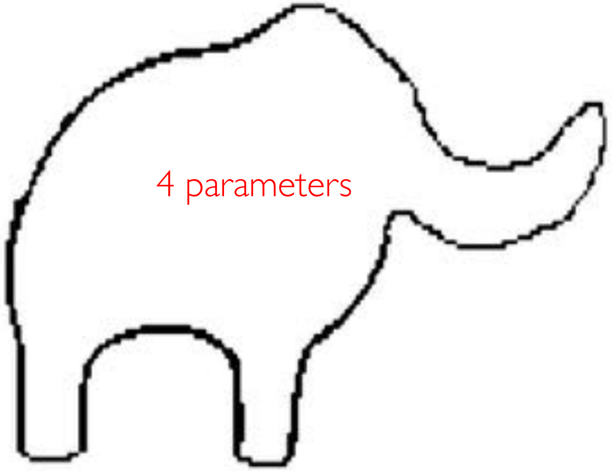
Deep neural networks have an inbuilt Occam's razor

Ard Louis

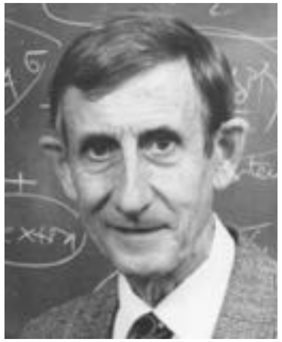
UNIVERSITY OF OXFORD

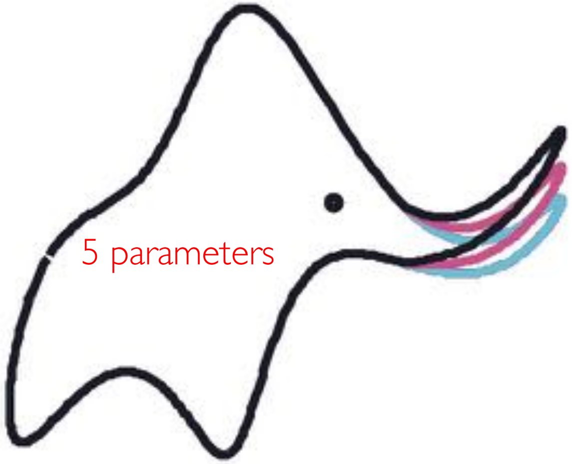
Physicists are taught: more parameters than data points is bad

F. Dyson, *A meeting with Enrico Fermi,* Nature. **427**, 287 (2004)




Enrico Fermi
1901-1954


Freeman Dyson
1923-2020

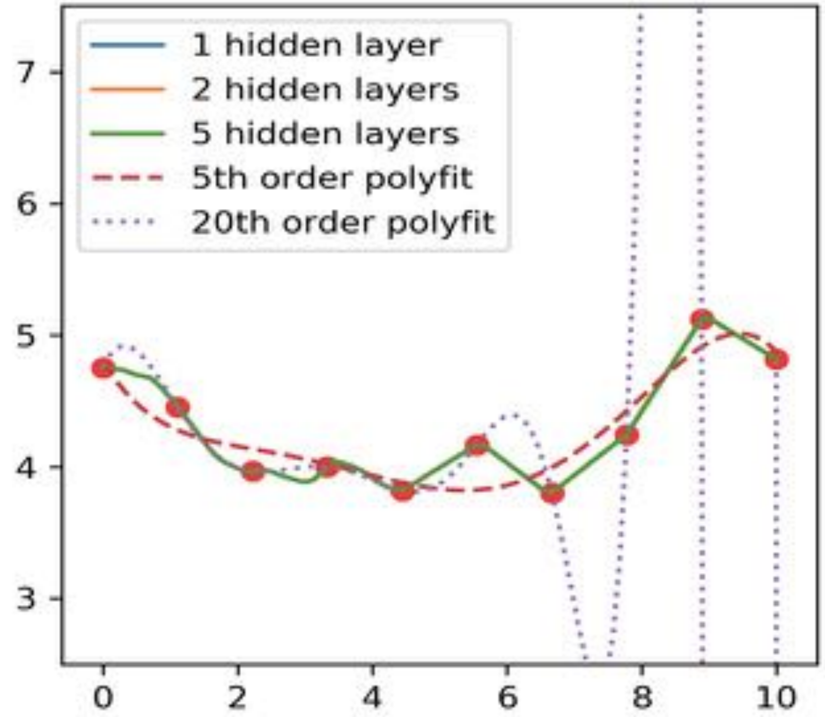



John von Neumann
1903-1957

With four parameters I can fit an elephant, and with five I can make him wiggle his trunk
-- John von Neuman (according to Fermi)

Drawing an elephant with four complex parameters
Jürgen Mayer; Khaled Khairy; Jonathon Howard; American Journal of Physics 78, 648-649 (2010)

Deep neural networks (DNNs) are heavily overparameterized



polynomial fit : $y(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \ldots a_nx^n$

compared to

simple DNNs (FCN with layer width of 1000 units)

CENTRAL THEORETICAL CONUNDRUM of DNNs: Why do they generalise so well?

1) DNNs are used in the over-parameterised regime with many more parameters than data points.
2) DNNs are highly expressive (there is a universal approximation theorem (Cybenko, Hornik etc..)
3) Classical learning theory, based on model capacity, predicts poor generalisation. (bias-variance tradeoff)

Deep learning and Physics

Comment | Published: 26 May 2020

Understanding deep learning is also a job for physicists

Lenka Zdeborová

Nature Physics **16**, 602–604(2020) | Cite this article

4148 Accesses | **170** Altmetric | Metrics

Automated learning from data by means of deep neural networks is finding use in an ever-increasing number of applications, yet key theoretical questions about how it works remain unanswered. A physics-based approach may help to bridge this gap.

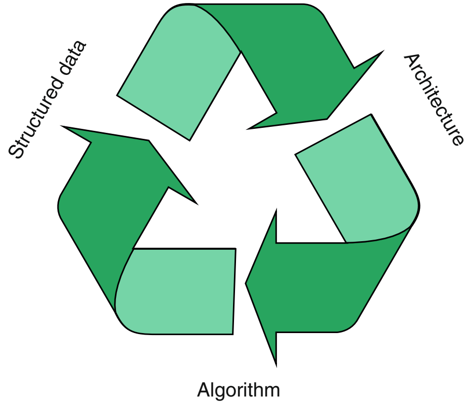


Fig. 1 | Interplay of key ingredients. Building theory of deep learning requires an understanding of the intrinsic interplay between the architecture of the neural network, the behaviour of the algorithm used for learning and the structure in the data.

To understand deep learning, the machine-learning community needs to fill the gap between the mathematically rigorous works and the end-product-driven engineering progress, all while keeping the scientific rigour intact. And this is where the physics approach and experience comes in handy. The virtue of physics research is that it strives to design and perform refined experiments that reveal unexpected (yet reproducible) behaviour, yet has a framework to critically re-examine and improve theories explaining the empirically observed behaviour.

In 1995, the influential statistician Leo Breiman summarized three main open problems in machine learning theory[7]: "Why don't heavily parameterized neural networks overfit the data?

While Breiman formulated thes questions 25 years ago, they are still open today and subject to most of the ongoing works in the learning-theory community,

Model problem: Supervised learning of a Boolean function with DNNs

Doctor's decision table for COVID-19

Boolean function

Send to hospital?	Fever?	Cough?	Lost sense of smell?	Over 50?	Heart problem?	Obese?	Diabetes?
1	1	1	1	1	1	1	1
1	1	1	1	0	1	1	1
0	1	1	1	0	0	0	0
1	1	1	1	1	0	1	1
0	1	1	0	1	0	1	1

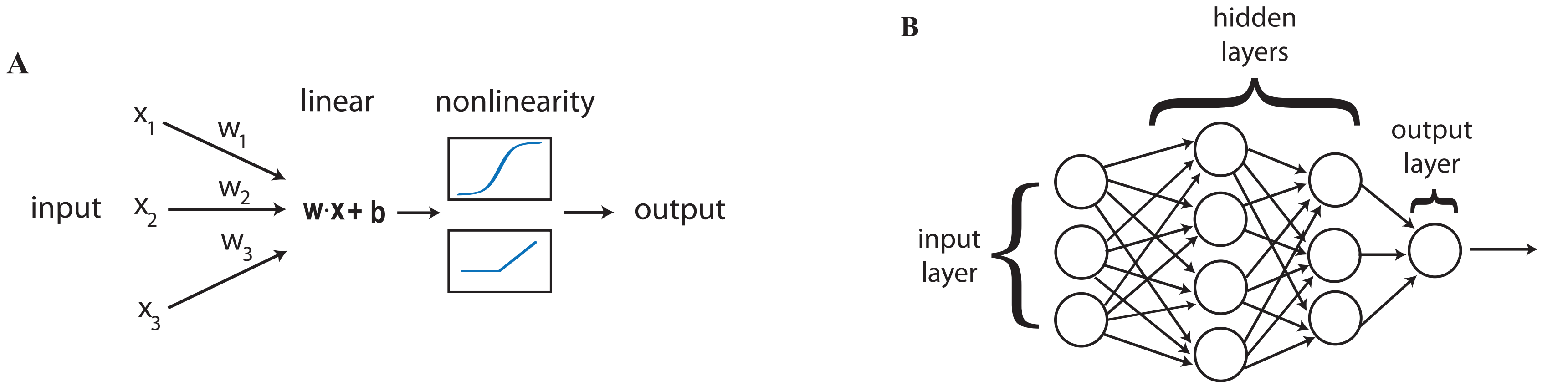
Given some examples, can we learn the rest of the function?

A function maps all possible answers to outputs.

n questions; 2^n possible answers; 2^{2^n} possible Boolean functions

For n=7 $2^7 = 128$ answers; $2^{128} = 3.4 \times 10^{38}$ possible functions

Parameter-function map



Let the space of functions that the model can express be $\mathcal{F}$. If the model has p real valued parameters, taking values within a set $\Theta \subseteq \mathbb{R}^p$,
the parameter-function map, $\mathcal{M}$, is defined as:

$$\begin{aligned} \mathcal{M} : \Theta &\to \mathcal{F} \\ \theta &\mapsto f_\theta \end{aligned}$$

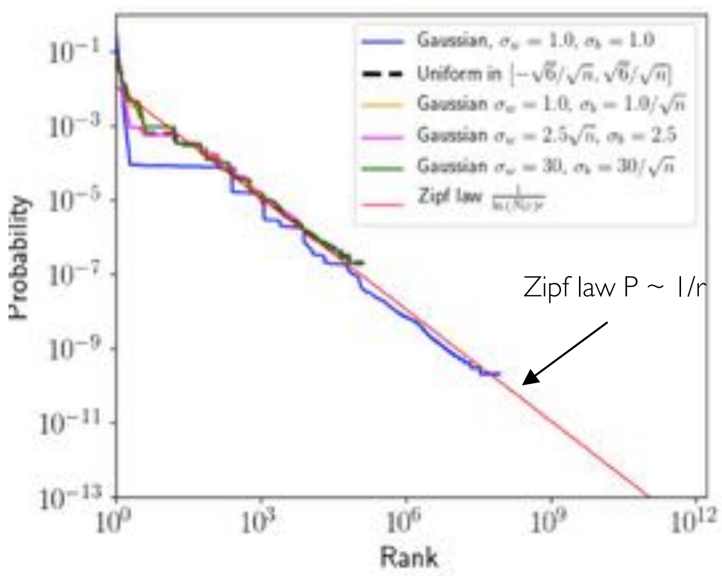
where f_θ is the function implemented by the model with choice of parameter vector θ.

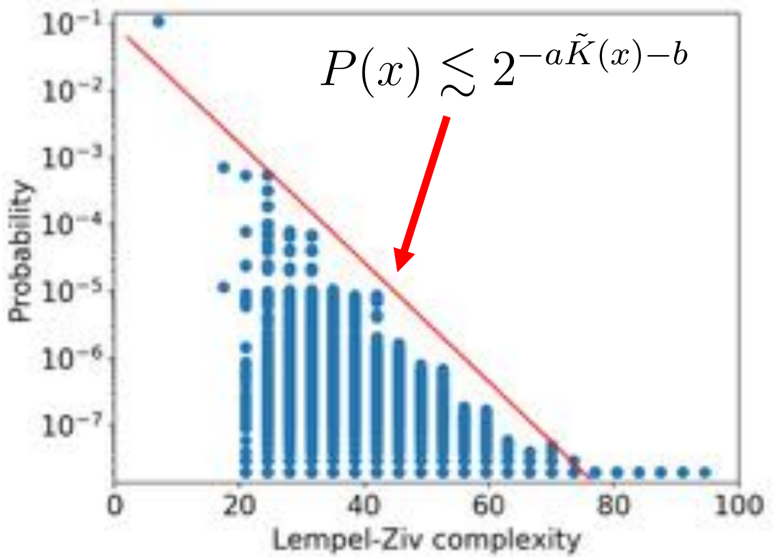
G. Valle-Perez, C. Camargo and A.A. Louis, arxiv:1805.08522 – ICLR 2019

Simplicity bias in the parameter-function map

Prior P(f): upon randomly sampling parameters, how likely to find Boolean function f?

Simple functions exponentially more likely to occur





10^8 samples of parameters (7,40,40,1) DNN (FCN) with ReLU.

