POLITECNICO DI TORINO
Repository ISTITUZIONALE

Learning through atypical phase transitions in overparameterized neural networks

Original
Learning through atypical phase transitions in overparameterized neural networks / Baldassi, C., Lauditi, C., Malatesta, E.M., Pacelli, R., Perugini, G., Zecchina, R.. - In: PHYSICAL REVIEW. E. - ISSN 2470-0053. - 106:1(2022), p. 014116. [10.1103/PhysRevE.106.014116]

Availability:
This version is available at: 11583/2983563 since: 2023-11-03T07:31:03Z

Publisher:
AMER PHYSICAL SOC

Published
DOI:10.1103/PhysRevE.106.014116

Terms of use:

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

Publisher copyright

(Article begins on next page)

26 November 2026

Learning through atypical "phase transitions" in overparameterized neural networks

Carlo Baldassi,[1] Clarissa Lauditi,[2] Enrico M. Malatesta,[1] Rosalba Pacelli,[2] Gabriele Perugini,[1] and Riccardo Zecchina[1]

[1]*Artificial Intelligence Lab, Bocconi University, 20136 Milano, Italy*
[2]*Department of Applied Science and Technology, Politecnico di Torino, 10129 Torino, Italy*

(Dated: June 14, 2022)

Current deep neural networks are highly overparameterized (up to billions of connection weights) and nonlinear. Yet they can fit data almost perfectly through variants of gradient descent algorithms and achieve unexpected levels of prediction accuracy without overfitting. These are formidable results that defy predictions of statistical learning and pose conceptual challenges for non-convex optimization. In this paper, we use methods from statistical physics of disordered systems to analytically study the computational fallout of overparameterization in non-convex binary neural network models, trained on data generated from a structurally simpler but "hidden" network. As the number of connection weights increases, we follow the changes of the geometrical structure of different minima of the error loss function and relate them to learning and generalization performance. A first transition happens at the so-called interpolation point, when solutions begin to exist (perfect fitting becomes possible). This transition reflects the properties of typical solutions, which however are in sharp minima and hard to sample. After a gap, a second transition occurs, with the discontinuous appearance of a different kind of "atypical" structures: wide regions of the weight space that are particularly solution-dense and have good generalization properties. The two kinds of solutions coexist, with the typical ones being exponentially more numerous, but empirically we find that efficient algorithms sample the atypical, rare ones. This suggests that the atypical phase transition is the relevant one for learning. The results of numerical tests with realistic networks on observables suggested by the theory are consistent with this scenario.

arXiv:2110.00683v2 [cs.LG] 11 Jun 2022

Machine learning has recently advanced in a totally unexpected way thanks to deep learning (DL), reaching unprecedented performance in many fields of data-driven research and applications. Impressive spin-offs are emerging not only in technological applications but also in a wide variety of basic scientific fields, from molecular biology and language processing, to the solution of partial differential equations for the study of materials and fluids, to name a few recent examples. At the same time theoretical research is trying to build a unifying framework that explains deep learning performance, enables its development based on first principles, and paves the way towards interdisciplinary methodological and modeling connections, including computational neuroscience.

Among the most disruptive aspects of deep learning models are their highly overparameterized and non-convex nature. Both of these aspects are a common trait of all the DL models and have led to unexpected results for classical statistical learning theory and non-convex optimization. Current deep neural networks (DNN) are composed of millions (or even billions) of connection weights and the learning process seeks to minimize the number of classification errors made by the DNN over a training set. This optimization problem is highly non-convex, and learning algorithms need to efficiently find good minima in a space of extremely high dimensionality without being trapped in local minima or saddle points for long times. Good minima are those that have good generalization capabilities, namely that do not suffer from overfitting given the inherent noisiness in the data and the huge number of parameters that can be adjusted. Surprisingly, this goal can often be achieved by relatively simple algorithms based on variants of the gradient descent method.

We are thus facing of two conceptually stimulating facts: (i) highly expressive neural network models can fit the training data via simple variants of algorithms originally designed for convex optimization; (ii) even if trained with little control over their statistical complexity, these models achieve high levels of prediction accuracy, contrary to what classical statistical intuitions (such as the bias-variance tradeoff) would suggest.

In this paper, we focus on the computational fallout of overparameterization in non-convex models. As the number of parameters increases, we study the changes in the geometric structure of the different minima of the error loss function and we relate this to learning performance.

Intuitively, one might be tempted to think that non-convex neural network models become effectively convex (with most of the weight volume of the minima associated to wide, accessible ones) when the number of weights becomes sufficiently large relative to the number of data to be classified. We will show analytically that this is not the case already in a simple one-layer binary weights overparametrized model. To the contrary, we find that an exponential number of sharp, isolated solutions with poor generalization properties exist even for very high levels of overparameterization. Indeed, these kind of solutions are by far in the majority, and algorithms that sample solutions with a flat measure find these typical ones (almost surely, in the limit of large system sizes); however, they also take an exponential amount of time in doing so. Thus, in practice, these typical solutions can only be found empirically in rather small networks. Efficient algorithms, that scale polynomially with the size of the problem, sample instead from wide regions of the space of the weights that are particularly dense with solutions and have good generalization properties.

Both kinds of solutions have been studied in simpler, non-overparameterized models, using tools from statistical physics of disordered systems: the typical solutions are the equilibrium ones [1, 2], and the atypical, highly entropic ones can be described by a large deviation technique [3–8] or by using a robustness bias [9]. Those techniques are non-rigorous, but a

few rigorous confirmation of some of the findings have been obtained [10–12].

Here, we extend those techniques to the study of the effect of overparameterization. We show that, contrary to what happens in overparameterized convex models, there are two transition points, separated by a gap. The first one is the information-theoretic interpolation threshold of the model: this is the point when zero-error solutions appear and perfect fitting of the data becomes possible. This point is obtained from an equilibrium computation and thus it is related to the typical, basically inaccessible solutions. The second transition point coincides with the sharp appearance of the highly locally entropic atypical solutions, that are attractive to learning algorithms. These dense regions stem from the development of new solutions which connect the preexisting ones.

We shall call this second transition the *Local Entropy* (LE) transition. This type of phase transition is not usually encountered in statistical physics, as it is driven by the appearance of rare structures in the solution space. Still it can be of basic relevance for learning processes (even very simple ones) that are not bound to try to sample from the dominating set of minima (i.e. are not designed to have the Gibbs distribution as stationary probability measure). This is indeed the case for all algorithms used for learning, which are subject to external perturbations, use ad hoc loss functions, and adopt peculiar optimization and initialization strategies, see also ref. [13].

Interestingly, the phase transitions to rare states have similarities to the localization phase transitions that are well known in quantum mechanics [14]. This fact is also consistent with the effectiveness of quantum annealing for learning problems similar to those discussed in this paper [15].

The paper is organized as follows. In sec. I we review some related literature, and introduce some basic non-convex analytically tractable versions of the random features models. In sec. II we study analytically the geometric structure of the loss landscape, derive Bayesian generalization bounds and the phase diagram for the interpolation and LE transitions. In sec. III, we report the results of numerical experiments on progressively less idealized and more realistic settings, validating the analytical findings, and confirming in particular that when the training algorithms start to be able to fit the data, they have already passed the interpolation point, and that they sample wide minima.

I. NON-CONVEX OVERPARAMETERIZED NEURAL CLASSIFIERS

Related work. The effects of overparameterization and the interpolation threshold have been recently studied in convex neural classifiers in which the input data are projected in a arbitrarily high dimensional space. These models are variants of the Random Features Model (RFM) which was first introduced as a tool to accelerate the training of Kernel machines [16–18]. More recently, the observation [19] that infinitely wide neural networks operate in the so-called "lazy regime", where the weights do not change much from their initial values during the gradient descent training dynamics, suggested that the behavior of neural networks can be approximated to some extent by random feature models, where the randomness in the features comes from the random initialization of the network weights (see for example [20] for a recent review). In the absence of specific regularization controls and for a given training set, as the size of the model increases the training and testing errors tend, initially, to decrease jointly. When the training error is about to reach perfect interpolation of the data, the test error begins to increase, giving rise to the famous U-shaped curve that describes the so called bias-variance trade-off in classical statistics. Not without surprise, if we keep adding parameters to the model, the test error behaves in a non-monotonic way: when the model exceeds the interpolation threshold, the training error remains zero and the test error starts to fall again, and tends to an absolute minimum in the regime of extreme overparameterization where the number of parameters is much larger than the number of samples. This phenomenon, called "double-descent" [21, 22], has been studied and reproduced in a number of different frameworks, ranging from rigorous computations [23] to statistical physics computations [24–27] in simple models of neural nets, to realistic architectures, see for example refs. [28, 29]. Subsequent numerical analysis of the Hessian of largely overparameterized models [30] showed that minimizers present many flat directions, and that it is not hard to find a path of zero training error connecting two solutions [31, 32]. In underparameterized neural networks, on the other hand, the authors of [33] showed that the landscape is very rough and dynamics is glassy. This led to think that the landscape of overparameterized networks where the dynamics is not glassy anymore, presents no "poor" minima at all [34]. According to our analysis, this is not the case. As we anticipated in the introduction, overparametrization has the effect of letting those connected regions appear at the LE transition, not letting "poor" minima completely disappear. Overparametrizing the network even further it is possible to increase the size of the connected region; "poor" or "sharp" solutions however remain the most numerous ones and dominate the Gibbs measure.

Overparameterized non-convex tractable model. Here we consider a non-convex RFM for binary classification with two layers. We consider random weights in the first layer (the "random features") and a second layer with N binary weights $\boldsymbol{w} \in \{-1, 1\}^N$ that are learned. Indeed, using binary weights suffices to make the overall learning problem highly non-convex.

In the model, each pattern $\boldsymbol{\xi}$ is generated on a hidden manifold, of dimension D, and projected as a pattern $\tilde{\boldsymbol{\xi}}$ on a visible feature space of dimension N. This models the common situation in which the raw input data is highly redundant, and its effective dimensionality is much lower. The projection is defined by:

$$\tilde{\xi}_i = \sigma\left(\frac{1}{\sqrt{D}}\sum_{k=1}^{D} F_{ki}\xi_k\right) \tag{1}$$

where F is a $D \times N$ feature matrix and $\sigma(\cdot)$ is a non-linear activation function. In the following we will consider for definiteness $\sigma(x) = \text{sign}(x)$ and a feature matrix of the Gaussian Orthogonal Ensemble (GOE) type, i.e. every element of F is a standard normal Gaussian; however our analytical results are valid for any $\sigma(\cdot)$ and every matrix having independent random entries with matching first and second moments, and that satisfy the hypothesis of the Gaussian Equivalence theorem [23, 24, 26, 35, 36]; see the details in the Supplementary Information (SI).

The corresponding output of the network is:

$$y_{\text{out}} \equiv \text{sign}\left(\frac{1}{\sqrt{N}}\sum_{i=1}^{N} w_i\,\tilde{\xi}_i\right) \tag{2}$$

We consider a training set composed of $P = \alpha N$ random patterns extracted from a standard normal distribution; the label y^μ corresponding to a given pattern $\boldsymbol{\xi}^\mu$ is assigned by a "teacher" network having random binary weights $\boldsymbol{w}^T \in \{-1,1\}^D$ as $y^\mu = \text{sign}\left(\frac{1}{\sqrt{D}}\sum_{k=1}^{D} w_k^T \xi_k^\mu\right)$. This intends to model the situation in which the true labels depend in a simple way from a latent representation, to which however the student network does not have access. The learning task consists in finding the weights $\boldsymbol{w}$ that fit all the data in the training set and that generalize well on the whole generative model.

II. GEOMETRY OF MINIMA VS OVERPARAMETERIZATION: THRESHOLD PHENOMENA

In the following we consider the primitive loss function that counts the number of misclassified patterns in the training set whose stability is greater than a given margin of $\kappa \geq 0$. For each pattern, the stability Δ^μ is defined as the product of the pre-activation of the output unit $\lambda^\mu(\boldsymbol{w})$ and the binary label of pattern $y^\mu = \pm 1$:

$$\Delta^\mu(\boldsymbol{w}) \equiv y^\mu \lambda^\mu(\boldsymbol{w}) \tag{3}$$

where

$$\lambda^\mu(\boldsymbol{w}) \equiv \frac{1}{\sqrt{N}}\sum_{i=1}^{N} w_i \tilde{\xi}_i \tag{4}$$

The loss function per pattern is defined as

$$\ell_{NE}\left(-\Delta^\mu(\boldsymbol{w});\kappa\right) = \Theta\left(-\Delta^\mu(\boldsymbol{w}) + \kappa\right) \tag{5}$$

where $\Theta(\cdot)$ is the Heaviside step function: $\Theta(x) = 1$ if $x > 0$ and zero otherwise. For $\kappa = 0$ this loss reduces to the one that counts the number of training errors; with a slight abuse of language we call it "number-of-errors loss" even if the margin is non-zero. For the analytical study, we will be interested in the large-size limit, where our calculations can be performed by asymptotic methods: $N, D, P \to \infty$ while keeping finite the ratios

$$\alpha \equiv \frac{P}{N}\ ,\ \alpha_T \equiv \frac{P}{D}\ ,\ \alpha_D \equiv \frac{D}{N}\,, \tag{6}$$

with $\alpha = \alpha_T \alpha_D$. In order to compute the typical properties of the solution space, the key quantity of interest is the averaged free entropy of the model, i.e.

$$\phi = \lim_{N,P,D\to\infty} \frac{1}{N}\langle \ln Z\rangle_{\xi,F} \tag{7}$$

where we denoted with $\langle\bullet\rangle_{\xi,F}$ the average over both the patterns (including the desired outputs and thus the teacher) and the features. Here Z denotes the partition function of the model which reads

$$Z(\beta) = \sum_{\boldsymbol{w}} e^{-\beta\sum_{\mu=1}^{P}\ell_{NE}(-\Delta^\mu(\boldsymbol{w});\kappa)} \tag{8}$$

For generic β, $Z(\beta)$ is the generating function in the variable $e^{-\beta}$ of the number of errors. In the analytical computations however we have only considered the large β limit, where the partition function reduces to counting the number of global minima, i.e. zero-error configurations (solutions) when they exist:

$$Z = \sum_{\boldsymbol{w}}\prod_{\mu=1}^{P}\Theta\left(\Delta^\mu(\boldsymbol{w}) - \kappa\right) \equiv \sum_{\boldsymbol{w}} \mathbb{X}_{\xi,F}(\boldsymbol{w};\kappa) \tag{9}$$

where $\mathbb{X}_{\xi,F}$ is the indicator function on $\boldsymbol{w}$ that all patterns are being correctly classified with the required robustness.

The averages of the logarithm in eq. (7), give access to the most probable number of solutions for a randomly chosen training set, and can be computed by asymptotic methods developed in the theory of disordered systems, either the so called replica method or the cavity method [37].

A first basic result of the analysis is that, for fixed α_T and κ, there is an $\alpha_{\text{max}}(\kappa,\alpha_T)$ for which $\phi \geq 0$, signalling that, with high probability, for $\alpha > \alpha_{\text{max}}(\kappa,\alpha_T)$ solutions with stability κ or larger cease to exist. In this context, supposing that the learning problem and thus α_T was fixed and that we are controlling the degree of overparameterization via α, the "interpolation threshold" $\alpha_c(\alpha_T)$ is the value of α for which *all* solutions disappear, i.e. $\alpha_c(\alpha_T) = \alpha_{\text{max}}(0,\alpha_T)$.

Conversely, we also define the maximum margin $\kappa_{\text{max}}(\alpha,\alpha_T)$ for fixed values of α,α_T as the value of κ for which $\phi = 0$. The solutions with maximum margin play a central role in our analysis, since they lie in the middle of dense regions whose breakup (as α increases) signals the LE phase transition. It is useful to point out that even if the entropy of the solutions vanishes at κ_{max}, their typical overlap (i.e. normalized dot product, also called cosine similarity) is still strictly smaller than 1.

Phase diagram. Before diving into analytical details, we anticipate how the geometry of the space of solutions changes as we increase the degree of overparameterization. The phase diagram of the model is reported in Fig. 1. The plane (α_T,α) is divided into three distinct regions:

(1) an UNSAT region when the value of the density of constraints exceeds the interpolation threshold: $\alpha > \alpha_c(\alpha_T)$. In this region there exists no configuration of weights that is

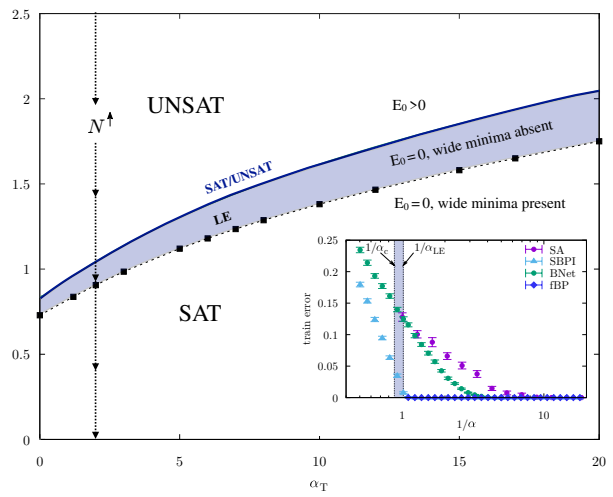

Figure 1. SAT/UNSAT interpolation threshold and Local Entropy transition versus α_T for the binary non-convex model of random features. For $\alpha_T \to 0$ we recover the critical capacity [1] and the local entropy transition [9] of a non-overparameterized binary perceptron trained on random patterns. In the inset we show the training error of SA, fBP and SBPI versus the degree of overparameterization $1/\alpha$ for $D = 201$ and $\alpha_T = 3$. Points are averages over 20 independent samples (except for fBP where we used 10 samples) and 5 independent runs per samples (3 for fBP). None of those algorithms is able to find solutions for $\alpha > \alpha_{\rm LE}$.

able to the whole training set. This threshold is independent of the learning algorithm, it depends only on the properties of the training data and of the architecture. On the other hand for $\alpha < \alpha_c(\alpha_T)$ we have a SAT region, so in principle the complexity of the model is sufficient to learn the data.

(2) for $\alpha_{\rm LE}(\alpha_T) < \alpha < \alpha_c(\alpha_T)$, despite the existence of configurations of weights that fit all the training set, they are either isolated or belong to minima that have a small characteristic size. These solutions turn out to be not easily accessible by learning algorithms.

(3) for $\alpha \leq \alpha_{\rm LE}(\alpha_T)$ highly entropic wide minima start to appear. These flat minima, though exponentially rare compared to the isolated solutions, are accessible by simple, efficient algorithms. The threshold $\alpha_{\rm LE}(\alpha_T)$ is thus the location of the Local Entropy transition, which we interpret as an upper bound for the effectiveness of learning algorithms.

As an experimental check of this picture, we show in the inset of Fig. 1 the train error of four algorithms that are representative of a spectrum of sampling strategies. On one extreme of the spectrum, we used Simulated Annealing (SA) [38], which samples from the equilibrium Gibbs distribution. On the opposite end, we used focusing Belief Propagation (fBP) [4], which is a modified version of the message-passing Belief Propagation (BP) algorithm [39] and is designed to target high local entropy regions (if present). The goal of the original BP algorithm is to perform statistical inference, and at convergence its messages allow to derive the marginal probabilities for each variable, computed for a uniform distribution over the solutions of the training task. The modification introduced by fBP consists in forcing the messages to progressively focus on the most dense regions, until they become peaked on a single configuration, thereby resulting in an efficient solver. The focusing process is controlled by fixing an overall "strength" $y > 1$ and by scheduling a parameter γ from 0 to ∞. Two more heuristic algorithms are specifically designed to work efficiently on binary architectures. One is the Stochastic BP-inspired (SBPI) algorithm [40], which can be regarded as a simple and fast approximate version of fBP. The other is BinaryNet (BNet) [41], which is a modified version of Stochastic Gradient Descent (SGD). As we can observe in the figure, none of these algorithms can find solutions below $1/\alpha_{\rm LE}$.

