More data or more parameters? Investigating the effect of data structure on generalization

Stéphane d'Ascoli*[1,2], Marylou Gabrié[3,4], Levent Sagun[1] and Giulio Biroli[2]

[1]Facebook AI Research, Paris
[2]Laboratoire de Physique de l'Ecole Normale Supérieure, Paris
[3]Flatiron Institute, New York
[4]Center for Data Science, NYU, New York

Abstract

One of the central features of deep learning is the generalization abilities of neural networks, which seem to improve relentlessly with over-parametrization. In this work, we investigate how properties of data impact the test error as a function of the number of training examples and number of training parameters; in other words, how the structure of data shapes the "generalization phase space". We first focus on the random features model trained in the teacher-student scenario. The synthetic input data is composed of independent blocks, which allow us to tune the saliency of low-dimensional structures and their relevance with respect to the target function. Using methods from statistical physics, we obtain an analytical expression for the train and test errors for both regression and classification tasks in the high-dimensional limit. The derivation allows us to show that noise in the labels and strong anisotropy of the input data play similar roles on the test error. Both promote an asymmetry of the phase space where increasing the number of training examples improves generalization further than increasing the number of training parameters. Our analytical insights are confirmed by numerical experiments involving fully-connected networks trained on MNIST and CIFAR10.

1 Introduction

Despite their remarkable achievements over the past decade [1–4], deep neural networks continue to puzzle rigorous understanding by their astonishing generalization capacities [5–7]. On one hand, classical wisdom says that a learning model should have just the right number of parameters to learn from a dataset without overfitting it. On the other hand, recent years have seen the emergence of massively overparametrized models [8, 9], which manage to generalize well, somehow avoiding the pitfall of overfitting.

Various works have given evidence of a double descent curve in deep learning [7, 10–12], whereby the test error first decreases as the number of parameters increases, then peaks, then decreases again monotonically. The overfitting peak occurs at the *interpolation threshold* where training error vanishes, and is well-studied in the statistical physics literature [13–17].

A puzzling observation one can make on the double descent phenomenon is the fact that it appears to occur both parameter-wise and sample-wise [18, 19]. Thus, the number of training parameters P and the number of training examples N play similar roles in generalization performance, being both beneficial, despite playing opposite roles in terms of fitting performance (train loss decreases with P and increases with N). In other words, a large model applied to a small dataset happens to give similar test loss to a small model applied to a large dataset.

This observation is illustrated in Fig. 1, where we show the train and test error of two-layer fully-connected networks on a downsampled version of CIFAR10. Both quantities are plotted in an (N, P)-diagram as a function of the numbers of training samples N and parameters P. The train error increases with N and decreases with P, vanishing at the interpolation threshold for P at a level slightly above N. However, the test error decreases when increasing either N or P, and the "generalization

*stephane.dascoli@ens.fr

phase space" is, in fact, more or less symmetric with respect to the $N = P$ line.

Generally speaking, generalization is allowed by the underlying structure of data. One aspect of data structure is the distribution of inputs: MNIST and CIFAR10 have same number of classes and images, yet generalization is harder for CIFAR10 because the images are more complex. The other aspect of structure is the rule between inputs and outputs: a random labelling of CIFAR10 can be learned by a neural network but offers no possibility of generalization [5]. Characterizing the structure of real-world data involves studying the interplay between these two aspects. In this direction, simple, interpretable models of structured data can prove useful to understand what underlies the behaviour of the generalization phase space outlined above.

Setting In this work we present a solvable model of learning which enables us to:

(i) vary P and N independently;

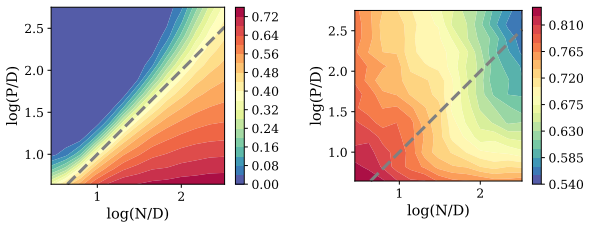
(ii) analyze classification tasks;

(iii) strudy the effect of data structure.

Many works have studied the double descent curve from an analytical point of view [13, 14, 20–22], yet to the best of our knowledge, none fulfils these three conditions simultaneously. For instance, linear models applied to regression problems with i.i.d. Gaussian inputs, which is a commonly analyzed setup, violates each of our three requirements.