Boolean functions for n=7. $2^7 = 128$ possible answers & $2^{128} \simeq 3.4 \times 10^{38}$ possible functions.

"Entropy" of simpler functions is larger than that of complex functions.

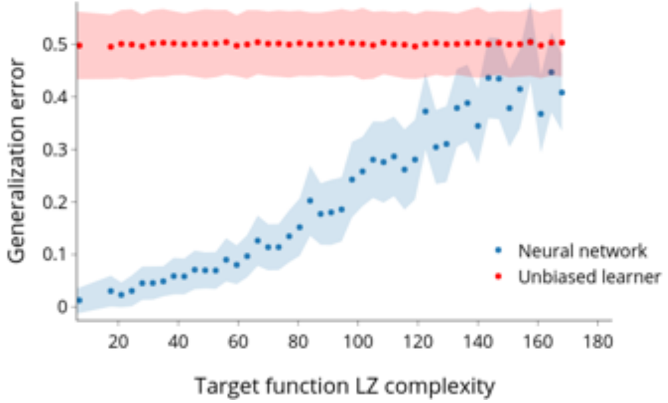
Boolean system is a key simplified model, akin to the Ising model is in physics.

Guillermo Valle Perez

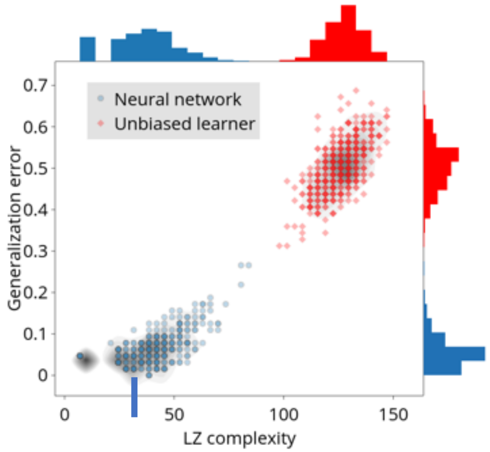
G. Valle Perez, C. Camargo and A.A. Louis, arxiv:1805.08522 – ICLR 2019

Simplicity bias aids generalisation (Occam)

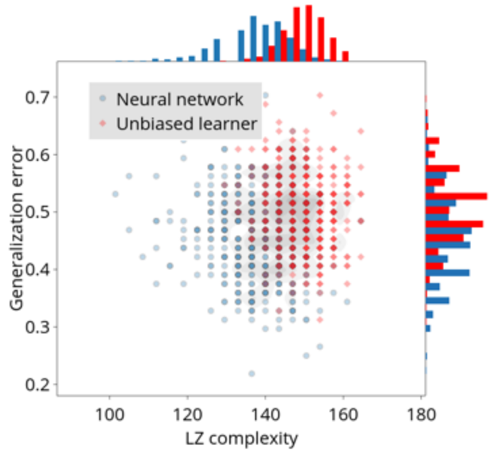
Supervised learning: 1) Pick a target function; 2) Train with SGD to zero training error on half the inputs; 3) Measure the error for the other half of inputs.



(a) Generalization error of learned functions



(a) Target function LZ complexity: 38.5



(b) Target function LZ complexity: 164.5

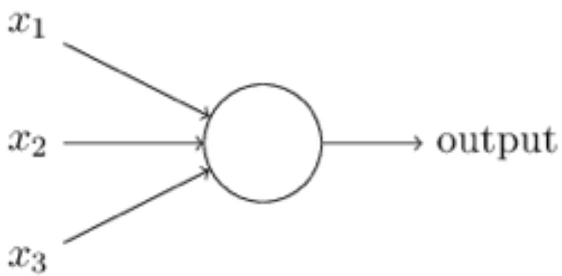
DNN works much better than an unbiased learner

DNN works well on simple target functions, but less well on complex functions

DNNs have an inbuilt "Occam's razor" – they work well on structured data.

Proving simplicity bias in the parameter-function map

P(f): If we randomly sample parameters θ, how likely are we to produce a particular function f?



Chris Mingard

Theorem 4.1. *For a perceptron f_θ with $b = 0$ and weights w sampled from a distribution which is symmetric under reflections along the coordinate axes, the probability measure $P(\theta : \mathcal{T}(f_\theta) = t)$ is given by*

$$P(\theta : \mathcal{T}(f_\theta) = t) = \begin{cases} 2^{-n} & \text{if } 0 \leq t < 2^n \\ 0 & \text{otherwise} \end{cases} .$$

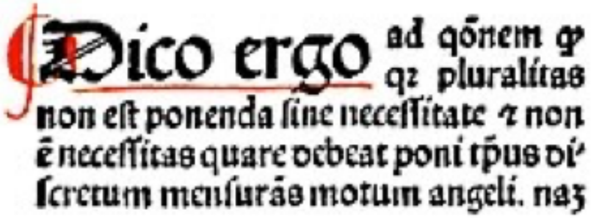
We can also prove theorems that bias towards simple function gets stronger with more layers!

Neural networks are a priori biased towards Boolean functions with low entropy, Chris Mingard, Joar Skalse, Guillermo Valle-Pérez, David Martínez-Rubio, Vladimir Mikulik, Ard A. Louis arxiv:1909.11522

Ockham's Razor and DNNs

Entities are not to be multiplied without necessity"
Ockham, according John Punch's 1639 commentary on Duns Scotus.

What Ockham actually said:


Dico ergo ad qõnem q qz pluralitas non est ponenda sine necessitate ⁊ non ē necessitas quare debeat poni tp̄us di' scretum mensuras motum angeli. naʒ

Pluralitas non est ponenda sine necessitate"
"Plurality is not to be posited without necessity"

William of Ockham
1287-1347

-possibly at Merton?

Modern approaches (the rabbit hole is deep …)

Bayes (e.g. David MacKay "*Information Theory, Inference, and Learning Algorithms*", ch 28)

AIT (e.g. Solomonoff, Hutter etc.. (AIT), but see Tom Sterkenberg for a critique)

Philosophers disagree …..Aristotle → Elliot Sober

Is this simplicity bias more universal ?

Why do DNNs exhibit an inbuilt Occam's razor?

(why the simplicity bias?)

MONKEY INTUITION:

What is the probability that a monkey types out M digits of **π** on an N key typewriter ?

P(X) = (1/N)^(M+1)

3.1415926535897932384626433832795028841971 6939
937510582097494459230781640628620899862803482
534211706798214808651328230664709384460955058
223172535940812848111745028410270193852110555
964462294895493038196442

But what if the monkey types into C ?

P(M) $\lesssim$ (1/N)^133

133 character (obfuscated) C code to calculate first 15,000 digits of **π**

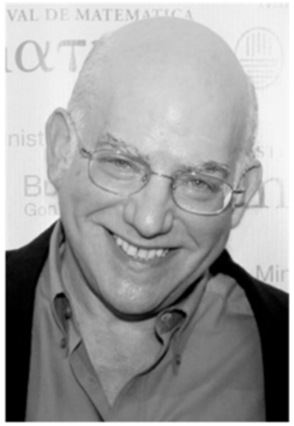
```
a[52514],b,c=52514,d,e,f=1e4,g,h;
main(){for(;b=c-=14;h=printf("%04d", e+d/f))
for(e=d%=f;g=--b*2;d/=g)d=d*b+f*(h?a[b]:f/5),a[b]=d%--g;}
```

$$\pi = \sum_{i=0}^{\infty} \frac{(i!)^2 2^{i+1}}{(2i+1)!}$$

C program due to Dik Winter and Achim Flammenkamp (See Unbounded Spigot Algorithms for the Digits of Pi, by Jeremy Gibbons (Oxford CS), Math. Monthly, April 2006, pages 318-328.)

Formalising the Monkey Intuition AIT: Kolmogorov Complexity

AIT = Algorithmic Information Theory



A.N. Kolgomorov 1903-1987 G.J. Chaitin 1947--

Kolmogorov/Chaitin complexity K(X) is the length in bits of the shortest program on a UTM that generates X

K is universal, (not UTM dependent) because you can always write a compiler => O(1) terms.

$K\_U(X) = K\_W(X) + O(1) \approx K(X)$ asymptotically

K is not computable due to Halting problem.

010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101 simple

01110101001100101111011110110101000010001011101010100110101111101110100101000111011101101101110 10 complex

Warning: you don't know for sure that it is complex, t could be encoding π= 3.14159265358979323846 …..

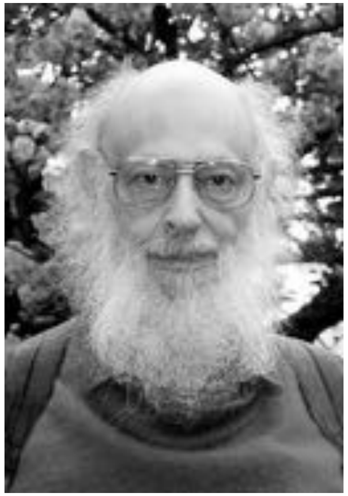
new intuitions

-- A random number is one for which $K(X) \gtrsim |X|$

-- The complexity of a set can be << than complexity of elements of the set

Formalising the Monkey Intuition using AIT: Algorithmic Probability

Algorithmic Probability P(x) = probability a random program on a (prefix) UTM generates x



R. Solomonoff
1926-2009

$$P_U(X) = \sum_{l:U(l)=X} 2^{-l} = 2^{-K(X)} + \dots$$

First term is the biggest one

Sum all binary codes that generate X on a prefix machine

Intuitively: simpler (small K(X)) outputs are much more likely to appear

It seems to me that the most important discovery since Gödel was the discovery by Chaitin, Solomonoff and Kolmogorov of the concept called Algorithmic Probability,. Everybody should learn all about that and spend the rest of their lives working on it.

Marvin Minsky (2014)

https://www.youtube.com/watch?v=DfY-DRsE86s&feature=youtu.be&t=1h30m02s

Solomonoff, R., "A Preliminary Report on a General Theory of Inductive Inference", Report V-131, Zator Co., Cambridge, Ma. Feb 4, 1960, revision, Nov., 1960.

Formalising the Monkey Intuition using AIT: Levin's Coding Theorem

We should teach this much more widely!

L. Levin, 1948 --

$$2^{-K(x)} \leq P(x) \leq 2^{-K(x)+O(1)}$$

Intuitively: simpler (small K(X)) outputs are much more likely to appear

Serious problems for applying coding theorem

1) Many systems of interest are not Universal Turing Machines
2) Kolmogorov complexity K(x) is formally incomputable
3) Only holds in in the asymptotic limit of large x…

L. A. Levin. Laws of information conservation (non-growth) and aspects of the foundation of probability theory. Problems of Information Transmission, 10:206–210, 1974.

Formalising the Monkey Intuition using AIT: a new coding theorem for non UTM maps

Proof sketch:

1) For simple maps **f**, with input size *n* we can calculate the whole set of input → output pairs at *O(1)* cost (complexity of a set << elements of set)
2) Encode this with a Shannon-Fano-Elias (SFE) code for which P(x) ~ ½^length
3) This procedure gives a bound on the Kolmogorov complexity, **given f and n:** K(x|f,n)

$$K(x|f,n) \leq l(E(x)) + O(1)$$
$$= \log_2 \left(\frac{1}{P(x)} \right) + O(1)$$
$$\Rightarrow \boxed{P(x) \leq 2^{-K(x|f,n)+O(1)}}$$

← NOTE: upper bound only!

K. Dingle, C. Camargo and A.AL, Nature Communications 9, 761 (2018)

Simplicity bias for computable input-output maps

Kamal Dingle

(2 Dphils of work)

$$P(x) \lesssim 2^{-a\tilde{K}(x)-b}$$

NOTE: upper bound only!