Typical solutions. Using the replica method in its replica symmetric (RS) version (see SI), the averaged free entropy in eq. (7) turns out to depend on the "order parameters" q, p, p_d, r and their conjugate Lagrange multipliers $\hat{q}$, $\hat{p}$, $\hat{p}_d$, $\hat{r}$. Geometrically q represent the typical overlap between a pair of solutions; p is the typical overlap between a pair of solutions projected in the teacher space (which has dimension D), the projection being performed simply by using the feature matrix F_{ki}; p_d is the typical squared norm of a projected solution and finally r denotes the typical overlap between a projected solution and the teacher.

Eventually, ϕ can be found with the saddle point method, by optimizing over eight order parameters

$$\phi = \max_{q,\hat{q},p,\hat{p},p_d,\hat{p}_d,r,\hat{r}} \phi_{RS}\,(q,\hat{q},p,\hat{p},p_d,\hat{p}_d,r,\hat{r}) \tag{10}$$

where ϕ_{RS} is the RS expression for ϕ (see SI). Knowing the order parameters for which the function ϕ_{RS} is maximal allows to compute not only the entropy but also other quantities of interest, such as the generalization error ϵ_g, defined as the probability of wrongly classifying a new (unseen) pattern

$$\epsilon_g = \mathbb{E}_\xi\, \ell_{NE}(-\Delta(\boldsymbol{w});\kappa)\,. \tag{11}$$

We find

$$\epsilon_g = \frac{1}{\pi}\arccos\left(\frac{M}{\sqrt{Q_d}}\right) \tag{12}$$

where $M \equiv \mu_1 r$, $Q_d \equiv \mu_\star^2 + \mu_1^2 p_d$, and μ_1, $\mu_\star$ are constants that depend only on the nonlinear function σ (see SI for their expressions). From the solutions of the saddle point equations we can also compute the probability that the average of the outputs of students sampled from the posterior on a random new pattern has different sign than that given by the teacher (see the SI for the definition); this turns out to be equivalent to computing the generalization error of the barycenter of typical solutions. All the details of the computation are reported in the SI; here we report the final result

$$\epsilon_g^B = \frac{1}{\pi}\arccos\left(\frac{M}{\sqrt{Q}}\right)\,, \tag{13}$$

where $Q \equiv \mu_\star^2 q + \mu_1^2 p$. In Fig. 2 we show the plot of the generalization error of typical solutions with zero, non-zero

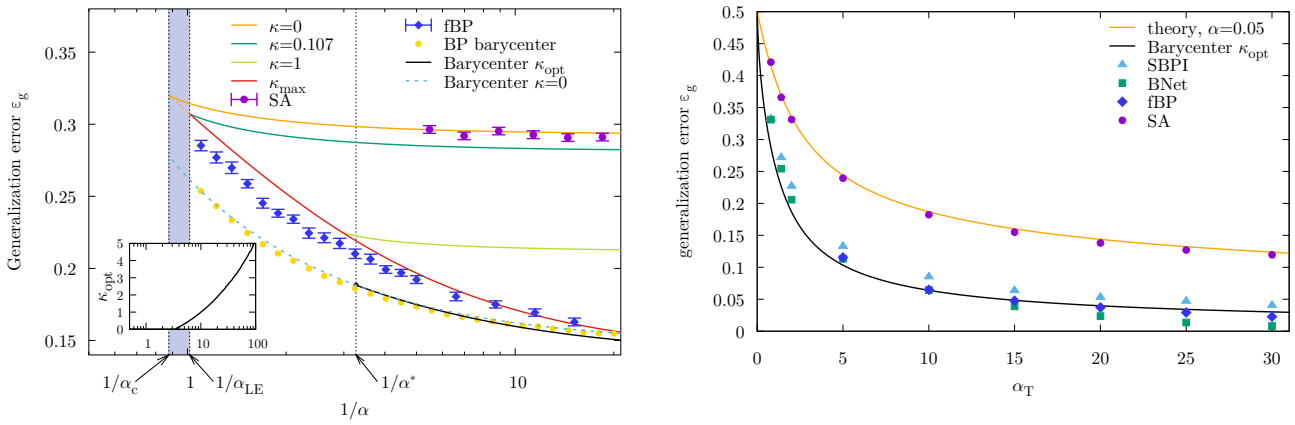

Figure 2. (Left panel) Generalization error as a function of the degree of overparameterization $1/\alpha$, for $\alpha_T = 3$. Vertical dashed lines denote the SAT-UNSAT transition α_c^{-1}, the local entropy transition $\alpha_{\rm LE}^{-1}$, and the $\kappa_{\rm opt}$ transition $(\alpha^*)^{-1}$. We show the generalization error of typical solutions with fixed margin $\kappa = 0, 0.107, 1$ and with maximum margin $\kappa_{\rm max}(\alpha, \alpha_T)$. The dashed turquoise curve is the error of the barycenter of typical solutions having zero margin, whereas the black line represents the generalization error of the "best" barycenter which was found by optimizing the margin. In the inset we show that this optimal margin $\kappa_{\rm opt}$ undergoes a transition when crossing α^*. We also show numerical results ($D = 201$) of two representative algorithms: SA (violet points) and fBP (blue points). When SA is able to find solutions, the corresponding generalization error is compatible to the one obtained by typical zero-margin configurations. In the large overparameterization regime fBP behaves similarly to the generalization error of the barycenter of zero-margin solutions. Yellow points (error bars not shown for clarity) represent the barycenter of zero margin solutions as computed by using the BP estimation of the posterior distribution. (Right panel) Generalization error versus α_T in the large overparameterization regime ($\alpha = 0.05$, $D = 201$). While SA gives the same generalization error of typical zero-margin solutions, SBPI, BNet and fBP, that do not target or sample from the Gibbs measure, perform much better. All points are averages over 5 independent samples, 2 independent runs per sample.

and maximum possible margin versus the degree of overparameterization $1/\alpha$, together with the generalization error of the barycenter of typical solutions having zero margin. All those curves are monotonically decreasing.

Moreover we show that the margin $\kappa_{\rm opt}$ that should be imposed in order to minimize the generalization error of the barycenter undergoes a transition from zero (for $\alpha > \alpha^*$) to non-zero values (for $\alpha < \alpha^*$) whenever we increase the degree of overparameterization. The value of the optimal margin $\kappa_{\rm opt}$ is plotted in the inset of Fig. 2.

Numerical Checks. In order to corroborate the analytical findings, we have performed some numerical experiments (see Fig. 2) using the four algorithms mentioned above: SA, fBP, SBPI and BNet. Similarly to what happens in spin glass models, we found that for sufficiently low α (i.e. for relatively small system sizes) SA is able to escape from local minima and find solutions that have generalization error which matches the one obtained by replica theory. We also found a perfect agreement between the theoretical results and the numerical experiments when we computed the distribution of the stabilities of typical configurations (see SI). We remark that the ability of SA to find solutions for low values of α is due to finite-size effects: indeed we show in the SI that scaling up the sizes while keeping α, α_T fixed, at a certain point SA is no longer able to find solutions. We find that fBP, SBPI and BNet, despite being mildly affected by finite-size effects as well, converge to entropic states that have a much better generalization error, as also predicted by the theory.

The entropy landscape around a typical solution. Having established that algorithms find solutions with different generalization properties, it remains to understand in which regions of the landscape those solutions end up and how they arise in terms of the degree of overparameterization.

A way to answer those questions is by studying the *local entropy* landscape by the computation of the so-called *Franz-Parisi* potential [42]. This technique has been introduced as a tool to study the role of metastable states in spin glasses [42] and recently [2, 5] it was used to show that, in one and two-layer binary neural networks, typical solutions with zero margin are organized as clusters with vanishing internal entropy, a scenario that has been called *frozen*-1RSB.

Given a configuration $\tilde{\boldsymbol{w}}$ with margin $\tilde{\kappa}$ that we call the "reference", the local entropy is the log of the number of configurations $\mathcal{N}(\tilde{\boldsymbol{w}}, d; \kappa)$ that are solutions with margin κ and that are constrained to be at a given normalized Hamming distance d from $\tilde{\boldsymbol{w}}$:

$$\mathcal{N}(\tilde{\boldsymbol{w}}, d; \kappa) = \sum_{\boldsymbol{w}} \mathbb{X}_{\xi,F}(\boldsymbol{w}; \kappa)\, \delta\left(N(1-2d) - \sum_i w_i \tilde{w}_i\right) . \tag{14}$$

The properties of the landscape around typical references can then be investigated by studying their average local entropy, which is called *Franz-Parisi* free entropy [2, 42]

$$\phi_{\rm FP}(d; \tilde{\kappa}, \kappa) = \left\langle \frac{1}{Z} \sum_{\tilde{\boldsymbol{w}}} \mathbb{X}_{\xi,F}(\tilde{\boldsymbol{w}}; \tilde{\kappa}) \ln \mathcal{N}(\tilde{\boldsymbol{w}}, d; \kappa) \right\rangle_{\xi,F} . \tag{15}$$

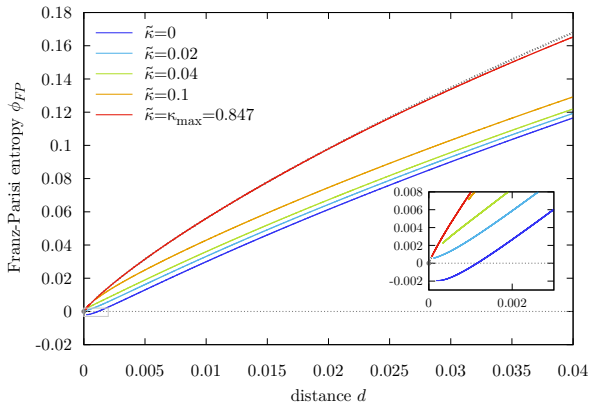

Figure 3. Local entropy of solutions with zero margin $\kappa = 0$ as a function of the distance d, evaluated for typical references having different values of the margin $\tilde{\kappa}$. Here $\alpha = 0.5$ and $\alpha_T = 8$. The (barely visible) dotted gray line at the top represents the total number of configurations at that given distance, which is a geometrical upper bound for the local entropy. The maximum margin that can be imposed is $\kappa_{\max} \simeq 0.847$ and corresponds to the curve with the largest local entropy. The inset refers to the entropy curves for small distances (they are not complete due to numerical issues). For all curves $\phi_{\rm FP}(d = 0; \tilde{\kappa}, \kappa) = 0$, so for $\tilde{\kappa} = 0$ the curve is non-monotonic in a neighborhood of $d = 0$, while for $\tilde{\kappa} > 0$ the profiles shown are all positive.

This quantity can be again computed by the replica method with a double analytic continuation (details in the SI).

Here, following ref. [9], we are chiefly interested in the behavior of this quantity when $\kappa = 0$. For a given value of $\alpha < \alpha_c$, the local entropy curves of the references exhibit different characteristics as $\tilde{\kappa}$ varies. This is shown in Fig. 3. The overall picture closely resembles that of simpler models [9], and we point out some noteworthy results (here and in the following we omit α_T, which we consider to be fixed, for simplicity):

(1) Zero-margin references are isolated: $\phi_{\rm FP}(d; \tilde{\kappa} = 0, \kappa = 0)$ is always negative in a neighborhood of $d = 0$. This explains the poor performance, both in terms of efficiency in finding a solution and in terms of the generalization properties of the solution it finds, of the SA algorithm, which directly targets the Gibbs measure. Even for small non-zero values of $\tilde{\kappa}$ the local entropy is negative for some distances d, or it is non-monotonic, denoting the existence of small isolated clusters of solutions.

(2) Fixing a small enough value of α and keeping increasing the margin of the reference configuration one eventually reaches a threshold value $\tilde{\kappa} = \kappa_u(\alpha)$ that separates a region for $\tilde{\kappa} < \kappa_u(\alpha)$ where the local entropy is non-monotonic (as described in the previous point) from a phase where the local entropy is monotonic (for $\tilde{\kappa} > \kappa_u(\alpha)$). This means that those references are located inside a dense region of solutions that extends to very large scales. The monotonic local entropy phase extends up to $\tilde{\kappa} = \kappa_{\max}(\alpha)$. As shown in Fig. 3, the highest curve in terms of local entropy is found by the typical configurations having maximum margin $\tilde{\kappa} = \kappa_{\max}(\alpha)$. These large-scale regions are apparently targeted by efficient solvers, which also have lower generalization errors than SA.

Local Entropy transition. A fundamental question that remains to be answered is how those (atypical) dense regions change when increasing α. In previously studied convex models those regions tend to shrink continuously and they reduce to a point at the SAT/UNSAT transition. This is not the case here: similarly to what happens in previously studied teacher-student non-convex models, those regions shrink when increasing α, until a critical value $\alpha_{\rm LE} < \alpha_c$ is reached, beyond which they fracture in multiple pieces. This is the LE transition. For $\alpha > \alpha_{\rm LE}$ no algorithm is seemingly able to find a solution efficiently, whereas below it efficient algorithms with good scaling properties only find solutions in non-isolated regions. Thus, $\alpha_{\rm LE}$ can be regarded as a fairly good upper bound to the algorithmic capacity for the most efficient algorithms.

Different approaches have been devised in order to estimate analytically $\alpha_{\rm LE}$. The first one is based on the use of a large deviations analysis [3, 4] which however leads to a quite heavy formalism for the models under study. We have thus adopted a recently introduced simpler method [9] which gives similar results to the large deviations approach. It is based on the observation that, by definition of $\alpha_{\rm LE}$, references located in the large-scale dense region should not exist anymore when $\alpha > \alpha_{\rm LE}$. We can therefore estimate $\alpha_{\rm LE}$ by the condition

$$\kappa_u(\alpha_{\rm LE}) = \kappa_{\max}(\alpha_{\rm LE}) \tag{16}$$

meaning that $\alpha_{\rm LE}$ is the value of α after which not even maximum margin solutions have a monotonic local entropy profile: all $\tilde{\kappa}$-margin solutions are located in disconnected balls in configuration space (see Fig. 4). This is a stricter condition than the one obtained from the large deviation analysis, which uses the criterion that *all* solutions have non-monotonic profiles; thus, it likely slightly under-estimates the true $\alpha_{\rm LE}$, but this difference is smaller than the resolution that can be detected by our numerical experiments.

III. NUMERICAL EXPERIMENTS

In order to assess the relevance of the analysis presented above to more realistic cases, we have performed a series of numerical studies that consider progressively less idealized scenarios. First, we investigated the simplest non-convex continuous overparameterized model, namely a tree-like committee machine trained on randomly generated and randomly projected data, again with labels provided by a random teacher. Second, we moved to deeper networks: we studied a fully-connected multi-layer network with a fixed number of variable-width layers, trained with gradient descent using the popular ADAM optimizer, and a deep convolutional networks trained with both SGD and the ADAM optimizer. These deep models have been trained respectively on the first 10 principal components of a reduced version of the MNIST dataset and on images of CIFAR10.

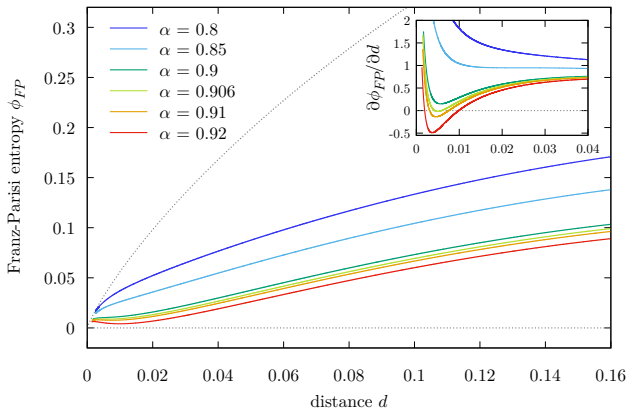

Figure 4. Estimating the Local Entropy transition $\alpha_{\rm LE}$ by looking to the local entropy profiles of maximum margin references and its derivative with respect to distance (inset). Here $\alpha_T = 2$ and the critical capacity is $\alpha_c \simeq 1.045$. For low values of α, e.g. 0.8, 0.85 and 0.9 typical references with maximum margin are located inside a dense region extending to a very long scale, since the local entropy is monotonic. Near $\alpha = \alpha_{\rm LE} \simeq 0.906$ the derivative of the local entropy develops a new zero for small distances. This signals a transition in the geometrical structure of the dense regions. For $\alpha > \alpha_{\rm LE}$, the local entropy is not monotonic anymore, meaning that typical maximum margin references (as well as all other typical solutions with smaller margin) are located in disconnected balls in configuration space.

Across these tests, we found some common characteristics, compatible with the analytical findings. In order to find a solution, the networks require a minimum number of parameters that is larger than the size of the input. When that degree of overparameterization is achieved, we observe that indeed we have already passed the "interpolation" point, since many solutions exist (even after having accounted for the permutation and rescaling symmetries in the networks) and they are located far apart from each other and belong to a flat region. As we increase the amount of overparameterization, the solutions that we found grow further apart in distance[1], their local landscapes become even flatter, and their generalization properties improve. We also observe that within such flat regions, different algorithms sample solutions of different types, more or less barycentric, and with a different flatness (as estimated by their local energy profiles, defined below).

Overparameterized tree committee machine: We studied an overparameterized tree-committee architecture with K hidden units, trained on random patterns. The teacher and the patterns are generated in the same way as for the perceptron of eq. 2, and in particular the device receives binary inputs $\tilde{\boldsymbol{\xi}}$ of length N obtained by projecting randomly-generated D-dimensional inputs $\boldsymbol{\xi}$ through a random matrix F and a non-linearity σ, as in eq. (1). Again, we choose $\sigma = \text{sign}$. We now consider only values of N divisible by K, and divide the inputs into groups of N/K, each of which is fed to one of the K hidden units; the final output is then decided by majority voting, as:

$$y_{\rm out} \equiv \text{sign}\left(\sum_{h=1}^{K} \text{sign}\left(\frac{1}{\sqrt{N/K}} \sum_{i=(h-1)\frac{N}{K}+1}^{h\frac{N}{K}} w_i \tilde{\xi}_i\right)\right) \qquad (17)$$

Beside the architecture, one major difference with the perceptron case is that here the weights $\boldsymbol{w}$ are assumed to be continuous. Due to the sign activation function, each unit is invariant to scaling, and thus we normalize the weights of the units by fixing their norms to 1.

We consider two learning algorithms for this architecture. The first one is a version of focusing-BP (fBP) that operates with continuous weights [7]. The implementation exploits the central limit theorem and thus it only works well for relatively large values of N/K; furthermore, even in the large N limit, it is only approximately correct on the tree-committee machine architecture. Despite this, in practice it produces excellent results.

The second algorithm is Stochastic Gradient Descent with cross-entropy loss. Following ref. [7], we substituted the (non-differentiable) units' activation function in eq. 17, $\text{sign}(\Delta)$, with $\tanh(\beta\Delta)$. The new parameter β can be regarded as taking the role of the norm of the unit's weights, since we keep the weights normalized at each step. We explicitly schedule this parameter, letting it start from a small value and making it diverge during the training, thereby recovering the original sign activation at the end [2]. Analogously, we also schedule a parameter γ that has the role of the norm of the (fixed) weights in the second layer, and that we can simply plug in the cross-entropy (see the Materials and Methods).

Here, we report the result of tests performed on a committee machine with $K = 9$ hidden units, trained on $P = 10005$ patterns produced in $D = 2001$ dimensions, thus at a fairly large $\alpha_T = P/D = 5$, while varying the degree of overparameterization $N = P\alpha^{-1}$ (see the Materials and Methods for the details of the settings used for the training). Our results are reported in Fig. 5. We found that both fBP and SGD fail to find a solution below $\alpha^{-1} \approx 0.36$, which we thus take to be a plausible estimate for the algorithmic threshold $\alpha_{\rm LE}^{-1}$ where the phase of the robust solutions presumably changes. This is corroborated by the study of the overlaps: for any given training set, the SGD algorithm finds different solutions when started from different random initial conditions (this is not true for fBP due to its deterministic nature). We measured the average overlap (cosine similarity) between the solutions, $\langle \frac{w^a \cdot w^b}{N} \rangle$, and found that when reducing α^{-1} the overlap grows, but it does not tend to 1 as α tends to $\alpha_{\rm LE}$, which one would expect if the solutions shrank to a single interpolation point

[1] The distance appears to plateau at a value strictly lower than the geometrical bound.

[2] Note that the divergence of the norms would occur naturally anyway in standard SGD with the cross-entropy loss.

