POLITECNICO DI TORINO
Repository ISTITUZIONALE

Universal mean-field upper bound for the generalization gap of deep neural networks

*Original*
Universal mean-field upper bound for the generalization gap of deep neural networks / Ariosto, S., Pacelli, R., Ginelli, F., Gherardi, M., Rotondo, P.. - In: PHYSICAL REVIEW. E. - ISSN 2470-0053. - 105:6(2022). [10.1103/PhysRevE.105.064309]

*Availability:*
This version is available at: 11583/2983564 since: 2023-11-03T07:50:24Z

*Publisher:*
APS

*Published*
DOI:10.1103/PhysRevE.105.064309

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*
APS postprint/Author's Accepted Manuscript e postprint versione editoriale/Version of Record

This article appeared in PHYSICAL REVIEW. E, 2022, 105, 6, and may be found at ttp://dx.doi.org/10.1103/PhysRevE.105.064309. Copyright 2022 American Physical Society

(Article begins on next page)

26 November 2026

# Universal mean field upper bound for the generalisation gap of deep neural networks

S. Ariosto,[1,2] R. Pacelli,[3] F. Ginelli,[1,2] M. Gherardi,[4,2] and P. Rotondo[4,2]

[1] *Dipartimento di Scienza e Alta Tecnologia and Center for Nonlinear and Complex Systems, Università degli Studi dell'Insubria, Via Valleggio 11, 22100 Como, Italy*
[2] *I.N.F.N. Sezione di Milano, Via Celoria 16, 20133 Milano, Italy*
[3] *Dipartimento di Scienza Applicata e Tecnologia, Politecnico di Torino, 10129 Torino, Italy*
[4] *Università degli Studi di Milano, Via Celoria 16, 20133 Milano, Italy*

arXiv:2201.11022v2 [cond-mat.dis-nn] 2 Mar 2022

Modern deep neural networks (DNNs) represent a formidable challenge for theorists: according to the commonly accepted probabilistic framework that describes their performance, these architectures should overfit due to the huge number of parameters to train, but in practice they do not. Here we employ results from replica mean field theory to compute the generalisation gap of machine learning models with quenched features, in the teacher-student scenario and for regression problems with quadratic loss function. Notably, this framework includes the case of DNNs where the last layer is optimised given a specific realisation of the remaining weights. We show how these results – combined with ideas from statistical learning theory – provide a stringent asymptotic upper bound on the generalisation gap of fully trained DNN as a function of the size of the dataset $P$. In particular, in the limit of large $P$ and $N_{\rm out}$ (where $N_{\rm out}$ is the size of the last layer) and $N_{\rm out} \ll P$, the generalisation gap approaches zero faster than $2N_{\rm out}/P$, for any choice of both architecture and teacher function. Notably, this result greatly improves existing bounds from statistical learning theory. We test our predictions on a broad range of architectures, from toy fully-connected neural networks with few hidden layers to state-of-the-art deep convolutional neural networks.

## I. INTRODUCTION

In the last ten years deep neural networks (DNNs) [1] revolutionised the field of Machine Learning, outperforming traditional methods in tasks that include image classification, speech recognition and time series prediction. Despite the enormous success in applications, the size of these architectures represents a puzzle for theorists. When Alexnet won the ImageNet competition in 2012 [2], it had roughly 60 million parameters. In the following years, the VGG network delivered the state-of-the-art performance with more than 138 million parameters [3]. Nowadays, convolutional DNNs such as ResNet or Inception work by training about 10 million weights [4, 5]. According to common intuition, models with such a high number of degrees of freedom should overfit the training data, and perform poorly on previously unseen data samples. Statistical learning theory (SLT) [6], the established probabilistic framework to quantify the generalisation performance in machine learning, does not provide any guarantee that such severely overparametrised models should have any predictive power on test data [7, 8].

Overcoming this conceptual puzzle engages computer scientists, mathematicians and physicists alike [9–11]. In Ref. [12], the authors provide a mean field view of the stochastic gradient dynamics of one-hidden layer networks by using the theory of gradient flows in Wasserstein spaces [13]. Unfortunately, it is challenging to extend this approach to deeper networks. Other groups are studying the role of overparametrisation and related phenomena such as the double descent in the regime of lazy training [14–20]. Also the statistical physics of kernel learning (originally started in [21]) has undergone a revival in the last few years [22], mainly due to the discovery of the Neural Tangent Kernel (NTK) limit of deep neural networks —a mathematical equivalence between neural networks and a certain kernel that arises in the limit of large layer size [23, 24]. Despite all these major conceptual advances in the field, it is fair to say that a unified framework to investigate and understand the generalisation performance of DNNs is still missing.

On the mathematical side, it is instructive to rationalise why the theorems proven in the framework of statistical learning theory often yield very loose bounds when applied to practical problems (as brilliantly put forward in Ref. [25] by Bottou or in the recent review [26] by Belkin). The goal of theorems in SLT is to provide *distribution-independent uniform* bounds on the deviation between the generalisation and training errors. The formulation and the derivation of these theorems reveal a source of possible reasons for their poor quantitative performance: (i) empirically relevant data distributions may lead to smaller typical deviations than the worst possible case [27–31]; (ii) uniform bounds hold for all possible functions in the model, but better bounds may hold when one restricts the analysis to functions that perform well on specific (and significative) training sets.

In this manuscript, we build upon these considerations to develop a mean field theory for the generalisation gap (GG) of deep neural networks. Firstly, we employ non-rigorous but standard statistical physics tools of disordered systems [32] to compute the generalisation and training errors of machine learning models with *quenched features*, obtaining simple formulas in the regime of large training dataset size $P$. In particular these results hold in the teacher-student scenario for a broad class of input-output distributions when the employed loss function is the mean squared error. Notably, this setup includes the case of DNNs where the $N_{\rm out}$ weights in the last layer are optimised given any specific instance of the remain-

ing weights. Analogous results have been derived in the recent literature [20, 22, 33]; here we show how to employ them to derive a universal mean field upper bound for the generalisation gap of fully trained DNNs. In the limit $N_{\rm out} \ll P$ (a condition satisfied by most state-of-the-art DNNs, even in the overparametrised regime where $P$ is small compared to the total number of weights), a simple asymptotic upper bound emerges, according to which the gap should approach zero faster than $2N_{\rm out}/P$. This is our central result; in the large $P$ limit, greatly improves existing estimates. Finally, we check the validity of our mean field bound against several synthetic and empirical data distributions and across a variety of different architectures, ranging from toy DNNs with few fully-connected layers to state-of-the-art ones employed for challenging computer vision problems.

Although these results lacks the mathematical rigor of formal theorems, it takes a concrete step towards understanding why overparametrised DNNs work in practice, and may guide to the formulation of more informed and accurate bounds for the generalisation gap of modern DNNs.

## II. GENERALISATION GAP OF QUENCHED FEATURES MODELS

We start by briefly describing the setting of the supervised learning problem that we will study throughout the manuscript. Let us consider a training set $\mathcal{T}$ made of $P$ independent identically distributed (IID) random observations, $\mathcal{T} = \{(\mathbf{x}^\mu, y^\mu)\}_{\mu=1}^P$, where the $\mathbf{x}^\mu$'s are $D$-dimensional vectors drawn by an input probability distribution $\rho(\mathbf{x})$ and the $y^\mu$'s are scalar outputs provided by a real-valued teacher function $f_{\rm T}$, i.e. $y^\mu = f_{\rm T}(\mathbf{x}^\mu)$. Under these assumptions, the joint input/output probability distribution $\rho_{\rm I/O}(\mathbf{x}, y)$ is given by $\rho_{\rm I/O}(\mathbf{x}, y) = \rho(\mathbf{x})\delta\left(y - f_{\rm T}(\mathbf{x})\right)$.

Our first goal is to compute the generalisation performance of a model (the so-called *student*) of the following form

$$f_{\rm S}(\mathbf{x}) = \sum_{\alpha=1}^{N} v_\alpha \phi_\alpha(\mathbf{x}) = \mathbf{v}\cdot\boldsymbol{\phi}(\mathbf{x}), \tag{1}$$

where $\boldsymbol{\phi}$ is an $N$-dimensional feature map and $\mathbf{v}$ is an $N$-dimensional vector of real weights to be optimised. The average generalisation and training errors are defined as:

$$\epsilon_{\rm g} = \left\langle \int d^D x \rho(\mathbf{x}) \left[f_{\rm T}(\mathbf{x}) - f_{\rm S}^*(\mathbf{x})\right]^2 \right\rangle_{\mathcal{T}}, \tag{2}$$

$$\epsilon_{\rm t} = \left\langle \frac{1}{P}\sum_{\mu=1}^{P} \left[f_{\rm T}(\mathbf{x}^\mu) - f_{\rm S}^*(\mathbf{x}^\mu)\right]^2 \right\rangle_{\mathcal{T}}, \tag{3}$$

where $\langle\cdot\rangle_{\mathcal{T}}$ indicates the average over all the possible realisations of a training set of size $P$, and the optimised function $f_{\rm S}^*$ corresponds to the choice of the vector $\mathbf{v}^*$ that minimises the quadratic loss $P^{-1}\sum_{\mu=1}^P \left[f_{\rm T}(\mathbf{x}^\mu) - f_{\rm S}(\mathbf{x}^\mu)\right]^2$ for a given instance of the dataset $\mathcal{T}$. Although in principle this approach can be developed for arbitrary loss functions [20], here we will only consider quadratic loss and regression problems, which are considerably simpler to deal with analytically [22, 34, 35].

