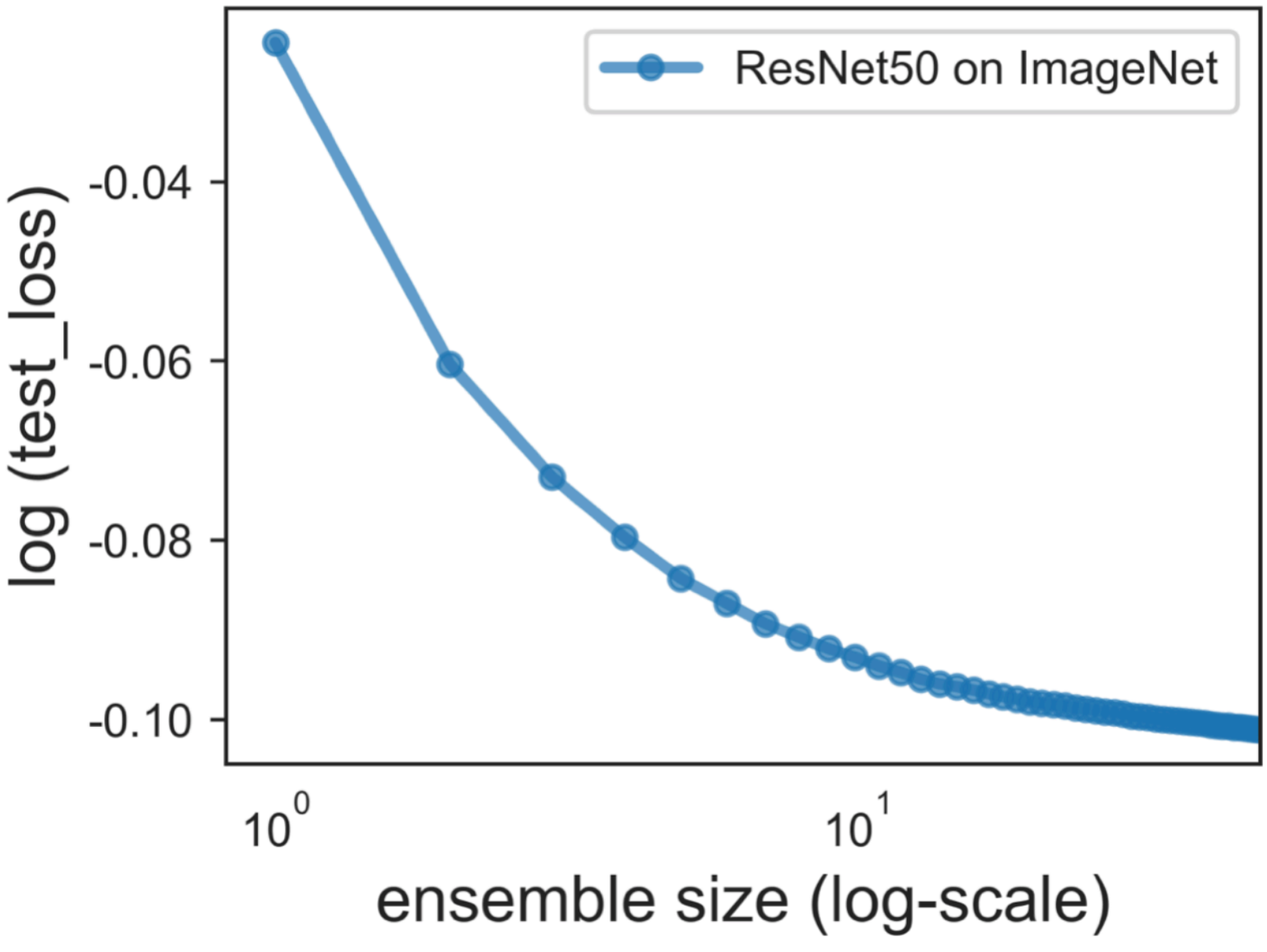
# Deep Ensembles

**What is the maximal possible performance that can be achieved by using infinitely large DE of networks with given structure?**