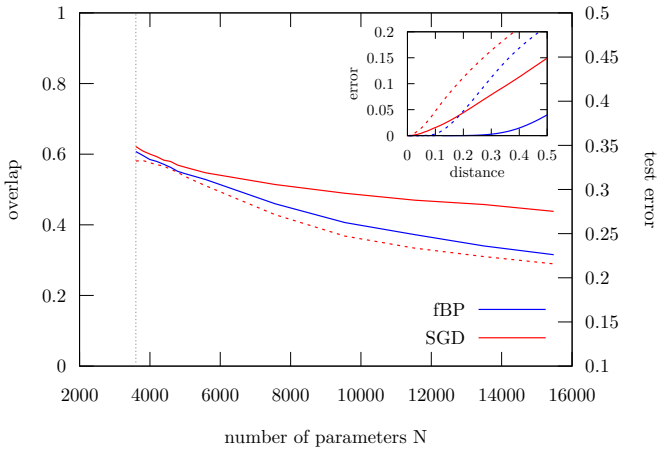

Figure 5. Results for the overparameterized continuous tree-like committee machine. Tests performed with $D = 2001$, $\alpha_T = 5$, $K = 9$. All points are averages over 5 samples, with 5 independent runs per sample for SGD. Dashed red line: mean overlap between two SGD solutions on the same sample, as a function of the number of parameters N. Solid lines: test error for BP and SGD. The vertical dashed grey line at $N = 3600$ ($\alpha^{-1} \approx 0.36$) denotes the algorithmic threshold below which the algorithms solve fewer than 50% of the samples. The overlaps are still far from 1 at this point. Inset: Local energy profiles, i.e. average train error as a function of the Euclidean distance from a solution. The dashed lines are measured at $N = 3996$ ($\alpha^{-1} \approx 0.4$), the solid lines at $N \approx 13500$ ($\alpha^{-1} \approx 1.35$).

like in convex models. The generalization error behaves as expected, decreasing monotonically with N; SGD is slightly worse than fBP in this regard. We also measured the flatness of the minima found by the algorithm by plotting the "average local energy"[3], i.e. the average training error profile of the landscape surrounding each solution, as a function of the distance. This can be estimated straightforwardly and robustly by randomly perturbing the weights, with a varying degree of multiplicative noise (the weights are still renormalized after the perturbation). In Fig. 5, we show two sets of curves, one at $\alpha^{-1} \approx 0.4$, close to $\alpha_{\rm LE}^{-1}$, and one at $\alpha^{-1} \approx 1.35$, at the opposite end. As expected, close to the threshold the minima are generally sharper, but in all cases both SGD and fBP have flat profiles for small distances, reflecting the fact that both algorithms are inherently biased towards wide flat minima [7]. The bias is stronger for the fBP algorithm, which was explicitly designed for this purpose, and its profiles are indeed flatter.

Overall, all the phenomenological features that we could measure on this model are compatible with the theoretical analysis of the previous section on the binary perceptron, despite the more complex architecture and the continuous weights, even in the context of gradient-based learning.

Comparing solutions in Deep architectures: removing symmetries

When discussing the space of configurations of standard multi-layer architectures, we need to be more careful compared to the simple models discussed so far, due to the presence of additional symmetries [44].

First, the ReLU activation function that is commonly used in deep learning models has the property that $\text{ReLU}(ax) = a\,\text{ReLU}(x)$, which implies that if we scale all the input weights of a hidden unit by a factor a^{-1} and all its output weights by a factor a the network's output will be unaffected. By setting the factor a to the norm of the input weights, one can normalize a hidden unit by simply "pushing up" its norm to the next layer. Furthermore, when a network is used for classification tasks, the output label is determined by an argmax operation, which is invariant to scaling. Thus, normalizing the last layer too is possible without affecting the classification properties of the network. In the full configuration space, each neural network has infinitely many parameter representations, and the error rate landscape has some trivially null directions. This issue can be avoided by normalization, which can be performed simply by starting from the first layer and moving up, as described above.

There is also a second, discrete symmetry, since networks are invariant to permutations of the units inside any hidden layer. If we failed to take into account this, we could measure a non-zero distance between networks which are just permuted versions of each other and thus functionally equivalent. One natural way to break this symmetry is to normalize and align the networks before comparing their weights. In our tests, we adopted again a sequential approach for aligning two given networks. Starting from the first hidden layer, we find the permutation of the second networks' units that minimizes the distance between the weights of the two networks for that layer[4], apply it, and proceed to the following layer.

In the following paragraphs, we present experiments on deep architectures that have both these symmetries. We use standard techniques to train them, and thus do not explicitly keep the norms and permutations under control. We do however normalize and align them when we compare two solutions, either to compare their error rates or to measure their distance.

Multi-layer neural network We studied a simple fully-connected multi-layer perceptron inspired by ref. [29]. The network has a fixed number H of hidden layers ($H = 5$ in the numerical experiments) whose width is varied in order to increase the number of model parameters. The model is required to perform a binary classification task on the parity of digits of 10000 MNIST images, using as inputs only the first 10 principal components of each image. We trained the model using full batch gradient descent with ADAM optimization, both with random orthogonal initialization [45] and with adversarial initialization [46]. The results are reported in Fig. 6 where it can be seen that in general different optimization schemes

[3] Notice that the local energy of a configuration is highly correlated with its local entropy, see e.g. [43]

[4] This can be accomplished by a matching algorithm, which is $O(H^3)$ if H is the number of hidden units of the layer. In practical terms, it is typically much quicker than the training.

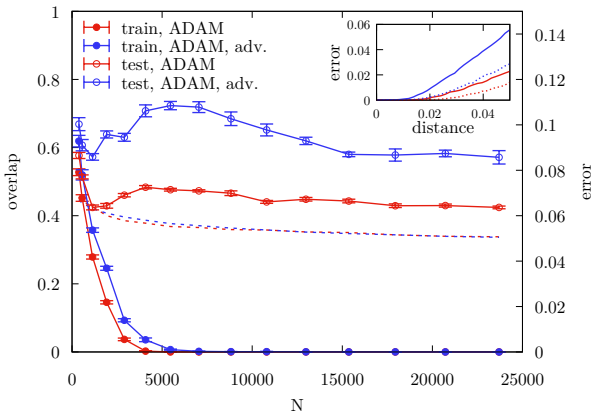

Figure 6. Train (full points) and test (empty points) error as a function of the network parameters N for a fully-connected network with 5 hidden layers. All points are averages over 10 independent runs. The network is trained using full batch gradient descent with ADAM optimization, with random orthogonal initialization (red curves) and adversarial initialization (blue curves). When the algorithms start finding zero errors solutions, the mean overlap between solutions is far below 1 (dashed lines). Inset: local energy profiles for both algorithms at two different values of the overparameterization ($N \simeq 9 \cdot 10^3$ (full lines) and $N \simeq 2 \cdot 10^4$ (dashed lines)).

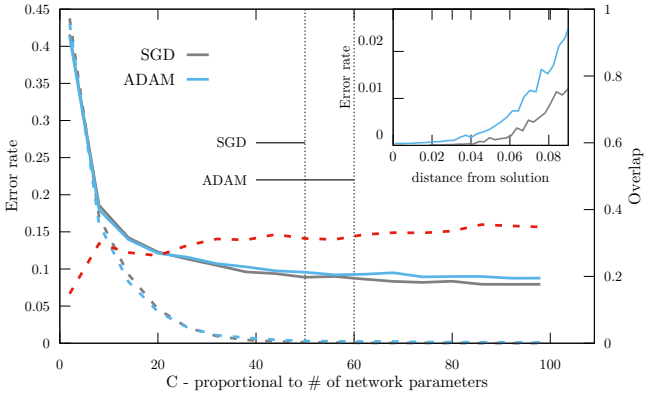

Figure 7. Train (dashed) and test (solid) errors of solutions obtained with SGD and ADAM (gray and blue lines respectively) as a function of C. Each point is the average on 5 independent samples. Red dotted line: average overlap between 3 different pairs of solutions (ADAM). Inset: local energy as a function of the distance from solutions (gray: SGD with $C = 54$; blue: ADAM with $C = 64$).

lead to different generalization error plateaus and different algorithmic thresholds. When the model starts to fit the training set, the solution is not unique and as a consequence the mean overlap between independent instances is lower than 1. The inset shows that the solutions are indeed robust to noise perturbations even in the proximity of the algorithmic threshold, and they become flatter as the overpameterization increases.

Convolutional networks As a second representative case of deep architectures we analyze a 5-layer NN, with 4 convolutional layers followed by a fully connected one, as in ref. [28]. After each 2d convolution, a batch normalization is performed before applying a ReLU nonlinearity. The overparameterization in this model is adjusted via a parameter C: in layer ℓ there are $2^\ell \cdot C$, 3x3 convolutional filters. This CNN is trained on CIFAR10 for 200 epochs, using two different learning algorithms: ADAM with momentum and SGD with a learning rate $\eta = 10^{-2}$. For real datasets like the one we considered, while it is relatively easy to achieve a very low training error, getting to precisely 0 error requires a disproportionate amount of additional computation. For this reason, we consider as solutions all configurations that misclassify at most 1 pattern (<0.0017% training error), and estimate the algorithmic LE threshold according to this criterion. In Fig. 7, train (dashed line) and test (solid line) errors are shown for both optimizers. SGD and ADAM begin to fit the training set data at $C = 50$ and $C = 60$ respectively, while the generalization error is monotonically decreasing with the number of network parameters. It is worth noticing that these architectures work in a relatively lazy training regime: on one hand, the first layer is less affected by the training, while the following layers change progressively more; on the other hand, the distance of the trained configurations from their initial conditions does not drop to zero as the number of parameters diverges (see details in the SI). Consistently with the analytical results, and similarly to other networks we have studied, we find that the learning algorithms, namely SGD and ADAM, find different solutions depending on initial conditions. In particular, even when we get close to the algorithmic threshold, we observe that the overlaps, in contrast to what happens at the interpolation threshold in convex networks, do not tend to their maximum value of one (red curve in Fig. 7). The solutions found by SGD and ADAM show similar geometric properties, i.e., they belong to flat regions of the training error landscape. This can be seen in the inset of Fig. 7, where we display the results of the numerical analysis of the average (local energy) landscape around a given solution. Like for the other models, this was measured by random sampling, perturbing the solutions with multiplicative noise (see the SI for details). The first derivative at short distances is essentially zero.

IV. CONCLUDING REMARKS

Our results characterize the interplay between overparameterization and nonconvexity in neural networks learning data generated by structurally different networks randomly chosen from a natural distribution. In particular, we identify a new phenomenon, namely the existence of a phase transition driven by the appearance of solution sets that are statistically atypical but which seem to be the ones targeted by learning algorithms. For the same systems we are able to derive the generalization error of different types of solutions and predict how to optimize the Bayesian error. The analytical techniques also suggest a number of numerical verifications that can be done on deep

networks and for different learning algorithms. The consistency of the results is very good, suggesting that the scenario identified in the analytically tractable models, i.e., the essential role played by highly entropic atypical solutions, may in fact be general.

There are several natural future directions. On the one hand, in-depth numerical studies of large deep networks should be conducted, and algorithms should be further optimized building on the information derived from the structure of solutions. Most algorithms already do this as a result of the tuning process that has been put in practice during the last decade. Still further progress appear to be possibile, and some steps in this direction have already been taken. On the other hand, it would be important to corroborate our results with rigorous bounds, for more general data distributions, in order to reach a more complete mathematical theory for learning in non-convex overparameterized systems. Finally, a theoretical confirmation that dynamics of a broad class of algorithms is indeed attracted to these structures would be of great interest (an analysis that shows that SGD is biased towards flat minima can be found in [13], but some of the assumptions are justified phenomenologically).

From a physics and modeling perspective, it seems to us that having identified that atypical states play a key role in learning processes opens the way toward a fertile connection between out-of-equilibrium physics and modeling of learning systems. Indeed, such states are inherently atypical with respect to algorithmic dynamics that tend to sample energy with a Gibbs measure and where the basic energy function, or loss function, is defined directly on the data as the number of errors.

ACKNOWLEDGEMENTS

We gratefully thank Fabrizio Pittorino for sharing with us his code implementing matching of CNN.

[1] Werner Krauth and Marc Mézard. Storage capacity of memory networks with binary couplings. *Journal de Physique*, 50(20):3057–3066, 1989. doi:/10.1051/jphys:0198900500200305700.

[2] Haiping Huang and Yoshiyuki Kabashima. Origin of the computational hardness for learning with binary synapses. *Phys. Rev. E*, 90:052813, Nov 2014. doi:10.1103/PhysRevE.90.052813.

[3] Carlo Baldassi, Alessandro Ingrosso, Carlo Lucibello, Luca Saglietti, and Riccardo Zecchina. Subdominant dense clusters allow for simple learning and high computational performance in neural networks with discrete synapses. *Phys. Rev. Lett.*, 115:128101, Sep 2015. doi:10.1103/PhysRevLett.115.128101.

[4] Carlo Baldassi, Christian Borgs, Jennifer T. Chayes, Alessandro Ingrosso, Carlo Lucibello, Luca Saglietti, and Riccardo Zecchina. Unreasonable effectiveness of learning neural networks: From accessible states and robust ensembles to basic algorithmic schemes. *Proceedings of the National Academy of Sciences*, 113(48):E7655–E7662, 2016. doi:10.1073/pnas.1608103113.

[5] Carlo Baldassi, Enrico M. Malatesta, and Riccardo Zecchina. Properties of the geometry of solutions and capacity of multilayer neural networks with rectified linear unit activations. *Phys. Rev. Lett.*, 123:170602, Oct 2019. doi:10.1103/PhysRevLett.123.170602.

[6] Carlo Baldassi, Riccardo Della Vecchia, Carlo Lucibello, and Riccardo Zecchina. Clustering of solutions in the symmetric binary perceptron. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(7):073303, 2020. doi:10.1088/1742-5468/ab99be.

[7] Carlo Baldassi, Fabrizio Pittorino, and Riccardo Zecchina. Shaping the learning landscape in neural networks around wide flat minima. *Proceedings of the National Academy of Sciences*, 117(1):161–170, 2020. doi:10.1073/pnas.1908636117.

[8] Carlo Baldassi, Enrico M Malatesta, Matteo Negri, and Riccardo Zecchina. Wide flat minima and optimal generalization in classifying high-dimensional gaussian mixtures. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(12):124012, 2020. doi:10.1088/1742-5468/abcd31.

[9] Carlo Baldassi, Clarissa Lauditi, Enrico M Malatesta, Gabriele Perugini, and Riccardo Zecchina. Unveiling the structure of wide flat minima in neural networks. *Physical Review Letters*, 127(27):278301, 2021. doi:10.1103/PhysRevLett.127.278301.

[10] Will Perkins and Changji Xu. Frozen 1-rsb structure of the symmetric ising perceptron. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1579–1588, 2021. doi:10.1145/3406325.3451119.

[11] Emmanuel Abbe, Shuangping Li, and Allan Sly. Proof of the contiguity conjecture and lognormal limit for the symmetric perceptron. *arXiv preprint arXiv:2102.13069*, 2021. arXiv:2102.13069.

[12] Emmanuel Abbe, Shuangping Li, and Allan Sly. Binary perceptron: efficient algorithms can find solutions in a rare well-connected cluster. *arXiv preprint arXiv:2111.03084*, 2021. arXiv:2111.03084.

[13] Yu Feng and Yuhai Tu. The inverse variance–flatness relation in stochastic gradient descent is critical for finding flat minima. *Proceedings of the National Academy of Sciences*, 118(9), 2021. doi:10.1073/pnas.2015617118.

[14] Rahul Nandkishore and David A Huse. Many-body localization and thermalization in quantum statistical mechanics. *Annu. Rev. Condens. Matter Phys.*, 6(1):15–38, 2015. doi:10.1146/annurev-conmatphys-031214-014726.

[15] Carlo Baldassi and Riccardo Zecchina. Efficiency of quantum vs. classical annealing in nonconvex learning problems. *Proceedings of the National Academy of Sciences*, 115(7):1457–1462, 2018. doi:10.1073/pnas.1711456115.

[16] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, pages 1177–1184, 2007. URL: https://papers.nips.cc/paper/2007/hash/013a006f03dbc5392effeb8f18fda755-Abstract.html.

[17] Radford M. Neal. *Bayesian Learning for Neural Networks*. Lecture Notes in Statistics. Springer-Verlag, New York, NY, 1996. doi:10.1007/978-1-4612-0745-0.

[18] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S. Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017. cite arxiv:1711.00165Comment: Pub-

lished version in ICLR 2018. 10 pages + appendix. URL: http://arxiv.org/abs/1711.00165.

[19] Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL: https://papers.nips.cc/paper/2018/hash/5a4be1fa34e62bb8a6ec6b91d2462f5a-Abstract.html.

[20] Mario Geiger, Leonardo Petrini, and Matthieu Wyart. Perspective: A phase diagram for deep learning unifying jamming, feature learning and lazy training, 2020. arXiv:2012.15110.

[21] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019. doi:10.1073/pnas.1903070116.

[22] Stefano Spigler, Mario Geiger, Stéphane d'Ascoli, Levent Sagun, Giulio Biroli, and Matthieu Wyart. A jamming transition from under- to over-parametrization affects generalization in deep learning. *Journal of Physics A: Mathematical and Theoretical*, 52(47):474001, Oct 2019. doi:10.1088/1751-8121/ab4c8b.

[23] Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics*, n/a(n/a), 2019. doi:https://doi.org/10.1002/cpa.22008.

[24] Sebastian Goldt, Marc Mézard, Florent Krzakala, and Lenka Zdeborová. Modeling the influence of data structure on learning in neural networks: The hidden manifold model. *Phys. Rev. X*, 10:041044, Dec 2020. doi:10.1103/PhysRevX.10.041044.

[25] Stèphane d'Ascoli, Maria Refinetti, Giulio Biroli, and Florent Krzakala. Double trouble in double descent : Bias and variance(s) in the lazy regime, 2020. arXiv:2003.01054.

[26] Federica Gerace, Bruno Loureiro, Florent Krzakala, Marc Mezard, and Lenka Zdeborova. Generalisation error in learning with random features and the hidden manifold model. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3452–3462. PMLR, 13–18 Jul 2020. URL: http://proceedings.mlr.press/v119/gerace20a.html.

[27] Jason W Rocks and Pankaj Mehta. Memorizing without overfitting: Bias, variance, and interpolation in over-parameterized models. *arXiv preprint arXiv:2010.13933*, 2020. arXiv:2010.13933.

[28] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021. doi:10.1088/1742-5468/ac3a74.

[29] Mario Geiger, Arthur Jacot, Stefano Spigler, Franck Gabriel, Levent Sagun, Stéphane d'Ascoli, Giulio Biroli, Clément Hongler, and Matthieu Wyart. Scaling description of generalization with number of parameters in deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(2):023401, feb 2020. doi:10.1088/1742-5468/ab633c.

[30] Levent Sagun, Utku Evci, V Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017. arXiv:1706.04454.

[31] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL: https://papers.nips.cc/paper/2018/hash/a41b3bb3e6b050b6c9067c67f663b915-Abstract.html.

[32] Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1309–1318. PMLR, 10–15 Jul 2018. URL: http://proceedings.mlr.press/v80/draxler18a.html.

[33] Marco Baity-Jesi, Levent Sagun, Mario Geiger, Stefano Spigler, Gérard Ben Arous, Chiara Cammarota, Yann LeCun, Matthieu Wyart, and Giulio Biroli. Comparing dynamics: Deep neural networks versus glassy systems. In *International Conference on Machine Learning*, pages 314–323. PMLR, 2018.

[34] S Spigler, M Geiger, S d'Ascoli, L Sagun, G Biroli, and M Wyart. A jamming transition from under- to over-parametrization affects generalization in deep learning. *Journal of Physics A: Mathematical and Theoretical*, 52(47):474001, oct 2019. doi:10.1088/1751-8121/ab4c8b.

[35] Sebastian Goldt, Bruno Loureiro, Galen Reeves, Florent Krzakala, Marc Mezard, and Lenka Zdeborova. The gaussian equivalence of generative models for learning with shallow neural networks. In Joan Bruna, Jan Hesthaven, and Lenka Zdeborova, editors, *Proceedings of the 2nd Mathematical and Scientific Machine Learning Conference*, volume 145 of *Proceedings of Machine Learning Research*, pages 426–471. PMLR, 16–19 Aug 2022. URL: https://proceedings.mlr.press/v145/goldt22a.html.