The generalisation power of a machine learning model can be measured by its *generalisation gap*,

$$\Delta\epsilon = \epsilon_{\rm g} - \epsilon_{\rm t}\,, \tag{4}$$

which expresses the average performance difference of a trained model between its training dataset and unseen data drawn from the same distribution. Crucially, it is possible to express the generalization and training errors as a function of the features; as we will discuss in the following, this ingredient is fundamental to provide insight on the generalisation gap of fully-trained DNNs. Here the calculation of the average generalisation and training errors is performed using the well-known replica method, a standard statistical physics technique developed to study disordered systems [32]. Optimisation is addressed introducing an effective Hamiltonian given by the sum of the training loss with a regularisation term:

$$\mathcal{L} = \frac{1}{2}\sum_{\mu=1}^{P}\left[f_{\rm T}(\mathbf{x}^\mu) - f_{\rm S}(\mathbf{x}^\mu)\right]^2 + \frac{\lambda}{2}\sum_{\alpha=1}^{N}(v_\alpha)^2\,, \tag{5}$$

gauged by the regularisation parameter $\lambda > 0$. Given the convexity of our problem, we can make use of the replica symmetric ansatz, which is known to deliver the correct result for convex optimisation when $P \gg 1$ [21]. The ground state of the effective Hamiltonian, given by the optimised $f_{\rm S}^*$, is finally evaluated from the replicated partition function in the large $P$ limit via the saddle point method. This approach is rather similar to that developed for the random features model (RFM) [15] and for kernel regression [22] and it is discussed in detail in the appendices.

It turns out that the analytical expressions for the generalisation and training errors depend on the following integrals over the input probability distribution:

$$\begin{aligned} J_\alpha &= \int d^D x \rho(\mathbf{x}) f_{\rm T}(\mathbf{x})\phi_\alpha(\mathbf{x})\,,\\ \Phi_{\alpha\beta} &= \int d^D x \rho(\mathbf{x})\phi_\alpha(\mathbf{x})\phi_\beta(\mathbf{x})\,,\\ T &= \int d^D x \rho(\mathbf{x}) f_{\rm T}^2(\mathbf{x})\,. \end{aligned} \tag{6}$$

with $\alpha, \beta = 1, \dots N$. The vector $\mathbf{J}$ and the matrix $\boldsymbol{\Phi}$ depend on the specific choice of the feature map $\boldsymbol{\phi}$ and respectively represent a teacher-feature and a feature-feature overlap, whereas the scalar quantity $T$ is by definition the trivial predictor [36] of the regression problem and it provides a natural scale to compare different learning problems. Using these definitions, we find the

following compact representation for the generalisation and training errors (valid in the thermodynamic limit of large $D, N, P$ as discussed in appendix A):

$$\begin{aligned}\epsilon_{\rm g} &= \frac{\epsilon_{\rm g}^{\rm R} + (\kappa\lambda)^2 \mathbf{J}^T \mathbf{\Phi}^{-1} \mathbf{G}^{-2} \mathbf{J}}{1 - P {\rm Tr}\left(\mathbf{\Phi}^2 \mathbf{G}^{-2}\right)}\,, \\ \epsilon_{\rm t} &= \frac{\epsilon_{\rm g}^{\rm R}}{\kappa} + \frac{\epsilon_{\rm g}}{\kappa}\left(\frac{\kappa-1}{\kappa} - \frac{N}{P}\right)\,,\end{aligned} \tag{7}$$

where the matrix $\mathbf{G} = \kappa\lambda\mathbb{1} + P\mathbf{\Phi}$ is invertible and the variable $\kappa$ is self-consistently defined via the following equation:

$$\kappa = 1 + \kappa {\rm Tr}\left(\mathbf{\Phi}\mathbf{G}^{-1}\right)\,. \tag{8}$$

The residual generalisation error $\epsilon_{\rm g}^{\rm R}$ corresponds to the best possible performance of the quenched model on the dataset, under the assumption of full knowledge of the input/output probability distribution, and it is given by:

$$\epsilon_{\rm g}^{\rm R} = T - \mathbf{J}^T \mathbf{\Phi}^{-1} \mathbf{J}\,. \tag{9}$$

It is worth notincing that for strictly infinite size of the dataset $P$, $\kappa \to 1$ and it is easy to prove that $\epsilon_{\rm g} \to \epsilon_{\rm g}^{\rm R}$ and $\epsilon_{\rm t} \to \epsilon_{\rm g}$, which provide a first consistency check of the validity of the mean field theory. Additionally, the self-consistent definition of $\kappa$ and the way it enters in the expression for the generalisation error are the same as in the recent work on kernel regression by Pehlevan's group [22]. This should not come as a surprise, since one could specialise the general quenched features that we employ here to the case of polynomial, Gaussian or NTK kernels . For these particular choices of the quenched features, Eq. (7) just provide a different representation of the generalisation error formula given in [22]. A generalization of Eq. (7) has been recently proved in [20].

Starting from Eq. (7) it is possible to perform an asymptotic analysis for large size of the training set $P$. This is particularly simple if we assume that both $P$ and $N$ are large, $N \ll P$ and the regularisation parameter $\lambda$ is finite. In this case we easily obtain that $\kappa \sim 1 + N/P$ and $\mathbf{G} \sim P\mathbf{\Phi}$. Using these asymptotic expressions, and normalising by the natural scale of the problem – i.e. the trivial predictor $T$ defined in Eq. (6) – we can compute the normalised generalisation gap:

$$\Delta\tilde{\epsilon} \equiv \frac{\Delta\epsilon}{T} \simeq 2\frac{\epsilon_{\rm g}^{\rm R}}{T}\frac{N}{P}\,. \tag{10}$$

Note that Eq. (10) does not necessarily imply a linear scaling with $N$ since $\epsilon_{\rm g}^{\rm R}$ may still retain a dependence on $N$ (as implied from Eq. (9)).

## III. GENERALISATION GAP OF FULLY TRAINED DNNS

Let us now suppose that the quenched features of the model under consideration are given by a DNN. For instance, in the special case of a fully-connected architecture with one hidden layer, we have that the function implemented is $f_{\rm 1HL}(\mathbf{x}) = \sum_{\alpha=1}^{N} v_\alpha \sigma\left(W_\alpha \cdot \mathbf{x}\right)$, where for simplicity we have set all the biases to zero. The $W_\alpha$'s are the $D$-dimensional vector weights of the hidden layer and $\sigma$ is a generic well-behaved activation function. Here the quenched features are given by $\phi_\alpha^{\rm 1HL}(\mathbf{x}, W) = \sigma\left(W_\alpha \cdot \mathbf{x}\right)$. More in general, let us consider a DNN with a fully-connected last layer. The specific architecture of the first layers is uninfluential. This class includes all the relevant state-of-the-art architectures. Let us fix the dimension of the last layer to $N = N_{\rm out}$ and let us split the weights $\vartheta$ of the network as $\vartheta = \{\mathbf{v}, \mathcal{W}\}$ where $\mathbf{v}$ is the vector of $N_{\rm out}$-dimensional weights of the last fully-connected layer, whereas $\mathcal{W}$ is a short notation for all the remaining weights of the DNN. We introduce the feature map notation $\phi_\alpha^{\rm DNN}(\mathbf{x}, \mathcal{W})$ to indicate the corresponding quenched features.

We now reconsider the results provided by Eq. (7) when specialised to the quenched features of a DNN. The crucial observation is that mean field theory provides the average generalisation and training errors for *each* realisation of the weights $\mathcal{W}$. In other words, given a specific configuration $\bar{\mathcal{W}}$, our theory predicts the corresponding average generalisation and training errors, supposing that the weights $\mathbf{v}$ of the last layer are set to the optimal value that minimises the training loss at fixed $\bar{\mathcal{W}}$. From now on, we use the notation $\epsilon_{\rm g}\left(\mathcal{W}\right)$, $\epsilon_{\rm g}^{\rm R}\left(\mathcal{W}\right)$, $\epsilon_{\rm t}\left(\mathcal{W}\right)$ to stress that the generalisation and training errors depend on $\mathcal{W}$ via the teacher-feature $\mathbf{J}$ and feature-feature $\mathbf{\Phi}$ overlaps.

Therefore, the result in Eq. (10) holds for each given realisation of the weights $\mathcal{W}$ of the DNN if we assume perfect training over the last layer. See also the recent conjecture put forward in [20]. In particular, this equivalence holds for a fully trained configuration $\theta^* \equiv \{v^*, \mathcal{W}^*\}$ that is a *local* minimum of the loss. Unfortunatley, such local minimum may depend on the size $P$ of the training set, and so it does $\epsilon_{\rm g}^{\rm R}\left(\mathcal{W}\right)$. However, since the residual generalisation error (9) is positive by definition and bounded by $T$, it follows that $0 \leq \epsilon_{\rm g}^{\rm R}\left(\mathcal{W}\right)/T \leq 1$ for every $\mathcal{W}$. As such, this provides us with the following asymptotic mean field upper bound for the (normalised) generalisation performance of a DNN:

$$\Delta\tilde{\epsilon}\left(\mathcal{W}\right) \leq \frac{2N_{\rm out}}{P}\,, \tag{11}$$

which is the central result of this manuscript.

## IV. NUMERICAL EXPERIMENTS

### A. Toy DNNs with synthetic datasets.

We start by testing our bound on a fully-connected architecture with one-hidden layer and ReLU activations. We have chosen three different teacher classes of increasing complexity: (i) a linear function $f_{\rm T}(\mathbf{x}) = \mathbf{t} \cdot \mathbf{x}$; (ii) a quadratic polynomial $f_{\rm T}(\mathbf{x}) = \mathbf{t} \cdot \mathbf{x} + (\mathbf{t} \cdot \mathbf{x})^2$; (iii) a fully-connected one-hidden layer (1HL) architecture with