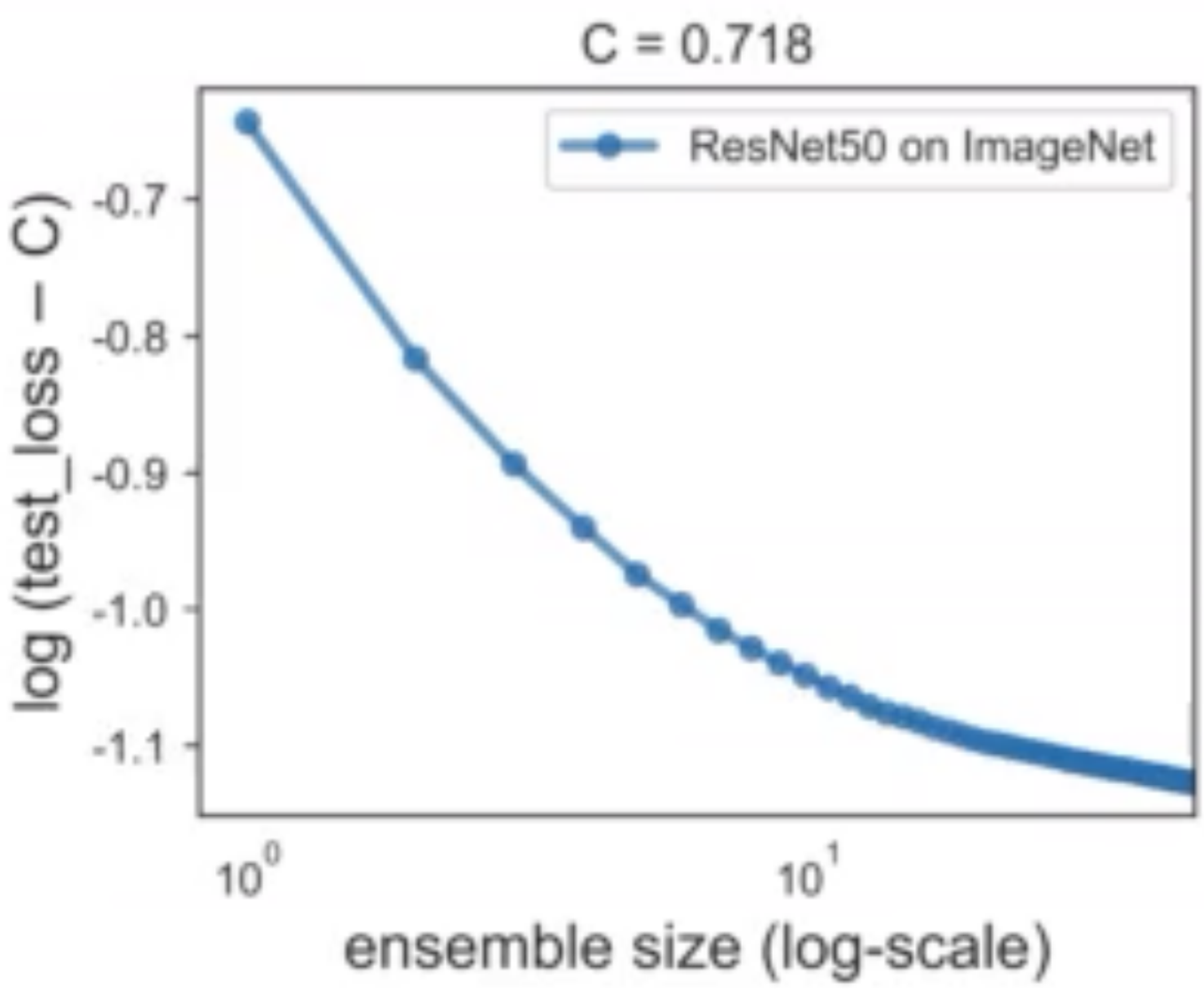
# Deep Ensembles

**What is the maximal possible performance that can be achieved by using infinitely large DE of networks with given structure?**