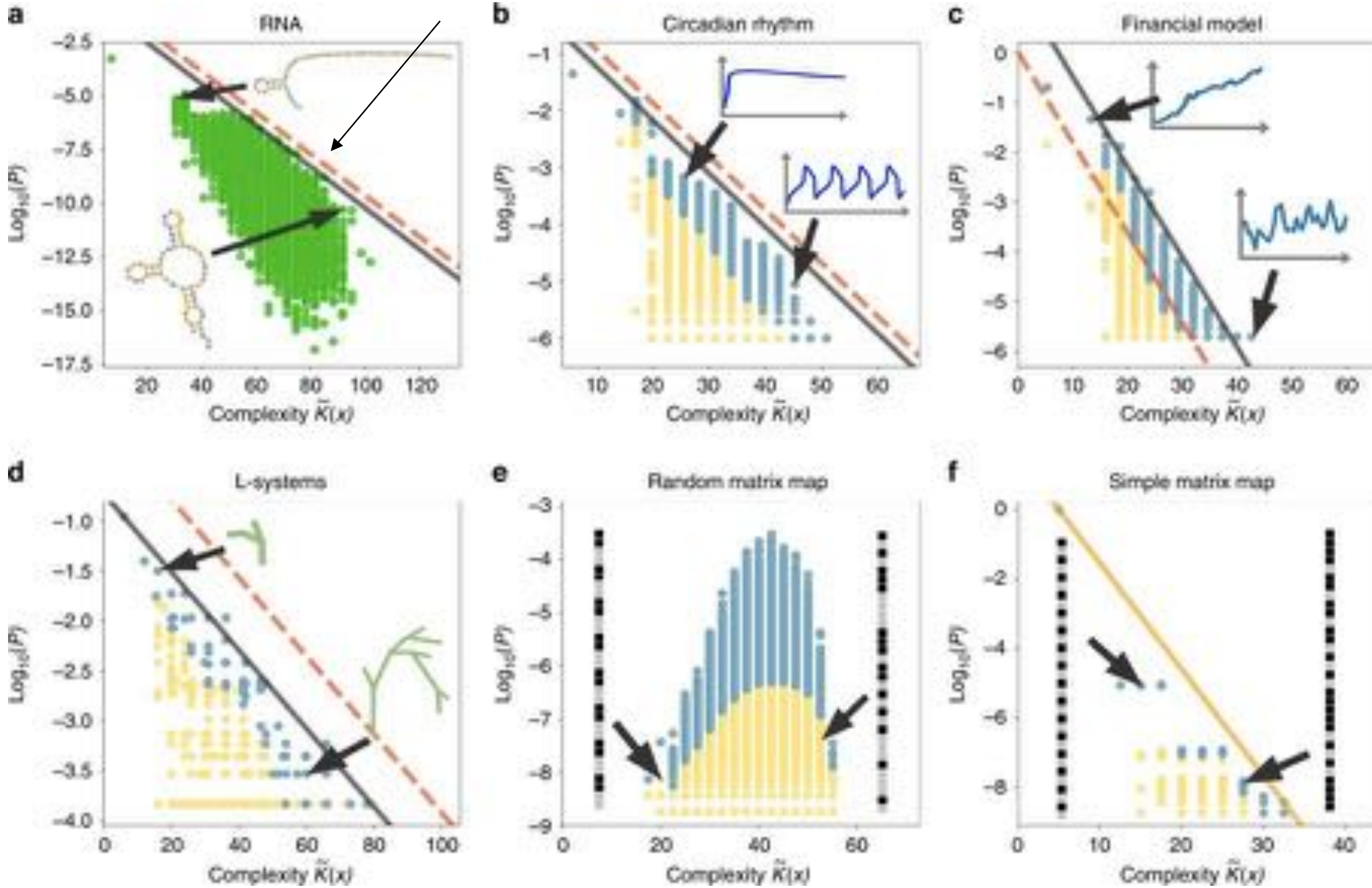
Chico Camargo

1) Computable input-output map f: I → O
2) Map f must be simple – e.g. K(f) grows slowly with system size – then $K(x|f,n) \approx K(x) + O(1)$
3) K(x) is approximated, for example by Lempel Ziv compression or some other suitable measure.
4) Bound is tight for most inputs, but not most outputs.
5) Maps must be a) simple, b) redundant, c) non-linear, d) well-behaved (e.g. not a pseudorandom number generator) – many maps satisfy these conditions.
6) There is also a statistical lower bound.

K. Dingle, C. Camargo and A.AL, Nature Comm **9**, 761 (2018); K. Dingle, G. Valle-Perez, AAL, Sci. Rep. 10, 4415 (2020)

Simpliciy bias works in many different maps

$P(x) \lesssim 2^{-a\tilde{K}(x)-b}$ = black line (red dashed with b=0)



K. Dingle, C. Camargo and A.AL, Nature Communications 9, 761 (2018)

Evolution has an inbuilt Occam's razor

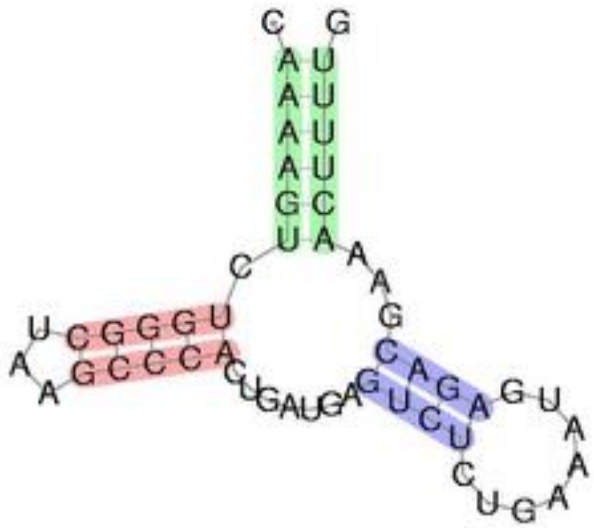
Mapping from RNA sequences to RNA structures

GAAAGUCUGGGCUAAGCCACUGAUGGUGUCUGAAAUGAGAGGAAAACUUUUG

Hammerhead ribozyme

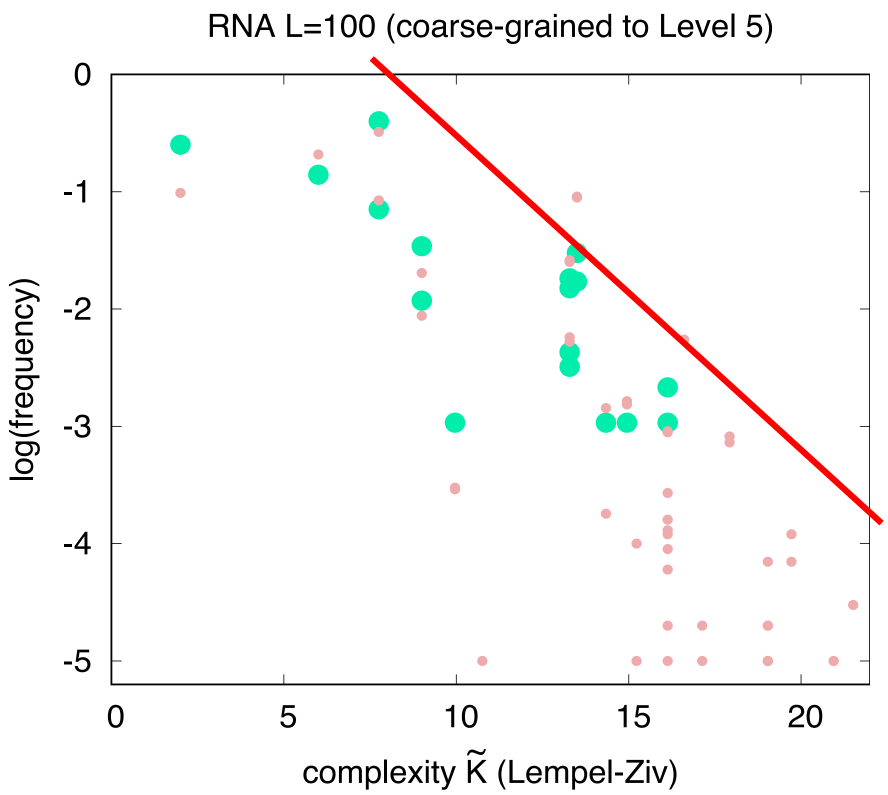
Tertiary structure (3D)

Folding
GP map



Secondary structure
(who bonds to whom)

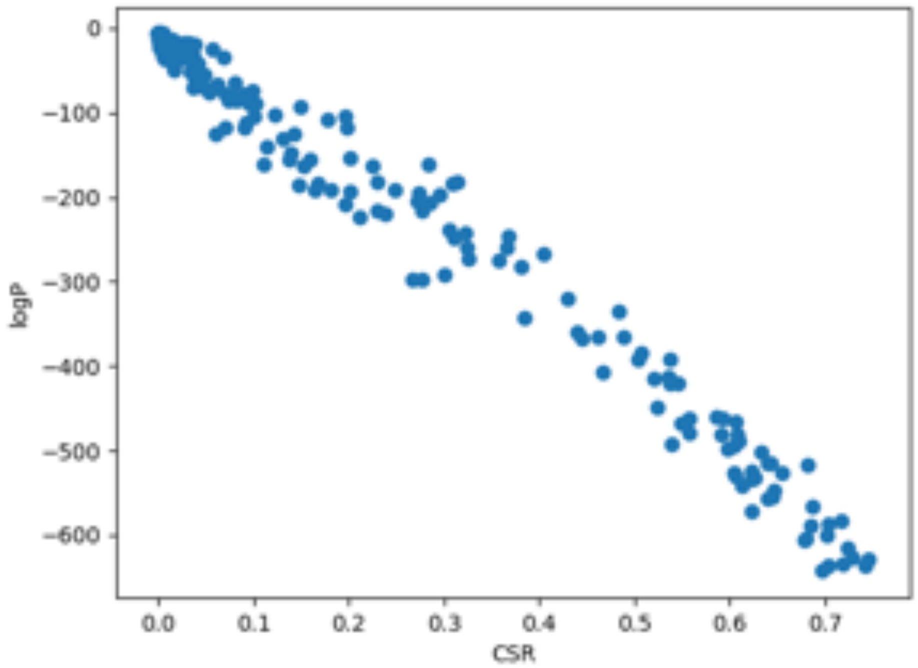
Evolution has an inbuilt Occam's razor



932 non-coding functional RNA of length 100 found in nature (from fRNAdb bioinformatic database)

K Dingle, F. Ghaddar, P. Sulc and AA Louis, bioarxiv /2020.12.03.410605

Simpliciy bias is found in DNNs



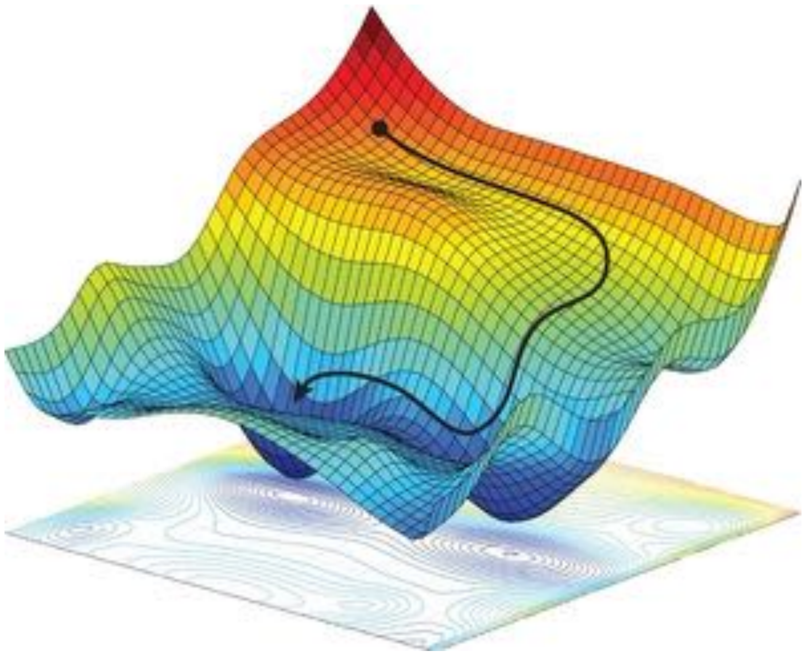
(a)

Simplicity bias for a CNN on CIFAR10

(a) Probability (using GP approximation) versus critical sample ratio (CSR) of labelings of 1000 random CIFAR10 inputs, produced by 250 random samples of parameters. The network is a 4 layer CNN.

G. Valle Perez, C. Camargo and A.A. Louis, arxiv:1805.08522 – ICLR 2019

Hold on: why should parameter function map predict DNN outcomes?



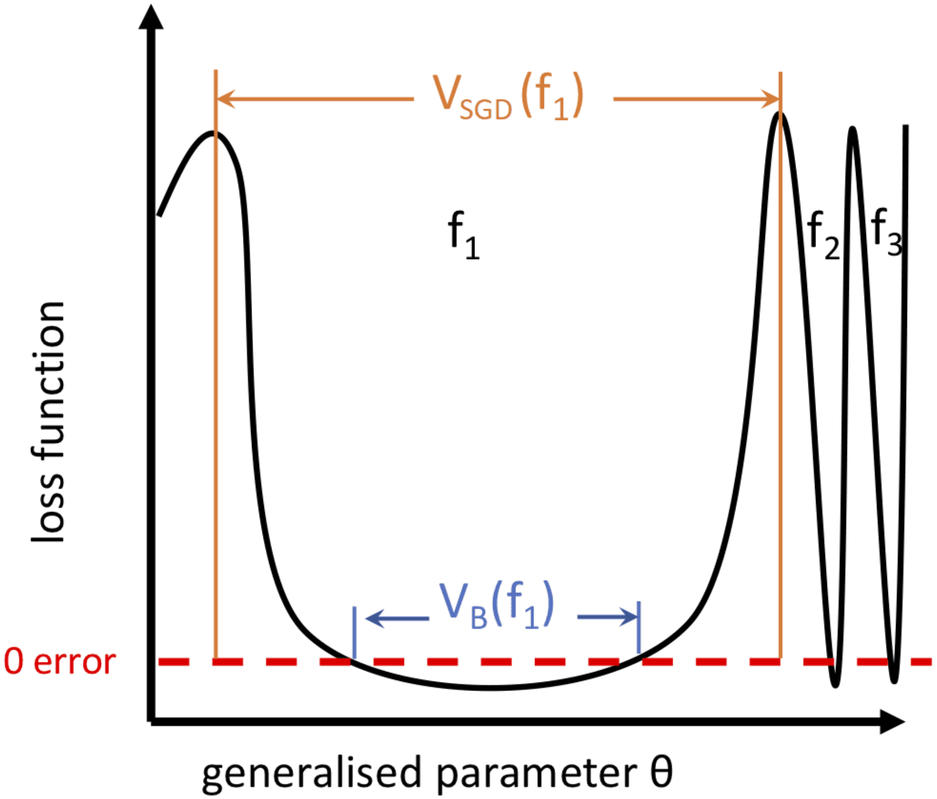
DNNs are trained using Stochastic gradient descent (SGD) on a loss function.