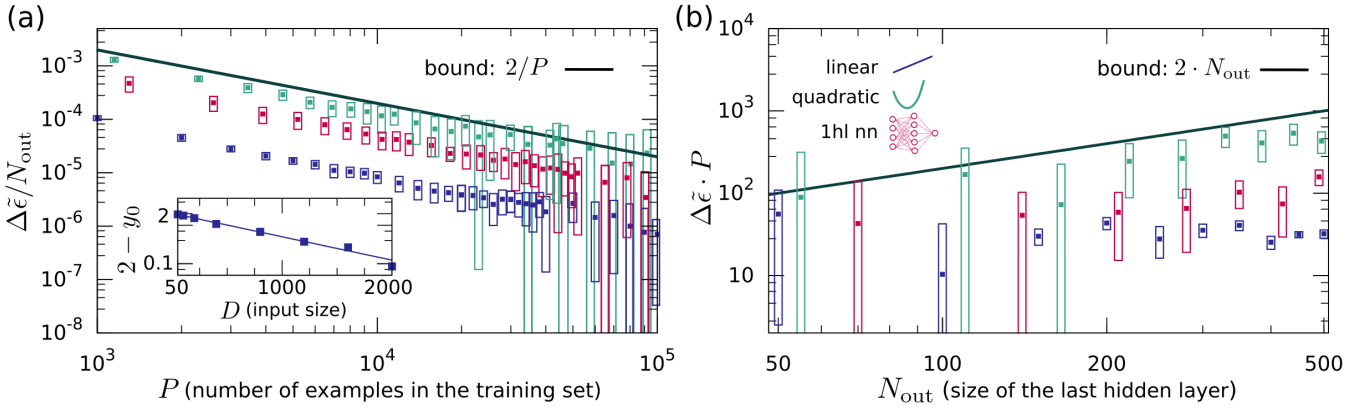


FIG. 1. **Generalisation gap in the lazy-training regime.** The behaviour of the normalized GG for a one-hidden layer student architecture is displayed for the three different classes of teacher outlined in the main text: linear (blue symbols), quadratic (green) and one-hidden layer (red). The solid black line marks our mean field upper bound. (a) Normalised GG (rescaled by $N_{\rm out}$) as a function of the training dataset $P$. Data points are the result of an average over 50 different realisations of the teacher and of the input (of dimension $D = 50$) with $N_{\rm out} = 400$. We observe that the functional form of the rescaled GG $\Delta\tilde{\epsilon}$ is compatible with $y_0/P$ over two decades. In the inset we consider different input dimension for the linear teacher case, showing that by increasing $D$, the residual generalisation gap converges exponentially fast to the bound, i.e. the prefactor $y_0$ (which is obtained by a standard fitting procedure) converges exponentially from below to 2 as $D \to \infty$ with a rate $\gamma \approx 0.0014$ (see also appendix B). (b) The normalised GG (multiplied by $P$) is shown as a function of the size of the hidden layer $N_{\rm out}$. Simulations are performed with $P$ equal to $2\cdot10^4$ (linear teacher), $4\cdot10^4$ (quadratic) and $8\cdot10^4$ (1HL), and by averaging over 20 different teacher and input realisations ($D = 50$). Error bars in both panels correspond to one standard error. Typical (rescaled by $T$) training errors in the lazy-training regime are of order $10^{-1}$ and are systematically smaller for the linear teacher.

ReLU activations, $f_{\rm T}(\mathbf{x}) = \sum_{\alpha=1}^{M} q_\alpha \mathrm{ReLU}\left(\mathbf{S}_\alpha \cdot \mathbf{x}\right)$. See also appendix B for details on these architectures and on the inputs choice.

We first consider DNNs where only the last layer is trained and the remaining weights are kept fixed to their initialisation values $\bar{\mathcal{W}}$, i.e. we consider the lazy training regime. Since the $\bar{\mathcal{W}}$'s do not change during training, the residual generalisation error $\epsilon_{\rm g}^{\rm R}\left(\bar{\mathcal{W}}\right)$ is independent of $P$ and one expects not only the bound (11) to hold but also the generalisation gap to scale precisely as $1/P$ for $P$ large enough. This is verified in Fig. 1a for the three different teacher classes introduced above. On the other hand, as already noted, $\epsilon_{\rm g}^{\rm R}\left(\bar{\mathcal{W}}\right)$ may still retain a dependence on the last layer size $N_{\rm out}$. In particular, one may expect that increasing $N_{\rm out}$ will decrease the residual generalisation error, as this increases the number of functions available to approximate the target $f_{\rm T}(\mathbf{x})$. Therefore, for large $P$ and $N_{\rm out}$ we expect $\Delta\tilde{\epsilon}$ to increase *at most* linearly as a function of $N_{\rm out}$. The numerical behaviour of the GG as a function of $N_{\rm out}$ is shown in Fig. 1b. Once again, our mean field bound holds, but different scaling with $N_{\rm out}$ can be appreciated. In particular, the GG is almost constant for the linear teacher, whereas it has an approximately linear behaviour for the quadratic one, reflecting different dependencies of the residual generalisation error from the last layer size. Note also that in both panels of Fig. 1 the GG is systematically lower for the linear teacher case, confirming the intuitive expectation that the linear problem should be the easiest to learn.

It is worth remarking that the input dimension $D$ only enters the theory through the residual generalisation error (9). Interestingly, as one increases the input dimension $D$, the normalised generalisation gap seems to saturate the unit bound with an exponential convergence in $D$, as shown in the inset of Fig. 1a for the linear teacher case (more details in appendix B). Currently we have no theory for this.

We next consider fully-trained DNNs. Here the weights $\mathcal{W}$ are trained and the residual generalisation error $\epsilon_{\rm g}^{\rm R}\left(\mathcal{W}\right)$ may dipend on the training set size $P$. Suppose for instance that there exists a configuration of the weights $\mathcal{W}^\dagger$ such that the residual generalisation error $\epsilon_{\rm g}^{\rm R}\left(\mathcal{W}^\dagger\right) = 0$, i.e. the DNN can learn the teacher function perfectly. Intuitively, as the size of the dataset $P$ grows, we expect that the DNN will be capable of finding configurations $\mathcal{W}$ that are closer to the optimal one $\mathcal{W}^\dagger$. This means that the residual generalisation error will decrease in some way towards zero as a function of $P$ for $P$ large enough, and that the the GG will drecrease faster than $1/P$.

In the complementary case where the DNN is not capable of learning the target function, there will be a nonzero residual generalisation error $\epsilon_{\rm g}^{\rm R}\left(\mathcal{W}\right) = \hat{\epsilon}^{\rm R}$ even for $P \to \infty$: the $1/P$ scaling of the GG will thus be restored asymptotically.

Numerical simulations for fully-trained DNNs are shown in Figure 2. As expected from the considerations raised above, and differently from the lazy training regime, here we do not observe a simple $1/P$ scaling of

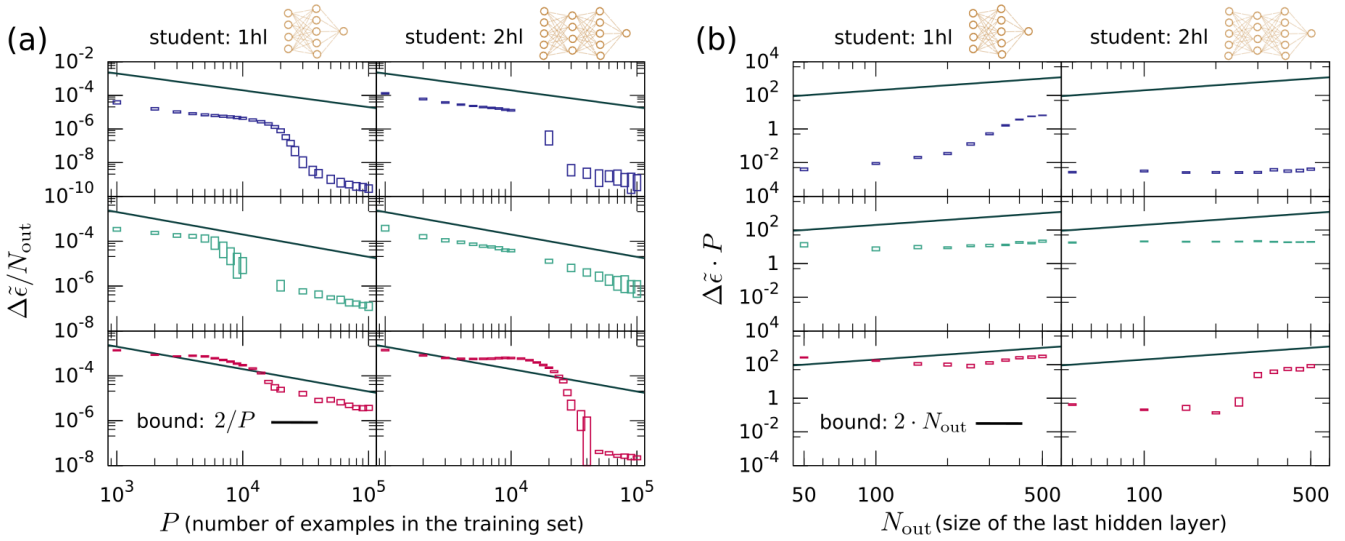


FIG. 2. **Generalisation gap of fully-trained toy DNNs** with one (left columns in (a)-(b)) and two (right columns) fully-connected hidden layers for the three different teacher classes outlined in the main text. From top to bottom: linear (blue symbols), quadratic (green) and 1HL teacher architectures (red). The solid black line marks the mean field upper bound. (a) Normalised GG (rescaled by $N_{\rm out}$ as a function of the size of the training set $P$. Data points are the result of an average over 50 realisations of the teacher and of the input ($D = 50$) with $N_{\rm out} = 100$. (b) Normalised GG (multiplied by $P$ vs. the size of the last hidden layer $N_{\rm out}$. Data is averaged over 20 different teacher and input ($D = 50$) realisations with $P = 4 \cdot 10^4$. Error bars in both panels correspond to one standard error. Typical training errors are of the order of $10^{-6}$ for each teacher/student pair, except in the case 1HL/1HL where the training error is of order $10^{-3}$.

the gap (Fig. 2a). On the contrary, the GG curves display two learning stages as a function of $P$, with the GG falling systematically below the mean field bound above $P \sim 10^4$. Once we enter in the second learning stage (for $P$ of the order of $10^4$) the gap however seems to approach zero as fast as $1/P$ for both one and two hidden layer architectures and across the different synthetic datasets, suggesting a fine constant residual generalisation error in this second learning stage. In Fig. 2b we analyse the generalisation performance as the width of the last layer $N_{\rm out}$ grows. According to our predictions, the bound holds and a linear or sub-linear degradation of the generalisation performance is systematically observed across the different student and teachers architectures.

### B. State-of-the-art architectures.

As an additional and more challenging test, we present the results for the generalisation gap obtained by training three different state-of-the-art convolutional architectures (ResNet18, DenseNet121 and VGG-11) on the MNIST dataset of handwritten digits [37]. Notice that this problem is in principle a classification problem, but our theory has been formulated for regression. For this reason we implemented the learning problem as a regression task: for each digit vector $\mathbf{x}$, the associated output is simply the integer number, between 0 and 9, corresponding to its class. Coherently, the performance of the network is not measured using the standard accuracy (i.e., the fraction of correctly classified digits), but as the mean square deviation between the network's output and the class index.