To satisfy *(i)*, we consider a random feature model [23], which was shown to exhibit double descent in Mei & Montanari [24]. To satisfy *(ii)*, we follow the lines of [25] and use an approach from statistical physics, enabling to generalize the result of [24] to any convex loss function, as long as the labels are generated by a linear ground truth, i.e. $y = f^0(\boldsymbol{\beta} \cdot \boldsymbol{x})$ with $\boldsymbol{\beta} \in \mathbb{R}^D$ a teacher weight vector. Finally, to satisfy *(iii)*, we introduce a block model in which the input space is subdivided into various subspaces with different variances (*saliency*) and different correlation (*relevance*) with the teacher vector $\boldsymbol{\beta}$. This model was studied for linear regression under the name of *strong and weak features model* [26].

Contributions Our main analytical contribution is the expression of the generalization error of a random feature model trained on data generated by the strong and weak features model, in the limit where



(a) Train error (b) Test error

Figure 1: **The generalization phase space of CIFAR10: number of parameters and training examples play opposite roles in terms of train error but similar roles in terms of generalization.** We trained 2-layer ReLU networks of varying width on CIFAR10, downscaled to $D = 10 \times 10$ images. N is the number of examples in the training set and P denotes the total number of parameters of the network. *Left:* Train error at convergence. *Right:* Test error at convergence. The phase space is almost symmetric along the line $N = P$: inverting P and N reaches similar test error.

the input dimension D, the number of training examples N and the number of parameters P are sent to infinity with their ratios fixed. Our analysis is valid both for regression and classification tasks. We confirm numerically that the asymptotic prediction matches simulations even at moderately large sizes.

This allows us to study analytically the effect of the data anisotropy on the double descent curves. We show in particular that salient relevant features allow the model to "beat the curse of dimensionality", whereas salient irrelevant features can act as a form of regularization.

We then investigate the generalization phase space in the same model. In the case of isotropic noiseless data, we observe a quasi-symmetric phase space. We find that noise in the labels or irrelevant features in the inputs tends to make the phase space less symmetric in favor of more data rather than more parameters. Finally, we present numerical investigations of the phase space for two-layer networks trained on MNIST and CIFAR10, and observe a similar phenomenology, demonstrating the relevance of the model considered.

Reproducibility We provide an open-source implementation of our experiments. It is available at https://github.com/sdascoli/strong-weak.

Related Work Although most theoretical studies of generalization focus on structureless data, a few exceptions exist. The strong and weak features model has recently been studied in the context of least-squares regression, both empirically [22] and theoretically [26–28]. Several intriguing observation emerge in strongly anisotropic setup: (i) several overfitting peaks can be seen [22, 29]; (ii) the optimal ridge regularization parameter γ can become negative [28, 30], as it becomes helpful to encourage the weights to have very different magnitudes; (iii) extra features acting as pure noise can actually play a beneficial role by inducing some implicit regularization [26].

The strong and weak features model also includes the setup studied in Ghorbani *et al.* [31], called the spiked covariates model. It involves a small block of size D^η scaling polynomially in the input dimension ($\eta < 1$) and a large "junk" block with no correlation with the labels. The question is then: can kernel methods learn to discard the junk features, hence "beat the curse of dimensionality" in the way neural networks do? The answer was shown to depend on the strength of the junk features: when the variance of these features small, they are not problematic and the kernel method ignores them, effectively learning a task of effective dimensionality $D^\eta \ll D$.

More general models for the structure of data were considered in other works. In the special case of random Fourier Feature regression, Liao *et al.* [32] derived the train and test error for a general input distribution. Canatar *et al.* [33] achieved a similar result in the non-parametric setting of kernel regression, which can be viewed as the limiting case of random feature regression when the number of random features P goes to infinity.

The few works studying classification analytically have mostly focused on linear models, trained on linearly separable data [34–36] or gaussian mixtures [37, 38]. One of the challenges in classification (in comparison to regression) is the large set of available loss functions [39, 40]. In particular, the choice of the loss impacts the location of the interpolation threshold. A notable exception is Gerace *et al.* [25], on which our paper builds. Using tools from statistical mechanics, the authors were able to derive the generalization loss of random features model for *any* loss function, yet their results focus on isotropic data.

In a more realistic deep learning setup, Li *et al.* [41] studied the effect of "hard-to-fit, easy-to-generalize" features, which are very akin to our strong and weak features model. They show that large learning rates favor learning easy-to-fit, hard-to-generalize features (what we would call highly salient but little relevant). In contrast, a small learning rate does the opposite, motivating the use of learning rate decay.