~~Dominant hypothesis in the field is that SGD has special properties that enhance generalization~~

Is SGD a Bayesian sampler? Well, almost, Chris Mingard, Guillermo Valle-Pérez, Joar Skalse, Ard A. Louis, arxiv.org: 2006.15191

Problem: why should parameter function map predict outcomes?

Intuition: for very strong bias: Basin of attraction ~ Basin size (P(f))



Similar effect in evolutionary theory under strong bias:
The arrival of the frequent: how bias in genotype-phenotype maps can steer populations to local optima
Steffen Schaper and Ard A. Louis, PLoS ONE 9 (2): e86635 (2014)

Is SGD a Bayesian sampler? Well, almost, Chris Mingard, Guillermo Valle-Pérez, Joar Skalse, Ard A. Louis, Journal of Machine Learning Research 22 (79), 1-64 (2021)

A function based picture

Definition 2.2 (Representation of Functions)**.** Consider a DNN $\mathcal{N}$, a training set $S = \{(x_i, y_i)\}_{i=1}^{m}$ and test set $E = \{(x'_i, y'_i)\}_{i=1}^{|E|}$. We *represent* the function $f(\mathbf{w})$ with parameters $\mathbf{w}$ associated with $\mathcal{N}$ as a string of length $(|S| + |E|)$, where the values are the labels $\hat{y}_i$ and $\hat{y}'$ that $\mathcal{N}$ produces on the concatenation of training inputs and testing inputs.

Example on 5 MNIST inputs:

f(**w**) = (5,0,4,1,9) (0 errors)
f(w) = (5,0,4,7,9) (1 error)

Bayesian function picture for supervised learning on S

Posterior for functions conditioned on training set S follows from Bayes rule

$$P(f|S) = \frac{P(S|f)P(f)}{P(S)},$$

Prior over functions $P(f)$

If we wish to infer (i.e. no noise) at some points, then we need a 0-1 likelihood on training data $S = \{(x_i, y_i)\}_{i=1}^{m}$

$$P(S|f) = \begin{cases} 1 \text{ if } \forall i, \quad f(x_i) = y_i \\ 0 \text{ otherwise} . \end{cases}$$

P(S) = marginal likelihood or evidence

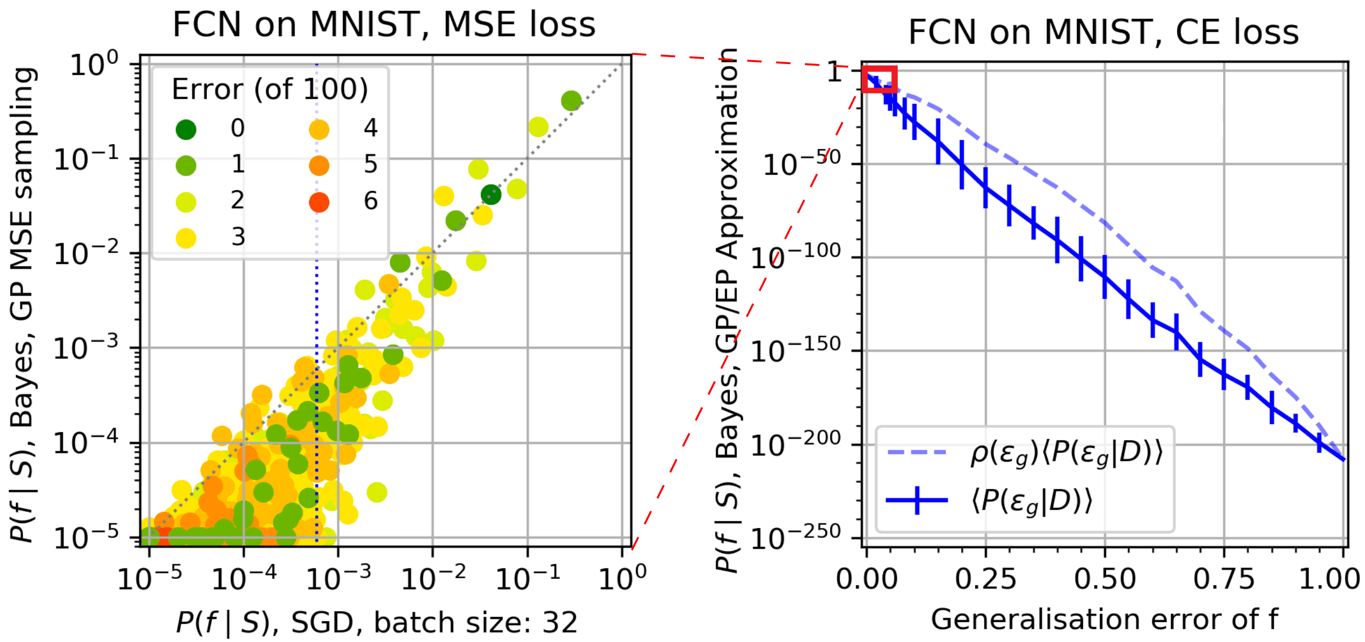
Functions that fit S

$$P(S) = \sum_f P(S|f)P(f) = \sum_{f \in C(S)} P(f)$$

P(f|S) = P(f)/P(S) or 0, so bias in prior translates over to bias in posterior

Is SGD a Bayesian sampler? Well, almost, Chris Mingard, Guillermo Valle-Pérez, Joar Skalse, Ard A. Louis, Journal of Machine Learning Research 22 (79), 1-64 (2021)

SGD acts like a Bayesian optimiser ….



(a) $P_{\mathrm{B}}(f|S)$ v.s. $P_{\mathrm{SGD}}(f|S)$ (b) $P_{\mathrm{B}}(f|S)$ v.s. ϵ_G

FCN on binarized MNIST – training set=10,000, test set=100 images 2^{100} = 10^{30} possible functions fit the test set.

We use Gaussian Processes (GP)s to calculate $P_B(f|S)$ –

Is SGD a Bayesian sampler? Well, almost, Chris Mingard, Guillermo Valle-Pérez, Joar Skalse, Ard A. Louis, Journal of Machine Learning Research 22 (79), 1-64 (2021)

Two kinds of questions about generalisation:

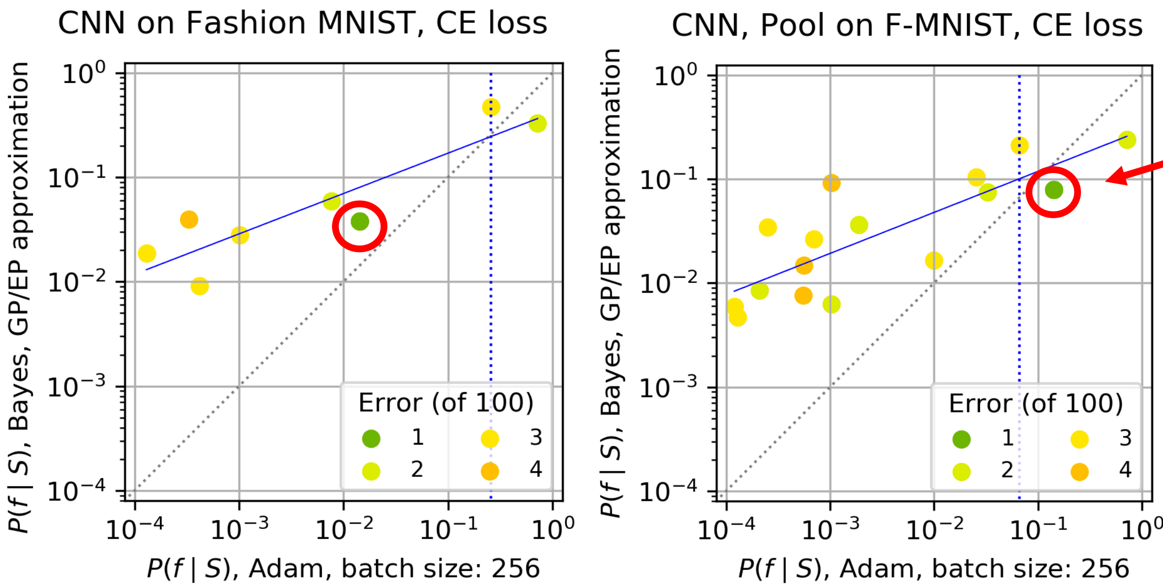
1) Why do DNNs generalise at all in the overparameterised regime?

Because the parameter-function map is highly biased towards simple solutions.

2) Given DNNs that generalise, can we further fine-tune the hyperparameters to improve generalisation? (engineers).

Is SGD a Bayesian sampler? Well, almost, C. Mingard, G.Valle-Pérez, J. Skalse, AAL, arxiv.org: 2006.15191

2^{nd} order effects beyond simplicity bias: changing the network



(b) $P_{\mathrm{B}}(f|S)$ v.s. $P_{\mathrm{Adam}}(f|S)$ (c) $P_{\mathrm{B}}(f|S)$ v.s. $P_{\mathrm{Adam}}(f|S)$

With max-pooling probability of low error function increases

CNN on binarized Fashion-MNIST – training set=10,000, test set=100 images
$2^{100} = 10^{30}$ possible functions fit the test set.

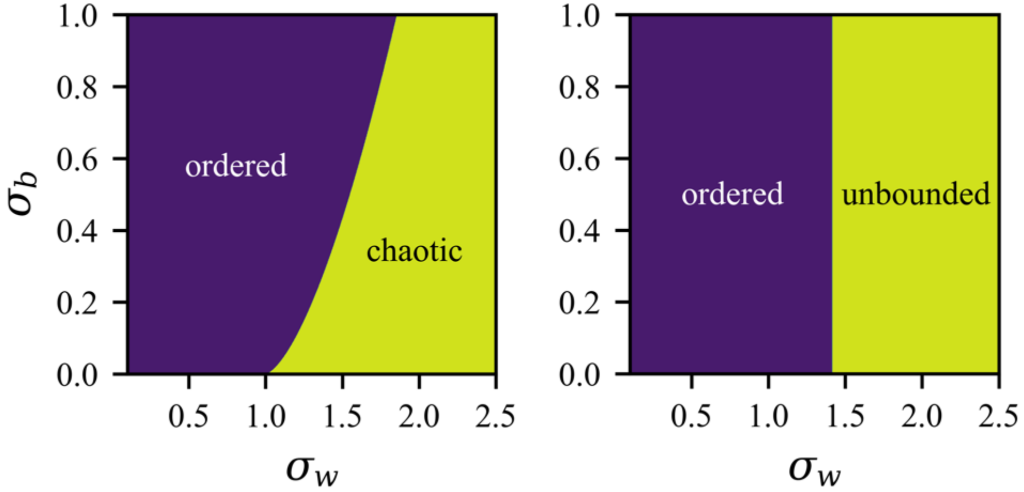
Similar results for CNN, LSTM, other data sets, etc….

Is SGD a Bayesian sampler? Well, almost, Chris Mingard, Guillermo Valle-Pérez, Joar Skalse, Ard A. Louis, Journal of Machine Learning Research 22 (79), 1-64 (2021)

Can we do any control experiments?

1)Can we break the simplicity bias?

DNNs can exhibit an order-to-chaos transition



(a) *Tanh* **(b)** *ReLU*

Figure 3: *Mean field phase diagrams for tanh and ReLU activation functions showing various phase regimes as a function of σ_w and σ_b.*

Chaotic regime for some activation functions (not ReLU!) – for wider initial parameters

Deep Information Propogation, S. S. Schoenholz et al. arXiv:1611.01232

Chaotic regime reduces bias in prior P(f)