A summary of the simulations is found in Fig. 3 (details on the learning protocols are provided in appendix B). Remarkably, our bound is also satisfied in the case of state-of-the-art architectures trained on a dataset of practical relevance for computer vision. Moreover, notice that in the regime we are exploring the generalisation gap approaches zero faster than $1/P$.

## V. DISCUSSION AND FUTURE PERSPECTIVES

Our mean field analysis, while lacking the full rigor of theorems, establishes a much more stringent bound for the generalisation gap w.r.t. the ones obtained in the strict context of statistical learning theory. For instance, in the case of classification problems, SLT predicts an upper bound roughly proportional to $\sqrt{N_{\rm tot}/P}$ with $N_{\rm tot}$ being the *total* number of network parameters [38] (more correctly $N_{\rm tot}$ should be replace by the so-called Vapnik-Chervonenkis dimension of the DNN), while our upper bound depends on the number of parameters of the network only through the width of the last layer; this may lead to speculate that the width of the last layer plays a special role in DNN architectures. The reason why we have been able to obtain this improved non-rigorous upper bound on the generalisation gap is related to tak-

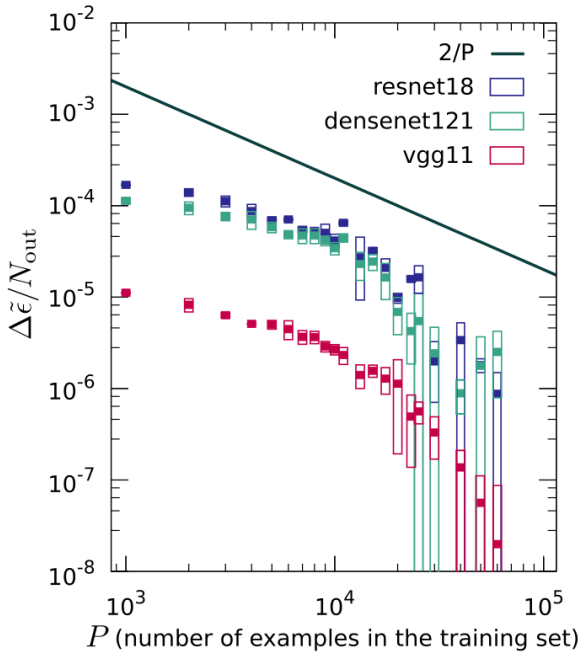


FIG. 3. **Generalisation gap for three state-of-the-art architectures trained on the MNIST dataset of handwritten digits.** The dependence of the normalised generalisation gap on the size of the training set $P$ is qualitatively similar for ResNet18 (blue symbols), DenseNet121 (green) and VGG-11 (red). The solid black line marks the mean field upper bound. Notice that by rescaling the GG by $N_{\rm out}$ we can better compare architectures with different last layer size. Averages have been performed over three different initial conditions for the architecture weights, and error bars measure one standard error. Typical training errors are of order $10^{-4}$.

ing optimisation into account, at least in the last layer. Whereas SLT attempts to find results for the generalisation gap that hold for every function in the model class, here we are restricting to special (but significative) elements of the class. In particular, for the specific case of a DNN with parameters $\vartheta = \{\mathbf{v}, \mathcal{W}\}$, classic SLT bounds hold for every realisation of $\vartheta$. On the contrary, our approach assumes that at fixed $\mathcal{W}$ the weights $\mathbf{v}$ of the last layer are optimised w.r.t. the training set.

It is fair to stress the major limitations of our mean field bound: (i) crucially the bounds in SLT hold for any size of the dataset and of the architecture. On the contrary, here we have no control on finite-size corrections, since our results hold only in the limit of large $P$ and $N_{\rm out}$. (ii) As observed in Fig. 2, there are cases in which the bound starts to hold only after a threshold $P_{\rm t} \approx 10^4 - 10^5$, which the theory does nothing to predict. However, our numerical experiments (e.g. those in Fig. 3) show that the threshold is rather small in many empirically relevant cases. (iii) A Gaussian approximation was performed at the level of the replicated partition function (for details see appendix A). Despite similar assumptions have been successfully employed in the past to study the statistical physics of kernels and of random feature models (where excellent agreement with numerics has also been found) [21, 22], we can not guarantee that this approximation is always quantitatively correct. (iv) These results hold for regression only. It would be interesting to understand whether the same non-rigorous tools could also be used to study classification problems [20].

Some of these drawbacks may be addressed by a more rigorous approach, for instance by the use of random matrix theory to avoid replicas (as done for instance in the case of kernel learning in [33, 39–42]). This may shed light on the dependence of the GG on the input dimension $D$, which we have shown numerically in the lazy training regime to saturate exponentially to the mean field bound.

In conclusion, we would like to point out an apparently intriguing consequence of our findings: since our mean field bound suggests a (linear or sub-linear) degradation of the generalisation performance with the last layer size $N_{\rm out}$, we might be led to surmise that, to improve generalisation performance, it may be convenient to design architectures with a small last layer. A more systematic investigation on state-of-the-art architectures is needed to understand whether this insight may lead to design more performant deep neural networks in the future.

## ACKNOWLEDGMENTS

The authors would like to thank V. Erba, B. Loureiro and M. Pastore for feedback on the manuscript and for pointing out recent related work. R. P. would like to thank her colleagues at Bocconi University for discussions. P. R. acknowledges funding from the Fellini program under the H2020-MSCA-COFUND action, Grant Agreement No. 754496, INFN (IT).

## Appendix A: Sketch of the replica symmetric calculation

We now discuss the salient aspects of the replica calculation. Further technical details can be found in appendix C. Our goal is to evaluate the generalisation and training errors defined in Eqs. (2), (3) for arbitrary teacher function $f_{\rm T}(\mathbf{x})$ and using as a student a function of the form (1).

### Replicated partition function

In order to evaluate these observables, one introduces a Gibbs distribution $p_G(\mathbf{v}) = \frac{1}{Z} e^{-\beta \mathcal{L}(\mathbf{v})}$, where $\mathcal{L}$ is the effective Hamiltonian defined in Eq. (5) and $\beta$ can be thought as the inverse of an effective temperature. The partition function $Z$ is given by:

$$Z = \int \mathrm{d}^N \mathbf{v} e^{-\frac{\beta}{2}\sum_\mu^P (f_{\mathrm{T}}(\mathbf{x}^\mu) - f_{\mathrm{S}}(\mathbf{v},\mathbf{x}^\mu))^2 - \frac{\beta}{2}\lambda\|\mathbf{v}\|^2} \tag{A1}$$

In the $\beta \to \infty$ limit, the Gibbs measure is dominated by the minimum of the Hamiltonian, which corresponds to the minimum of the training loss. As for many other problems in disordered systems, meaningful results can be obtained only by *quenched* averages, that is by averaging the logarithm of the partition function over all the possible realisations of the training set $\mathcal{T}$. Physically, this amounts to optimising first the student weights for any given instance of the dataset $\mathcal{T}$ and then averaging over all dataset realisations[43]. In order to perform this quenched average we exploit the standard replica method [32],

$$\langle \log Z \rangle_{\mathcal{T}} = \lim_{m\to 0} \frac{\langle Z^m \rangle_{\mathcal{T}} - 1}{m}\,, \tag{A2}$$

where one firstly computes the average of $Z^m$ for an integer number of replica $m$ and only later one performs the analytical continuation to $m \to 0$. Since the inputs are drawn as iid variables, the integral over the training set factorises to yield

$$\begin{aligned}\langle Z^m \rangle_{\mathcal{T}} = &\int \prod_{a=1}^m \mathrm{d}^N \mathbf{v}^a e^{-\frac{\beta}{2}\lambda\sum_a^m \|\mathbf{v}^a\|^2} \\ &\times \left[\int \mathrm{d}^D x\, \rho(\mathbf{x}) e^{-\frac{\beta}{2}\sum_a^m (q^a)^2}\right]^P\end{aligned} \tag{A3}$$

where $a$ is a replica index and we introduced a set of auxiliary random variables $q^a \equiv f_{\mathrm{T}}(\mathbf{x}) - \mathbf{v}^a \cdot \boldsymbol{\phi}(\mathbf{x})$ with mean $\mu_q^a(\{\mathbf{v}^a\}) = \langle q^a \rangle_\rho$ and covariance matrix:

$$Q_{ab}(\{\mathbf{v}^a\}) = \langle q^a q^b \rangle_\rho = T + (\mathbf{v}^a)^T\, \mathbf{\Phi}\, \mathbf{v}^b - \mathbf{J}^T \cdot (\mathbf{v}^a + \mathbf{v}^b)\,, \tag{A4}$$

(with $\mathbf{J}$, $\mathbf{\Phi}$ and $T$ given by Eq. (6)). To proceed further we note that each of the random variables $q^a$ is the sum of $N$ random variables. For a large last layer size $N$ and input dimension $D$ we approximate their probability distribution with a multivariate Gaussian with mean $\mu_q^a$ and covariance matrix $Q_{ab}$ (an order parameter which measures the overlap between replica $a$ and $b$). This allows us to perform the integration in the square brackets in Eq. (A3) to get

$$\begin{aligned}&\int \mathrm{d}^D x\, \rho(\mathbf{x}) e^{-\frac{\beta}{2}\sum_a^m (q^a)^2} = \int \prod_a^m dq^a e^{-\frac{\beta}{2}\sum_a^m (q^a)^2} \times \\ &\qquad \times \int d^D x \rho(\mathbf{x})\, \delta\left(q_a - \sum_\alpha^N v_\alpha^a \phi_\alpha(\mathbf{x}) + f_{\mathrm{T}}(\mathbf{x})\right) \\ &\simeq \int \prod_a^m dq^a\, \frac{e^{-\frac{\beta}{2}\sum_a (q^a)^2 - \frac{1}{2}\sum_{ab}(q^a)^T Q_{ab}^{-1} q^b}}{\sqrt{(2\pi)^m\, \det \mathbf{Q}}} = \\ &= \left(\det(\mathbb{I} + \beta \mathbf{Q})\right)^{-\frac{1}{2}}\end{aligned} \tag{A5}$$

where we have assumed $\mu_q^a = 0$ without loss of generality [22].