2 A solvable model of data structure

To explore the properties of the generalization phase space, we focus on the random features model[1] introduced in Rahimi & Recht [23]. Using tools from statistical physics, we derive the generalization error in a teacher-student scenario, including the strong and weak features model of structured data. Albeit non-rigorous in a strict mathematical sense, these methods are considered exact in theoretical physics and have been influential to statistical learning theory since the 80s, see Gabrié [44] and Bahri *et al.* [45] for recent reviews. Moreover, several of their conjectures have been proven exact in the last decade [46–49].

Random feature model The random features model can be viewed as a two-layer neural network whose first layer is fixed random matrix containing P random features $\{\boldsymbol{F}_i \in \mathbb{R}^D\}_{i=1\dots P}$:

$$\hat{y}_\mu = \hat{f}\left(\sum_{i=1}^{P} \boldsymbol{w}_i \sigma\left(\frac{\boldsymbol{F}_i \cdot \boldsymbol{x}_\mu}{\sqrt{D}}\right)\right) \equiv \hat{f}\left(\boldsymbol{w} \cdot \boldsymbol{z}_\mu\right), \tag{1}$$

where $\sigma(\cdot)$ is a pointwise activation function, $\hat{f}$ is an output function and we introduced the notation $\boldsymbol{z}_\mu \in \mathbb{R}^P$. Elements of $\boldsymbol{F}$ are drawn i.i.d from $\mathcal{N}(0,1)$. The labels are given by a linear ground truth, modulated by an output function f^0 and possibly corrupted by noise through a probabilistic channel $\mathcal{P}$:

$$y_\mu \sim \mathcal{P}\left(\cdot \,\middle|\, f^0\left(\frac{\boldsymbol{\beta} \cdot \boldsymbol{x}_\mu}{\sqrt{D}}\right)\right). \tag{2}$$

The second layer weights, i.e. the elements of $\boldsymbol{w} \in \mathbb{R}^P$, are trained by minimizing an ℓ_2-regularized loss

[1] This model is akin to the so-called lazy learning regime of neural networks where the weights stay close to their initial value [42]: assuming $f_{\theta_0} = 0$, we have $f_\theta(\mathbf{x}) \approx \nabla_\theta f_\theta(\mathbf{x})|_{\theta=\theta_0} \cdot (\theta - \theta_0)$ [43]. In other words, lazy learning amounts to a linear fitting problem with a random feature vector $\nabla_\theta f_\theta(\mathbf{x})|_{\theta=\theta_0}$.

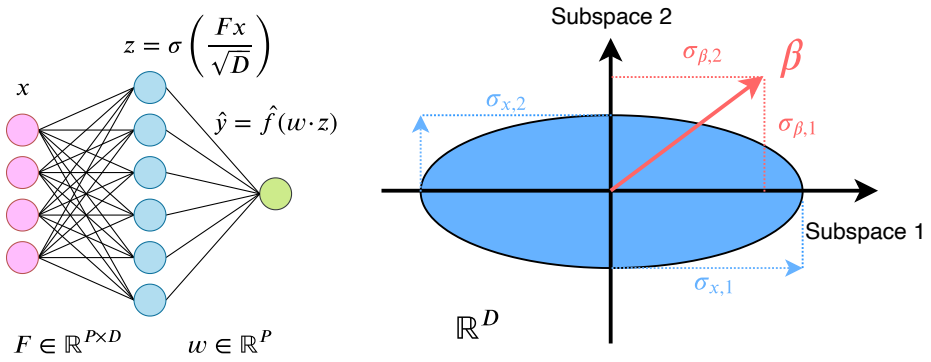


Figure 2: *Left:* Random feature model considered here, which can be viewed as a two-layer networks where only the second layer is trained. *Right:* Strong and weak features model considered here. Input space is decomposed into two subspaces, a *salient* one with strong variance $\sigma_{x,1}$ and the other with smaller variance $\sigma_{x,2}$. We can adjust the *relevance* of both subspaces by adjusting the variance of the teacher $\boldsymbol{\beta}$ which they contain. The task is easy when $\sigma_{\beta,1} > \sigma_{\beta,2}$, and hard in the opposite case.

on N training examples $\{\boldsymbol{x}_\mu \in \mathbb{R}^D\}_{\mu=1\dots N}$:

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \left[\epsilon_t(\boldsymbol{w})\right], \tag{3}$$

$$\epsilon_t(\boldsymbol{w}) = \sum_{\mu=1}^{N} \ell\left(y^\mu, \boldsymbol{w}\cdot\boldsymbol{z}_\mu\right) + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2. \tag{4}$$

We consider both regression tasks, where $\hat{f} = f^0 = \mathrm{id}$, and binary classification tasks, $\hat{f} = f^0 = \mathrm{sign}$. In the first case, ℓ is the square loss and the noise is additive. In the second case, ℓ can be any type of loss (square, hinge, logistic) and the noise amounts to random label flipping.