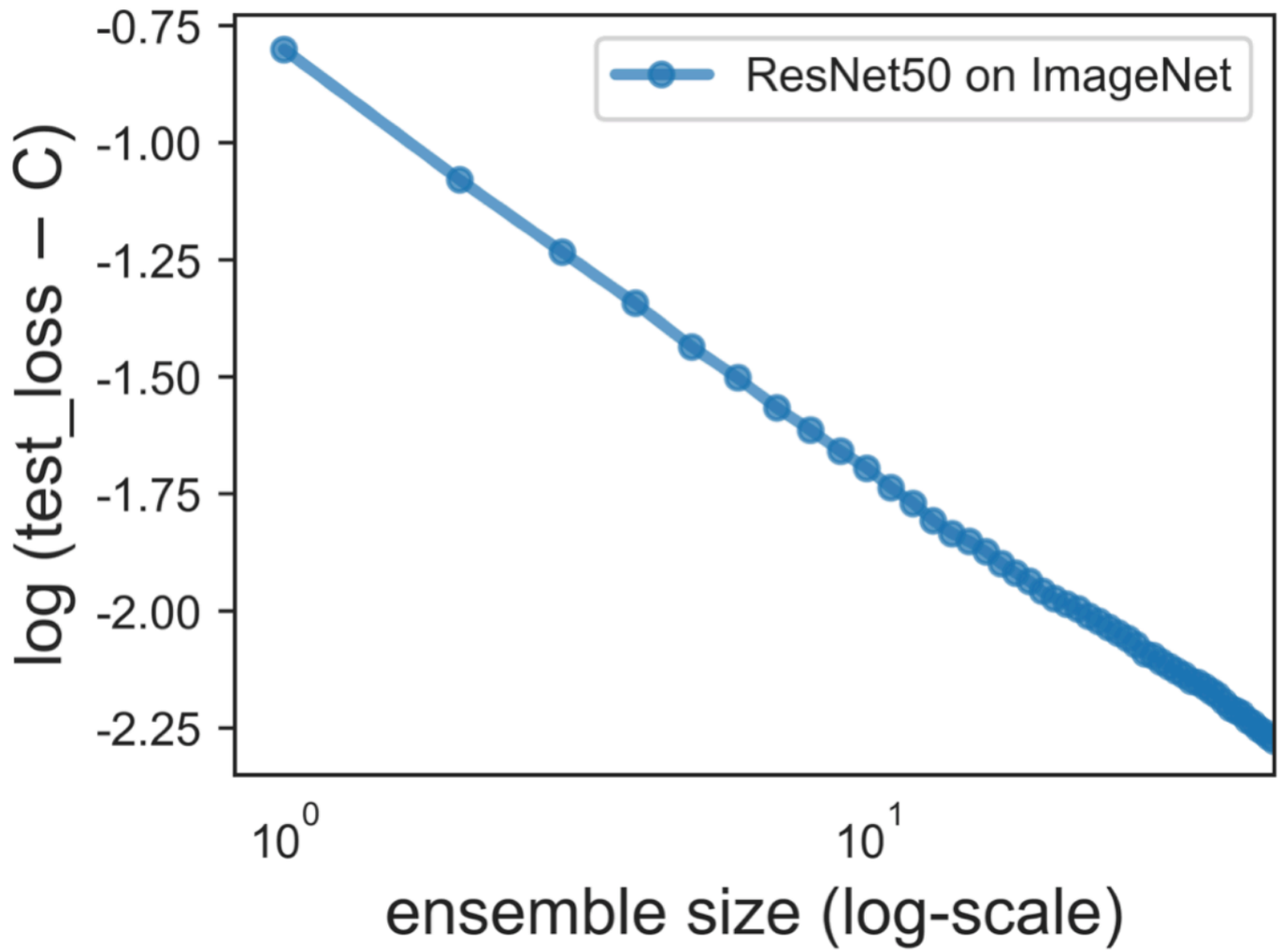
# Deep Ensembles

**What is the maximal possible performance that can be achieved by using infinitely large DE of networks with given structure?**



# Deep Ensembles

**What is the maximal possible performance that can be achieved by using infinitely large DE of networks with given structure?**