It is worth noticing that this Gaussian approximation is crucial in order to make progress, but rather uncontrolled, as we lack a formal result demonstrating its validity. Nonetheless similar non-rigorous approximations are quite standard in the literature on kernel learning, which include the seminal work on support vector machines by Dietrich, Opper and Sompolinsky [21], more recent findings on kernel regression [22] or works on the so-called random feature model, where this approximation goes under the name of *Gaussian equivalence principle* [15]. These ideas imply that the $N$ feature maps $\phi_\alpha(\mathbf{x})$ are somehow mutually weakly correlated, an assumption deemed resonable for a large class of relevant architectures in the thermodinamic limit of large $N$ and $D$ with finite $N/D$ [15].

The integration of Eq. (A3) over the weights $\mathbf{v}^a$, while rather convoluted, now follows a standard replica scheme as detailed in the more technical appendix C. Thanks to Eq. (A5) and making use of the identities

$$\begin{aligned}1 &= \int dQ_{ab}\, \delta\left(Q_{ab} - \langle q^a q^b \rangle_\rho\right) \\ &= \int dQ_{ab} \frac{d\hat{Q}_{ab}}{2\pi}\, e^{-i\hat{Q}_{ab}\left(Q_{ab} - \langle q^a q^b \rangle_\rho\right)}\end{aligned} \tag{A6}$$

one may express Eq. (A3) as an integral over the replica order parameter $Q_{ab}$ and its conjugated variables $\hat{Q}_{ab}$. Working in the replica symmetric ansatz, which holds for our convex problem, $Q_{ab} = Q_0\delta_{ab} + Q(1-\delta_{ab})$ (the same symmetry holding for the conjugated variable), one may compute the leading, linear order contribution in $m$ which determines the limit (A2), to get

$$\langle Z^m \rangle_{\mathcal{T}} \approx \int dQ_0 dQ d\hat{Q}_0 d\hat{Q}\; e^{-\frac{mP}{2} S_\beta(Q_0, Q, \hat{Q}_0, \hat{Q})} \tag{A7}$$

with the rather complicated expression for the action $S_\beta(Q_0, Q, \hat{Q}_0, \hat{Q})$ given by Eq. (C12) of appendix C.

We can finally solve this last integral by saddle-point method in the large $P$ limit. One has to solve a set of four saddle-point equations to find the action minimum that determines the two order parameters of the problem and their conjugated variables (see appendix C). In the $\beta \to \infty$ limit the order parameter has the rather simple saddle-point solution

$$Q^* = Q_0^* = \frac{T - P\mathbf{J}^T\left(2\kappa\lambda\mathbb{1} + P\mathbf{\Phi}\right)\mathbf{G^{-2}J}}{1 - P\mathrm{Tr}[\mathbf{\Phi^2 G^{-2}}]} \tag{A8}$$

where the variable $\kappa$ and the invertible matrix $\mathbf{G}$ have been introduced in the main text.

### Generalisation and training errors

The generalisation $\epsilon_g$ and training $\epsilon_t$ errors can be easily related to the saddle-point replica order parameters.

To see this, first consider the generalisation error. From Eqs. (2) and (1) in the main text

$$\begin{aligned}\epsilon_{\rm g} &= \int d^D x \rho(\mathbf{x}) \left(f_T(\mathbf{x}) - \mathbf{v}^* \cdot \boldsymbol{\phi}(\mathbf{x})\right)^2 \\ &= T - 2\mathbf{J}^T\mathbf{v}^* + {\mathbf{v}^*}^T \boldsymbol{\Phi}\mathbf{v}^* = Q^* \,.\end{aligned} \tag{A9}$$

where in the last equality we used Eq. (A4) and the replica symmetric ansatz. In the limit $P \to \infty$, it is easy to show that $\kappa \to 1$ and $\mathbf{G} \sim P\boldsymbol{\Phi}$ so that the generalisation error converges to the *residual generalisation error* introduced in the main text, Eq. (9)

$$\epsilon_{\rm g} \to \epsilon_{\rm g}^{\rm R} = T - \mathbf{J}^T \boldsymbol{\Phi}^{-1} \mathbf{J} \,. \tag{A10}$$

Notice that this result is not surprising and it provides a first consistency check of our replica mean field theory: one could also obtain it by directly minimising Eq. (A9) with respect to the parameters $\mathbf{v}$. In fact, $\partial_{\mathbf{v}} \epsilon_{\rm g} = -2\mathbf{J}^T + 2\boldsymbol{\Phi}\mathbf{v}$ implies $\mathbf{v}^* = \mathbf{J}^T\boldsymbol{\Phi}^{-1}$, so that $\epsilon_{\rm g}(\mathbf{v}^*) \equiv \epsilon_{\rm g}^{\rm R} = T - \mathbf{J}^T\boldsymbol{\Phi}^{-1}\mathbf{J}$. Furthermore, by isolating the residual generalisation error in Eq. (A8), we finally find the compact formula for the generalisation error quoted in Eq. (7) of the main text.

Using the theory developed so far, we can also have access to the average value of the training error. We notice that the average training error is by definition the average loss function defined in Eq. (5), evaluated in $\lambda = 0$ (up to a factor $P$). This means that one can extract its value by evaluating the action on the saddle point solution and performing the limit $\beta \to \infty$, i.e.:

$$\epsilon_{\rm t} = \lim_{\beta\to\infty} \frac{1}{\beta} \, S_\beta(Q^*, Q_0^*, \hat{Q}^*, \hat{Q}_0^*)\Big|_{\lambda=0} \,. \tag{A11}$$

After some straightforward but lengthy algebraic manipulations, one recovers the training error given in Eq. (7) of the main text.

## Appendix B: Numerical experiments details

In this section we give a detailed report of all our numerical procedures. The code to replicate our experiments can be found at *https://github.com/rosalba-p/Generalisation_DNN*.

### Teacher-student architectures

*Student architectures.* We considered six types of student architectures: two toy networks with one- and two-hidden layers (size of the second hidden layer $N_{\rm hid} = 200$) and three state-of-the-art convolutional ones (ResNet18, DenseNet121 and VGG11). The toy architectures have fully connected layers and ReLu activation functions at every layer but the last; the convolutional networks are the standard PyTorch [44] models modified to yield a scalar output suitable for regression through a last fully connected linear layer with parameters $\mathbf{v}$, instead of the $LogSoftMax$ that is employed for classification. All these architectures have several convolutional layers before a last fully connected one that counts respectively $N_{\rm out} = 512, 1024, 4096$ hidden units. The total number of trainable parameters in the first two networks (weights and biases) is approximately 10 million, while vgg11 counts 10 times as many.

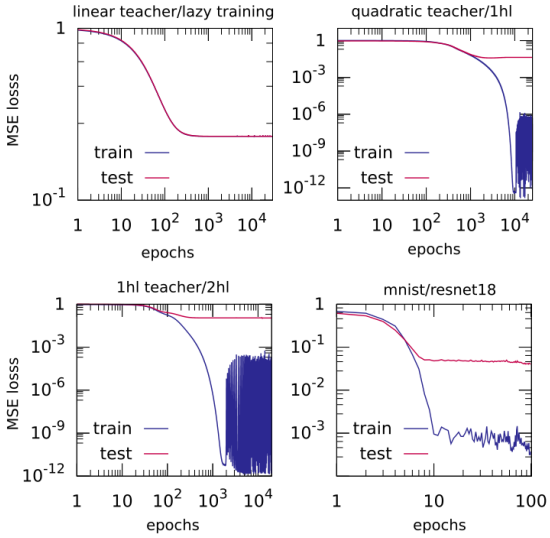


FIG. 4. Train (blue) and test (magenta) loss of different teacher/student tasks as a function of the training epochs. The test loss reaches a plateau even if the training loss is noisy, due to the different order of magnitude reached by the two. It is worth remarking that the losses are normalised with the trivial predictor and therefore at epoch 1 are $O(1)$.

*Teacher architectures and inputs for toy DNNs.* Each linear or quadratic teachers (see Results) is defined by a random uniform vector $\mathbf{t} \in \mathbb{R}^D$ of unitary norm. 1HL teachers are defined by parameters $q_\alpha \in \mathbb{R}$and $\mathbf{S}_\alpha \in \mathbb{R}^D$ ($\alpha = 1, \ldots, M = 200$). They are drawn from a normal distribution with zero mean and variance (respectively) $1/M$ and $1/D$. Inputs $\mathbf{x} \in \mathbb{R}^D$ are also drawn from normal distribution with zero mean and unit variance.

### Learning and generalization

*Learning algorithm.* All the architectures are trained with Adam optimiser [45] and a small weight decay ($w_d = 10^{-5}$), while the learning rate $\alpha$ is set to $10^{-3}$ for the toy architectures and $10^{-4}$ for the convolutional ones. The weight decay $w_d$ is related to the regularisation parameter $\lambda$ via $w_d = \lambda\alpha$, and we have verified that our results do not change quantitatively by varying $\lambda$ in a reasonable range ($\lambda \leq 10^{-1}$). The regression loss employed is the

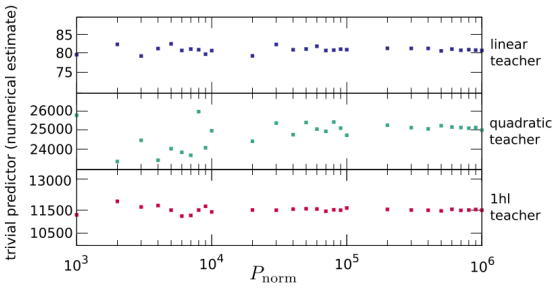


FIG. 5. Numerical estimate of the trivial predictor $T$ as a function of the number of points $P_{\rm norm}$ used in the approximation. This values describe a single realisation of a random teacher function (linear, quadratic, 1hl). $T$ converges to a fixed value around $P_{\rm norm} = 10^6$, that is the value chosen for all our numerical experiments.