The generalization error is given by

$$\epsilon_g = \frac{1}{2^k}\,\mathbb{E}_{\boldsymbol{x},y}\left[\left(\hat{y}\left(\boldsymbol{x}\right) - y\right)^2\right] \tag{5}$$

with $k = 1$ for regression tasks (mean-square error) and $k = 2$ for binary classification tasks (zero-one error).

Strong and weak features model To impose structure on the dataset, we introduce a block model in which the elements of the inputs $\boldsymbol{x} \in \mathbb{R}^D$ and the teacher $\boldsymbol{\beta} \in \mathbb{R}^D$ are sampled from centered Gaussian distributions whose covariances have the following block structure :

$$\boldsymbol{x} \sim \mathcal{N}(0, \Sigma_x), \quad \Sigma_x = \begin{bmatrix} \sigma_{x,1}\mathbb{I}_{\phi_1 D} & 0 & 0 \\ 0 & \sigma_{x,2}\mathbb{I}_{\phi_2 D} & 0 \\ 0 & 0 & \ddots \end{bmatrix},$$

$$\boldsymbol{\beta} \sim \mathcal{N}(0, \Sigma_\beta), \quad \Sigma_\beta = \begin{bmatrix} \sigma_{\beta,1}\mathbb{I}_{\phi_1 D} & 0 & 0 \\ 0 & \sigma_{\beta,2}\mathbb{I}_{\phi_2 D} & 0 \\ 0 & 0 & \ddots \end{bmatrix}.$$

Although our theory is valid for an arbitrary number of blocks, we will focus for interpretability on the special case where we only have two blocks of sizes $\phi_1 D$ and $\phi_2 D$, with $\phi_1 + \phi_2 = 1$ (see Fig. 2). We will typically be interested in the strongly anisotropic setup where the first subspace is much smaller $\phi_1 \ll 1$, but potentially has higher *saliency* $r_x = \sigma_{x,1}/\sigma_{x,2} \gg 1$ and higher *relevance* $r_\beta = \sigma_{\beta,1}/\sigma_{\beta,2} \gg 1$.

Main analytical result Using the replica method from statistical physics [50] and the Gaussian Equivalence Theorem [24, 49, 51, 52], we derive the generalization error in the high-dimensional limit where D, N and $P \to \infty$ with fixed ratios. Our result extends the result of Gerace *et al.* [25] for isotropic data to anisotropic data; we additionally derive the effect of label flipping.

The asymptotic generalization error is given by

$$\lim_{N\to\infty} \epsilon_g = \frac{1}{2^k}\,\mathbb{E}_{\lambda,\nu}\left[\left(f^0(\nu) - \hat{f}(\lambda)\right)^2\right], \tag{6}$$

where λ, ν are jointly Gaussian scalars with mean zero and covariance:

$$\Sigma_{\lambda,\nu} = \begin{pmatrix} \rho & M \\ M & Q \end{pmatrix} \in \mathbb{R}^2, \tag{7}$$

with

$$\rho = \sum_i \phi_i \sigma_{\beta,i}\sigma_{x,i}, \quad M = \kappa_1 \sum_i \sigma_{x,i} m_{s,i},$$
$$\text{and} \quad Q = \kappa_1^2 \sum_i \sigma_{x,i} q_{s,i} + \kappa_\star^2 q_w. \tag{8}$$

where the parameters $\kappa_1, \kappa_\star$ are related to the activation function: denoting $r = \sum_i \phi_i \sigma_{x,i}$ and $\xi \sim \mathcal{N}(0, r)$, one has

$$\kappa_1 = \frac{1}{r}\mathbb{E}_\xi[\xi\sigma(\xi)]], \quad \kappa_\star = \sqrt{\mathbb{E}_\xi\left[\sigma(\xi)^2\right] - r\kappa_1^2}. \tag{9}$$