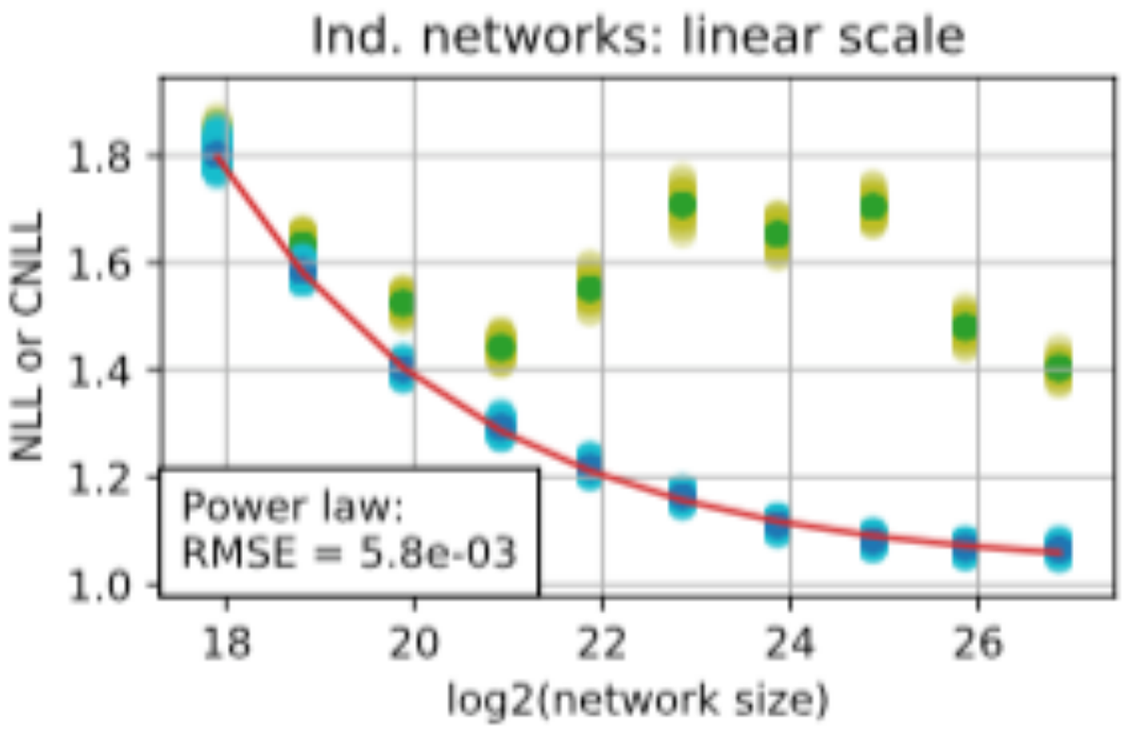
# Power law wrt width of DNN

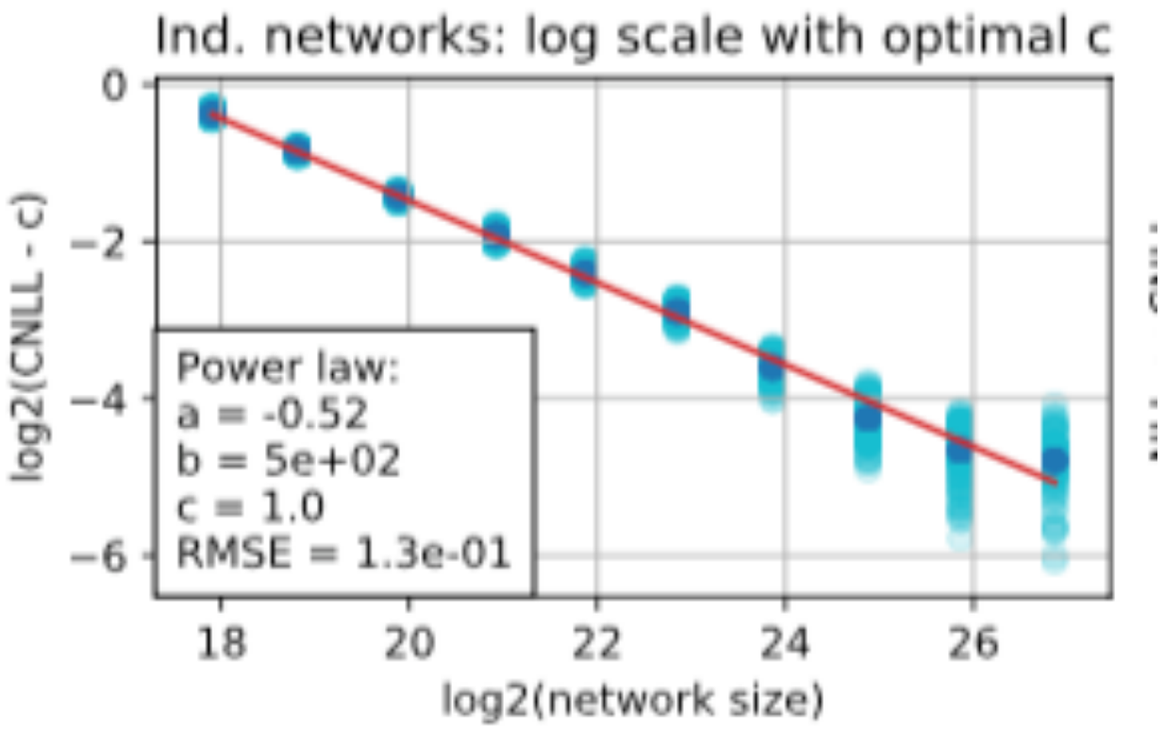
After temperature calibration we can see a clear power law