[36] Hong Hu and Yue M. Lu. Universality laws for high-dimensional learning with random features, 2020. URL: https://arxiv.org/abs/2009.07669, doi:10.48550/ARXIV.2009.07669.

[37] Marc Mézard, Giorgio Parisi, and Miguel Virasoro. *Spin glass theory and beyond: An Introduction to the Replica Method and Its Applications*, volume 9. World Scientific Publishing Company, 1987. doi:10.1142/0271.

[38] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983. doi:10.1126/science.220.4598.671.

[39] Jonathan S Yedidia, William T Freeman, Yair Weiss, et al. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, 8:236–239, 2003. URL: https://www.cs.huji.ac.il/course/2005/pmai/tirguls/TR2001-22.pdf.

[40] Carlo Baldassi, Alfredo Braunstein, Nicolas Brunel, and Riccardo Zecchina. Efficient supervised learning in networks with binary synapses. *Proceedings of the National Academy of Sciences*, 104(26):11079–11084, 2007. doi:10.1073/pnas.0700324104.

[41] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. *Advances in neural information processing systems*, 29, 2016. URL: https://papers.nips.cc/paper/2016/hash/d8330f857a17c53d217014ee776bfd50-Abstract.html.

[42] Silvio Franz and Giorgio Parisi. Recipes for metastable states in spin glasses. *Journal de Physique I*, 5(11):1401–1415, 1995. doi:10.1051/jp1:1995201.

[43] Fabrizio Pittorino, Carlo Lucibello, Christoph Feinauer, Gabriele Perugini, Carlo Baldassi, Elizaveta Demyanenko, and

Riccardo Zecchina. Entropic gradient descent algorithms and wide flat minima. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124015, dec 2021. doi:10.1088/1742-5468/ac3ae8.

[44] Fabrizio Pittorino, Antonio Ferraro, Gabriele Perugini, Christoph Feinauer, Carlo Baldassi, and Riccardo Zecchina. Deep networks on toroids: Removing symmetries reveals the structure of flat regions in the landscape geometry. *arXiv preprint arXiv:2202.03038*, 2022.

[45] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013. arXiv:1312.6120.

[46] Shengchao Liu, Dimitris Papailiopoulos, and Dimitris Achlioptas. Bad global minima exist and sgd can reach them. *Advances in Neural Information Processing Systems*, 33, 2020. arXiv:1906.02613.

[47] E Gardner. The space of interactions in neural network models. *Journal of Physics A: Mathematical and General*, 21(1):257–270, jan 1988. doi:10.1088/0305-4470/21/1/030.

[48] E Gardner and B Derrida. Optimal storage properties of neural network models. *Journal of Physics A: Mathematical and General*, 21(1):271–284, jan 1988. doi:10.1088/0305-4470/21/1/031.

[49] E Gardner and B Derrida. Three unfinished works on the optimal storage capacity of networks. *Journal of Physics A: Mathematical and General*, 22(12):1983–1994, jun 1989. doi:10.1088/0305-4470/22/12/004.

[50] Géza Györgyi. First-order transition to perfect generalization in a neural network with binary synapses. *Phys. Rev. A*, 41:7097–7100, Jun 1990. doi:10.1103/PhysRevA.41.7097.

[51] Andreas Engel and Christian Van den Broeck. *Statistical mechanics of learning*. Cambridge University Press, 2001.

[52] Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them, 2019. arXiv:1912.02178.

[53] Fabrizio Pittorino, Carlo Lucibello, Christoph Feinauer, Gabriele Perugini, Carlo Baldassi, Elizaveta Demyanenko, and Riccardo Zecchina. Entropic gradient descent algorithms and wide flat minima. In *International Conference on Learning Representations*, 2021. URL: https://openreview.net/forum?id=xjXg0bnoDmS.

[54] Carlo Baldassi. Generalization learning in a perceptron with binary synapses. *Journal of Statistical Physics*, 136(5):902–916, 2009. doi:10.1007/s10955-009-9822-1.

Appendix A: Some preliminary definitions

We denote by w_k^T the weights of a teacher that lives in a D-dimensional space ($k = 1, \dots, D$). The teacher assigns to i.i.d. standard normal random input variables ξ_k^μ (with $\mu = 1, \dots, P$) a label, via

$$y^\mu = \text{sign}\,(u^\mu) = \text{sign}\left(\frac{1}{\sqrt{D}}\sum_{k=1}^{D} w_k^T \xi_k^\mu\right). \tag{A1}$$

The student sees a projection of the patterns in an N dimensional space plus a non linearity σ. The dimensionality of the space N can either be higher or lower than the true dimension D. The projection is therefore identified by an $D \times N$ feature matrix F_{ki}, so that

$$\tilde{\xi}_i^\mu = \sigma\left(\frac{1}{\sqrt{D}}\sum_{k=1}^{D} F_{ki}\xi_k^\mu\right) \tag{A2}$$

The student then classifies the projected patterns with its weights as

$$\hat{y}^\mu = \text{sign}\,(\lambda^\mu) = \text{sign}\left(\frac{1}{\sqrt{N}}\sum_{i=1}^{N} w_i\tilde{\xi}_i^\mu\right) = \text{sign}\left(\frac{1}{\sqrt{N}}\sum_{i=1}^{N} w_i\,\sigma\left(\frac{1}{\sqrt{D}}\sum_{k=1}^{D} F_{ki}\xi_k^\mu\right)\right) \tag{A3}$$

Notice that we have denoted with u^μ and λ^μ the preactivation of pattern μ for the teacher and the student respectively. The only assumptions we make on the feature matrix are

$$\frac{1}{D}\sum_{k=1}^{D} F_{ki}^2 = 1\,, \qquad \forall i \tag{A4a}$$

$$\frac{1}{\sqrt{D}}\sum_{k=1}^{D} F_{ki}F_{kj} = O(1)\,, \qquad \forall i \neq j \tag{A4b}$$

$$s_{k_1,\dots,k_s}^{a_1,\dots,a_n} \equiv \frac{1}{\sqrt{N}}\sum_{i=1}^{N} w_i^{a_1}\dots w_i^{a_n}F_{k_1 i}\dots F_{k_s i} = O(1)\,, \qquad \forall n, s \geq 1 \tag{A4c}$$

In particular the second requirement tells us that two different sub-perceptrons are almost uncorrelated. In general, choosing the entries of the matrix F_{ki} to be random i.i.d. standard Gaussian random variables or random i.i.d. binary ones will do the job.

The partition function of the model is, therefore,

$$Z = \int \prod_i dw_i \, P_w(\boldsymbol{w}) \, e^{-\beta \sum_{\mu=1}^{P} \ell(-y^\mu \lambda^\mu)} \tag{A5}$$

where $\ell(\cdot)$ is a loss function per pattern and $P_w(\boldsymbol{w})$ represents the prior over the weights and identifies their space of definition. We will adopt a similar probability density $P_{w^T}(\boldsymbol{w}^T)$ for the teacher weights. In the following we will study analytically the non convex "binary" problem, where both the teacher and the student are ± 1; the "spherical", problem where the weights live on the sphere, is convex and has been already studied in the literature [26]. In this paper we will focus on the loss that simply counts the number of patterns in the training set whose stability $y^\mu \lambda^\mu$ is larger than a given positive margin κ

$$\ell_{NE}(-x;\kappa) = \Theta(-x+\kappa) \tag{A6}$$

where $\Theta(\cdot)$ is the Heaviside step function: $\Theta(x) = 1$ if $x > 0$ and zero otherwise. For $\kappa = 0$ this loss reduces to the one that counts the number of errors; with a slight abuse of notation we call it number of errors loss even if the margin is non-zero. In the following we will be interested in the binary weights case; we will compute the free entropy of solution in the thermodynamic limit

$$N, D, P \to \infty \qquad \text{fixing } \alpha \equiv \frac{P}{N} \text{ and } \alpha_T = \frac{P}{D} \equiv \frac{\alpha}{\alpha_D} \,. \tag{A7}$$

We will also limit ourselves to the case of random i.i.d. standard Gaussian features F_{ki}.

Appendix B: Replica Method

Introducing replicas we get

$$Z^n = \int \prod_{ia} dw_i^a \, P_w(\boldsymbol{w}^a) \, e^{-\beta \sum_{\mu=1}^{P} \sum_{a=1}^{n} \Theta(-y^\mu \lambda^{\mu a} + \kappa)} \tag{B1}$$

We now enforce the definitions of the preactivations of the teacher and the student by using delta functions

$$\mathbb{E}_{\{\boldsymbol{\xi}^\mu\}}[Z^n] = \int \prod_{ia} dw_i^a \, P_W(\boldsymbol{w}^a) \int \prod_\mu du^\mu \prod_{\mu a} d\lambda_a^\mu \, e^{-\beta \sum_{\mu=1}^{P} \sum_{a=1}^{n} \Theta\left(-\mathrm{sign}(u^\mu)\lambda_a^\mu + \kappa\right)}$$
$$\times \mathbb{E}_{\{\boldsymbol{\xi}^\mu\}} \left[\prod_\mu \delta\left(u^\mu - \frac{1}{\sqrt{D}} \sum_{k=1}^{D} w_k^T \xi_k^\mu \right) \prod_{\mu a} \delta\left(\lambda_a^\mu - \frac{1}{\sqrt{N}} \sum_{i=1}^{N} w_i^a \, \sigma\left(\frac{1}{\sqrt{D}} \sum_{k=1}^{D} F_{ki} \xi_k^\mu \right) \right) \right] . \tag{B2}$$

Notice that if the distributions of patterns and features are symmetric, we can perform the gauge transformation $\xi_k^\mu \to w_k^T \xi_k^\mu$, $F_{ki} \to w_k^T F_{ki}$, so that we can consider $w_k^T = 1, \forall k$, without loss of generality.

1. Average over the disorder: Gaussian equivalence theorem

When σ is linear, the average can be easily computed by using the integral representation of the delta function. When σ is non-linear the computation of the average is more involved. We can, however, compute the moments of the variables u^μ and λ_a^μ as defined in equation (B2). One can show that in the thermodynamic limit (A7), the moments are those of a multivariate Gaussian random variable [23, 24]. This result is equivalent to the central limit theorem that is easy to derive in the classical models without (random) feature projections [47–50] and has been renamed as "Gaussian equivalence theorem". In the following we will compute explicitly the first two moments of the random variables u^μ and λ_a^μ, and we will refer to [24] for the computation of the fourth moment.

We start defining the following useful quantities

$$\mu_0 = \int Dz \, \sigma(z) \tag{B3a}$$

$$\mu_1 = \int Dz \, z \, \sigma(z) = \int Dz \, \sigma'(z) \tag{B3b}$$

$$\mu_2 = \int Dz \, \sigma^2(z) \tag{B3c}$$

$$\mu_\star^2 = \mu_2 - \mu_1^2 - \mu_0^2 \tag{B3d}$$

where $Dz \equiv \frac{e^{-z^2/2}}{\sqrt{2\pi}} dz$. The mean of u^μ is trivial

$$\mathbb{E}_{\boldsymbol{\xi}}\left[u^\mu\right] = 0 \tag{B4}$$

whereas that of λ_a^μ is

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\xi}^\mu}\left[\lambda_a^\mu\right] &= \int \prod_{i=1}^N \frac{dv_i^\mu d\hat{v}_i^\mu}{2\pi} e^{i\sum_i \hat{v}_i^\mu v_i^\mu} \left[\frac{1}{\sqrt{N}}\sum_{i=1}^N w_i^a \sigma\left(v_i^\mu\right)\right] \prod_k \mathbb{E}_{\xi_k^\mu} e^{-i\frac{\xi_k^\mu}{\sqrt{D}}\left(\sum_i \hat{v}_i^\mu F_{ki}\right)} \\ &= \int \prod_{i=1}^N \frac{dv_i^\mu d\hat{v}_i^\mu}{2\pi} e^{i\sum_i \hat{v}_i^\mu v_i^\mu} \left[\frac{1}{\sqrt{N}}\sum_{i=1}^N w_i^a \sigma\left(v_i^\mu\right)\right] e^{-\frac{1}{2}\sum_{ij}\left(\frac{1}{D}\sum_k F_{ki}F_{kj}\right)\hat{v}_i^\mu \hat{v}_j^\mu}\end{aligned} \tag{B5}$$

We now use (A4a) and (A4b) obtaining

$$\mathbb{E}_{\boldsymbol{\xi}^\mu}\left[\lambda_a^\mu\right] = \frac{1}{\sqrt{N}}\sum_{i=1}^N w_i^a \int \frac{dv_\mu d\hat{v}_\mu}{2\pi} e^{i\hat{v}_\mu v_\mu} \sigma\left(v_\mu\right) e^{-\frac{\hat{v}_\mu^2}{2}} = \mu_0 \frac{1}{\sqrt{N}}\sum_i w_i^a \,. \tag{B6}$$

The second moment of u^μ is

$$\mathbb{E}_{\boldsymbol{\xi}^\mu}\left[u_\mu^2\right] = \frac{1}{D}\sum_{k=1}^D \left(w_k^T\right)^2 = 1 \tag{B7}$$

whereas that of λ_a^μ is

$$\mathbb{E}_{\boldsymbol{\xi}^\mu}\left[\lambda_\mu^a \lambda_\mu^b\right] = \int \prod_{i=1}^N \frac{dv_i^\mu d\hat{v}_i^\mu}{2\pi} e^{i\sum_i \hat{v}_i^\mu v_i^\mu} \left[\frac{1}{N}\sum_{ij} w_i^a w_j^b \sigma\left(v_i^\mu\right)\sigma\left(v_j^\mu\right)\right] e^{-\frac{1}{2}\sum_{ij}\left(\frac{1}{D}\sum_k F_{ki}F_{kj}\right)\hat{v}_i^\mu \hat{v}_j^\mu} \,. \tag{B8}$$

Now we split $i = j$ and $i \neq j$ contributions. Because of (A4b), $\frac{1}{D}\sum_k F_{ki}F_{kj}$ with $i \neq j$ is of order $1/\sqrt{D}$; we can therefore expand the exponential. We have

$$\begin{aligned} e^{i\hat{v}_i^\mu v_i^\mu + i\hat{v}_j^\mu v_j^\mu - \left(\frac{1}{D}\sum_k F_{ki}F_{kj}\right)\hat{v}_i^\mu \hat{v}_j^\mu} &\simeq e^{i\hat{v}_i^\mu v_i^\mu + i\hat{v}_j^\mu v_j^\mu}\left[1 - \frac{1}{2}\left(\frac{1}{D}\sum_k F_{ki}F_{kj}\right)\hat{v}_i^\mu \hat{v}_j^\mu\right] \\ &= \left[1 + \left(\frac{1}{D}\sum_k F_{ki}F_{kj}\right)\frac{d}{dv_i^\mu}\frac{d}{dv_j^\mu}\right] e^{i\hat{v}_i^\mu v_i^\mu + i\hat{v}_j^\mu v_j^\mu}\end{aligned} \tag{B9}$$

so that, performing the integrals we have

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\xi}^\mu}\left[\lambda_\mu^a \lambda_\mu^b\right] &= \mu_2 \frac{1}{N}\sum_i w_i^a w_i^b + \frac{1}{N}\sum_{i\neq j} w_i^a w_j^b \int Dv_i^\mu Dv_j^\mu \sigma\left(v_i^\mu\right)\sigma\left(v_j^\mu\right)\left[1 + \left(\frac{1}{D}\sum_k F_{ki}F_{kj}\right) v_i^\mu v_j^\mu\right] \\ &= \left(\mu_2 - \mu_1^2 - \mu_0^2\right)\frac{1}{N}\sum_i w_i^a w_i^b + \frac{\mu_0^2}{N}\sum_i w_i^a \sum_j w_j^b + \mu_1^2 \frac{1}{D}\sum_{k=1}^D s_k^a s_k^b\end{aligned} \tag{B10}$$

where we have defined the "projected" weights s_k^a as

$$s_k^a \equiv \frac{1}{\sqrt{N}}\sum_{i=1}^N F_{ki} w_i^a \,. \tag{B11}$$

The covariance is therefore

$$\mathbb{E}_{\boldsymbol{\xi}^\mu}\left[\lambda_\mu^a \lambda_\mu^b\right] - \mathbb{E}_{\boldsymbol{\xi}^\mu}\left[\lambda_\mu^a\right]\mathbb{E}_{\boldsymbol{\xi}^\mu}\left[\lambda_\mu^b\right] = \mu_\star^2 \frac{1}{N}\sum_i w_i^a w_i^b + \mu_1^2 \frac{1}{D}\sum_{k=1}^D s_k^a s_k^b \,. \tag{B12}$$

We also define the "projected" teacher weights as

$$s_i^T \equiv \frac{1}{D}\sum_{k=1}^D F_{ki} w_k^T \,. \tag{B13}$$

Using again assumptions (A4a) and (A4b) we get for the cross term

$$\begin{aligned}\mathbb{E}_{\xi^\mu}\left[u_\mu \lambda_\mu^a\right] &= \int \prod_i \frac{dv_i^\mu d\hat{v}_i^\mu}{2\pi}\frac{du_\mu d\hat{u}_\mu}{2\pi} e^{i\sum_i \hat{v}_i^\mu v_i^\mu + i\hat{u}_\mu u_\mu}\left[\frac{u_\mu}{\sqrt{N}}\sum_i w_i^a \sigma\left(v_i^\mu\right)\right] e^{-\frac{\hat{u}_\mu^2}{2}-\frac{1}{2}\sum_{ij}\left(\frac{1}{D}\sum_k F_{ki}F_{kj}\right)\hat{v}_i^\mu \hat{v}_j^\mu - \hat{u}_\mu \sum_i \hat{v}_i^\mu s_i^T} \\ &= \frac{1}{\sqrt{N}}\sum_i w_i^a \int \frac{dv_\mu d\hat{v}_\mu}{2\pi}\frac{du_\mu d\hat{u}_\mu}{2\pi} e^{i\hat{v}_\mu v_\mu + i\hat{u}_\mu u_\mu}\left[u_\mu \sigma\left(v_\mu\right)\right] e^{-\frac{\hat{u}_\mu^2}{2}-\frac{\hat{v}_\mu^2}{2}-\hat{u}_\mu \hat{v}_\mu s_i^T} \\ &= \mu_1 \frac{1}{\sqrt{N}}\sum_i w_i^a s_i^T = \mu_1 \frac{1}{D}\sum_{k=1}^D s_k^a w_k^T \,.\end{aligned} \tag{B14}$$

The distribution of random variables u^μ and λ_a^μ therefore can be written as a multivariate Gaussian. The final result reads

$$P\left(u^\mu, \{\lambda_a^\mu\}\right) = \frac{1}{\sqrt{2\pi \det \Sigma}} e^{-\frac{1}{2}\sum_{\gamma,\delta=0}^n (\Upsilon_\gamma^\mu - \rho_\gamma)(\Sigma^{-1})_{\gamma\delta}(\Upsilon_\delta^\mu - \rho_\delta)} \tag{B15}$$

where $\Upsilon_0^\mu \equiv u^\mu$ and $\Upsilon_a^\mu \equiv \lambda_a^\mu$, $\forall a = 1, \dots, n$. The mean vector is $\rho_0 = 0$ and $\rho_a = \frac{\mu_0}{\sqrt{N}}\sum_i w_i^a$ for $a = 1, \dots, n$; the covariance is

$$\Sigma \equiv \begin{pmatrix} 1 & M_a \\ M_a & Q_{ab} \end{pmatrix} \tag{B16}$$

where

$$M_a = \mu_1 \frac{1}{D}\sum_{k=1}^D s_k^a w_k^T \equiv \mu_1 r_a \tag{B17a}$$

$$Q_{ab} = \mu_\star^2 \frac{1}{N}\sum_{i=1}^N w_i^a w_i^b + \mu_1^2 \frac{1}{D}\sum_{k=1}^D s_k^a s_k^b \equiv \mu_\star^2 q_{ab} + \mu_1^2 p_{ab} \tag{B17b}$$

The average over the replicated partition function therefore takes the form

$$\mathbb{E}_{\{\xi^\mu\}}[Z^n] = \int \prod_{ia} dw_i^a \, P_w(\boldsymbol{w}^a) \int \prod_\mu du^\mu \prod_{\mu a} d\lambda_a^\mu \, e^{-\beta \sum_{\mu=1}^P \sum_{a=1}^n \Theta\left(-\mathrm{sign}(u^\mu)\lambda_a^\mu + \kappa\right)} P\left(u^\mu, \{\lambda_a^\mu\}\right) \tag{B18}$$

or, equivalently

$$\begin{aligned}\mathbb{E}_{\{\xi^\mu\}}[Z^n] = \int \prod_{ia} dw_i^a \, P_w(\boldsymbol{w}^a) \int \prod_\mu \frac{du^\mu d\hat{u}^\mu}{2\pi} \prod_{\mu a} \frac{d\lambda_a^\mu d\hat{\lambda}_a^\mu}{2\pi} \, e^{-\beta \sum_{\mu=1}^P \sum_{a=1}^n \Theta\left(-\mathrm{sign}(u^\mu)\lambda_a^\mu + \kappa\right)} \\ \times \prod_\mu e^{iu^\mu \hat{u}^\mu + i\sum_a (\lambda_a^\mu - \rho_a)\hat{\lambda}_a^\mu - \frac{(\hat{u}^\mu)^2}{2} - \frac{1}{2}\sum_{ab} Q_{ab}\hat{\lambda}_a^\mu \hat{\lambda}_b^\mu - \sum_a M_a \hat{u}^\mu \hat{\lambda}_a^\mu} \,.\end{aligned} \tag{B19}$$

Therefore the analytical expression of the average over patterns is similar to the one of the non-overparameterized teacher-student scenario [49, 50], except for two important differences. Firstly, M_a, i.e. the overlap between the teacher and the student with replica index a has a different definition (see eq. (B17a)) since the two architectures live in spaces with different dimensions. Secondly, also the definition of the overlap matrix Q_{ab} changes (see eq. (B17b)). In particular notice that an additional matrix of overlaps p_{ab} appears; this represents the overlap between the projection (in the teacher space) of the weights of two students with replica indexes a and b.