FCN on Boolean system

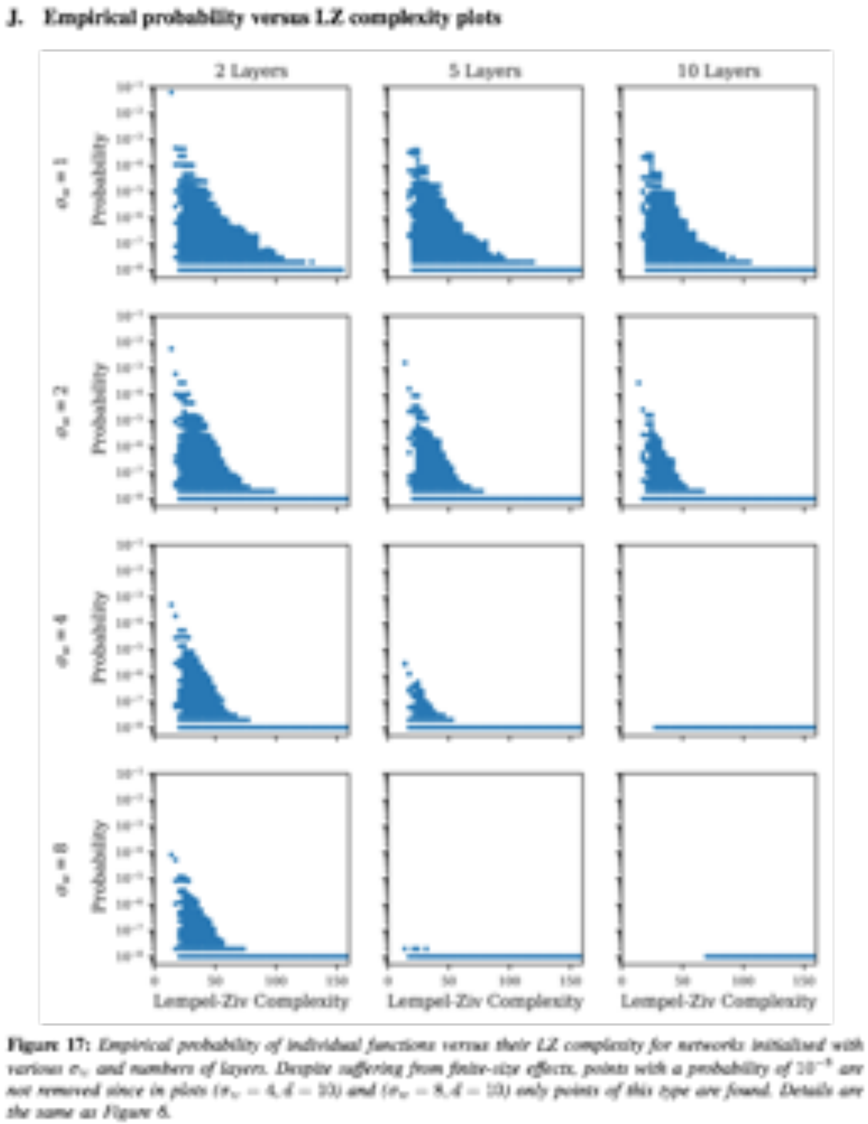

J. Empirical probability versus LZ complexity plots

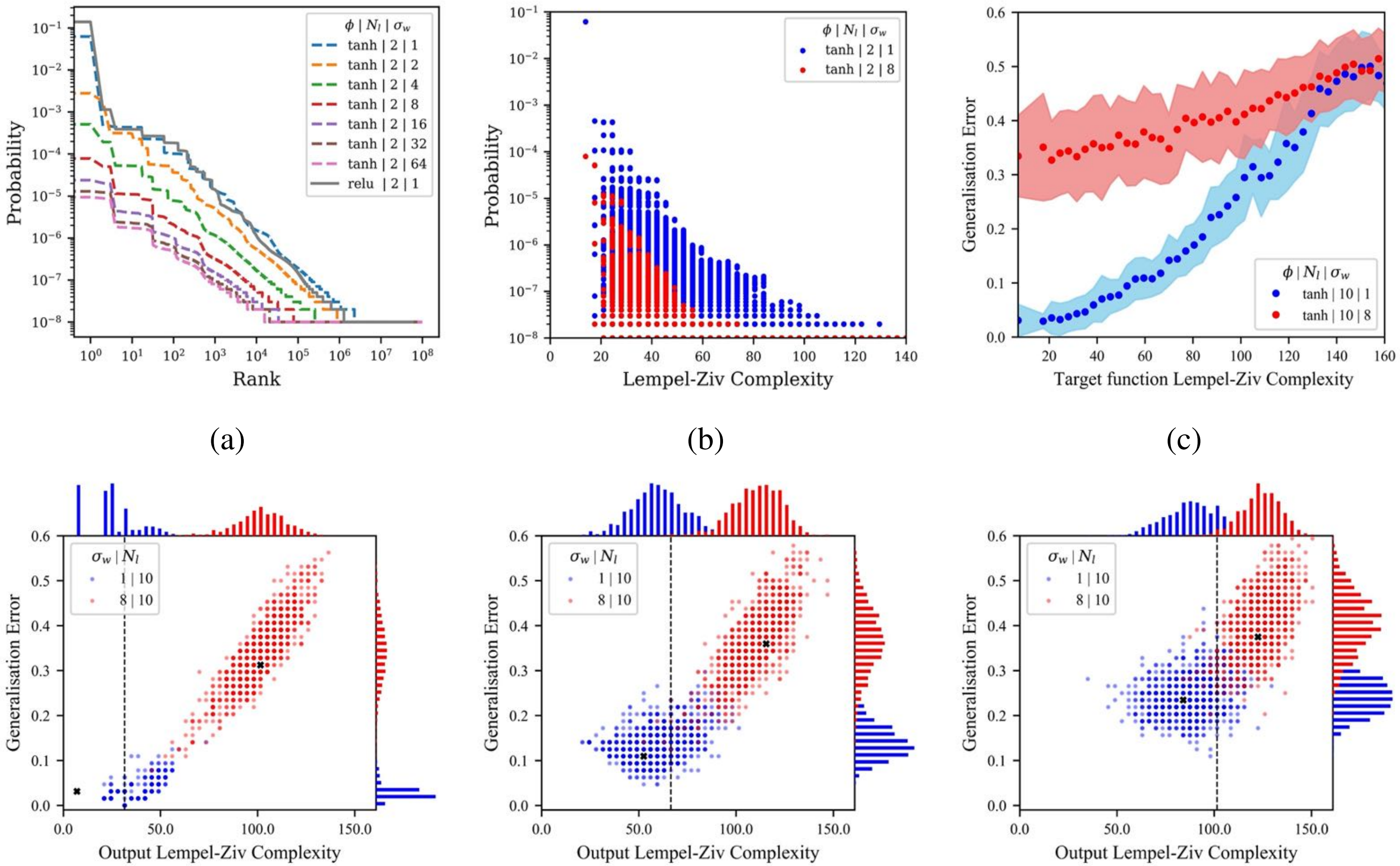
Figure 17: Empirical probability of individual functions versus their LZ complexity for networks initialised with various σ_w and numbers of layers. Despite suffering from finite-size effects, points with a probability of 10^{-8} are not removed since in plots ($\sigma_w = 4, d = 10$) and ($\sigma_w = 8, d = 10$) only points of this type are found. Details are the same as Figure 6.

More biased

Less biased
No Occam

Greg Yang and Hadi Salman. A fine-grained spectral perspective on neural networks. arXiv preprint arXiv:1907.10599, 2019.

Chaotic regime changes the bias in prior P(f)



(a) (b) (c)

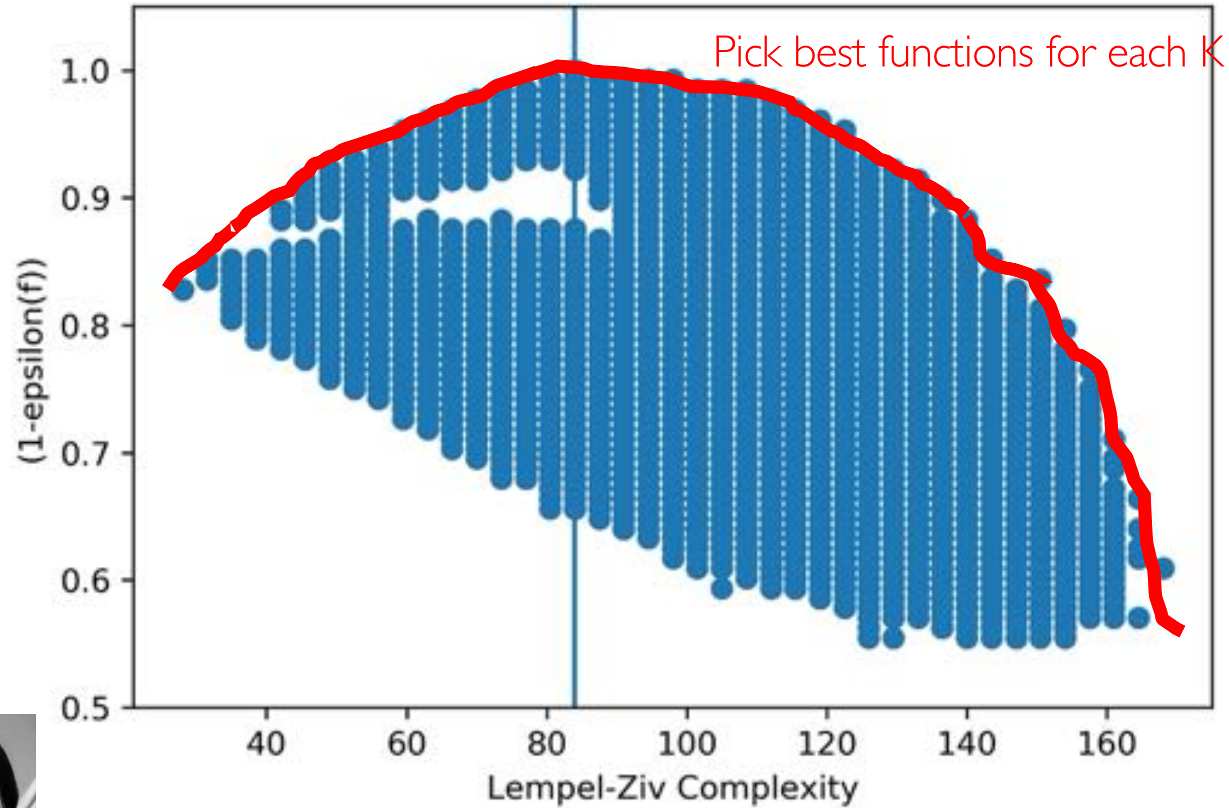
(d) Target function LZ = 31.5 (e) Target function LZ = 66.5 (f) Target function LZ = 101.5

Greg Yang and Hadi Salman. A fine-grained spectral perspective on neural networks. arXiv preprint arXiv:1907.10599, 2019.

Bayesian picture and the data

$$\langle P(f|\mathcal{S})\rangle_S = P(f)\left\langle \frac{P(\mathcal{S}_i|f)}{P(\mathcal{S}_i)}\right\rangle_{\mathcal{S}_i} \approx \frac{P(f)(1-\epsilon(f))^m}{\langle P(\mathcal{S})\rangle}$$

$(1 - \varepsilon(f))$



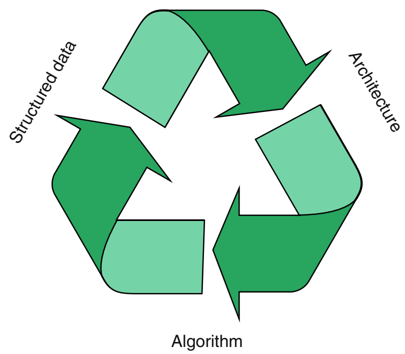
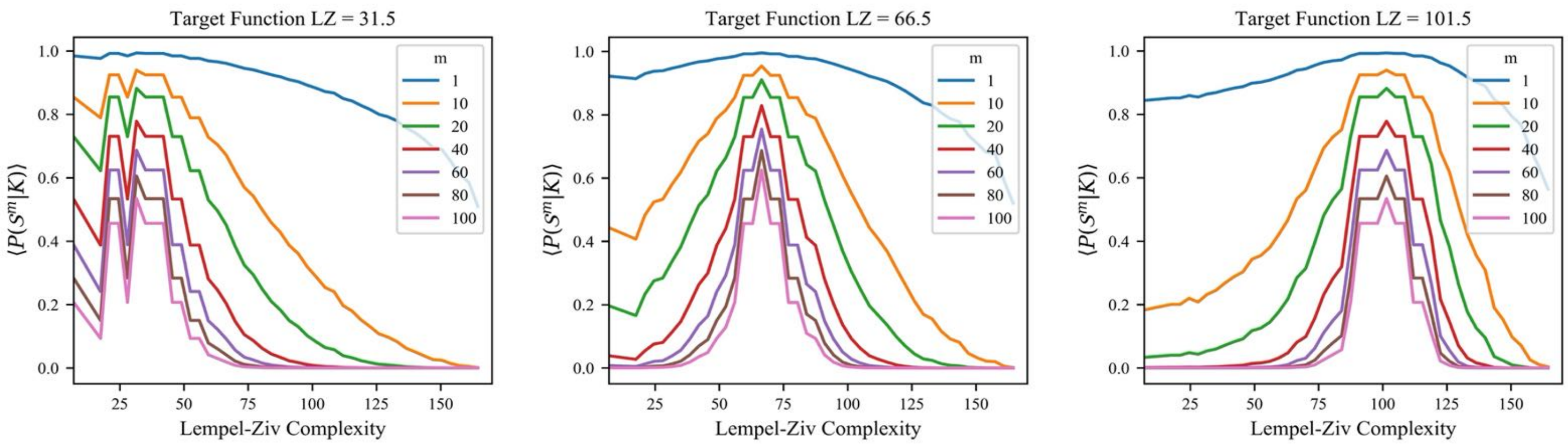


Fig. 1 | Interplay of key ingredients. Building

Lenka Zdeborová. Nature Physics **16**, 602 (2020)

Bayesian picture and data

$$\langle P(f|\mathcal{S})\rangle_S = P(f)\left\langle \frac{P(\mathcal{S}_i|f)}{P(\mathcal{S}_i)}\right\rangle_{\mathcal{S}_i} \approx \frac{P(f)\,(1-\epsilon(f))^m}{\langle P(\mathcal{S})\rangle}$$