$$\vec{p}_{cal}(D \mid \tau) = \text{softmax}\left(\frac{1}{\tau} \log \vec{p}_{unc}(D)\right)$$

$$\tau = \arg\max_{\tau} \log p(D_{val} \mid \tau)$$

Interestingly calibration removes double descent behaviour



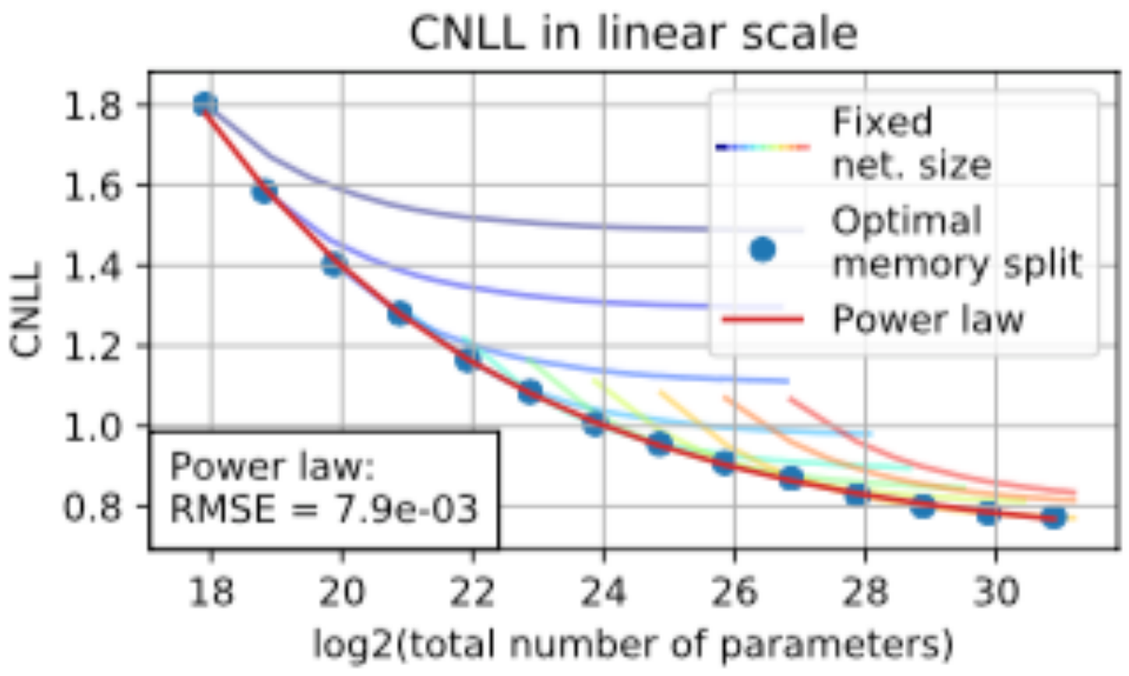