Notice that equation (B18) can be obtained starting from (B2), also by using the following mapping (Gaussian covariate model)

$$\tilde{\xi}_i^\mu = \sigma\left(\frac{1}{\sqrt{D}}\sum_{k=1}^D F_{ki}\xi_k^\mu\right) = \mu_0 + \frac{\mu_1}{\sqrt{D}}\sum_{k=1}^D F_{ki}\xi_k^\mu + \mu_\star \eta_i^\mu \tag{B20}$$

where $\eta_i \sim \mathcal{N}(0,1)$ are i.i.d. standard Gaussian random variables. This means that in the thermodynamic limit (A7), the statistical properties of the random feature model are equivalent to a Gaussian covariate model, in which each projected pattern $\tilde{\boldsymbol{\xi}}^\mu$ is a *linear* combination of the patterns components ξ_k^μ plus noise. The strength of the noise depends on the degree of non-linearity of the activation function σ. This was already noticed in [23].

In the following we will limit ourselves to the case $\sigma(x) = \mathrm{sign}(x)$, but our analytical results are valid to the class of functions $\sigma(\cdot)$ for which $\mu_0 = 0$; this will also impose $\rho_a = 0$, reducing the number of terms in the calculations.

2. Average over features and introduction of the order parameters

Inserting the definition of the projected weights (B11) using delta functions it becomes easy to perform the average over random Gaussian features. We get a terms of the following form

$$\int \prod_{ka} \frac{ds_k^a d\hat{s}_k^a}{2\pi} e^{i\sum_{ka} s_k^a \hat{s}_k^a} \prod_{ki} \mathbb{E}_{F_{ki}} \left[e^{-i\frac{F_{ki}}{\sqrt{N}} \sum_a \hat{s}_k^a w_i^a} \right] = \int \prod_{ka} \frac{ds_k^a d\hat{s}_k^a}{2\pi} e^{i\sum_{ka} s_k^a \hat{s}_k^a - \frac{1}{2}\sum_{ab,k} \hat{s}_k^a \hat{s}_k^b \left(\frac{1}{N}\sum_i w_i^a w_i^b\right)} . \tag{B21}$$

Next we can safely impose the definitions of the order parameters

$$q_{ab} \equiv \frac{1}{N}\sum_i w_i^a w_i^b , \qquad p_{ab} \equiv \frac{1}{D}\sum_k s_k^a s_k^b , \qquad r_a \equiv \frac{1}{D}\sum_k s_k^a w_k^T , \tag{B22}$$

Notice that $q_{aa} = 1$ since we have binary weights. Denoting by $\overline{\cdots}$ the average over both patterns and random features, the final result reads

$$\overline{Z^n} = \int \prod_{a<b} \frac{dq_{ab} d\hat{q}_{ab}}{2\pi} \prod_{a\leq b} \frac{dp_{ab} d\hat{p}_{ab}}{2\pi} \prod_a \frac{dr_a d\hat{r}_a}{2\pi} e^{N\phi} \tag{B23}$$

where

$$\phi = -\sum_{a<b} q_{ab}\hat{q}_{ab} - \frac{\alpha_D}{2}\sum_{ab} p_{ab}\hat{p}_{ab} - \alpha_D \sum_a r_a \hat{r}_a + G_{SS} + \alpha_D G_{SE} + \alpha G_E \tag{B24a}$$

$$G_{SS} = \ln \int \prod_a dw_a \, P_w(w_a) \, e^{\frac{1}{2}\sum_{a\neq b} \hat{q}_{ab} w_a w_b} \tag{B24b}$$

$$G_{SE} = \ln \int \prod_a \frac{ds_a d\hat{s}_a}{2\pi} e^{i\sum_a s_a \hat{s}_a + \sum_a \hat{r}_a s_a + \frac{1}{2}\sum_{ab} \hat{p}_{ab} s_a s_b - \frac{1}{2}\sum_{ab} q_{ab} \hat{s}_a \hat{s}_b} \tag{B24c}$$

$$G_E = \ln \int \prod_a \frac{d\lambda_a d\hat{\lambda}_a}{2\pi} \frac{du d\hat{u}}{2\pi} e^{iu\hat{u} + i\sum_a \lambda_a \hat{\lambda}_a - \beta \sum_a \Theta(-\mathrm{sign}(u)\lambda_a + \kappa) - \frac{\hat{u}^2}{2} - \frac{1}{2}\sum_{ab} Q_{ab}\hat{\lambda}_a\hat{\lambda}_b - \hat{u}\sum_a M_a \hat{\lambda}_a} \tag{B24d}$$

and M_a, Q_{ab} are defined in terms of q_{ab}, p_{ab}, r_a in (B17), and as usual $\alpha_D \equiv D/N$. Notice that G_{SS} is the usual "entropic" contribution in a perceptron storing random patterns, whereas G_E is the usual "energetic" contribution in the teacher student setting. G_{SE} is a new term that we call "entropic-energetic" since it depends on both overlaps q_{ab} and conjugated ones $\hat{p}_{ab}$, $\hat{r}_a$. Notice that G_{SE} can be computed analytically, since it contains only Gaussian integrals. It reads

$$G_{SE} = -\frac{1}{2}\ln\det\left(\mathbb{I} - q\hat{p}\right) + \frac{1}{2}\sum_{ab} \hat{r}_a \left[\left(\mathbb{I} - q\hat{p}\right)^{-1} q\right]_{ab} \hat{r}_b . \tag{B25}$$

3. Replica-Symmetric ansatz

We impose a Replica-Symmetric (RS) ansatz for the order parameters: $q_{ab} = \delta_{ab} + q\,(1-\delta_{ab})$, $\hat{q}_{ab} = \hat{q}\,(1-\delta_{ab})$; $p_{ab} = p_d\delta_{ab} + p\,(1-\delta_{ab})$, $\hat{p}_{ab} = -\hat{p}_d\delta_{ab} + \hat{p}\,(1-\delta_{ab})$ and $r_a = r$, $\hat{r}_a = \hat{r}$.

We obtain

$$\mathcal{G}_{SS} \equiv \frac{\hat{q}}{2} + \lim_{n\to 0}\frac{G_{SS}}{n} = \int Dx \, \ln 2\cosh\left(\sqrt{\hat{q}}x\right) \tag{B26a}$$

$$\mathcal{G}_{SE} \equiv \lim_{n\to 0}\frac{G_{SE}}{n} = -\frac{1}{2}\frac{q}{1-q} - \frac{1}{2}\ln\left[1 + (\hat{p}+\hat{p}_d)(1-q)\right] + \frac{1}{2}\frac{(\hat{p}+\hat{r}^2)(1-q) + \frac{q}{1-q}}{1 + (\hat{p}+\hat{p}_d)(1-q)} \tag{B26b}$$

$$\mathcal{G}_E \equiv \lim_{n\to 0}\frac{G_E}{n} = 2\int Dx \, H\left(-\frac{Mx}{\sqrt{Q-M^2}}\right) \ln H_\beta\left(\frac{\kappa - \sqrt{Q}x}{\sqrt{Q_d - Q}}\right) \tag{B26c}$$

where $M \equiv \mu_1 r$, $Q \equiv \mu_\star^2 q + \mu_1^2 p$, $Q_d \equiv \mu_\star^2 + \mu_1^2 p_d$. We have also defined

$$H(x) \equiv \frac{1}{2}\mathrm{Erfc}\left(\frac{x}{\sqrt{2}}\right) , \tag{B27a}$$

$$H_\beta(x) \equiv e^{-\beta} + \left(1 - e^{-\beta}\right) H(x) . \tag{B27b}$$

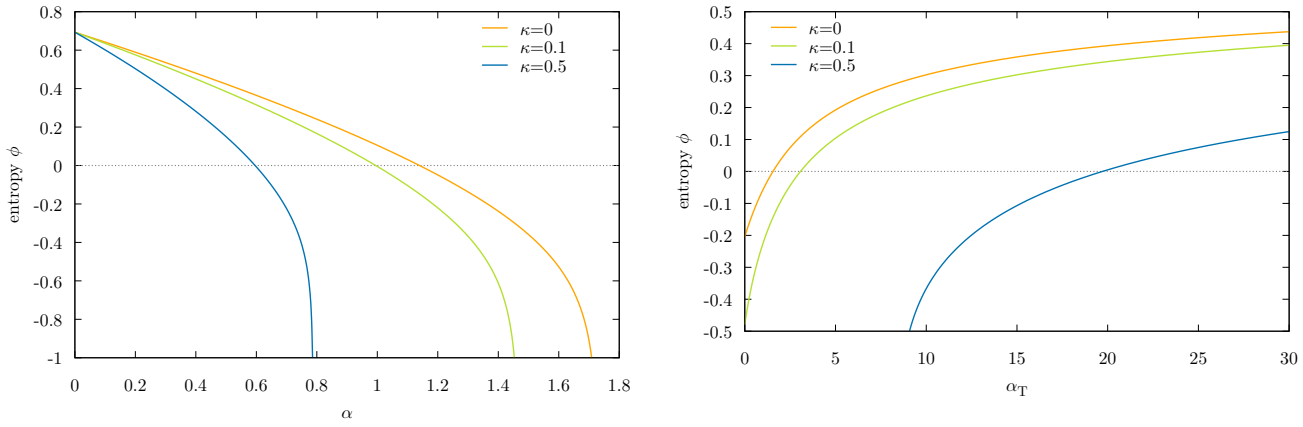

Figure 8. Plot of the entropy as a function of α for fixed $\alpha_T = 3$ (left panel) and as a function of α_T for fixed $\alpha = 1$ (right panel) for different values of the margin κ.

The free entropy of the system is

$$\phi = -\frac{\hat{q}}{2}(1-q) + \frac{\alpha_D}{2}\left(p_d\hat{p}_d + p\hat{p}\right) - \alpha_D r\hat{r} + \mathcal{G}_{SS} + \alpha_D\mathcal{G}_{SE} + \alpha\mathcal{G}_E \,. \tag{B28}$$

The order parameters q, p, p_d, r, $\hat{q}$, $\hat{p}$, $\hat{p}_d$, $\hat{r}$ are found by saddle point equations

$$\begin{aligned} q &= 1 - 2\frac{\partial\mathcal{G}_{SS}}{\partial\hat{q}}\,, & p &= -2\frac{\partial\mathcal{G}_{SE}}{\partial\hat{p}}\,, & p_d &= -2\frac{\partial\mathcal{G}_{SE}}{\partial\hat{p}_d}\,, & r &= \frac{\partial\mathcal{G}_{SE}}{\partial\hat{r}}\,, \\ \hat{q} &= -2\alpha_D\frac{\partial\mathcal{G}_{SE}}{\partial q} - 2\alpha\frac{\partial\mathcal{G}_E}{\partial q}\,, & \hat{p} &= -2\alpha_T\frac{\partial\mathcal{G}_E}{\partial p}\,, & \hat{p}_d &= -2\alpha_T\frac{\partial\mathcal{G}_E}{\partial p_d}\,, & \hat{r} &= \alpha_T\frac{\partial\mathcal{G}_E}{\partial r}\,, \end{aligned} \tag{B29}$$

As in the simple binary perceptron [1], the "interpolation threshold" or critical capacity is found by looking to the value of α for which the RS free entropy vanishes. We show in 8 the behaviour of the entropy (i.e. the free entropy in the $\beta \to \infty$ limit) as a function of α (for a fixed value of α_T) and α_T (for a fixed value of α) for different margins.

a. Generalization error

To compute the generalization error, we extract a new pattern $\boldsymbol{\xi}^\star$ and label $y^\star$ and we compute the average number of errors. Denoting by $\langle\cdot\rangle$ the ensemble average, we have

$$\begin{aligned} \epsilon_g &\equiv \langle \mathbb{E}_{\boldsymbol{\xi}^\star}\Theta\left(-y^\star\hat{y}^\star\right)\rangle \\ &= \int du d\lambda\,\Theta\left(-u\lambda\right)\mathbb{E}_{\boldsymbol{\xi}^\star}\left\langle\delta\left(u - \frac{1}{\sqrt{D}}\sum_{k=1}^{D} w_k^T\xi_k^\star\right)\delta\left(\lambda - \frac{1}{\sqrt{N}}\sum_{i=1}^{N} w_i\,\sigma\left(\frac{1}{\sqrt{D}}\sum_{k=1}^{D} F_{ki}\xi_k^\star\right)\right)\right\rangle \\ &= \int \frac{du d\hat{u}}{2\pi}\frac{d\lambda d\hat{\lambda}}{2\pi}\,\Theta\left(-u\lambda\right)e^{iu\hat{u}+i\lambda\hat{\lambda}-\frac{\hat{u}^2}{2}-\frac{1}{2}Q_d\hat{\lambda}^2-M\hat{u}\hat{\lambda}} = 2\int_0^\infty Du\,H\left(-\frac{Mu}{\sqrt{Q_d - M^2}}\right). \end{aligned} \tag{B30}$$

Performing the last integral we finally obtain

$$\epsilon_g = \frac{1}{\pi}\arccos\left(\frac{M}{\sqrt{Q_d}}\right), \tag{B31}$$

which is nothing but the standard formula of the generalization error for the classical teacher-student problem, but written in terms of the "projected" overlap with the teacher M and the "projected" norm of the weights Q_d.

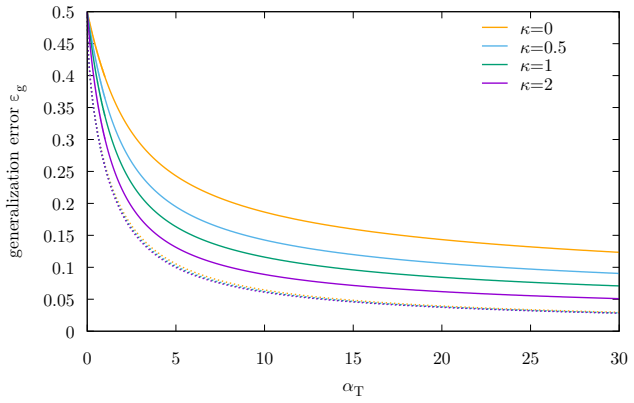

Figure 9. Generalization error in the large overparameterization limit ($\alpha \to 0$) versus $\alpha_T = P/D$ for different values of the margin κ. The dotted lines represent the corresponding generalization error of the barycenter of typical solutions (see section D)

b. *Storage problem*

When the dimension of the teacher is much larger than the number of patterns in the training set $\alpha_T = \frac{P}{D} \ll 0$, the problem is as if the student sees random patterns. Indeed the saddle point equations (B29) reduce in the limit $\alpha_T \to 0$ to

$$\begin{aligned} q &= 1 - 2\frac{\partial \mathcal{G}_{SS}}{\partial \hat{q}}\,, \quad p = q\,, \quad p_d = 1\,, \quad r = 0\,, \\ \hat{q} &= -2\alpha\frac{\partial \mathcal{G}_E}{\partial q}\,, \quad \hat{p} = 0\,, \quad \hat{p}_d = 0\,, \quad \hat{r} = 0\,. \end{aligned} \tag{B32}$$

Therefore $Q = \left(\mu_\star^2 + \mu_1^2\right) q$, $Q_d = \mu_\star^2 + \mu_1^2$ and the free entropy reduces to

$$\phi = -\frac{\hat{q}}{2}(1-q) + \mathcal{G}_{SS} + \alpha \mathcal{G}_E\,. \tag{B33}$$

with

$$\mathcal{G}_{SS} = \int Dx\ \ln\cosh\left(\sqrt{\hat{q}}x\right) \tag{B34a}$$

$$\mathcal{G}_E = \int Dx\ \ln H_\beta\left(-\sqrt{\frac{q}{1-q}}x\right)\,. \tag{B34b}$$

This is exactly the free entropy of the storage problem as derived by Gardner [47, 48]. Notice that this limit is achieved independently of the non-linearity $\sigma(\cdot)$ used.

c. *Overparameterization limit*

Here we want to address analytically the infinite overparameterization limit, i.e. $\alpha \to 0$ for a fixed value of α_T. In this limit also $\alpha_D = \frac{\alpha}{\alpha_T}$ is vanishing, therefore from saddle point equations (B29) we see that $\hat{q} \to 0$ and consequently $q \to 0$, meaning that typical solutions are uncorrelated in the space of the students. However there is still information about the teacher, so the corresponding overlap in the space of the teacher is not zero. Furthermore we can eliminate all other conjugated parameters $\hat{p}$, $\hat{p}_d$ and $\hat{r}$ by expressing them in terms of the other order parameters. The entropy can be written as

$$\phi \simeq \ln 2 + \frac{\alpha}{\alpha_T}\delta\phi \tag{B35}$$

where

$$\delta\phi = \frac{1}{2}\left(1 - p_d - \frac{r^2}{p_d - p}\right) + \mathcal{G}_{SE} + \alpha_T \mathcal{G}_E \tag{B36a}$$

$$\mathcal{G}_{SE} = \frac{1}{2}\left(\frac{p}{p_d - p} + \ln(p_d - p)\right) \tag{B36b}$$

$$\mathcal{G}_E = 2\int Dx\, H\left(-\frac{rx}{\sqrt{p - r^2}}\right) \ln H_\beta\left(\frac{\kappa - \sqrt{p}x}{\sqrt{\frac{\mu_\star^2}{\mu_1^2} + p_d - p}}\right) \tag{B36c}$$

Notice that $\mathcal{G}_E$ apart for the dependence on $\sigma(\cdot)$ is identical to the energetic term of the classical teacher-student problem. Instead $\mathcal{G}_{SE}$ is identical to the entropic term of a spherical perceptron storing random patterns.