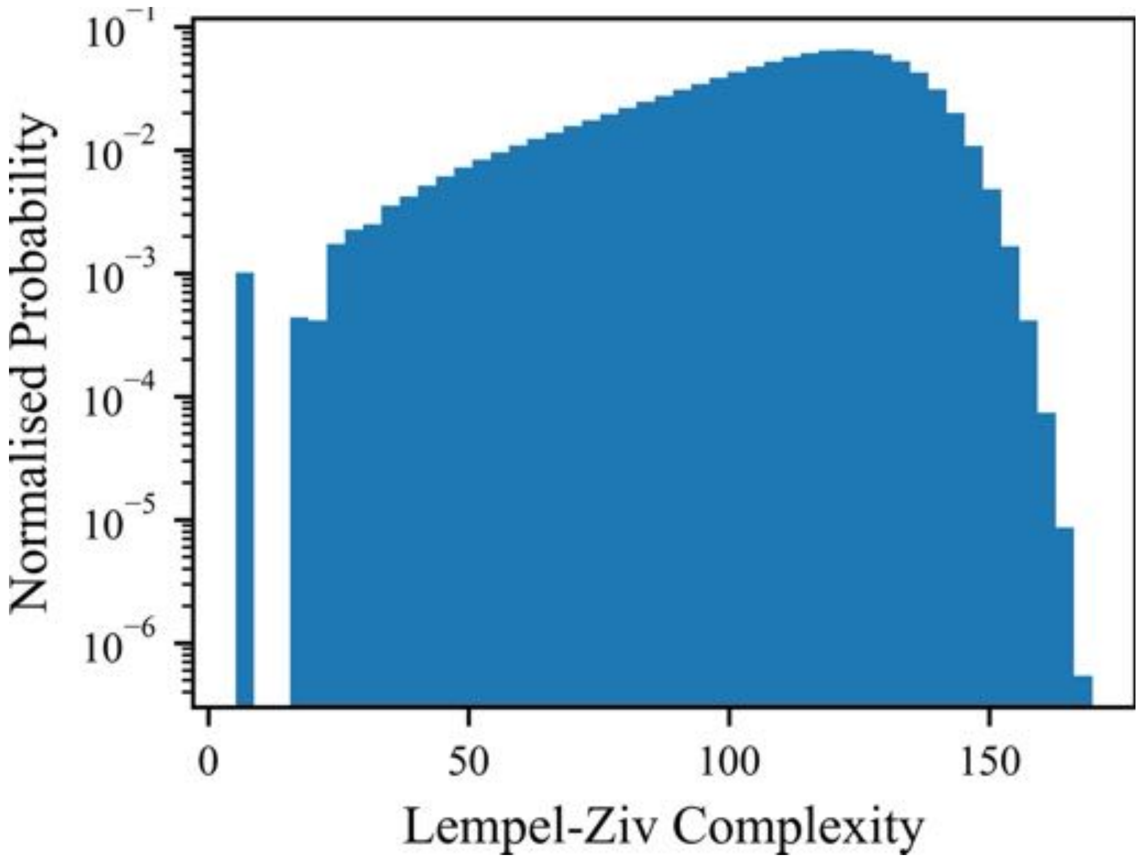
Bayesian picture and prior P(K)

$$\langle P(f|\mathcal{S})\rangle_S = P(f)\left\langle \frac{P(\mathcal{S}_i|f)}{P(\mathcal{S}_i)}\right\rangle_{\mathcal{S}_i} \approx \frac{P(f)(1-\epsilon(f))^m}{\langle P(\mathcal{S})\rangle}$$

$P(K)$

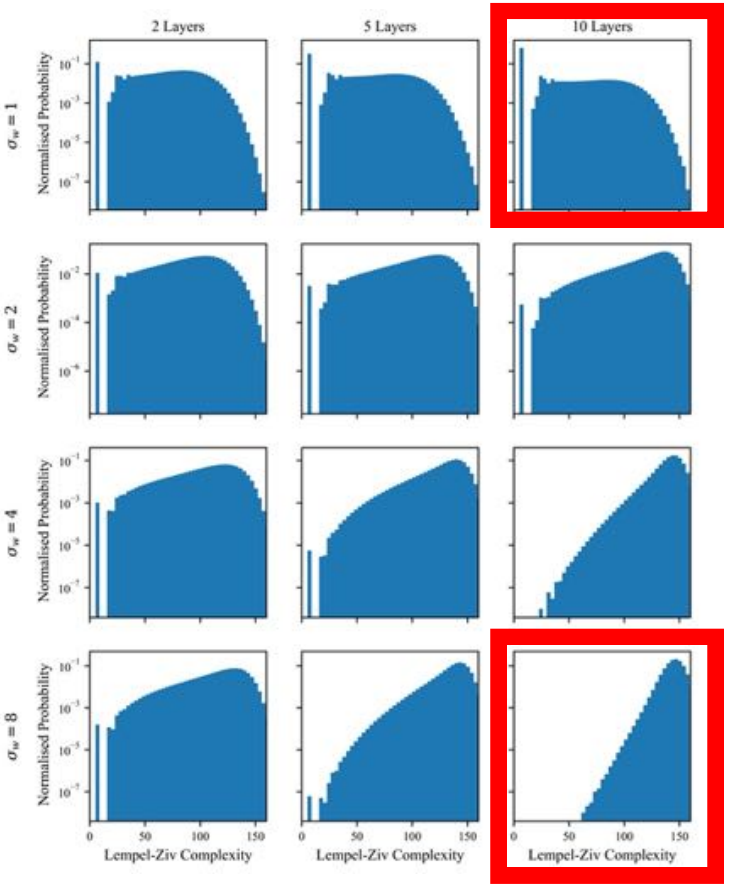
Instead of

$P(f)$



Bayesian picture and prior P(K)

$$\langle P(f|\mathcal{S})\rangle_S = P(f)\left\langle \frac{P(\mathcal{S}_i|f)}{P(\mathcal{S}_i)}\right\rangle_{\mathcal{S}_i} \approx \frac{P(f)\,(1-\epsilon(f))^m}{\langle P(\mathcal{S})\rangle}$$



Ordered Regime:

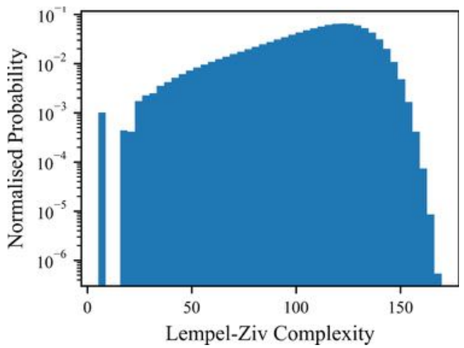
10 Layers, σ_w = 1.0

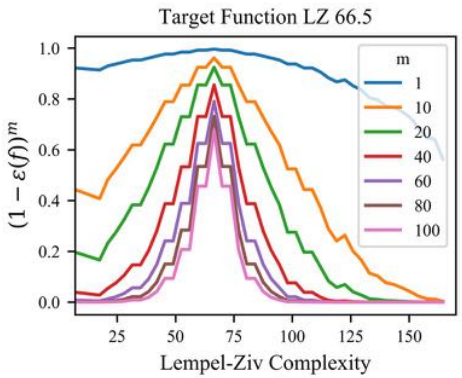
Chaotic Regime:

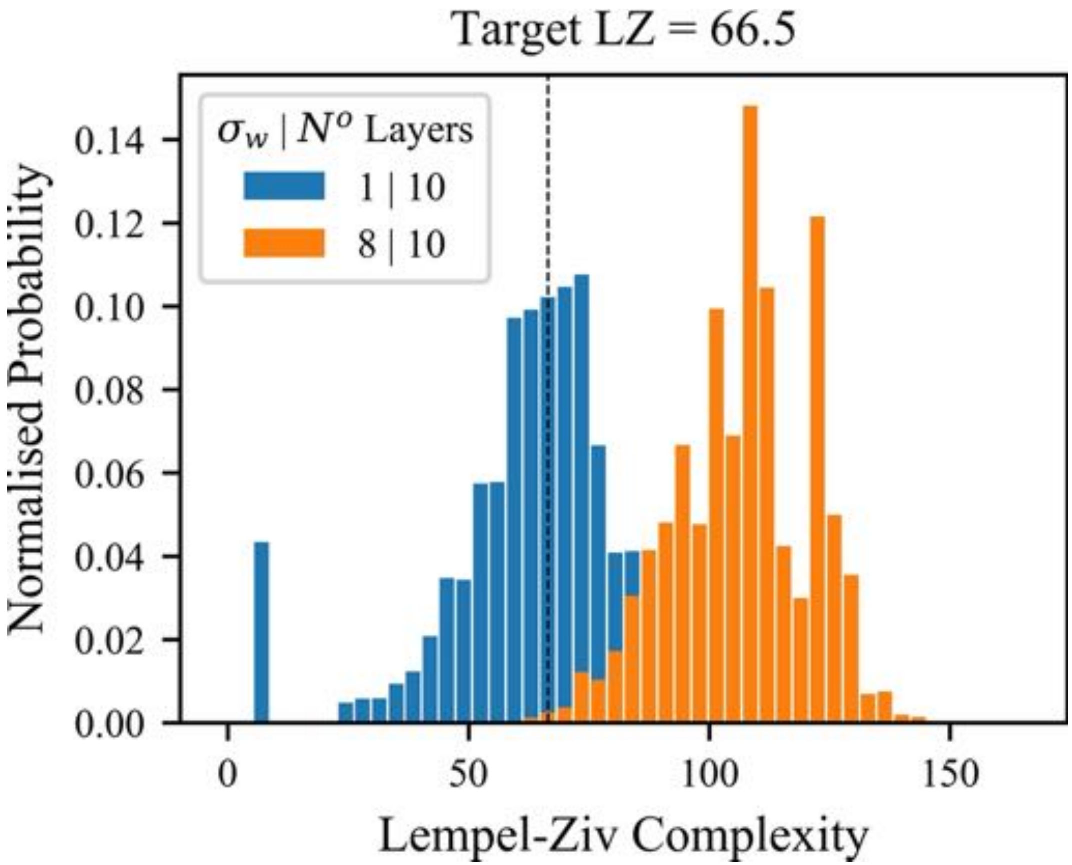
10 Layers, σ_w = 8.0

Bayesian picture: combining data and prior

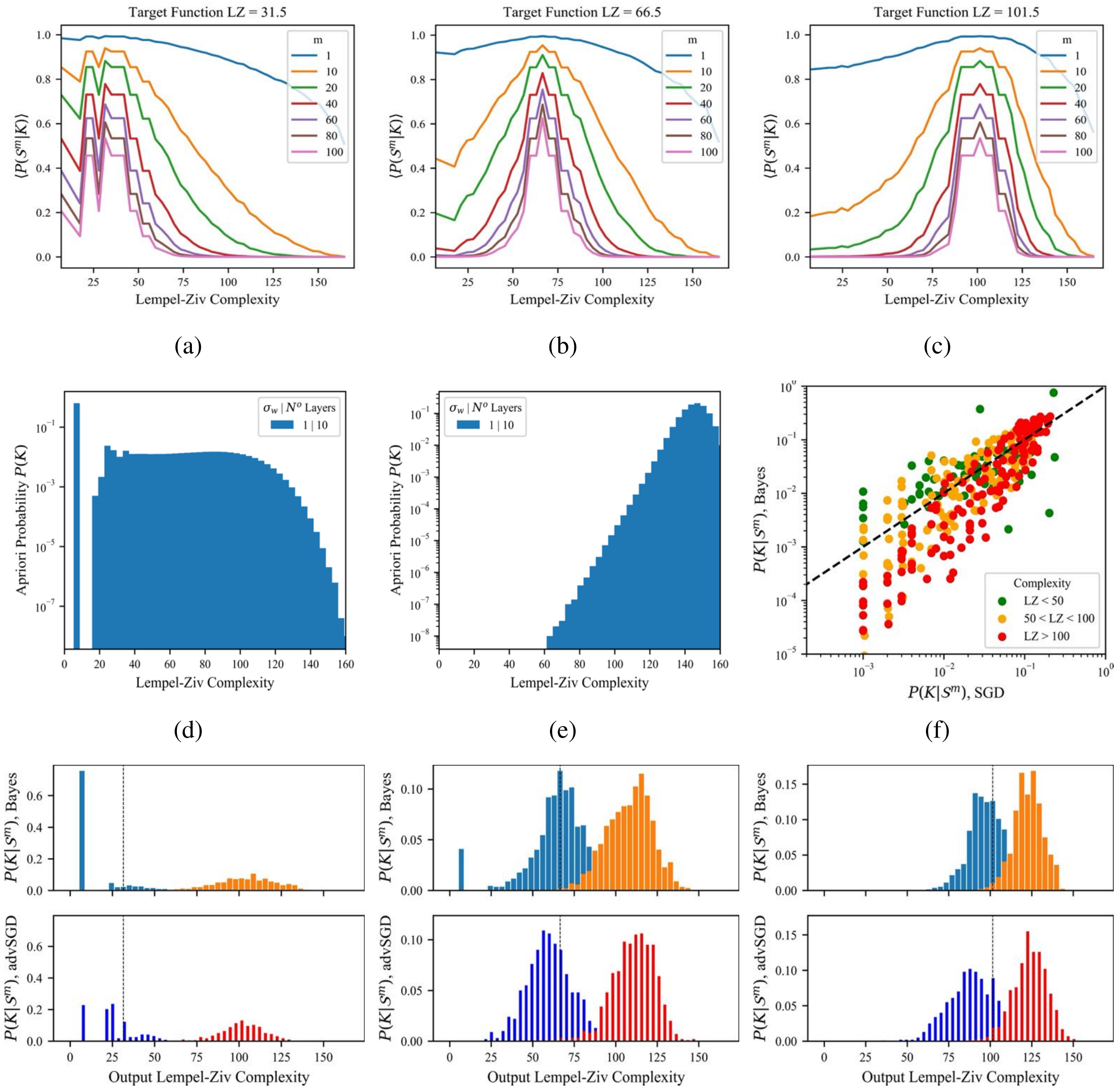
$$\langle P(f|S)\rangle_S = P(f)\left\langle \frac{P(S_i|f)}{P(S_i)} \right\rangle_{S_i} \approx \frac{P(f)\,(1-\epsilon(f))^m}{\langle P(S)\rangle} \propto P(K)(1-\epsilon(f))^m$$