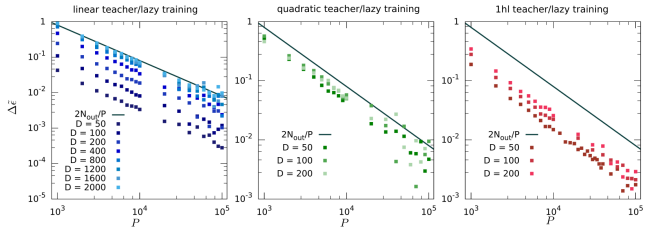


FIG. 6. Generalisation gap of a lazy training architecture as a function of the number of examples in the training set for different input sizes $D$. From left to right the teacher functions are respectively linear, quadratic and 1hl. The size of the hidden layer is kept fixed to $N_{\rm out} = 400$.

standard mean squared error,

$$\text{MSE} = \frac{1}{P}\sum_{\mu=1}^{P}\left(f_{\rm T}(x^\mu) - f_{\rm S}(x^\mu)\right)^2 \tag{B1}$$

For the MNIST dataset, we consider the labels as integers: $f_{\rm S}(x^\mu) = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]$, and the MSE loss is computed as in the other cases.

*Estimation of the generalisation gap.* We first bring the training procedure to convergence, i.e. we ensure that the train and test loss have reached a plateau. For synthetic datasets, this requires a large number of training epochs ($3 \cdot 10^4$), while MNIST is learnt faster (100 epochs). In one epoch the network is fed all the dataset, arranged in minibatches only when full batch learning is prohibited by memory limitations. Once the train loss is steady with respect to the test loss, we retrieve the generalisation gap as the average over the last 100 epochs (50 for MNIST). The size of the train set is $P_{\rm test} = 10^4$ is all cases.
Several plots of train and test loss vs the number of epochs are shown in fig 4: for different teacher-student pairs the plateau in the test loss is always reached: even if some noise is still visible in the train loss, its oscillations are too small to affect the test loss and therefore the generalisation gap.

### Trivial predictor.

To compare results obtained from different teacher/student pairs, we need to normalise the loss by its natural scale, i.e. the trivial predictor $T$ defined in Eq. (6). By doing this, we make sure that the train and test loss are always of $O(1)$ for a random architecture (i.e. at epoch 0). $T$ is a property of the dataset, and its computation changes accordingly.

*MNIST.* The MNIST dataset is intrinsically suitable for classification problems, since it has 10 classes with discrete labels, that are simply the first 10 integers. For this reason, the integral that describes the trivial predictor here becomes a summation over the classes. Another simplification can be made considering that MNIST is balanced over the training labels: the $P = 6 \cdot 10^4$ training examples are equally distributed over the 10 classes. In this case the computation of the trivial predictor is simple and can be performed analytically:

$$T = \frac{1}{P}\sum_{n=0}^{9}\frac{P}{\#\text{ classes}}n^2 = \frac{1}{\#\text{ classes}}\sum_{n=0}^{9}n^2 = 28.5 \tag{B2}$$

*Synthetic datasets.* For the other teachers that we employed $T$ cannot be computed analytically. We therefore perform a numerical estimation in the following approximation:

$$T \sim \frac{1}{P_{\rm norm}}\sum_{\mu=1}^{P_{\rm norm}} f_{\rm T}^2(x^\mu) \tag{B3}$$

Operatively, we draw an independent data sample of $P_{\rm norm}$ elements $\{x^\mu\}_{\mu=1\dots P_{\rm norm}}$, and average the respective squared true labels $f_{\rm T}^2(x^\mu)$. $P_{\rm norm} = 10^6$ was chosen as a safe compromise between computational time and consistency of the estimate. The convergence of $T$ to a fixed value when $P_{\rm norm}$ grows is shown in Figure 5.

### Higher dimensional synthetic inputs.

In the main text we show the generalisation gap scaling for fixed input size $D = 50$, with the exception of the inset in Fig 1. Here we report the full analysis of higher dimensional input size and clarify the procedure used to obtain the inset. In Figure 6 we plot the generalisation gap of of a lazy training architecture learning the three function considered in the main text (linear, quadratic and 1hl), as a function of the number of training examples and for different sizes of the input $D$. Increasing $D$ does not change the asympthotic behaviour of the curves, but has the effect of pushing them closer to the bound given the same trainset size $P$. To assert how fast the curves approach the bound, we have performed a linear fit of

the generalisation gap as a function of $P$:

$$\log \Delta\tilde{\epsilon} = m \cdot \log P + y_0 \tag{B4}$$

Note that for the bound $m \equiv -1$ and $y_0 \equiv \log 2N_{\text{out}}$. For different values of $D$, it is verified that $m \sim -1$, while $y_0 - 2N_{\text{out}}$ approaches 0 exponentially fast (see also inset in Fig. 1 of the main text).

### Appendix C: Replica symmetric ansatz and saddle point equations

Thanks to the Gaussian approximation (A5) and making use of the identities (A6), the replicated partition function (A3) may be recasted in the following form

$$\langle Z^m \rangle_{\mathcal{T}} = \int dQ_{ab} d\hat{Q}_{ab}\; K(\beta, Q)^{-\frac{P}{2}} e^{iQ_{ab}\hat{Q}_{ab}} \times \\ \times \int d^N \mathbf{v}^a e^{-\frac{\beta\lambda}{2}\sum_a \|\mathbf{v}^a\|^2 - i\sum_{a\le b}\hat{Q}_{ab}\left(T + (\mathbf{v}^a)^T \mathbf{\Phi}\, \mathbf{v}^b - \mathbf{J}^T(\mathbf{v}^a + \mathbf{v}^b)\right)} \tag{C1}$$

where

$$K(\beta, \mathbf{Q}) \equiv \det(\mathbb{I} + \beta \mathbf{Q})\,. \tag{C2}$$

The idea is now to explicitly perform the integration over the replicated weights $\mathbf{v}^a$ and to evaluate the integrals over $Q_{ab}$ and $\hat{Q}_{ab}$ with the saddle-point method. As it occurs in standard spin glass models, we are left with the complication of performing the tricky limit $m \to 0$, which may lead to the breaking of the so-called replica symmetry of the matrix $Q_{ab}$. However in this specific case the underlying optimisation problem that we are dealing with is convex and this guarantees that the simplest replica symmetric ansatz for the matrix $Q_{ab}$ will provide the exact solution to the saddle-point equations in the limit $m \to 0$ [21].

The assumption of replica symmetry amounts to require that the matrices $Q_{ab}/\hat{Q}_{ab}$ take the following form:

$$Q_{ab} = \begin{cases} Q_0 & a = b \\ Q & a \neq b \end{cases} \qquad \hat{Q}_{ab} = \begin{cases} \hat{Q}_0 & a = b \\ \hat{Q} & a \neq b\,. \end{cases} \tag{C3}$$

With a slight abuse of notation, we can specify our analysis to the replica symmetric ansatz already at the level of Eq. (C1), thus obtaining:

$$\langle Z^m \rangle_{\mathcal{T}} = \int dQ_0\; dQ\; d\hat{Q}_0\; d\hat{Q}\; K^{-\frac{P}{2}} e^{im\left(\hat{Q}_0(Q_0 - T) + \frac{m-1}{2}\hat{Q}(Q-T)\right)} \times \\ \times \int d^N \mathbf{v}^a e^{-\frac{\beta\lambda}{2}\sum_{\alpha,a}(v^a_\alpha)^2 - i\left(\hat{Q}_0 - \frac{\hat{Q}}{2}\right)\sum_a (\mathbf{v}^a)^T \mathbf{\Phi} \mathbf{v}^a} e^{-i\frac{\hat{Q}}{2}\sum_{ab}(\mathbf{v}^a)^T \mathbf{\Phi}\mathbf{v}^b + 2i\left(\hat{Q}_0 - (1-m)\frac{\hat{Q}}{2}\right)\sum_a \mathbf{J}^T\mathbf{v}^a}\,. \tag{C4}$$

By performing the Gaussian integrals over the replicated weights and using standard mathematical manipulations, we finally get the following result:

$$\langle Z^m \rangle_{\mathcal{T}} = \int dQ_0\; dQ\; d\hat{Q}_0\; d\hat{Q}\; K^{-\frac{P}{2}} e^{-\frac{m}{2}\left(\hat{Q}_0(Q_0 - T) + (m-1)\hat{Q}(Q-T)\right)} \times \\ \times (2\pi)^{\frac{m}{2}} e^{-\frac{m-1}{2}\mathrm{Tr}\left[\log\left(\beta\lambda\mathbb{1} - (\hat{Q}_0 - \hat{Q})\mathbf{\Phi}\right)\right] - \frac{1}{2}\mathrm{Tr}\left[\log\left(\beta\lambda\mathbb{1} - (\hat{Q}_0 - (1-m)\hat{Q})\mathbf{\Phi}\right)\right]} e^{\frac{m}{2}\left(\hat{Q}_0 - (1-m)\hat{Q}\right)^2 \mathbf{J}^T\left(\beta\lambda\mathbb{1} - (\hat{Q}_0 - (1-m)\hat{Q})\mathbf{\Phi}\right)^{-1}\mathbf{J}} \tag{C5}$$

At this point we have managed to integrate both on the dataset $\mathcal{T}$ and on the replicated weights. To find the solution of the last integrals over the order parameters, we exploit the saddle-point method, which will deliver the exact solution in the limit $P \to \infty$.