By solving the corresponding saddle point equations, we are able to numerically compute the plateau of the generalization error; this is plotted as a function of α_T in Fig. 9 for different values of the margin.

d. Stability distribution

The stability of the weights $\boldsymbol{w}$ given a pattern ξ^μ and its corresponding label y^μ is defined as

$$\Delta^\mu \equiv y^\mu \lambda^\mu = \frac{y^\mu}{\sqrt{N}} \sum_{i=1}^{N} w_i \sigma\left(\frac{1}{\sqrt{D}} \sum_{k=1}^{D} F_{ki} \xi_k^\mu\right). \tag{B37}$$

We restrict for simplicity to the case of zero margin κ. Once the saddle point equations (B29) are solved, we can compute the stability distribution

$$P(\Delta) \equiv \langle \delta\left(\Delta - \Delta^\mu\right)\rangle = \overline{\frac{1}{Z}\int \prod_i dw_i\, P_w(\boldsymbol{w})\, e^{-\beta \sum_{\mu=1}^{P} \Theta(-y^\mu \hat{y}^\mu)}\, \delta\left(\Delta - \Delta^\mu\right)} \tag{B38}$$

using the replica method. We obtain

$$P(\Delta) = \lim_{n\to 0} \int \frac{du d\hat{u}}{2\pi} \prod_a \frac{d\lambda_a d\hat{\lambda}_a}{2\pi}\, e^{iu\hat{u} + i\sum_a \lambda_a \hat{\lambda}_a - \frac{\hat{u}^2}{2} - \beta \sum_a \Theta(-u\lambda_a) - \frac{1}{2}\sum_{ab} Q_{ab} \hat{\lambda}_a \hat{\lambda}_b - \hat{u}\sum_a M_a \hat{\lambda}_a}\, \delta\left(\Delta - \mathrm{sign}(u)\lambda_1\right), \tag{B39}$$

that in the RS ansatz reduces to

$$P(\Delta) = \frac{2e^{-\beta\Theta(-\Delta)}}{\sqrt{Q_d - Q}} \int Dx\, G\left(\frac{\Delta - \sqrt{Q}x}{\sqrt{Q_d - Q}}\right) \frac{H\left(-\frac{Mx}{\sqrt{Q - M^2}}\right)}{H_\beta\left(-\sqrt{\frac{Q}{Q_d - Q}}x\right)}, \tag{B40}$$

where $G(x) \equiv \frac{e^{-x^2/2}}{\sqrt{2\pi}}$.

In Fig. 12 we show the distribution of stabilities of typical solutions for different values of α_T. The maximum of the distribution appears to be near the origin, especially for low values of α_T; this has been already noted to be a characteristic of "sharp" solutions in one-layer [51] and two-layer neural networks [5] in contrast to "flat" or high local entropy ones, for which it is usually noted that a low probability of having a small stability (i.e. targeting high local entropy regions induces a soft margin).

Appendix C: Agreement with numerical simulations

We have performed some numerical simulations in order to corroborate analytical results of typical solutions. We have used very simple algorithms that have the Gibbs distribution as stationary probability measure such as the zero-temperature Monte Carlo (MCT0) and the Simulated Annealing algorithm (SA) [38].

Both algorithms have difficulties in finding solutions since the dominant set of minima consist of isolated point like clusters with vanishing internal entropy. Nevertheless, for finite size systems, in the highly overparameterized regime those algorithms

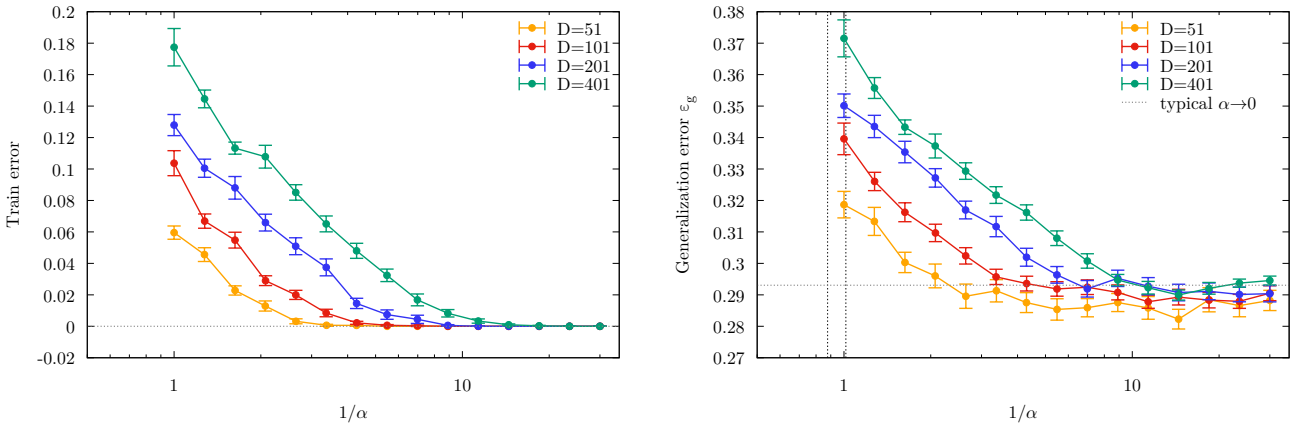

Figure 10. Train (left panel) and test error (right panel) of the SA algorithm as a function of $1/\alpha$. Different colors represent the result of the simulations with different values of D, while maintaining fixed $\alpha_T = 3$. The points are averages over 40 samples for $D = 51, 101$, 20 samples for $D = 201, 401$, and 2 independent runs per sample. Approaching the thermodynamic limit, it is harder to find a solution. Nonetheless for a fixed system size we can reach zero or sufficiently small the training errors in the overparameterized regime. The corresponding generalization error matches the replica theory result (dotted horizontal line).

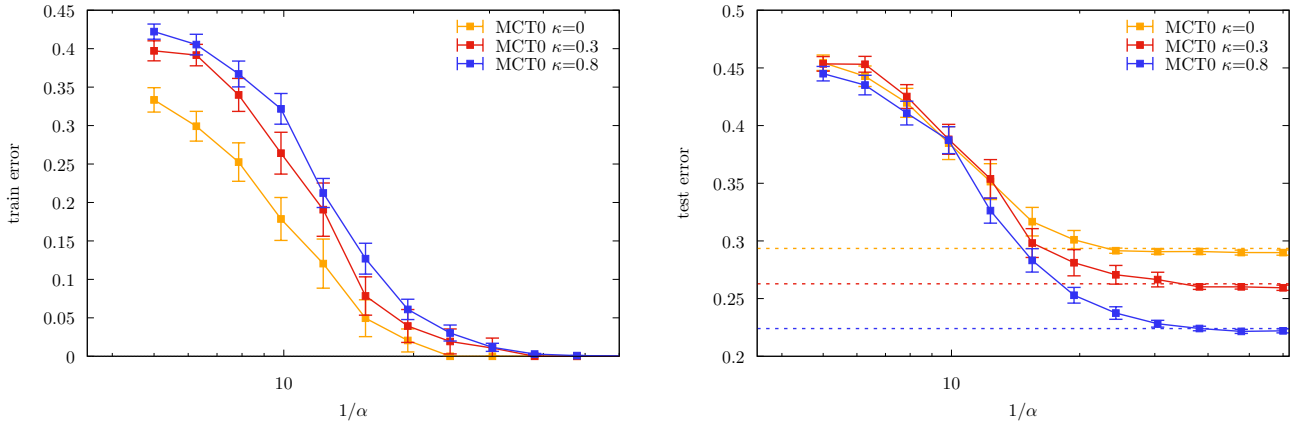

Figure 11. Train (left panel) and test error (right panel) of zero-temperature Monte Carlo algorithm as a function of $1/\alpha$ for different values of margins. In the simulations we fixed $D = 201$ and $P = 603$ i.e. $\alpha_T = 3$, and we ran the algorithm for a fixed number of sweeps (200). Points are averages over 10 samples and 2 random restarts for each sample. We show also in dashed the analytical predictions coming from replica theory.

are able to find solutions (see left panel of Fig. 10 for the behaviour of the train error as a function of $1/\alpha$ obtained by using the SA algorithm for different system sizes). The statistical properties of those solutions are in agreement with the predictions of the replica theory (see right panel of Fig. 10). The same results hold for MCT0: in Fig. 11 we show that the generalization error obtained by replica theory for several values of the margin is in perfect agreement with that obtained by numerical simulations. Notice how increasing the margin makes finding the solution more difficult (since they are rarer); however when solutions start to be accessible, increasing the margin increases the accuracy on the test set. This is consistent with the fact that even if high margin solutions lie in flat regions of the loss landscape (see main text) they are still isolated between each other. Finally in Fig. 12 we show the agreement between the analytical (see equation (B40)) and numerical distribution of stabilities for different values of α_T obtained by MCT0.

Appendix D: Bayesian generalization error

We want to compute the average probability that the ensemble of students generalizes correctly with respect to the teacher, i.e. the probability that the average of the outputs of the students on a random new pattern has different sign than that given by the

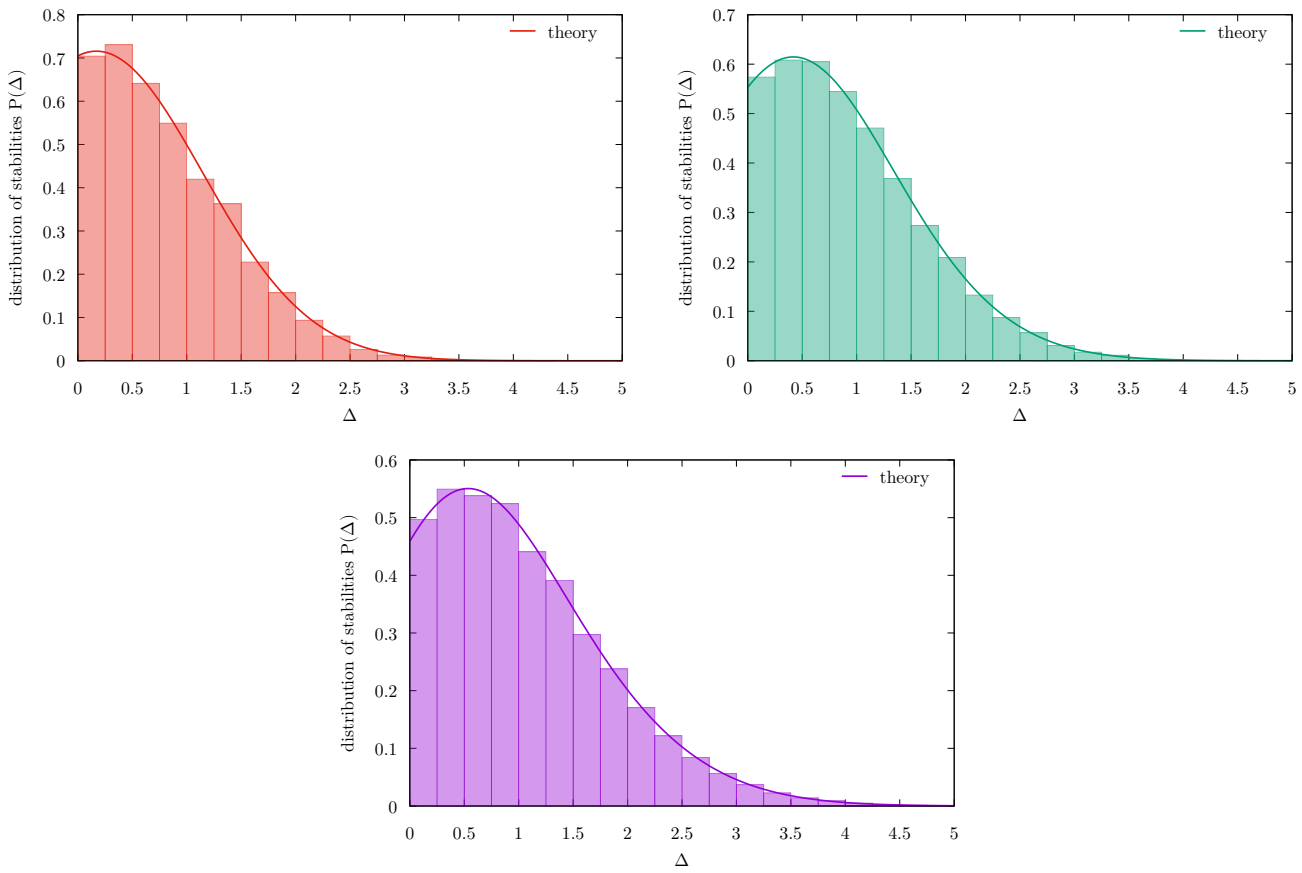

Figure 12. Histograms of the distribution of stabilities for zero temperature Monte Carlo solutions for $\alpha_T = 1.4$ (top left), $\alpha_T = 5$ (top right) and $\alpha_T = 10$ (bottom). The algorithm was ran until a solution is found or a maximum number of sweeps (200) is reached. The histograms are averaged over 40 samples and 2 random restarts for each sample. The full line is the replica prediction of the distribution for typical solutions in the thermodynamic limit given by equation (B40).

teacher:

$$\epsilon_g^B = \mathbb{E}_{\boldsymbol{\xi}^\star} \overline{\Theta\left(-y^\star \langle \hat{y}^\star \rangle_{\boldsymbol{w}|\{\boldsymbol{\xi}^\mu\}}\right)} \tag{D1}$$

In the previous equation $\boldsymbol{\xi}^\star$, $y^\star$ are respectively a test pattern and its corresponding label (computed using equation (A1)); $\hat{y}^\star$ is the output of the student given input $\boldsymbol{\xi}^\star$ as in (A3); $\overline{\cdots}$ is the average over training patterns $\{\boldsymbol{\xi}^\mu\}$ and random features F, and finally $\langle\cdot\rangle_{\boldsymbol{w}|\{\boldsymbol{\xi}^\mu\}}$ is the average over the posterior distribution, namely the average over the probability distribution of student weights given training data

$$\langle \hat{y}^\star \rangle_{\boldsymbol{w}|\{\boldsymbol{\xi}^\mu\}} = \frac{1}{Z} \int \prod_i dw_i \, P_w(\boldsymbol{w}) \, \mathrm{sign}\left(\frac{1}{\sqrt{N}} \sum_i w_i \sigma\left(\frac{1}{\sqrt{D}} \sum_k F_{ki} \xi_k^\star \right) \right) \times e^{-\beta \sum_\mu \Theta\left[-\left(\frac{1}{\sqrt{D}} \sum_k \xi_k^\mu \right) \left(\frac{1}{\sqrt{N}} \sum_i w_i \sigma\left(\frac{1}{\sqrt{D}} \sum_k F_{ki} \xi_k^\mu \right) \right) \right]} \tag{D2}$$

We have used again $w_k^T = 1$ without loss of generality. We start the computation by extracting the definitions $\hat{y}^\star$ by using delta functions

$$\epsilon_g^B = \int \frac{dx d\hat{x}}{2\pi} \, e^{ix\hat{x}} \, \mathbb{E}_{\boldsymbol{\xi}^\star} \Theta\left[-\left(\frac{1}{\sqrt{D}} \sum_k \xi_k^\star \right) x \right] \overline{e^{-i\hat{x} \langle y^\star \rangle_{\boldsymbol{w}|\{\boldsymbol{\xi}^\mu\}}}} \, . \tag{D3}$$

Next, we use the following identity

$$e^{-i\hat{x} \langle y^\star \rangle_{\boldsymbol{w}|\{\boldsymbol{\xi}^\mu\}}} = \sum_{s=0}^{\infty} \frac{(-i\hat{x})^s}{s!} \langle y^\star \rangle_{\boldsymbol{w}|\{\boldsymbol{\xi}^\mu\}}^s \, , \tag{D4}$$

which enables us to perform the average over training patterns and random features. Two replica indexes are needed: the first one is due to the factor Z^{-1} in equation (D2) which can be re-written as $\frac{1}{Z} = \lim_{n\to 0} Z^{n-1}$; the second one is due to the power s in (D4). We use as before $a, b \in [n]$, whereas indexes $l, m \in [s]$ for the new replicas. We have

$$\left\langle y^\star \right\rangle^s_{\boldsymbol{w}\,|\,\{\xi^\mu\}} = \lim_{n\to 0} \int \prod_l \frac{d h_l d\hat{h}_l}{2\pi} e^{i\sum_l h_l \hat{h}_l} \prod_l \mathrm{sign}(h_l) \int \prod_{ial} dw_i^{la} P_w(\boldsymbol{w}^{la}) \\ \times e^{-\beta \sum_{\mu a} \Theta\left[-\left(\frac{1}{D}\sum_k \xi_k^\mu\right)\left(\frac{1}{\sqrt{N}}\sum_i w_i^{la}\sigma\left(\frac{1}{\sqrt{D}}\sum_k F_{ki}\xi_k^\mu\right)\right)\right] - \frac{i}{\sqrt{N}}\sum_l \hat{h}_l \sum_i w_i^{l1}\sigma\left(\frac{1}{\sqrt{D}}\sum_k F_{ki}\xi_k^\star\right)} \tag{D5}$$

The average over patterns and features is now straightforward using the central limit theorem of section B 1. We finally get

$$\overline{\left\langle y^\star \right\rangle^s_{\boldsymbol{w}\,|\,\{\xi^\mu\}}} = \lim_{n\to 0} \int \prod_l \frac{d h_l d\hat{h}_l}{2\pi} e^{i\sum_l h_l \hat{h}_l - \frac{\mu_\star^2}{2}\sum_{lm} q_{11}^{lm}\hat{h}_l\hat{h}_m} \prod_l \mathrm{sign}(h_l) \\ \times \int \prod_{\substack{a<b \\ lm}} \frac{dq_{ab}^{lm} d\hat{q}_{ab}^{lm}}{2\pi} \prod_{\substack{a\le b \\ lm}} \frac{dp_{ab}^{lm} d\hat{p}_{ab}^{lm}}{2\pi} \prod_{al} \frac{dr_a^l d\hat{r}_a^l}{2\pi} e^{N\phi'} \tag{D6}$$

where we have defined

$$\phi' = -\sum_{a<b}\sum_{lm} q_{ab}^{lm}\hat{q}_{ab}^{lm} - \frac{\alpha_D}{2}\sum_{ab} p_{ab}^{lm}\hat{p}_{ab}^{lm} - \alpha_D \sum_a r_a^l \hat{r}_a^l + G_{SS} + \alpha_D G'_{SE} + \alpha G_E \tag{D7a}$$

$$G_{SS} = \ln \int \prod_{al} dw_{la}\, P_w(w_{la})\, e^{\frac{1}{2}\sum_{a\neq b}\hat{q}_{ab}^{lm} w_{la} w_{mb}} \tag{D7b}$$

$$G'_{SE} = \frac{1}{D}\sum_k \ln \int \prod_{al} \frac{ds_k^{la} d\hat{s}_k^{la}}{2\pi}\, e^{i\sum_{la} s_k^{la}\hat{s}_k^{la} + \sum_{la}\hat{r}_a^l s_k^{la} + \frac{1}{2}\sum_{ab}\sum_{lm}\hat{p}_{ab}^{lm} s_k^{la} s_k^{mb} - \frac{1}{2}\sum_{ab}\sum_{lm} q_{ab}^{lm}\hat{s}_k^{la}\hat{s}_k^{mb} - i\frac{\mu_1\xi_k^\star}{\sqrt{D}}\sum_l \hat{h}_l s_k^{l1}} \tag{D7c}$$

$$G_E = \ln \int \prod_{la} \frac{d\lambda_{la} d\hat{\lambda}_{la}}{2\pi} \frac{du d\hat{u}}{2\pi}\, e^{iu\hat{u} + i\sum_{la}\lambda_{la}\hat{\lambda}_{la} - \beta\sum_{la}\Theta(-u\lambda_{la}) - \frac{\hat{u}^2}{2} - \frac{1}{2}\sum_{ab}\sum_{lm} Q_{ab}^{lm}\hat{\lambda}_{la}\hat{\lambda}_{mb} - \hat{u}\sum_{la} M_a^l \hat{\lambda}_{la}} \tag{D7d}$$