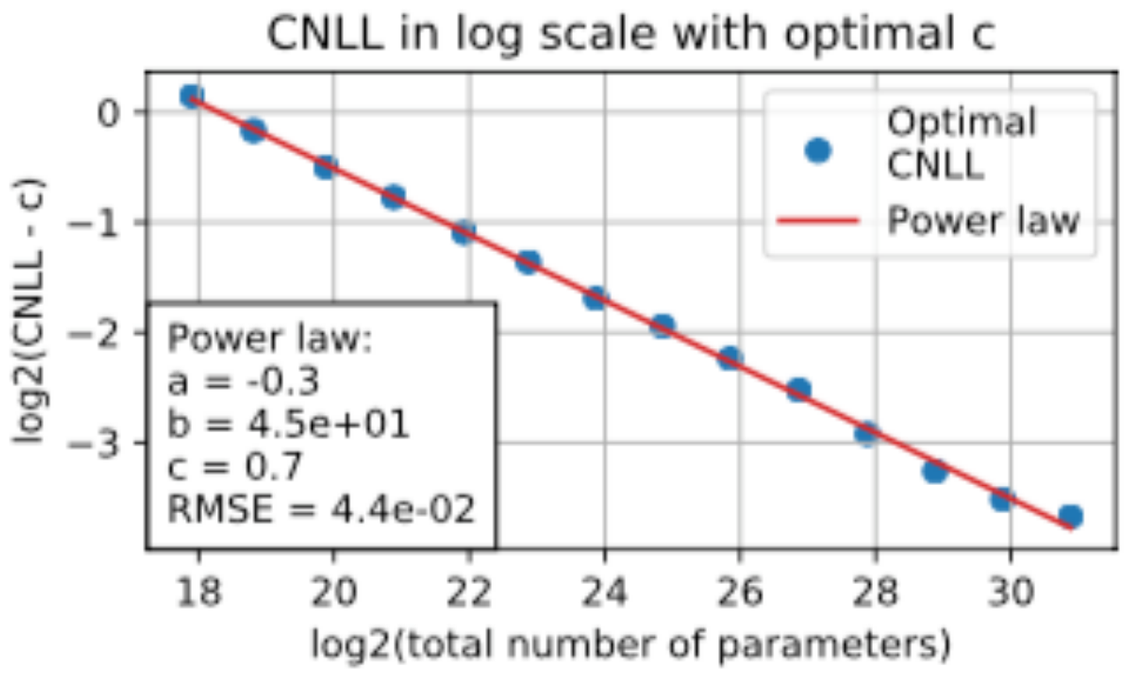
# Memory split effect

Ensembles of DNNs of different width have different power laws

Interestingly their lower envelope also obeys a power law

For a given memory budget there exists an optimal width and ensemble size