#### Replica symmetric ansatz

First we notice that assuming replica symmetry, the contribution $K$ to the partition function simplifies in the following way:

$$K(\beta, Q, Q_0) = \det[\mathbb{I} + \beta\mathbf{Q}] \\ = (1 + \beta(Q_0 - Q))^{m-1}(1 + \beta(Q_0 - (1-m)Q))\,. \tag{C6}$$

In order to extract the $m \to 0$ limit, we have to keep only those terms that are linear in $m$. This is easily done term by term:

$$e^{-\frac{P}{2}\log\left((1+\beta(Q_0-Q))^{m-1}(1+\beta(Q_0-(1-m)Q))\right)} \\ \approx e^{-\frac{P}{2}\, m\left(\frac{\beta Q}{1+\beta(Q_0-Q)} + \log(1+\beta(Q_0-Q))\right)}\,, \tag{C7}$$

$$e^{-\frac{m}{2}\left(\hat{Q}_0(Q_0-T)+(m-1)\hat{Q}(Q-T)\right)} \approx e^{-\frac{m}{2}\left(\hat{Q}_0(Q_0-T)-\hat{Q}(Q-T)\right)}, \tag{C8}$$

$$\begin{aligned} &e^{-\frac{m-1}{2}\mathrm{Tr}\left[\log\left(\beta\lambda\mathbb{1}-(\hat{Q}_0-\hat{Q})\mathbf{\Phi}\right)\right]-\frac{1}{2}\mathrm{Tr}\left[\log\left(\beta\lambda\mathbb{1}-(\hat{Q}_0-(1-m)\hat{Q})\mathbf{\Phi}\right)\right]} \\ &\approx e^{-\frac{m}{2}\mathrm{Tr}\left[\log\left(\beta\lambda\mathbb{1}-(\hat{Q}_0-\hat{Q})\mathbf{\Phi}\right)\right]+\frac{m\hat{Q}}{2}\mathrm{Tr}\left[\mathbf{\Phi}\left(\beta\lambda\mathbb{1}-(\hat{Q}_0-\hat{Q})\mathbf{\Phi}\right)^{-1}\right]}, \end{aligned} \tag{C9}$$

and

$$\begin{aligned} &e^{\frac{m}{2}\left(\hat{Q}_0-(1-m)\hat{Q}\right)^2\mathbf{J}^T\left(\beta\lambda\mathbb{1}-(\hat{Q}_0-(1-m)\hat{Q})\mathbf{\Phi}\right)^{-1}\mathbf{J}} \\ &\approx e^{\frac{m}{2}\left(\hat{Q}_0-\hat{Q}\right)^2\mathbf{J}^T\left(\beta\lambda\mathbb{1}-(\hat{Q}_0-\hat{Q})\mathbf{\Phi}\right)^{-1}\mathbf{J}}. \end{aligned} \tag{C10}$$

As such, the leading contribution to the average replicated partition function in the $m \to 0$ limit reads:

$$\begin{aligned} \langle Z^m\rangle_{\mathcal{T}} \sim \int dQ_0\, dQ\, d\hat{Q}_0\, d\hat{Q}\, &e^{-\frac{P}{2}\, m\left(\frac{\beta Q}{1+\beta(Q_0-Q)}+\log(1+\beta(Q_0-Q))\right)} e^{-\frac{m}{2}\left(\hat{Q}_0(Q_0-T)-\hat{Q}(Q-T)\right)} \\ &\times e^{-\frac{m}{2}\mathrm{Tr}\left[\log\left(\beta\lambda\mathbb{1}-(\hat{Q}_0-\hat{Q})\mathbf{\Phi}\right)\right]+\frac{m\hat{Q}}{2}\mathrm{Tr}\left[\mathbf{\Phi}\left(\beta\lambda\mathbb{1}-(\hat{Q}_0-\hat{Q})\mathbf{\Phi}\right)^{-1}\right]} e^{\frac{m}{2}\left(\hat{Q}_0-\hat{Q}\right)^2\mathbf{J}^T\left(\beta\lambda\mathbb{1}-(\hat{Q}_0-\hat{Q})\mathbf{\Phi}\right)^{-1}\mathbf{J}} \end{aligned} \tag{C11}$$

and by rescaling the parameters $\hat{Q} \to P\hat{Q}$ and $\hat{Q}_0 \to P\hat{Q}_0$, we recast the partition function in the form $\langle Z^m\rangle_{\mathcal{T}} = \int e^{-\frac{mP}{2}S_\beta}$, where the action $S_\beta$ is defined as:

$$\begin{aligned} S_\beta = &\frac{\beta Q}{1+\beta(Q_0-Q)} + \log(1+\beta(Q_0-Q)) + \hat{Q}_0(Q_0-T) + \\ &- \hat{Q}(Q-T) + \frac{1}{P}\mathrm{Tr}\left[\log\left(\beta\lambda\mathbb{1}-P(\hat{Q}_0-\hat{Q})\mathbf{\Phi}\right)\right] + \\ &- \hat{Q}\mathrm{Tr}\left[\mathbf{\Phi}\left(\beta\lambda\mathbb{1}-P(\hat{Q}_0-\hat{Q})\mathbf{\Phi}\right)^{-1}\right] + \\ &- P\left(\hat{Q}_0-\hat{Q}\right)^2\mathbf{J}^T\left(\beta\lambda\mathbb{1}-P(\hat{Q}_0-\hat{Q})\mathbf{\Phi}\right)^{-1}\mathbf{J} \end{aligned} \tag{C12}$$

### Saddle point equations

We can now move to the derivation of the saddle-point equations. Firstly we notice that direct differentiation with respect to $Q$ and $Q_0$ allows to find the explicit expressions for $\hat{Q}$ and $\hat{Q}_0$:

$$0 = \frac{\partial S_\beta}{\partial Q} = -\hat{Q} + \frac{\beta^2 Q}{(1+\beta(Q_0-Q))^2} \tag{C13}$$

which implies

$$\hat{Q} = \frac{\beta^2 Q}{(1+\beta(Q_0-Q))^2} \tag{C14}$$

and

$$0 = \frac{\partial S_\beta}{\partial Q_0} = \hat{Q}_0 + \frac{-\beta Q(\beta)}{(1+\beta(Q_0-Q))^2} + \frac{\beta}{1+\beta(Q_0-Q)} \tag{C15}$$

which gives

$$\begin{aligned} \hat{Q}_0 &= \frac{\beta^2 Q}{(1+\beta(Q_0-Q))^2} - \frac{\beta}{1+\beta(Q_0-Q)} = \\ &= \hat{Q} - \frac{\beta}{1+\beta(Q_0-Q)} \end{aligned} \tag{C16}$$

Let us look at the derivative of the action w.r.t. $\hat{Q}$:

$$\begin{aligned} 0 = &\frac{\partial S_\beta}{\partial \hat{Q}} = -(Q-T) + \frac{1}{P}\partial_{\hat{Q}}\mathrm{Tr}\left[\log\left(\tilde{\mathbf{G}}\right)\right] - \mathrm{Tr}\left[\mathbf{\Phi}\tilde{\mathbf{G}}^{-1}\right] + \\ &- \hat{Q}\partial_{\hat{Q}}\mathrm{Tr}\left[\mathbf{\Phi}\tilde{\mathbf{G}}^{-1}\right] + P\mathbf{J}^T\frac{P\mathbf{\Phi}(\hat{Q}_0-\hat{Q})^2+2\tilde{\mathbf{G}}(\hat{Q}_0-\hat{Q})}{\tilde{\mathbf{G}}^2}\mathbf{J} \end{aligned} \tag{C17}$$

that gives

$$\begin{aligned} Q = &T + \frac{1}{P}\partial_{\hat{Q}}\mathrm{Tr}\left[\log\left(\tilde{\mathbf{G}}\right)\right] - \mathrm{Tr}\left[\mathbf{\Phi}\tilde{\mathbf{G}}^{-1}\right] + \\ &- \hat{Q}\partial_{\hat{Q}}\mathrm{Tr}\left[\mathbf{\Phi}\tilde{\mathbf{G}}^{-1}\right] + P\mathbf{J}^T\frac{P\mathbf{\Phi}(\hat{Q}_0-\hat{Q})^2+2\tilde{\mathbf{G}}(\hat{Q}_0-\hat{Q})}{\tilde{\mathbf{G}}^2}\mathbf{J} \\ = &T + P\,\hat{Q}\,\mathrm{Tr}[(\mathbf{\Phi}\tilde{\mathbf{G}}^{-1})^2] + P\mathbf{J}^T\frac{P\mathbf{\Phi}(\hat{Q}_0-\hat{Q})^2+2\tilde{\mathbf{G}}(\hat{Q}_0-\hat{Q})}{\tilde{\mathbf{G}}^2}\mathbf{J} \end{aligned} \tag{C18}$$

where for convenience we have defined the $N \times N$ matrix $\tilde{\mathbf{G}} = \beta\lambda\mathbb{1} - P(\hat{Q}_0 - \hat{Q})\mathbf{\Phi}$. Finally we obtain the saddle point equation for $\hat{Q}_0$:

$$\begin{aligned} 0 = &\frac{\partial S_\beta}{\partial \hat{Q}_0} = (Q_0-T) + \frac{1}{P}\partial_{\hat{Q}_0}\mathrm{Tr}\left[\log\left(\tilde{\mathbf{G}}\right)\right] + \\ &- \hat{Q}\partial_{\hat{Q}_0}\mathrm{Tr}\left[\mathbf{\Phi}\tilde{\mathbf{G}}^{-1}\right] - P\mathbf{J}^T\frac{P\mathbf{\Phi}(\hat{Q}_0-\hat{Q})^2+2\tilde{\mathbf{G}}(\hat{Q}_0-\hat{Q})}{\tilde{\mathbf{G}}^2}\mathbf{J} \end{aligned} \tag{C19}$$

giving us

$$\begin{aligned} Q_0 &= T + \mathrm{Tr}\left[\mathbf{\Phi}\tilde{\mathbf{G}}^{-1}\right] + P\,\hat{Q}\,\mathrm{Tr}[\mathbf{\Phi}\tilde{\mathbf{G}}^{-1}\mathbf{\Phi}\tilde{\mathbf{G}}^{-1}] \\ &\quad + P\mathbf{J}^T\frac{P\mathbf{\Phi}(\hat{Q}_0-\hat{Q})^2+2\tilde{\mathbf{G}}(\hat{Q}_0-\hat{Q})}{\tilde{\mathbf{G}}^2}\mathbf{J} \\ &= Q + \mathrm{Tr}\left[\mathbf{\Phi}\tilde{\mathbf{G}}^{-1}\right] \end{aligned} \tag{C20}$$

Let us now consider the special combination $\kappa = 1 + \beta(Q_0 - Q)$. By using Eq. (C20), we obtain:

$$\begin{aligned} \kappa = &1+\beta(Q_0-Q) = 1+\beta\left(Q+\mathrm{Tr}\left[\mathbf{\Phi}\tilde{\mathbf{G}}^{-1}\right]-Q\right) \\ &= 1+\beta\,\mathrm{Tr}\left[\mathbf{\Phi}\tilde{\mathbf{G}}^{-1}\right], \end{aligned} \tag{C21}$$

whereas by considering the difference between Eq. (C14) and Eq. (C16), we easily show that the difference $\hat{Q}_0 - \hat{Q}$ depends on $\kappa$ only as:

$$\hat{Q}_0 - \hat{Q} = \hat{Q} - \frac{\beta}{1+\beta(Q_0-Q)} - \hat{Q} = -\frac{\beta}{1+\beta(Q_0-Q)} = -\frac{\beta}{\kappa} \tag{C22}$$