Apart for the different numbers of replicas G_{SS} and G_E have the same expression as before, see equations (B24). The entropic-energetic term instead is the same as before apart for an additional term that depends on the test pattern $\boldsymbol{\xi}^\star$; for this reason we denote it with a prime index,

$$G'_{SE} = G_{SE} + \frac{1}{D}\delta G_{SE} \tag{D8}$$

where G_{SE} is given in (B24c) and

$$\delta G_{SE} \equiv \sum_k \ln \langle\!\langle e^{-\frac{\mu_1\xi_k^\star}{\sqrt{D}}\sum_l \hat{h}_l s_k^{l1}} \rangle\!\rangle_k \tag{D9a}$$

$$\langle\!\langle \bullet \rangle\!\rangle_k \equiv \frac{\displaystyle\int \prod_{al} \frac{ds_k^{la} d\hat{s}_k^{la}}{2\pi}\, e^{i\sum_{la} s_k^{la}\hat{s}_k^{la} + \sum_{la}\hat{r}_a^l s_k^{la} + \frac{1}{2}\sum_{ab}\sum_{lm}\hat{p}_{ab}^{lm} s_k^{la} s_k^{mb} - \frac{1}{2}\sum_{ab}\sum_{lm} q_{ab}^{lm}\hat{s}_k^{la}\hat{s}_k^{mb}} \bullet}{\displaystyle\int \prod_{al} \frac{ds_k^{la} d\hat{s}_k^{la}}{2\pi}\, e^{i\sum_{la} s_k^{la}\hat{s}_k^{la} + \sum_{la}\hat{r}_a^l s_k^{la} + \frac{1}{2}\sum_{ab}\sum_{lm}\hat{p}_{ab}^{lm} s_k^{la} s_k^{mb} - \frac{1}{2}\sum_{ab}\sum_{lm} q_{ab}^{lm}\hat{s}_k^{la}\hat{s}_k^{mb}}} \tag{D9b}$$

Given that at first order in D, G'_{SE} is equal to G_{SE}, we therefore have the same saddle point equations for every ansatz over replicas, as expected. Next, we can expand for large D equation (D9a)

$$\delta G_{SE} = \ln \prod_k \left[1 - i\frac{\mu_1}{\sqrt{D}}\xi_k^\star \sum_l \langle\!\langle s_k^{l1} \rangle\!\rangle_k \hat{h}_l - \frac{\mu_1^2}{2D}(\xi_k^\star)^2 \sum_{lm} \langle\!\langle s_k^{l1} s_k^{m1} \rangle\!\rangle_k \hat{h}_l \hat{h}_m\right] . \tag{D10}$$

As can be seen from (D7), the measure $\langle\!\langle \bullet \rangle\!\rangle_k$ is related to the derivatives of G_{SE}; therefore we can use the saddle point equations and substitute the complicated integral expression with corresponding order parameters

$$\langle\!\langle s_k^{la} \rangle\!\rangle_k = \frac{\partial G_{SE}}{\partial \hat{r}_a^l} = r_a^l \tag{D11a}$$

$$\langle\!\langle s_k^{la} s_k^{mb} \rangle\!\rangle_k = \frac{\partial G_{SE}}{\partial \hat{p}_{ab}^{lm}} = p_{ab}^{lm} \tag{D11b}$$

Notice how the right-hand expressions do not depend on k anymore, since G_{SE} is factorized over this index. We obtain

$$\begin{aligned}\delta G_{SE} &= \ln \prod_k \left[1 - i\frac{\mu_1}{\sqrt{D}}\xi_k^\star \sum_l r_1^l \hat{h}_l - \frac{\mu_1^2}{2D}(\xi_k^\star)^2 \sum_{lm} p_{11}^{lm}\hat{h}_l\hat{h}_m\right] \\ &\simeq -i\mu_1\left(\frac{1}{\sqrt{D}}\sum_k \xi_k^\star\right)\sum_l r_1^l\hat{h}_l - \frac{\mu_1^2}{2}\left(\frac{1}{D}\sum_k(\xi_k^\star)^2\right)\sum_{lm}\left(p_{11}^{lm} - r_1^l r_1^m\right)\hat{h}_l\hat{h}_m\,.\end{aligned} \tag{D12}$$

In the RS ansatz we get

$$\delta G_{SE} \simeq -i\mu_1\left(\frac{1}{\sqrt{D}}\sum_k \xi_k^\star\right) r\sum_l \hat{h}_l - \frac{\mu_1^2}{2}\left(\frac{1}{D}\sum_k(\xi_k^\star)^2\right)\left[\left(p-r^2\right)\left(\sum_l \hat{h}_l\right)^2 + (p_d - p)\sum_l \hat{h}_l^2\right]. \tag{D13}$$

so that

$$\overline{\langle y^\star\rangle^s_{\boldsymbol{w}|\{\boldsymbol{\xi}^\mu\}}} = \int\prod_l \frac{dh_l d\hat{h}_l}{2\pi} e^{i\sum_l\left(h_l - \mu_1 m_\xi^\star r\right)\hat{h}_l - \frac{1}{2}\left(\mu_\star^2(1-q)+\mu_1^2\sigma_\xi^\star(p_d-p)\right)\sum_l\hat{h}_l^2 - \frac{1}{2}\left(\mu_1^2\sigma_\xi^\star(p-r^2)+\mu_\star^2 q\right)\left(\sum_l\hat{h}_l\right)^2}\prod_l \text{sign}(h_l) \tag{D14}$$

where

$$m_\xi^\star \equiv \frac{1}{\sqrt{D}}\sum_k \xi_k^\star\,, \tag{D15a}$$

$$\sigma_\xi^\star \equiv \frac{1}{D}\sum_k\left(\xi_k^\star\right)^2\,. \tag{D15b}$$

Using an Hubbard-Stratonovich transformation we finally obtain

$$\begin{aligned}\overline{\langle y^\star\rangle^s_{\boldsymbol{w}|\{\boldsymbol{\xi}^\mu\}}} &= \int Dz\left[\int\frac{dhd\hat{h}}{2\pi}e^{i\left(h-\mu_1 m_\xi^\star r - \sqrt{\mu_1^2\sigma_\xi^\star(p-r^2)+\mu_\star^2 q}\,z\right)\hat{h} - \frac{1}{2}\left(\mu_\star^2(1-q)+\mu_1^2\sigma_\xi^\star(p_d-p)\right)\hat{h}^2}\text{sign}(h)\right]^s \\ &= \int Dz\left[\int Dh\,\text{sign}\left(\mu_1 m_\xi^\star r + \sqrt{\mu_\star^2(1-q)+\mu_1^2\sigma_\xi^\star(p_d-p)}\,h + \sqrt{\mu_1^2\sigma_\xi^\star\left(p-r^2\right)+\mu_\star^2 q}\,z\right)\right]^s \\ &= \int Dz\left[\text{erf}\left(\frac{\mu_1 m_\xi^\star r + \sqrt{\mu_1^2\sigma_\xi^\star\left(p-r^2\right)+\mu_\star^2 q}\,z}{\sqrt{2\mu_\star^2(1-q)+2\mu_1^2\sigma_\xi^\star\left(p_d-p\right)}}\right)\right]^s\end{aligned} \tag{D16}$$

Inserting this expression into (D4) and (D3) we find

$$\begin{aligned}\epsilon_g^B &= \int Dz\,\mathbb{E}_{\boldsymbol{\xi}^\star}\Theta\left[-m_\xi^\star\,\text{erf}\left(\frac{\mu_1 m_\xi^\star r + \sqrt{\mu_1^2\sigma_\xi^\star\left(p-r^2\right)+\mu_\star^2 q}\,z}{\sqrt{2\mu_\star^2(1-q)+2\mu_1^2\sigma_\xi^\star\left(p_d-p\right)}}\right)\right] \\ &= \int DzDu\,\Theta\left[-u\left(\mu_1 ru + \sqrt{\mu_1^2\left(p-r^2\right)+\mu_\star^2 q}\,z\right)\right] = 2\int_0^\infty Du\,H\left(\frac{\mu_1 ru}{\sqrt{\mu_1^2\left(p-r^2\right)+\mu_\star^2 q}}\right) \\ &= \frac{1}{\pi}\arccos\left(\frac{\mu_1 r}{\sqrt{\mu_1^2 p+\mu_\star^2 q}}\right) = \frac{1}{\pi}\arccos\left(\frac{M}{\sqrt{Q}}\right)\end{aligned} \tag{D17}$$

Notice that if we want to compute the generalization error of the barycenter of typical solutions with a given margin κ, the formula above remains the same. The only dependence on the margin is implicit in the order parameters q, p and r.

The behaviour of the generalization error of the barycenter of typical solutions with vanishing and non-vanishing margins can be found in Fig. 9. As shown in the main text the barycenter achieving the minimal generalization error has a margin κ_{opt} that undergoes a transition when crossing the value $\alpha = \alpha^*$: $\kappa_{\text{opt}} = 0$ for $\alpha > \alpha^*$ whereas it becomes larger then zero when $\alpha < \alpha^*$.

Appendix E: Local entropy lanscape of solutions

1. Analytical approach: Franz-Parisi entropy

To study the local entropy landscape of solutions around a given typical configuration we use the Franz-Parisi approach [2, 42]. The Franz-Parisi free entropy is defined as

$$\Phi_{FP}(t_1) = \frac{1}{Z}\int \prod_i d\tilde{w}_i\, P_w(\tilde{\boldsymbol{w}}) e^{-\tilde{\beta}\sum_\mu \ell_{NE}\left(-y^\mu \tilde{\lambda}^\mu;\tilde{\kappa}\right)} \ln \mathcal{N}(\tilde{\boldsymbol{w}}, t_1) \tag{E1}$$

where $\mathcal{N}(\tilde{\boldsymbol{w}}, t_1)$ is the number of configurations $\boldsymbol{w}$ extracted from the Gibbs measure that have an overlap t_1 with the reference configuration $\tilde{\boldsymbol{w}}$

$$\mathcal{N}(\tilde{\boldsymbol{w}}, t_1) \equiv \int \prod_i dw_i\, P_w(\boldsymbol{w}) e^{-\beta\sum_\mu \ell_{NE}\left(-y^\mu \lambda^\mu;\kappa\right)} \delta\left(\sum_i w_i \tilde{w}_i - N t_1\right). \tag{E2}$$

In order to compute the Franz-Parisi free entropy, we introduce two sets of replicas, one for the partition function Z in the denominator of (E1) (replica index $a = 1, \dots, n$), and the other one for the logarithm in the same equation (replica index $c = 1, \dots, s$)

$$\begin{aligned}
\Phi_{FP}(t_1) &= \lim_{n\to 0}\lim_{s\to 0}\partial_s \int \prod_{ia} d\tilde{w}_i^a \prod_{a=1}^n P_w(\tilde{\boldsymbol{w}}^a) e^{-\tilde{\beta}\sum_{\mu a}\ell_{NE}\left(-y^\mu \tilde{\lambda}_a^\mu;\tilde{\kappa}\right)} \mathcal{N}^s(\tilde{\boldsymbol{w}}^{a=1}, t_1) \\
&= \lim_{\substack{n\to 0\\ s\to 0}} \partial_s \int \prod_{ia} d\tilde{w}_i^a \int \prod_{ia} dw_i^c \prod_{a=1}^n P_w(\tilde{\boldsymbol{w}}^a) \prod_{c=1}^s P_w(\boldsymbol{w}^a) \prod_c \delta\left(\sum_i w_i^c \tilde{w}_i^1 - N t_1\right) \\
&\times e^{-\tilde{\beta}\sum_{\mu a}\ell_{NE}\left(-y^\mu \tilde{\lambda}_a^\mu;\tilde{\kappa}\right) - \beta\sum_{\mu c}\ell_{NE}\left(-y^\mu \lambda_c^\mu;\kappa\right)}
\end{aligned} \tag{E3}$$

The computation is more involved, but proceeds in the same way as before; first of all we extract the teacher and student preactivations (both for the reference and constrained configurations)

$$\begin{aligned}
\Phi_{FP}(t_1) &= \lim_{\substack{n\to 0\\ s\to 0}} \partial_s \int \prod_\mu \frac{du^\mu d\hat{u}^\mu}{2\pi} \prod_{\mu a} \frac{d\tilde{\lambda}_a^\mu d\hat{\tilde{\lambda}}_a^\mu}{2\pi} \prod_{\mu c} \frac{d\lambda_c^\mu d\hat{\lambda}_c^\mu}{2\pi} \prod_\mu e^{iu_\mu \hat{u}_\mu + i\sum_a \tilde{\lambda}_a^\mu \hat{\tilde{\lambda}}_a^\mu + i\sum_c \lambda_c^\mu \hat{\lambda}_c^\mu} \\
&\times \int \prod_{ia} d\tilde{w}_i^a \prod_{ia} dw_i^c \prod_{a=1}^n P_w(\tilde{\boldsymbol{w}}^a) \prod_{c=1}^s P_w(\boldsymbol{w}^a) \prod_c \delta\left(\sum_i w_i^c \tilde{w}_i^1 - N t_1\right) \\
&\times \prod_\mu e^{-\tilde{\beta}\sum_a \ell_{NE}\left(-y^\mu \tilde{\lambda}_a^\mu;\tilde{\kappa}\right) - \beta\sum_c \ell_{NE}\left(-y^\mu \lambda_c^\mu;\kappa\right) - i\hat{u}_\mu \frac{1}{\sqrt{D}}\sum_k w_k^T \xi_k^\mu - i\sum_a \hat{\tilde{\lambda}}_a^\mu \frac{1}{\sqrt{N}}\sum_i w_i^a \tilde{\xi}_i^\mu - i\sum_c \hat{\lambda}_c^\mu \frac{1}{\sqrt{N}}\sum_i w_i^c \xi_i^\mu}.
\end{aligned} \tag{E4}$$

Then we average over the patterns and features, using the central limit theorem of Section B 1. We finally find

$$\begin{aligned}
\Phi_{FP}(t_1) &= \lim_{\substack{n\to 0\\ s\to 0}} \partial_s \int \prod_{a<b} \frac{d\tilde{q}_{ab} d\hat{\tilde{q}}_{ab}}{2\pi} \prod_{c<d} \frac{dq_{cd} d\hat{q}_{cd}}{2\pi} \prod_{a\le b} \frac{d\tilde{p}_{ab} d\hat{\tilde{p}}_{ab}}{2\pi} \prod_{c\le d} \frac{dp_{cd} d\hat{p}_{cd}}{2\pi} \prod_a \frac{d\tilde{r}_a d\hat{\tilde{r}}_a}{2\pi} \prod_c \frac{dr_c d\hat{r}_c}{2\pi} \\
&\times \int \prod_{ac} \frac{dk_{ac} d\hat{k}_{ac}}{2\pi} \prod_{c,a\neq 1} \frac{dt_{ac} d\hat{t}_{ac}}{2\pi} e^{N\phi_{FP}}
\end{aligned} \tag{E5}$$

where

$$\phi_{FP} = -\sum_{a<b}\tilde{q}_{ab}\hat{\tilde{q}}_{ab} - \frac{\alpha_D}{2}\sum_{ab}\tilde{p}_{ab}\hat{\tilde{p}}_{ab} - \alpha_D\sum_a \tilde{r}_a\hat{\tilde{r}}_a - \sum_{c<d} q_{cd}\hat{q}_{cd} - \frac{\alpha_D}{2}\sum_{cd} p_{cd}\hat{p}_{cd} - \alpha_D\sum_c r_c\hat{r}_c \\ - \alpha_D\sum_{ac} k_{ac}\hat{k}_{ac} - \sum_{ac} t_{ac}\hat{t}_{ac} + G_{SS} + \alpha_D G_{SE} + \alpha G_E \tag{E6a}$$

$$G_{SS} = \ln\int\prod_a d\tilde{w}_a\, P_w(\tilde{w}_a)\int\prod_c dw_c\, P_w(w_c)\, e^{\frac{1}{2}\sum_{a\neq b}\hat{\tilde{q}}_{ab}\tilde{w}_a\tilde{w}_b - \frac{1}{2}\sum_{a\neq b}\hat{q}_{cd}w_c w_d + \sum_{ac}\hat{t}_{ac}\tilde{w}_a w_c} \tag{E6b}$$

$$G_{SE} = \ln\int\prod_a\frac{d\tilde{s}_a d\hat{\tilde{s}}_a}{2\pi}\prod_c\frac{ds_c d\hat{s}_c}{2\pi}\, e^{i\sum_a\tilde{s}_a\hat{\tilde{s}}_a + i\sum_c s_c\hat{s}_c + \sum_a\hat{\tilde{r}}_a\tilde{s}_a + \sum_c\hat{r}_c s_c + \frac{1}{2}\sum_{ab}\hat{\tilde{p}}_{ab}\tilde{s}_a\tilde{s}_b + \frac{1}{2}\sum_{cd}\hat{p}_{cd}s_c s_d} \\ \times e^{-\frac{1}{2}\sum_{ab}\tilde{q}_{ab}\hat{\tilde{s}}_a\hat{\tilde{s}}_b - \frac{1}{2}\sum_{cd}q_{cd}\hat{s}_c\hat{s}_d + \sum_{ac}\hat{k}_{ac}\tilde{s}_a s_c - \sum_{ac}t_{ac}\hat{\tilde{s}}_a\hat{s}_c} \tag{E6c}$$

$$G_E = \ln\int\prod_a\frac{d\tilde{\lambda}_a d\hat{\tilde{\lambda}}_a}{2\pi}\prod_a\frac{d\lambda_c d\hat{\lambda}_c}{2\pi}\frac{du d\hat{u}}{2\pi}\, e^{iu\hat{u} + i\sum_a\tilde{\lambda}_a\hat{\tilde{\lambda}}_a + i\sum_c\lambda_c\hat{\lambda}_c - \tilde{\beta}\sum_c\Theta\left(-\mathrm{sign}(u)\tilde{\lambda}_c + \tilde{\kappa}\right) - \beta\sum_a\Theta(-\mathrm{sign}(u)\lambda_a + \kappa)} \\ \times e^{-\frac{\hat{u}^2}{2} - \frac{1}{2}\sum_{ab}\tilde{Q}_{ab}\hat{\tilde{\lambda}}_a\hat{\tilde{\lambda}}_b - \frac{1}{2}\sum_{cd}Q_{cd}\hat{\lambda}_c\hat{\lambda}_d - \hat{u}\sum_a\tilde{M}_a\hat{\tilde{\lambda}}_a - \hat{u}\sum_c M_c\hat{\lambda}_c - \sum_{ac}T_{ac}\hat{\tilde{\lambda}}_a\hat{\lambda}_c} \tag{E6d}$$

All the order parameters appearing in the previous formulas are

$$M_c = \mu_1 r_c\,, \qquad r_c = \frac{1}{D}\sum_{k=1}^{D} s_k^c\,, \qquad s_k^a = \frac{1}{\sqrt{N}}\sum_{i=1}^{N} F_{ki}w_i$$

$$\tilde{M}_a = \mu_1\tilde{r}_a\,, \qquad \tilde{r}_a = \frac{1}{D}\sum_{k=1}^{D}\tilde{s}_k^a\,, \qquad \tilde{s}_k^a = \frac{1}{\sqrt{N}}\sum_{i=1}^{N} F_{ki}\tilde{w}_i$$

$$Q_{cd} = \mu_1^2 p_{cd} + \mu_\star^2 q_{cd}\,, \qquad p_{cd} = \frac{1}{D}\sum_{k=1}^{D} s_k^c s_k^d\,, \qquad q_{cd} = \frac{1}{N}\sum_{i=1}^{N} w_i^c w_i^d$$

$$\tilde{Q}_{ab} = \mu_1^2\tilde{p}_{ab} + \mu_\star^2\tilde{q}_{ab}\,, \qquad \tilde{p}_{ab} = \frac{1}{D}\sum_{k=1}^{D}\tilde{s}_k^a\tilde{s}_k^b\,, \qquad \tilde{q}_{ab} = \frac{1}{N}\sum_{i=1}^{N}\tilde{w}_i^a\tilde{w}_i^b$$

$$T_{ac} = \mu_1^2 k_{ac} + \mu_\star^2 t_{ac}\,, \qquad k_{ac} = \frac{1}{D}\sum_{k=1}^{D}\tilde{s}_k^a s_k^c\,, \qquad t_{ac} = \frac{1}{N}\sum_{i=1}^{N}\tilde{w}_i^a w_k^c\,.$$

We have understood that $t_{1c} \equiv t_1$ as this condition is imposed by the delta function in equation (E2).