Bayesian picture combining data and prior

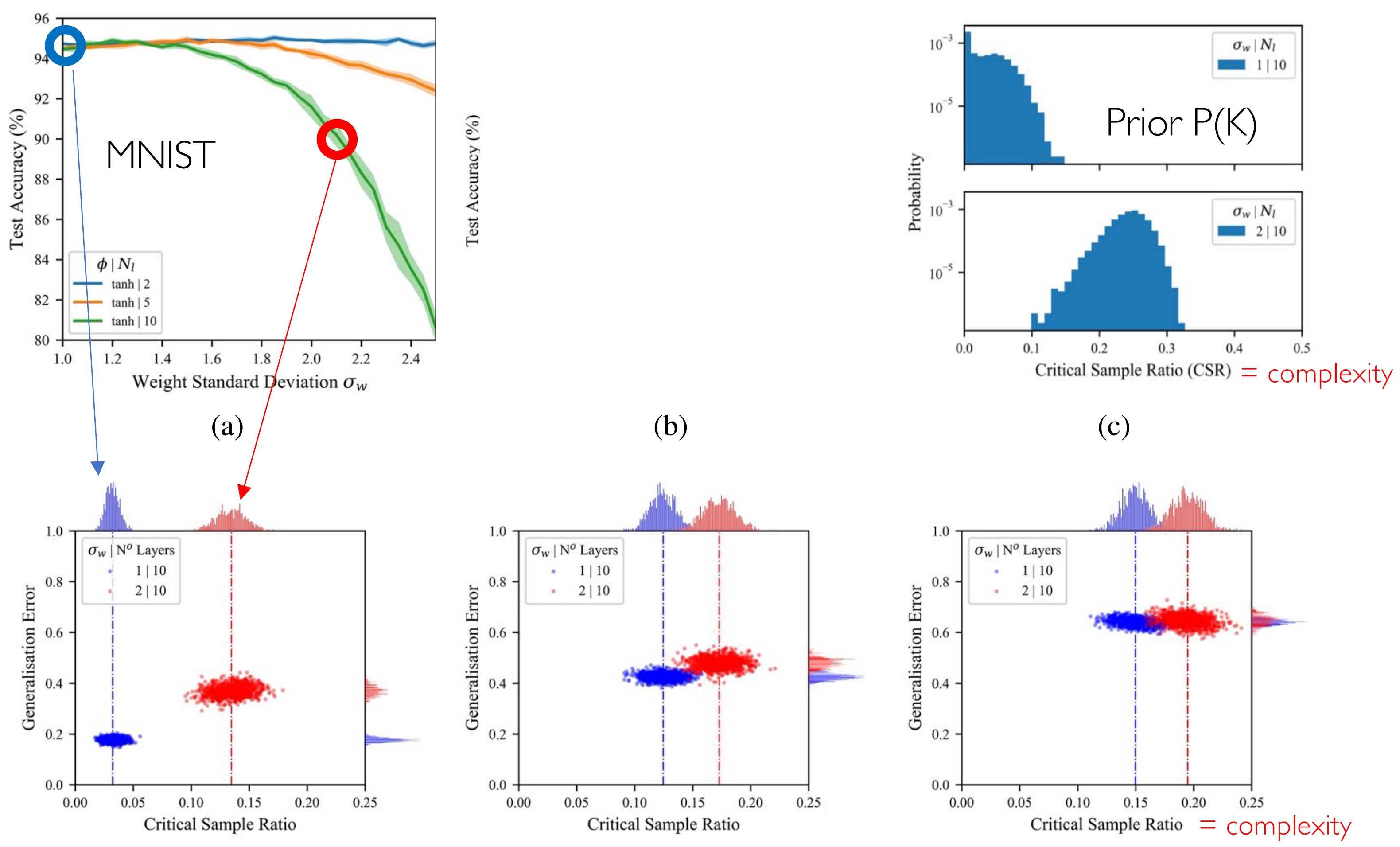


(a) (b) (c)

(d) (e) (f)

(g) Target function LZ = 31.5 (h) Target function LZ = 66.5 (i) Target function LZ = 101.5

Bayesian picture: prior and data for MNIST/CIFAR-10



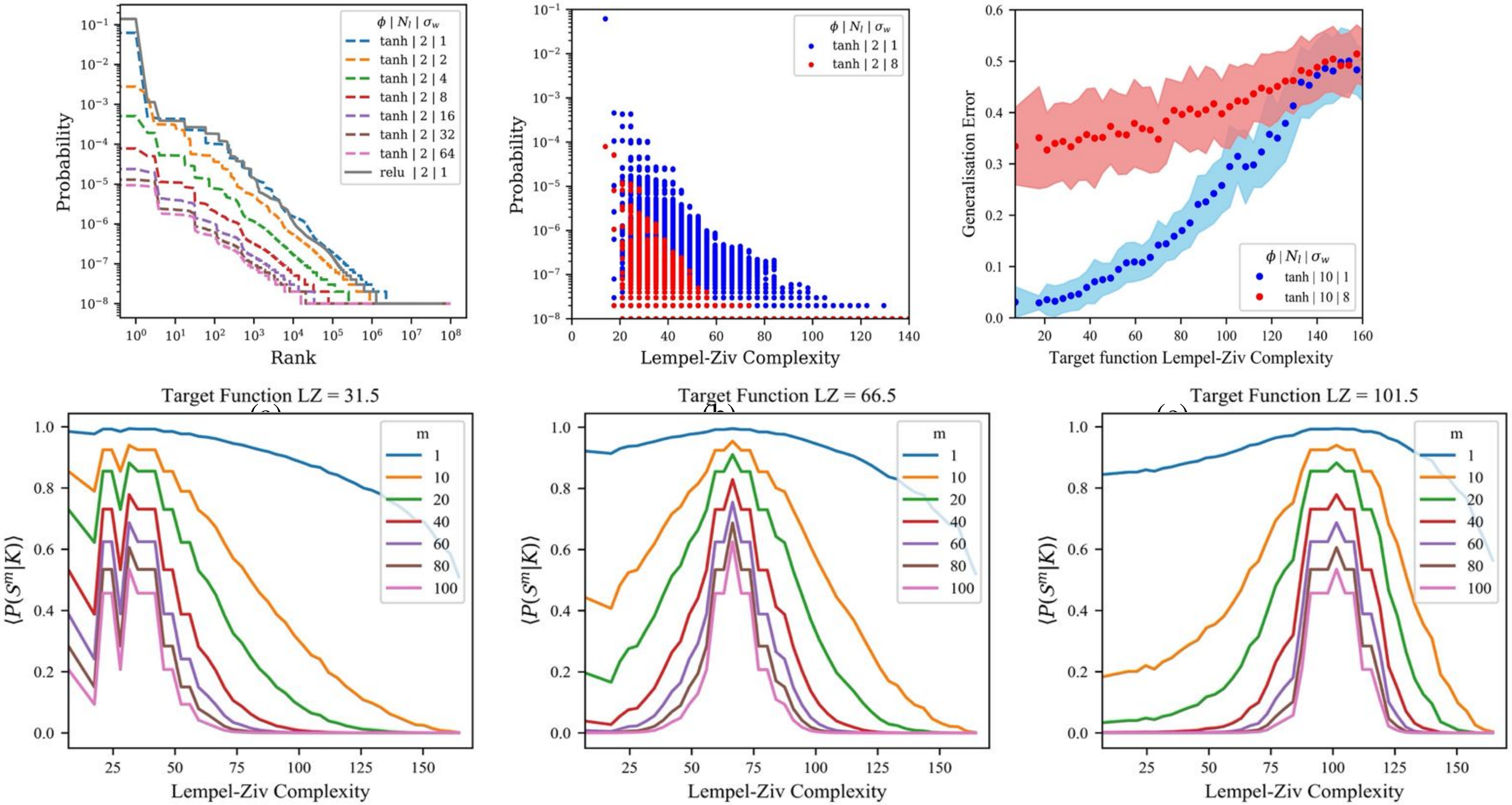
(a) (b) (c)

(d) Uncorrupted (e) 25% corruption (f) 50% corruption

Summary: Bayesian picture: prior and data

Average posterior over training sets

$$\langle P(f|\mathcal{S})\rangle_S = P(f)\left\langle \frac{P(\mathcal{S}_i|f)}{P(\mathcal{S}_i)} \right\rangle_{\mathcal{S}_i} \approx \frac{P(f)\,(1-\epsilon(f))^m}{\langle P(\mathcal{S})\rangle}$$



Function based picture and generalisation bounds

Big literature on bounds –
Concepts such as PAC learning, VC dimension, Rademacher complexity etc….

$$\forall \mathcal{D},\ \mathbf{P}_{S\sim\mathcal{D}^m}\left[\sup_{h\in\mathcal{H}}|\epsilon(h)-\hat{\epsilon}(h)| \leq C\sqrt{\frac{\mathrm{VC}(\mathcal{H})+\ln\frac{1}{\delta}}{m}}\right] \geq 1-\delta$$

Big review paper on generalization bounds, includes 7 desiderata bounds should satisfy and a classification

Generalization bounds for deep learning Guillermo Valle-Pérez and AAL, arxiv:arXiv:2012.04115

Function based picture and generalisation bounds

	Algorithm-independent (section 4.1)		Algorithm-dependent (section 4.2)	
	Based on uniform convergence	Based on non-uniform convergence		Other
Data-independent	VC dimension bound[*] (section 4.1.1)	SRM-based bounds[†] (section 4.2.1.1)	-	uniform stability bounds[‡] and compression bounds[§] (section 4.3.1)
Data-dependent	Rademacher complexity bound[¶] (section 4.1.2)	data-dependent SRM-based bounds[**] (section 4.2.1.1)	margin bounds[††] (4.2.1.2), sensitivity-based bounds[‡‡] (section 4.2.1.4), NTK-based bounds[§§] (section 4.2.1.3), other PAC-Bayes bounds[¶¶] (section 4.2.2)	non-uniform stability bounds[***] (section 4.3.1), marginal-likelihood PAC-Bayes bound[†††] (section 5)

Table 1: Classification of the main types of generalization bounds treated in this paper. Roughly speaking, the number of assumptions grows going from left to right, and from top to bottom. Note that, as we discussed in section 3.3.4, algorithm dependent bounds based on non-uniform convergence are automatically data-dependent, which is why there is an empty cell.

[*] Vapnik and Chervonenkis (1974); Blumer et al. (1989); Harvey et al. (2017)
[†] Vapnik (1995); McAllester (1998)
[‡] Bousquet and Elisseeff (2002); Hardt et al. (2016); Mou et al. (2018)
[§] Littlestone and Warmuth (1986); Brutzkus et al. (2018)
[¶] Bartlett and Mendelson (2002)
[**] Shawe-Taylor et al. (1998); Shawe-Taylor and Williamson (1997)
[††] Bartlett (1997, 1998); Bartlett et al. (2017); Neyshabur et al. (2018a); Golowich et al. (2018); Neyshabur et al. (2018b); Barron and Klusowski (2019)
[‡‡] Neyshabur et al. (2017); Dziugaite and Roy (2017); Arora et al. (2018); Banerjee et al. (2020)
[§§] Arora et al. (2019); Cao and Gu (2019)
[¶¶] Zhou et al. (2018); Dziugaite and Roy (2018)
[***] Kuzborskij and Lampert (2017)
[†††] Valle-Pérez et al. (2018)

Guillermo
Valle Perez

Big review paper on generalization bounds, includes 7 desiderata bounds should satisfy and a classification

Generalization bounds for deep learning Guillermo Valle-Pérez and AAL, arxiv:arXiv:2012.04115

Function based picture and PAC-Bayes bounds

PAC-Bayes bound

$$\forall \mathcal{D},\ \mathbf{P}_{S\sim\mathcal{D}^m}\left[\forall Q\ KL(\mathbf{E}_{h\sim Q}[\epsilon(h)], \mathbf{E}_{h\sim Q}[\hat{\epsilon}(h)]) \leq \frac{KL(Q||P) + \ln\frac{1}{\delta} + \ln(2m)}{m-1}\right] \geq 1-\delta \tag{13}$$

where $KL(Q||P)$ is the KL-divergence between Q and P. On the left hand side we use the standard abuse of notation to define $KL(a,b) \equiv a\ln(a/b) + (1-a)\ln((1-a)/(1-b))$, for $a, b \in [0,1]$.