By inserting this result into the definition of $\tilde{\mathbf{G}}$ we can define the rescaled matrix $\mathbf{G} = \kappa\tilde{\mathbf{G}}/\beta$:

$$\mathbf{G} = \frac{\kappa}{\beta}\left[\beta\lambda\mathbb{1} - P(\hat{Q}_0 - \hat{Q})\mathbf{\Phi}\right] = \kappa\lambda\mathbb{1} + P\mathbf{\Phi}\,. \tag{C23}$$

These observations allow us to show that the new variable $\kappa$ satisfies the following self-consistency equation:

$$\begin{aligned}\kappa =& 1 + \beta\ \mathrm{Tr}\left[\mathbf{\Phi}\left(\frac{\beta}{\kappa}\mathbf{G}\right)^{-1}\right] = 1 + \kappa\mathrm{Tr}\left[\mathbf{\Phi}\mathbf{G}^{-1}\right] \\ =& 1 + \kappa\ \mathrm{Tr}\left[\frac{\mathbf{\Phi}}{\kappa\lambda\mathbb{1} + P\mathbf{\Phi}}\right]\end{aligned} \tag{C24}$$

and allow to recast the solution of the saddle-point equations in the following very convenient form (notice that from now on the solutions of the saddle-point equations will be indicated with an asterisk):

$$\begin{aligned}\hat{Q}_0^* &= \hat{Q}^* - \frac{\beta}{\kappa} \\ \hat{Q}^* &= \frac{\beta^2 Q^*}{\kappa^2} \\ Q_0^* &= Q^* + \frac{\kappa - 1}{\beta} \\ Q^* &= \frac{T - P\mathbf{J}^T\frac{2\kappa\lambda + P\mathbf{\Phi}}{\mathbf{G}^2}\mathbf{J}}{1 - P\mathrm{Tr}[\mathbf{\Phi}^2\mathbf{G}^{-2}]}\,.\end{aligned} \tag{C25}$$

It is worth noticing that since $\mathbf{G}$ and $\kappa$ are independent on the inverse temperature $\beta$, the solution for the order parameter $Q^*$ is also independent on the temperature. Moreover, we easily get that $Q^* = Q_0^*$ in the limit $\beta \to \infty$.


[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[6] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 1999.

[7] V. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999.

[8] O. Bousquet, S. Boucheron, and G. Lugosi, *Introduction to Statistical Learning Theory*, pp. 169–207. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.

[9] L. Zdeborová, "Understanding deep learning is also a job for physicists," *Nature Physics*, vol. 16, pp. 602–604, 2020.

[10] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning (still) requires rethinking generalization," *Commun. ACM*, vol. 64, pp. 107–115, feb 2021.

[11] Y. Bahri, J. Kadmon, J. Pennington, S. S. Schoenholz, J. Sohl-Dickstein, and S. Ganguli, "Statistical mechanics of deep learning," *Annual Review of Condensed Matter Physics*, vol. 11, no. 1, pp. 501–528, 2020.

[12] S. Mei, A. Montanari, and P.-M. Nguyen, "A mean field view of the landscape of two-layer neural networks," *Proceedings of the National Academy of Sciences*, vol. 115, no. 33, pp. E7665–E7671, 2018.

[13] F. Santambrogio, "Euclidean, metric, and wasserstein gradient flows: an overview," *Bulletin of Mathematical Sciences*, vol. 7, no. 1, pp. 87–154, 2017.

[14] S. D'Ascoli, M. Refinetti, G. Biroli, and F. Krzakala, "Double trouble in double descent: Bias and variance(s) in the lazy regime," in *Proceedings of the 37th International Conference on Machine Learning* (H. D. III and A. Singh, eds.), vol. 119 of *Proceedings of Machine Learning Research*, pp. 2280–2290, PMLR, 13–18 Jul 2020.

[15] S. Goldt, M. Mézard, F. Krzakala, and L. Zdeborová, "Modeling the influence of data structure on learning in neural networks: The hidden manifold model," *Phys. Rev. X*, vol. 10, p. 041044, Dec 2020.

[16] S. Mei and A. Montanari, "The generalization error of random features regression: Precise asymptotics and the double descent curve," *Communications on Pure and Applied Mathematics*, 2019.

[17] F. Gerace, B. Loureiro, F. Krzakala, M. Mezard, and L. Zdeborova, "Generalisation error in learning with random features and the hidden manifold model," in *Proceedings of the 37th International Conference on Machine Learning* (H. D. III and A. Singh, eds.), vol. 119 of *Proceedings of Machine Learning Research*, pp. 3452–3462, PMLR, 13–18 Jul 2020.

[18] C. Baldassi, C. Lauditi, E. M. Malatesta, R. Pacelli, G. Perugini, and R. Zecchina, "Learning through atypical "phase transitions" in overparameterized neural networks," *arXiv preprint arXiv:2110.00683*, 2021.

[19] C. Baldassi, C. Lauditi, E. M. Malatesta, G. Perugini, and R. Zecchina, "Unveiling the structure of wide flat minima in neural networks," *Phys. Rev. Lett.*, vol. 127, p. 278301, Dec 2021.

[20] B. Loureiro, C. Gerbelot, H. Cui, S. Goldt, F. Krzakala, M. Mezard, and L. Zdeborová, "Learning curves of generic features maps for realistic datasets with a teacher-student model," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[21] R. Dietrich, M. Opper, and H. Sompolinsky, "Statistical mechanics of support vector networks," *Phys. Rev. Lett.*, vol. 82, pp. 2975–2978, Apr 1999.

[22] A. Canatar, B. Bordelon, and C. Pehlevan, "Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks," *Nature communications*, vol. 12, no. 1, pp. 1–12, 2021.

[23] A. Jacot, F. Gabriel, and C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, (Red Hook, NY, USA), p. 8580–8589, Curran Associates Inc., 2018.

[24] J. Lee, L. Xiao, S. Schoenholz, Y. Bahri, R. Novak, J. Sohl-Dickstein, and J. Pennington, "Wide neural networks of any depth evolve as linear models under gradient descent," in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.

[25] L. Bottou, *Making Vapnik–Chervonenkis Bounds Accurate*, pp. 143–155. 09 2015.

[26] M. Belkin, "Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation," *Acta Numerica*, vol. 30, p. 203?248, 2021.

[27] P. Rotondo, M. C. Lagomarsino, and M. Gherardi, "Counting the learnable functions of geometrically structured data," *Phys. Rev. Research*, vol. 2, p. 023169, May 2020.

[28] P. Rotondo, M. Pastore, and M. Gherardi, "Beyond the storage capacity: Data-driven satisfiability transition," *Phys. Rev. Lett.*, vol. 125, p. 120601, Sep 2020.

[29] M. Pastore, P. Rotondo, V. Erba, and M. Gherardi, "Statistical learning theory of structured data," *Phys. Rev. E*, vol. 102, p. 032119, Sep 2020.

[30] M. Gherardi, "Solvable model for the linear separability of structured data," *Entropy*, vol. 23, no. 3, 2021.

[31] M. Pastore, "Critical properties of the SAT/UNSAT transitions in the classification problem of structured data," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2021, p. 113301, nov 2021.

[32] M. Mézard, G. Parisi, and M. A. Virasoro, *Spin glass theory and beyond: An Introduction to the Replica Method and Its Applications*, vol. 9. World Scientific Publishing Company, 1987.

[33] E. Dobriban and S. Wager, "High-dimensional asymptotics of prediction: ridge regression and classification," *The Annals of Statistics*, vol. 46, no. 1, pp. 247–279, 2018.

[34] G. Mel and S. Ganguli, "A theory of high dimensional regression with arbitrary correlations between input features and target functions: sample complexity, multiple descent curves and a hierarchy of phase transitions," in *Proceedings of the 38th International Conference on Machine Learning* (M. Meila and T. Zhang, eds.), vol. 139 of *Proceedings of Machine Learning Research*, pp. 7578–7587, PMLR, 18–24 Jul 2021.

[35] A. C. C. Coolen, M. Sheikh, A. Mozeika, F. Aguirre-Lopez, and F. Antenucci, "Replica analysis of overfitting in generalized linear regression models," *Journal of Physics A: Mathematical and Theoretical*, vol. 53, p. 365001, aug 2020.

[36] B. Ghorbani, S. Mei, T. Misiakiewicz, and A. Montanari, "Linearized two-layers neural networks in high dimension," *The Annals of Statistics*, vol. 49, no. 2, pp. 1029 – 1054, 2021.

[37] Y. LeCun, "The mnist database of handwritten digits.," 1998.

[38] P. L. Bartlett, N. Harvey, C. Liaw, and A. Mehrabian, "Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 2285–2301, 2019.

[39] B. Bordelon, A. Canatar, and C. Pehlevan, "Spectrum dependent learning curves in kernel regression and wide neural networks," in *Proceedings of the 37th International Conference on Machine Learning* (H. D. III and A. Singh, eds.), vol. 119 of *Proceedings of Machine Learning Research*, pp. 1024–1034, PMLR, 13–18 Jul 2020.

[40] A. Jacot, B. Simsek, F. Spadaro, C. Hongler, and F. Gabriel, "Kernel alignment risk estimator: Risk prediction from training data," in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, eds.), vol. 33, pp. 15568–15578, Curran Associates, Inc., 2020.

[41] S. Goldt, B. Loureiro, G. Reeves, F. Krzakala, M. Mézard, and L. Zdeborová, "The gaussian equivalence of generative models for learning with shallow neural networks," *arXiv preprint arXiv:2006.14709*, 2020.

[42] H. Hu and Y. M. Lu, "Universality laws for high-dimensional learning with random features," *arXiv preprint arXiv:2009.07669*, 2020.

[43] E. Gardner, "The space of interactions in neural network models," *Journal of Physics A: Mathematical and General*, vol. 21, pp. 257–270, jan 1988.

[44] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, pp. 8024–8035, Curran Associates, Inc., 2019.

[45] D. Kingma and L. Ba, "A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.