Notice that, as in Section B 2, the entropic-energetic term G_{SE} is Gaussian, so it can be readily solved. Defining the quantities

$$\hat{\overline{r}}_\alpha \equiv \left(\hat{\tilde{r}}_a, \hat{r}_c\right) \in \mathbb{R}^{n+s}\,, \tag{E7a}$$

$$\overline{q}_{\alpha\beta} \equiv \begin{pmatrix}\tilde{q}_{ab} & t_{ac}\\ t_{ac} & q_{cd}\end{pmatrix} \in \mathbb{R}^{(n+s)\times(n+s)} \tag{E7b}$$

$$\hat{\overline{p}}_{\alpha\beta} \equiv \begin{pmatrix}\hat{\tilde{p}}_{ab} & k_{ac}\\ k_{ac} & \hat{p}_{cd}\end{pmatrix} \in \mathbb{R}^{(n+s)\times(n+s)} \tag{E7c}$$

it can be seen that (E6c) can be written in the same way as (B24c) in terms of $\hat{\overline{r}}_\alpha$, $\overline{q}_{\alpha\beta}$ and $\hat{\overline{p}}_{\alpha\beta}$, so that

$$G_{SE} = -\frac{1}{2}\ln\det\left(\mathbb{I} - \overline{q}\,\hat{\overline{p}}\right) + \frac{1}{2}\sum_{\alpha\beta}\hat{\overline{r}}_\alpha\left[\left(\mathbb{I} - \overline{q}\,\hat{\overline{p}}\right)^{-1}\overline{q}\right]_{\alpha\beta}\hat{\overline{r}}_\beta\,. \tag{E8}$$

a. RS ansatz

We impose an RS ansatz over the order parameters:

$$q_{ab} = \delta_{ab} + q\,(1-\delta_{ab}) \qquad \hat{q}_{ab} = \hat{q}\,(1-\delta_{ab}) \tag{E9a}$$
$$p_{ab} = p_d\delta_{ab} + p\,(1-\delta_{ab}) \qquad \hat{p}_{ab} = -\hat{p}_d\delta_{ab} + \hat{p}\,(1-\delta_{ab}) \tag{E9b}$$
$$r_a = r \qquad \hat{r}_a = r \tag{E9c}$$
$$k_{ac} = k_1\delta_{a1} + k_0(1-\delta_{a1}) \qquad \hat{k}_{ac} = \hat{k}_1\delta_{a1} + \hat{k}_0(1-\delta_{a1}) \tag{E9d}$$
$$t_{ac} = t_1\delta_{a1} + t_0(1-\delta_{a1}) \qquad \hat{t}_{ac} = \hat{t}_1\delta_{a1} + \hat{t}_0(1-\delta_{a1}) \tag{E9e}$$

A similar ansatz is imposed for the tilde order parameters.

b. Entropic-Entropic and Energetic terms

Let us start from the entropic-entropic contribution. This term is exactly equal to the entropic contribution of a storage problem [2]. It is equal to

$$\mathcal{G}_{SS} \equiv \frac{\hat{q}}{2} + \lim_{\substack{n\to 0\\ s\to 0}} \partial_s G_{SS} = \int Dx \frac{\sum_{\tilde{w}=\pm 1} e^{\sqrt{\hat{\tilde{q}}}\tilde{w}x} \int Dy \ln 2\cosh\left(\sqrt{\hat{q} - \frac{\hat{t}_0^2}{\hat{\tilde{q}}}}y + \frac{\hat{t}_0}{\sqrt{\hat{\tilde{q}}}}x + (\hat{t}_1 - \hat{t}_0)\tilde{w}\right)}{2\cosh\left(\sqrt{\hat{\tilde{q}}}x\right)} \tag{E10}$$

The energetic term is a bit more involved. It is however equal (apart for a redefinition of order parameters) to the energetic term that is obtained in a classic teacher-student problem. It is equal to

$$\mathcal{G}_E \equiv \lim_{\substack{n\to 0\\ s\to 0}} \partial_s G_E = 2\int DxDy\, \frac{H_{\tilde{\beta}}\left(u(x,y)\right)}{H_{\tilde{\beta}}\left(h(x)\right)} \int_{h(x)}^{\infty} Dz\, \ln H_\beta\left(v(x,y,z)\right) \tag{E11}$$

with

$$u(x,y) \equiv \frac{\tilde{M}\sqrt{\Gamma}\,(by - ax) - M\sqrt{\tilde{Q}}y}{\sqrt{(\tilde{Q}-\tilde{M}^2)(\Gamma - M^2) - (T_0 - M\tilde{M})^2}} \tag{E12a}$$
$$v(x,y,z) \equiv \frac{\kappa - \sqrt{\Gamma}(ay + bx) - \frac{T_1 - T_0}{\sqrt{\tilde{Q}_d - \tilde{Q}}}z}{\sqrt{Q_d - Q}} \tag{E12b}$$
$$h(x) \equiv \frac{\tilde{\kappa} - \sqrt{\tilde{Q}}x}{\sqrt{\tilde{Q}_d - \tilde{Q}}} \tag{E12c}$$
$$\Gamma \equiv Q - \frac{(T_1 - T_0)^2}{\tilde{Q}_d - \tilde{Q}} \tag{E12d}$$
$$b \equiv \frac{T_0}{\sqrt{\tilde{Q}\Gamma}}, \qquad a \equiv \sqrt{1-b^2} \tag{E12e}$$

c. Entropic-Energetic term

Following a series of algebraic manipulations, the entropic-energetic term reads

$$\begin{aligned}\mathcal{G}_{SE} \equiv \lim_{\substack{n\to 0\\ s\to 0}} \partial_s G_{SE} = &-\frac{1}{2}\ln\eta + \frac{1}{2\eta}\left[\left(\hat{p}+\hat{r}^2\right)(1-q) - (\hat{p}_d+\hat{p})\,q\right] + \\ &+ \frac{1}{2\eta\tilde{\eta}}\left\{(1-q)(\hat{k}_1-\hat{k}_0)^2\left[1-q+\frac{1}{\tilde{\eta}}\left((1-\tilde{q})^2(\hat{\tilde{p}}+\hat{\tilde{r}}^2)+q\right)\right] + 2(\hat{k}_1-\hat{k}_0)\left[(1-q)(1-\tilde{q})(\hat{k}_0+\hat{r}\hat{\tilde{r}})+t_0\right]\right\} \\ &+ \frac{1}{2\eta\tilde{\eta}}\left\{(\hat{p}_d+\hat{p})(t_1-t_0)^2\left[\hat{\tilde{p}}_d+\hat{\tilde{p}}-\frac{1}{\tilde{\eta}}\left(\hat{\tilde{p}}+\hat{\tilde{r}}^2+(\hat{\tilde{p}}_d+\hat{\tilde{p}})^2\tilde{q}\right)\right] + 2(t_1-t_0)\left[(\hat{p}_d+\hat{p})\left(\hat{\tilde{p}}_d+\hat{\tilde{p}}\right)t_0+\hat{k}_0+\hat{r}\hat{\tilde{r}}\right]\right\} \\ &+ \frac{(\hat{k}_1-\hat{k}_0)(t_1-t_0)}{\eta\tilde{\eta}}\left[1+\frac{1}{\tilde{\eta}}\left[(1-\tilde{q})\left(\hat{\tilde{p}}+\hat{\tilde{r}}^2\right)-(\hat{\tilde{p}}_d+\hat{\tilde{p}})\tilde{q}\right]\right]\end{aligned} \tag{E13}$$

where we have defined the quantities

$$\eta \equiv 1 + (\hat{p}_d+\hat{p})(1-q) \tag{E14a}$$

$$\tilde{\eta} \equiv 1 + (\hat{\tilde{p}}_d+\hat{\tilde{p}})(1-\tilde{q}) \tag{E14b}$$

d. Final expression of the free entropy

The RS Franz-Parisi free entropy is finally

$$\Phi_{FP}(t_1) = -\frac{\hat{q}}{2}(1-q) + \frac{\alpha_D}{2}\,(p_d\hat{p}_d + p\hat{p}) - \alpha_D r\hat{r} - \alpha_D(k_1\hat{k}_1 - k_0\hat{k}_0) - t_1\hat{t}_1 + t_0\hat{t}_0 + \mathcal{G}_{SS} + \alpha_D\mathcal{G}_{SE} + \alpha\mathcal{G}_E \tag{E15}$$

The tilde order parameters being those one characterizing the reference configuration will satisfy the RS saddle point equation analyzed in Section B 3. The order parameters q, $\hat{q}$, p_d, $\hat{p}_d$, p, $\hat{p}$, r, $\hat{r}$, k_1, $\hat{k}_1$, k_0, $\hat{k}_0$, $\hat{t}_1$, t_0 and $\hat{t}_0$ are found by solving the saddle point equations obtained by taking the corresponding derivatives of the Franz-Parisi entropy and imposing them to be equal to zero.

2. Numerical experiments

Local energy curves. To compare the geometrical structure of solutions found by different algorithms, we computed the local energy profiles [52, 53]. For all the architectures we have analyzed in the main text (continuous tree committee machine, multi-layer perceptrons, CNN), we have computed the local energy as follows. Given a solution to the learning problem, we perturbed it using a multiplicative Gaussian noise that acts on the network weights as follows

$$W \to W\,(1+\eta)$$

where W is a weight of the network while the variance of the noise η is tuned to obtain perturbed vectors that have increasing distance from W. After the perturbation, the networks are normalized as explained in the main text, and we measure their Euclidean distance from the original solution. We repeatedly perturb every solution for each level of noise, and collect the distances and training errors. The "local energy" curve is the average training error rate displayed as a function of the average distance.

In the main text we show curves for the multi-layer perceptron architecture and for the CNN.

Here we show in Fig. 13 the local energy curves for three different algorithms trained on the (binary) overparameterized perceptron: MCT0, SBPI and BP. In general one can argue that overparameterizing the network leads to higher flatness; however for the MCT0 algorithm, the solution is always sharp, as predicted by the replica theory.

Lazy regime description for CNNs trained on CIFAR10 dataset. These big architectures have been shown to work in the so-called lazy training regime: the initial layers changes less than the following ones. This can be quantified numerically studying the overlap between the single layer weight configuration at epoch 0 and at epoch t, as shown in Fig. 14. The overlap decreases for every layer, with a drop rate that depends on the layer position.

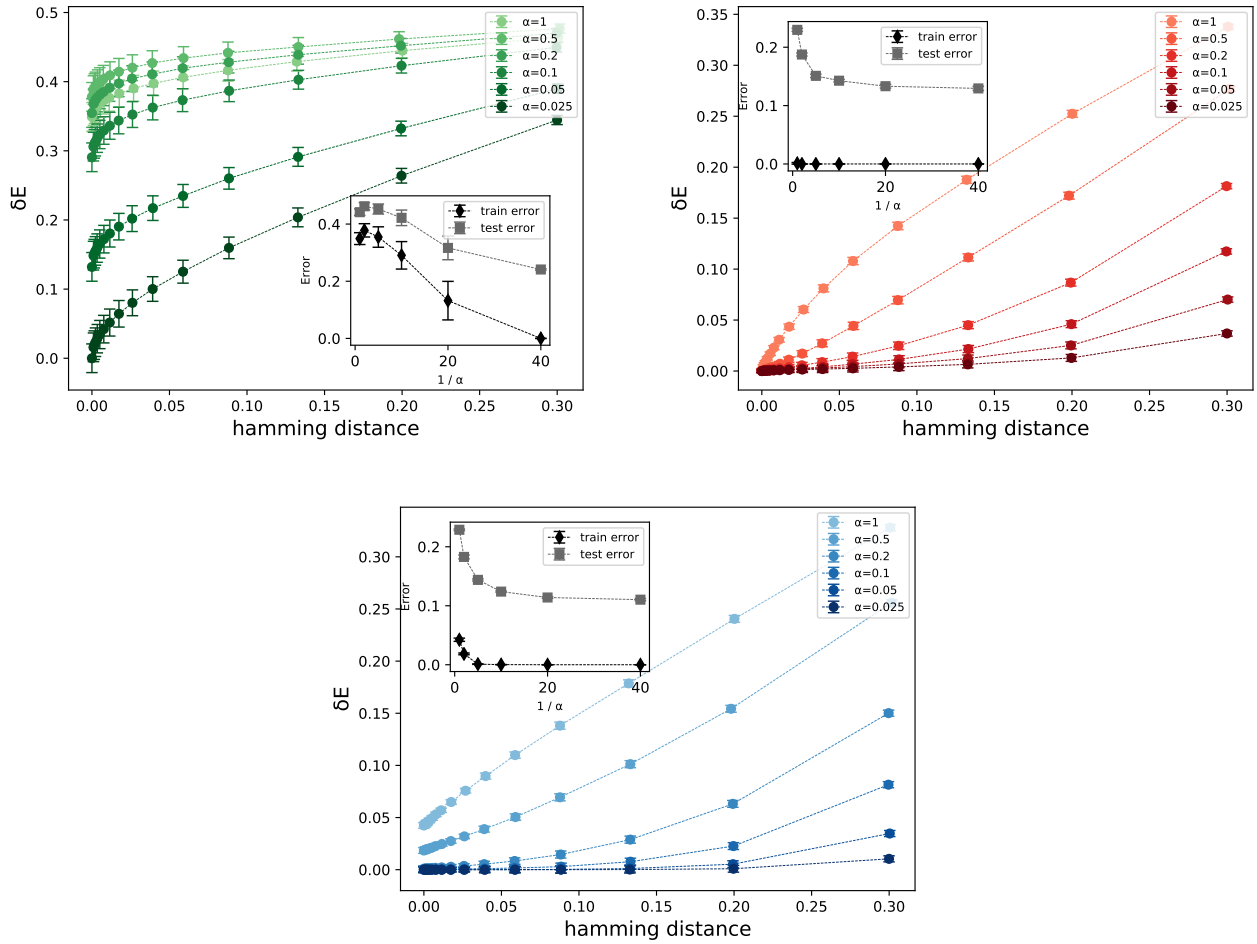

Figure 13. Local energy δE as a function of distance from the reference solution in the overparameterized perceptron model fixing $\alpha_T = 5$, for different values of α and for MCT0, SBPI and BP solutions. Train and test error are depicted in the insets. While MCT0 solutions are sharp, even in the $\alpha \to 0$ limit, other algorithms find solutions whose flatness increases as the student is more overparameterized.

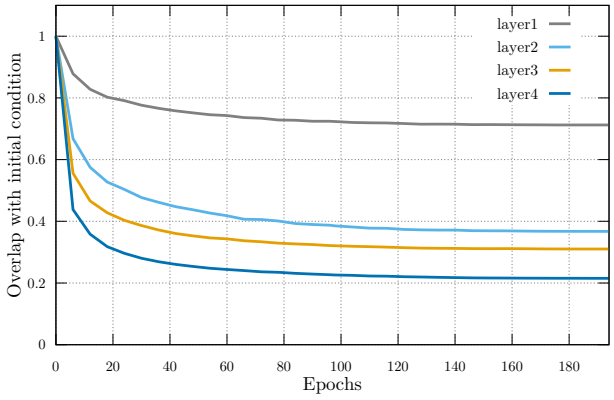

Figure 14. Overlap of a solution with its initial condition (epoch 0) as a function of the number of epochs. The overlap is represented layer by layer and it refers to ADAM optimizer, a number of filters proportional to $C = 20$, and a $lr = 0.01$.

Appendix F: Material and Methods

Gaussian Equivalence theorem. It has been shown by [23], that in the thermodynamic limit (A7), the statistical properties of the random feature model are equivalent to a Gaussian covariate model, in which each projected pattern $\tilde{\boldsymbol{\xi}}^\mu$ is a *linear* combination of the patterns components ξ_k^μ plus noise. The strength of the noise depends on the degree of non-linearity of the activation function σ. In mathematical terms the following mapping between different models holds

$$\tilde{\xi}_i^\mu = \sigma\left(\frac{1}{\sqrt{D}}\sum_{k=1}^{D} F_{ki}\xi_k^\mu\right) = \mu_0 + \frac{\mu_1}{\sqrt{D}}\sum_{k=1}^{D} F_{ki}\xi_k^\mu + \mu_\star \eta_i^\mu \tag{F1}$$

where $\eta_i \sim \mathcal{N}(0,1)$ are i.i.d. standard Gaussian random variables and $\mu_0 = \int Dz\, \sigma(z)$, $\mu_1 = \int Dz\, z\, \sigma(z)$, $\mu_2 = \int Dz\, \sigma^2(z)$, $\mu_\star^2 = \mu_2 - \mu_1^2 - \mu_0^2$ with $Dz \equiv \frac{e^{-z^2/2}}{\sqrt{2\pi}}$. We provide a sketch of the proof, based on the explicit computations of the moments, in the SI and refer to [23, 24] for more details.

Numerical Experiments on the Binary Perceptron. Here we report the details for the numerical experiments performed on the overparameterized binary perceptron. (SA) Simulated Annealing, based on a standard Metropolis algorithm that attempts one weight flip at a time, is run until either a solution is found or a maximum number of sweeps (4000) is reached, where a sweep consists of N attempted moves. We used an initial inverse temperature $\beta = 1.0$ that is increased at every sweep with a linear increment $\Delta\beta = 5 \cdot 10^{-3}$. (SBPI) For a complete description of the SBPI algorithm see ref. [54]. In the numerical experiments we set the maximum number of allowed iterations to 500 and used a threshold $\theta_{\rm m} = 2$ and a probability $p_{\rm s} = 0.3$ of updating the synapses with a stability $0 \leq \Delta^\mu \leq \theta_{\rm m}$. (BP) We used a standard BP implementation with damping $\delta = 0.5$ and a maximum number of updates fixed to 200. The magnetization are randomly initialized with a uniform distribution in the interval $[-\epsilon, \epsilon]$ with $\epsilon = 10^{-2}$. (fBP) For the focusing BP algorithm we used the same initialization of BP, and a damping factor $\delta = 0.9$. We set the number of virtual replicas to $y = 10$ and update the messages until convergence for 30 steps, each time increasing the coupling strength γ according to $\gamma = \text{atanh}\,(i/29)$ where $i = 0, \ldots, 29$ (using by convention $\gamma = 10$ for the last step). (BNet) We used the standard implementation of BinaryNet (see. ref [41]) using sign activation function and cross-entropy loss, without using batch normalization. We fixed the learning rate $\eta = 5 \cdot 10^{-3}$ and ran a full batch gradient update for 2000 epochs.

Numerical Experiments on the committee machine. Here we report the details for the numerical experiments performed on the overparameterized continuous tree-like committee machine, for the two algorithms used. (fBP) In all experiments, we set $y = 10$ and ranged γ between 0.5 and 30 with an exponential schedule divided into 30 steps; at each step, the algorithm was run (with damping $\delta = 0.1$) until convergence (with a convergence criterion set to $\epsilon = 10^{-2}$) or at most 200 iterations. (SGD) The expression for the cross-entropy loss in the binary classification case, with a scale parameter γ, is: $f_\gamma(x) = -\frac{x}{2} + \frac{1}{2\gamma}\log(2\cosh(\gamma x))$. This is just the standard expression but with the input x scaled by γ (which is equivalent to setting the norm of the input weights to γ) and the output scaled by $1/\gamma$ (equivalent to scaling the gradients by $1/\gamma$). In all the experiments, we set the batch size to 100, the maximum number of epochs to 700, and the learning rate to 10^{-2}. The weights were initialized from a uniform distribution and then normalized for each unit. The norm parameters γ and β were initialized at the values $\gamma = 10$, $\beta = 1$ and multiplied by $1 + 10^{-4}$, $1 + 10^{-2}$, respectively, after each epoch. The algorithm stopped as soon as it found a solution (this was determined using the desired architecture with sign activation and output functions, which is equivalent to letting $\beta, \gamma \to \infty$ and checking for a zero-error).

Numerical Experiments on the Multi-layer neural network Our implementation follows closely the one of ref. [29]. We set the learning rate to 10^{-4} and train the model with full batch gradient descent with ADAM optimization for a fixed number of epochs (5000). In all the simulations we used ReLU non-linearities and the square-hinge loss with a margin fixed to 1. In order to train the model with the adversarial initialization, we first trained the network using SGD with a fixed number of epochs (5000) (with minibatches of size 128 and learinig rate set to $5 \cdot 10^{-3}$) on a modified train set in which the labels have been randomized, then we used the resulting weights as the initial condition for ADAM.

Numerical Experiments on CNNs. We used standard PyTorch initialization (HE for ReLU activation function) and the cross-entropy loss function for all the experiments. We took 5 independent samples for both optimizers: SGD with momentum and ADAM. In all experiments, the learning rate was 10^{-2}, the batch size 50 and the number of epochs was 200. For SGD the momentum was set to 0.3. All the additional parameters were taken as in the default Pytorch settings. The number of network parameters was controlled by the value to C as defined in the main text.