David McAllester COLT (1998)

We prove that function based will (in principle) always be better than parameter based PAC-Bayes bounds

$$KL(Q||P) \leq KL(Q_{\mathrm{par}}||P_{\mathrm{par}})$$

Generalization bounds for deep learning Guillermo Valle-Pérez and AAL, arxiv:arXiv:2012.04115

Function based picture and PAC-Bayes bounds

Guillermo
Valle Perez

Theorem 5.1. ***(marginal-likelihood PAC-Bayes bound)***

For any distribution P on any hypothesis space $\mathcal{H}$ and any realizable distribution $\mathcal{D}$ on a space of instances we have, for $0 < \delta \leq 1$, and $0 < \gamma \leq 1$, that with probability at least $1-\delta$ over the choice of sample S of m instances, that with probability at least $1-\gamma$ over the choice of h:

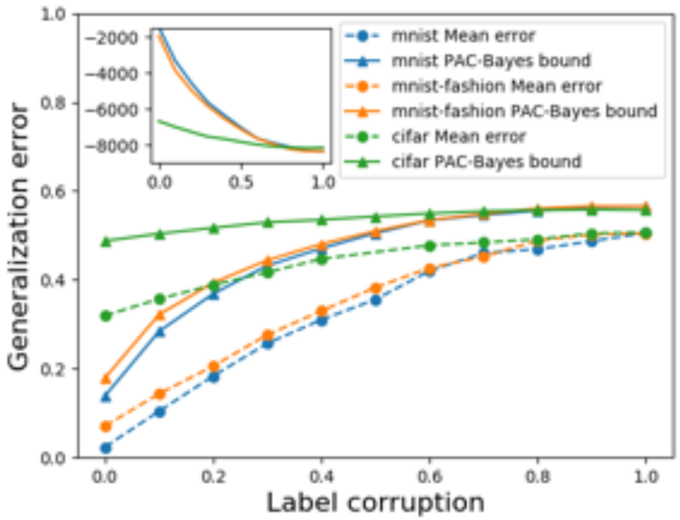
$$-\ln\left(1-\epsilon(h)\right) < \frac{\ln \frac{1}{P(C(S))} + \ln m + \ln \frac{1}{\delta} + \ln \frac{1}{\gamma}}{m-1}$$

where h is chosen according to the posterior distribution $Q(h) = \frac{P(h)}{\sum_{h\in C(S)} P(h)}$, $C(S)$ is the set of hypotheses in $\mathcal{H}$ consistent with the sample S, and where $P(C(S)) = \sum_{h\in C(S)} P(h)$

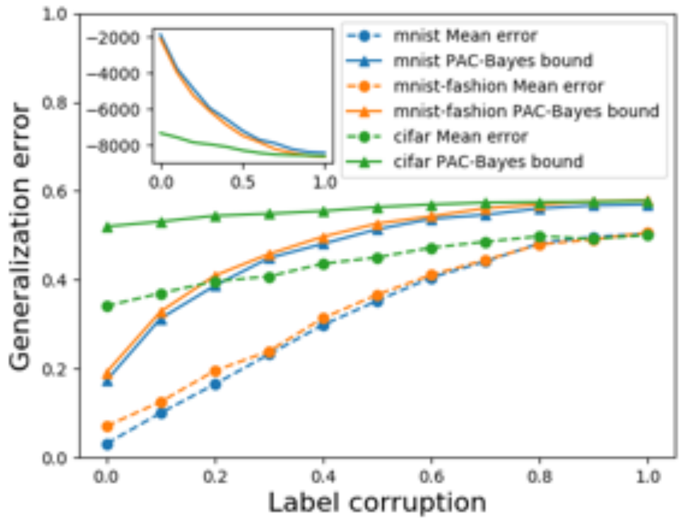
Marginal-likelihood = sum over functions (hypotheses) h

Generalization bounds for deep learning Guillermo Valle-Pérez and AAL, arxiv:arXiv:2012.04115

Tight PAC-Bayes bounds: error with complexity



(a) for a 4 hidden layers convolutional network



(b) for a 1 hidden layer fully connected network

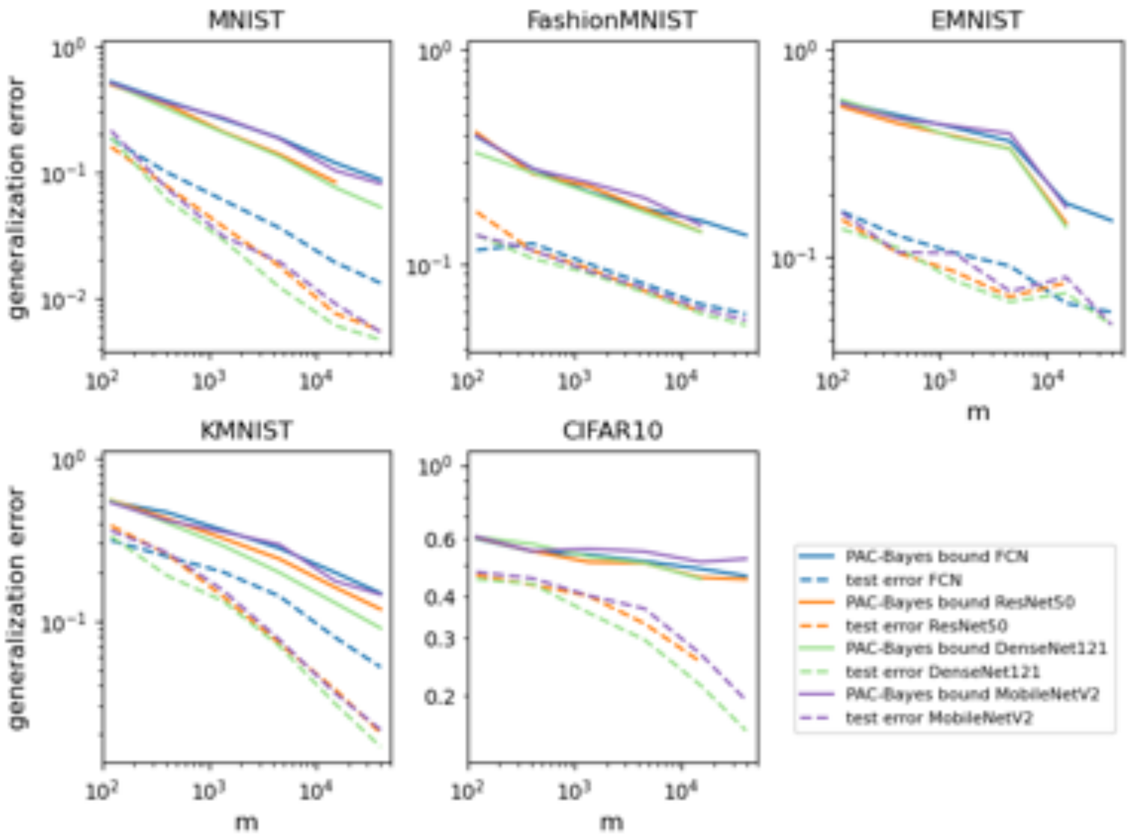
Guillermo
Valle Perez

Marginal-likelihood PAC-Bayes bound

$$-\ln\left(1-\epsilon(h)\right) < \frac{\ln \frac{1}{P(C(S))} + \ln m + \ln \frac{1}{\delta} + \ln \frac{1}{\gamma}}{m-1}$$

Generalization bounds for deep learning Guillermo Valle-Pérez and AAL, arxiv:arXiv:2012.04115

Tight PAC-Bayes bounds: learning curves with m

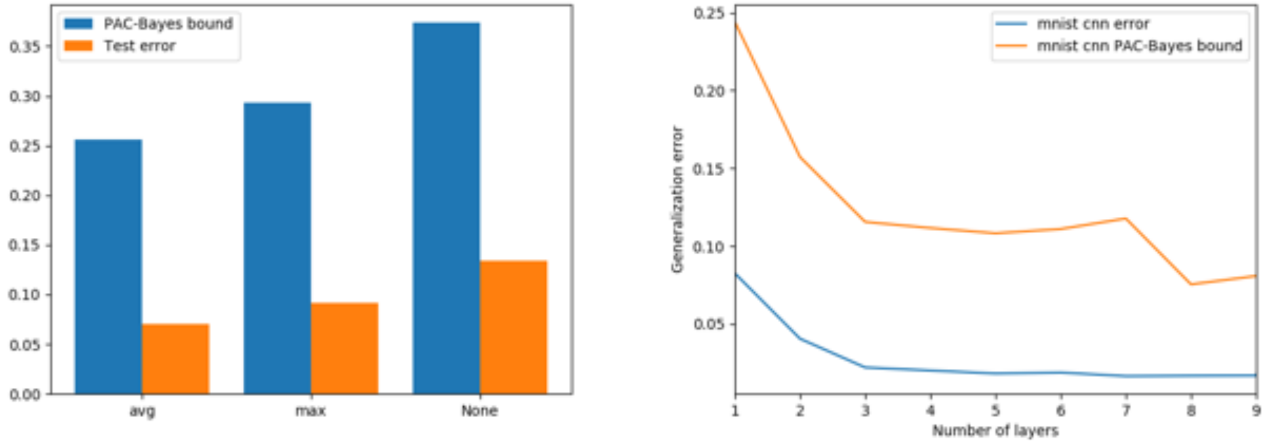


Guillermo
Valle Perez

$$-\ln\left(1-\epsilon(h)\right) < \frac{\ln \frac{1}{P(C(S))} + \ln m + \ln \frac{1}{\delta} + \ln \frac{1}{\gamma}}{m-1}$$

Generalization bounds for deep learning Guillermo Valle-Pérez and AAL, arxiv:arXiv:2012.04115

Tight PAC-Bayes bounds: comparing architectures



(a) (b)

Figure 6: **PAC-Bayes bound and generalization error versus different architecture hyperparameters**. (a) Error versus pooling type, for a CNN trained on a sample of 1k images from KMNIST. (b) Error versus number of layers for a CNN trained on a sample of size 10k from MNIST. Training set error is 0 in all experiments. We used SGD with batch 32 for both of these experiments.

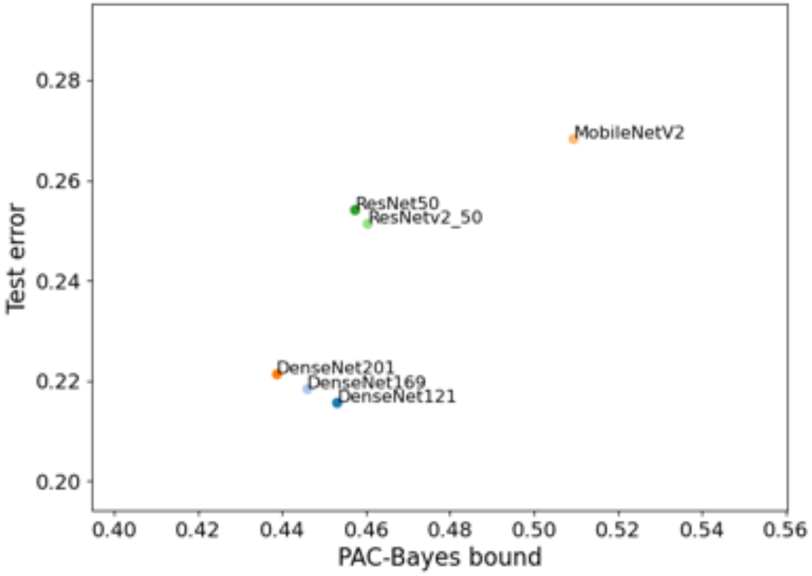
Guillermo
Valle Perez

Marginal-likelihood bound

$$-\ln\left(1-\epsilon(h)\right) < \frac{\ln\frac{1}{P(C(S))}+\ln m+\ln\frac{1}{\delta}+\ln\frac{1}{\gamma}}{m-1}$$

Generalization bounds for deep learning Guillermo Valle-Pérez and AAL, arxiv:arXiv:2012.04115

Tight PAC-Bayes bounds: comparing architectures



(a) CIFAR10

Guillermo
Valle Perez

Error at m=15,000 training set for some SOTA networks

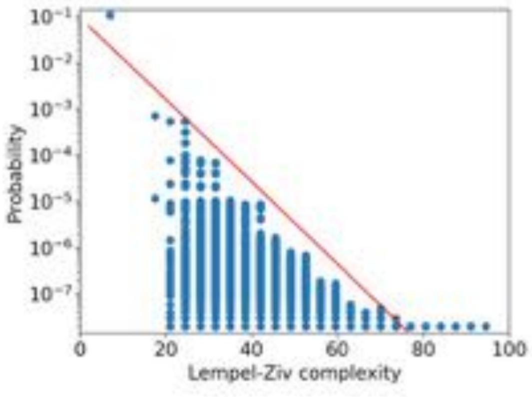
Marginal-likelihood bound

$$-\ln(1-\epsilon(h)) < \frac{\ln \frac{1}{P(C(S))} + \ln m + \ln \frac{1}{\delta} + \ln \frac{1}{\gamma}}{m-1}$$

Generalization bounds for deep learning Guillermo Valle-Pérez and AAL, arxiv:arXiv:2012.04115

Thanks!

$$P(x) \lesssim 2^{-a\tilde{K}(x)-b}$$



Occam's razor

Conclusions:

1) DNNs generalize because they have an implicit bias towards simple functions, as predicted by AIT
2) SGD acts as a Bayesian optimizer, it is not the source of the good generalization performance
3) Many common intuitions from learning theory, such as bias-variance tradeoff etc… don't work for DNNs, but:
4)Our marginal-likelihood PAC-Bayes bound performs well

Kamal Dingle

Chico Camargo

Guillermo Valle Perez

Shuofeng Zhang

Yoonsoo Nam

David Martinez

Ouns El Harzli

Henry Rees

Hertford College undergraduates

Chris Mingard

Joar Skalse

Vlad Mikulik

Isaac Reid