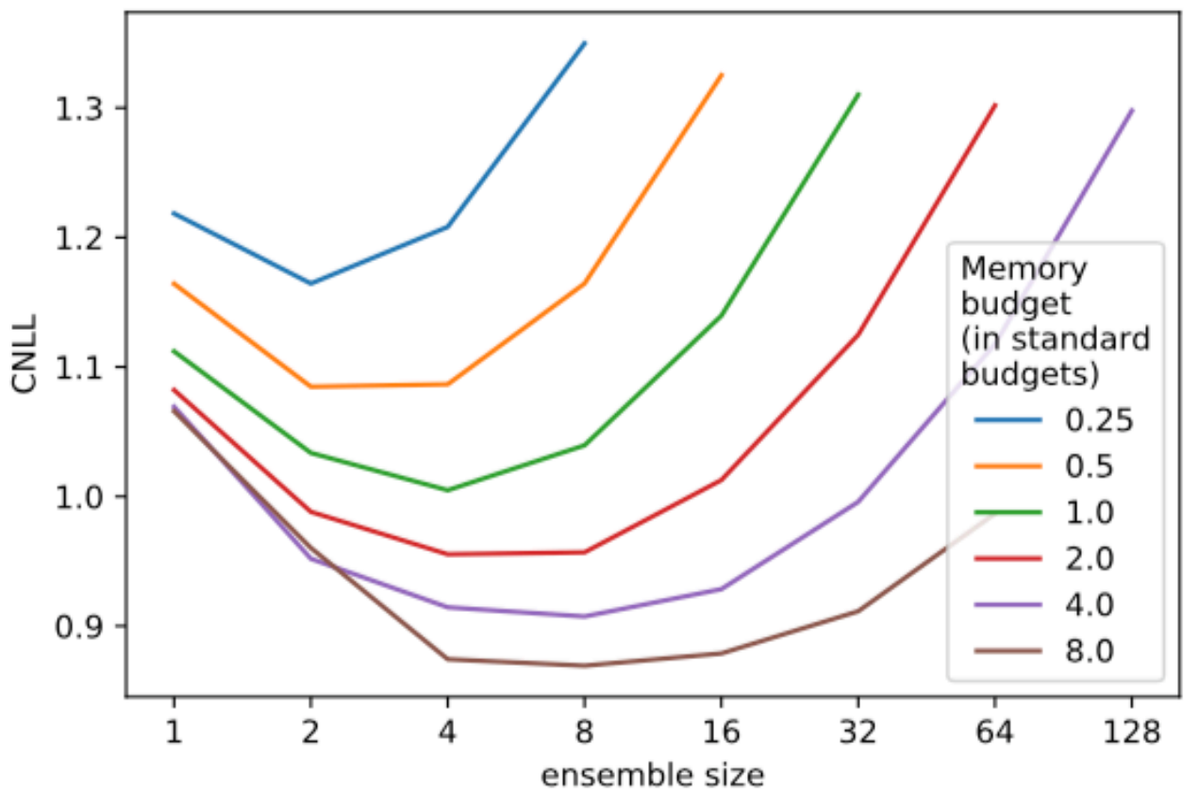



# Memory split effect

Ensembles of DNNs of different width have different power laws

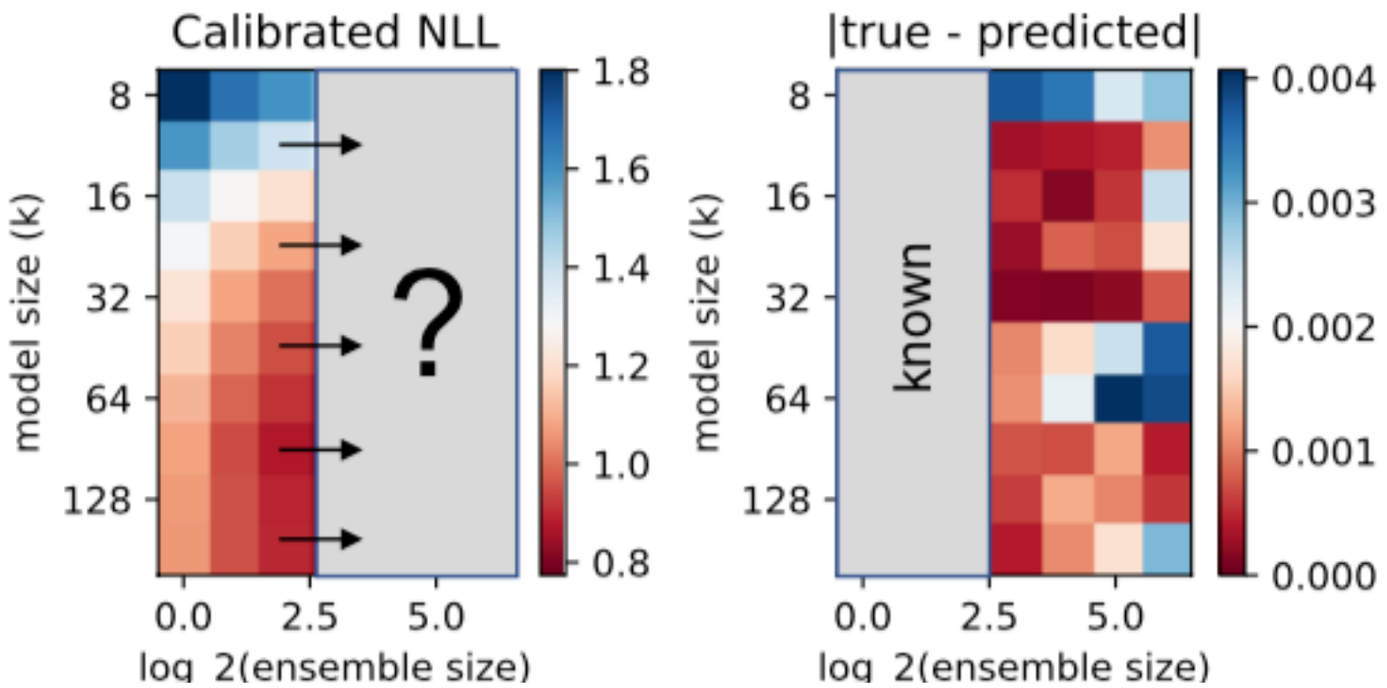
Interestingly their lower envelope also obeys a power law

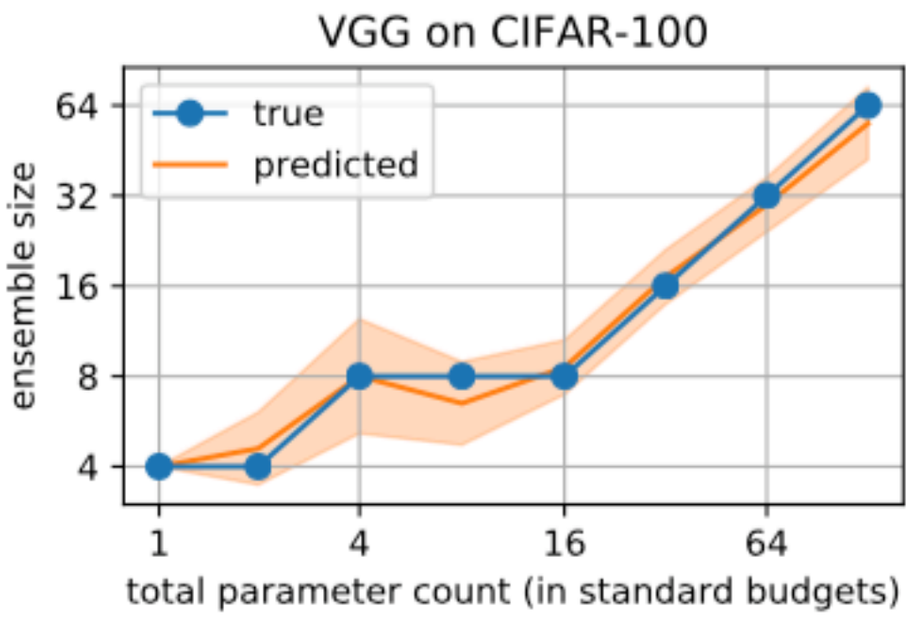
For a given memory budget there exists an optimal width and ensemble size

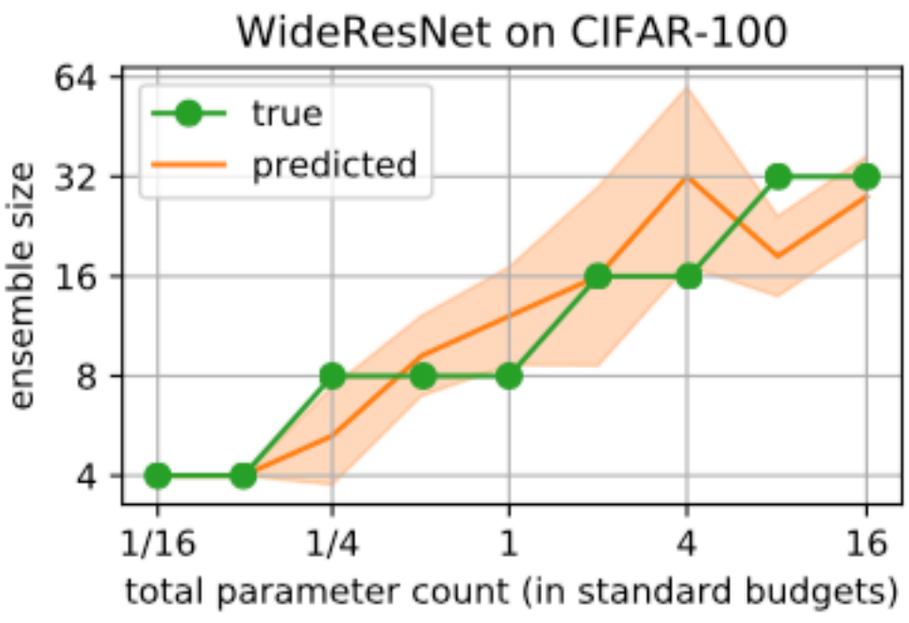



The larger is budget the more networks are required in ensemble

# Optimal split prediction



# Conclusion

- We understand very little of what's going on when we train DNNs

- SGD has an inductive bias towards good solutions. Still need to realize which one

- New framework is needed to predict generalization. The bounds should depend on optimization/sampling hyperparameters

- Larger models are somehow less complicated when trained with stochastic optimization

**E. Lobacheva, N. Chirkova, M. Kodryan, D. Vetrov. On Power Laws in Deep Ensembles. Submitted to NeurIPS2020**