Besides these constants and the ones defining the data structure $(\phi_i, \sigma_{\beta,i}, \sigma_{x,i})$, the key ingredients to obtain the covariance elements in (8) are the so-called *order parameters* m_s, q_s, q_w. They correspond to the high-dimensional limit of the following overlaps:

$$\begin{aligned} m_{s,i} &= \lim_{D\to\infty} \frac{1}{D}\mathbb{E}_{\mathcal{P}}\left[\boldsymbol{s}_i \cdot \boldsymbol{\beta}_i\right] \\ q_{s,i} &= \lim_{D\to\infty} \frac{1}{D}\mathbb{E}_{\mathcal{P}}\left[\boldsymbol{s}_i \cdot \boldsymbol{s}_i\right], \\ q_w &= \lim_{P\to\infty} \frac{1}{P}\mathbb{E}_{\mathcal{P}}\left[\hat{\boldsymbol{w}} \cdot \hat{\boldsymbol{w}}\right], \end{aligned}$$

where $\boldsymbol{s} = \frac{1}{\sqrt{P}} F\hat{\boldsymbol{w}} \in \mathbb{R}^D$ and $\boldsymbol{s}_i, \boldsymbol{\beta}_i \in \mathbb{R}^{\phi_i D}$ denote the orthogonal projections of $\boldsymbol{s}$ and $\boldsymbol{\beta}$ onto subspace $i \in \{1, 2\}$.

The distribution $\mathcal{P}$ over which the expectations are taken is the joint distribution of all random quantities in the problem: the teacher weights $\boldsymbol{\beta}$, the random features $\boldsymbol{F}$ and the training data $\boldsymbol{x}$.

The order parameters are one of the outputs of the replica computation deferred to SM B. They are obtained by solving a set of non-linear equations (see Section B.4 of the SM). The replica formalism allows also to obtain the asymptotic training error, in a similar fashion to Gerace *et al.* [25]. In what follows, to gain more intuition, we focus on two simple cases.

Square loss regression We first consider the square loss regression setup $\ell(x, y) = (x - y)^2$ where $\hat{f} = f^0 = \mathrm{id}$ and the labels are corrupted by additive noise, i.e.

$$y_\mu = \frac{\boldsymbol{\beta} \cdot \boldsymbol{x}_\mu}{\sqrt{D}} + \eta_\mu, \quad \eta_\mu \sim \mathcal{N}(0, \Delta). \tag{10}$$

The expressions of the test and train error become very informative :

$$\epsilon_g = \frac{1}{2}\left(\rho + Q - 2M\right), \tag{11}$$

$$\epsilon_t = \frac{\epsilon_g + \Delta/2}{(1+V)^2}, \tag{12}$$

with

$$\begin{aligned} V &= \kappa_1^2 \sum_i \sigma_{x,i} v_{s,i} + \kappa_\star^2 v_w, \\ v_{s,i} &= \lim_{D\to\infty} \frac{1}{D}\mathbb{V}_{\mathcal{P}}\left[\boldsymbol{s}_i \cdot \boldsymbol{s}_i\right], \\ v_w &= \lim_{P\to\infty} \frac{1}{P}\mathbb{V}_{\mathcal{P}}\left[\hat{\boldsymbol{w}} \cdot \hat{\boldsymbol{w}}\right] \end{aligned}$$

where $\mathbb{V}_{\mathcal{P}}$ denotes the variance over all random quantities in the problem. Similarly, the values of the order parameters v_s, v_w are obtained by solving the set of equations given by the replica calculation.

Note here the surprisingly simple relation between the train loss and the test loss. One can invert this expression and write $\epsilon_g = \epsilon_t(1+V)^2 - \Delta/2$, showing that V acts as a variance opening a generalization gap.

Square loss classification Our framework is valid for any convex loss function, however the replica equations need to be evaluated numerically in the general case. For this reason, we focus here on the square loss classification setup[2] i.e. $\ell(x, y) = (x - y)^2$ where $\hat{f} = f^0 = \mathrm{sign}$, in which case the replica equations have a closed form which is both interpretable and easy to evaluate. In this case, the labels are corrupted by random label flipping of probability Δ, i.e.

$$y_\mu = \eta_\mu \,\mathrm{sign}\left(\frac{\boldsymbol{\beta} \cdot \boldsymbol{x}_\mu}{\sqrt{D}}\right), \quad \eta_\mu = \begin{cases} 1 \text{ w. prob. } 1-\Delta \\ -1 \text{ w. prob. } \Delta. \end{cases} \tag{13}$$

In this case, the expressions of the train and test error are:

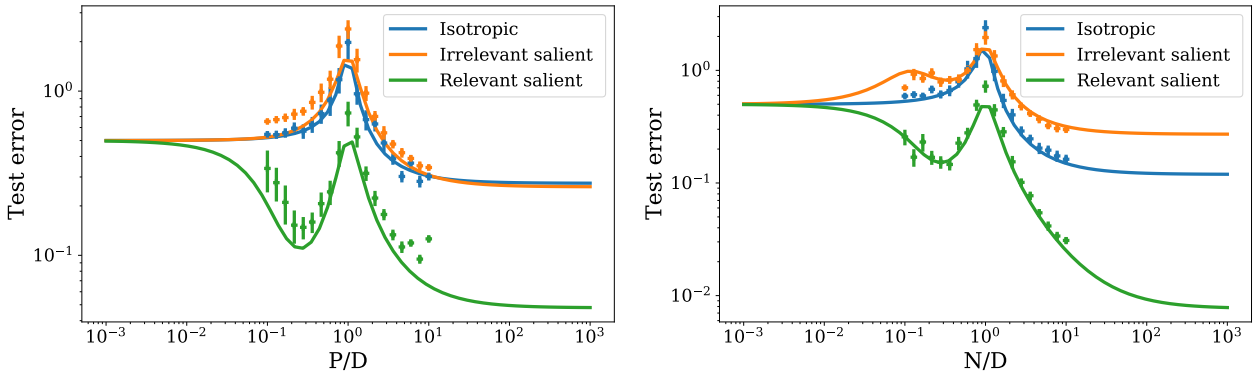
$$\epsilon_g = \frac{1}{\pi}\cos^{-1}\left(\frac{M}{\sqrt{\rho Q}}\right), \tag{14}$$

$$\epsilon_t = \frac{1}{2(1+V)^2}\left(\rho + Q - 2(1-2\Delta)M\right). \tag{15}$$

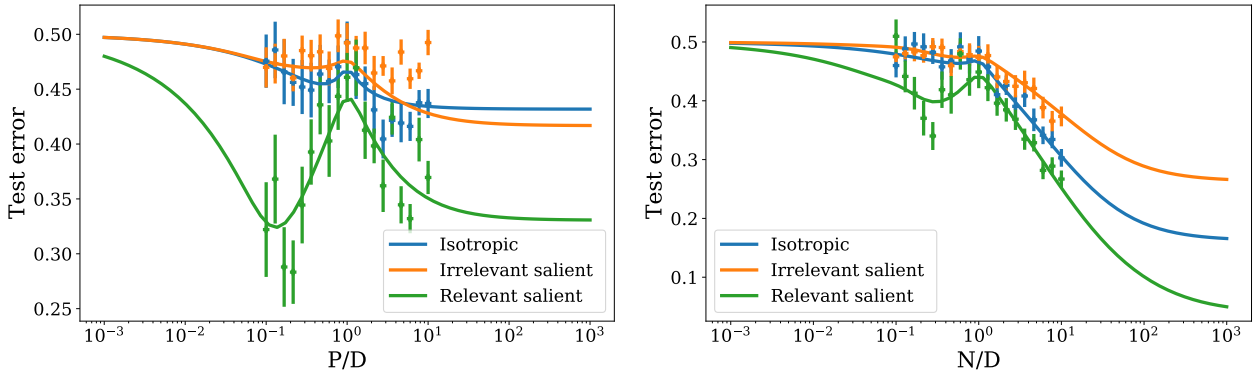
3 Effect of data structure on double descent

Before investigating the entire generalization phase space, we focus on "slices" by either fixing the number of training samples N and varying the number of

[2] Note that for datasets with few classes, recent works [53, 54] have shown that deep networks trained with MSE loss demonstrate comparable test-error performance to those trained with cross-entropy



(a) Regression



(b) Classification

Figure 3: **Anisotropic data strongly affects the double descent curves.** Theoretical results (solid curves) and numerical results (dots with vertical bars denoting standard deviation over 10 runs) agree even at moderate size $D = 100$ for both regression (upper row) and classification (lower row) tasks. We set $\sigma = \mathrm{Tanh}$, $\lambda = 10^{-3}$ and $\Delta = 0.3$. For the blue curve $r_x = r_\beta = 1$, for the orange curve $r_x = 10$ and $r_\beta = 0.01$, for the green curve $r_x = 10$ and $r_\beta = 100$. In all cases $\phi_1 = 0.1$. *Left:* parameter-wise profile at $N/D = 1$. *Right:* sample-wise profile at $P/D = 1$.

parameters P or conversely fixing P and varying N. The main issue here is to investigate how the data structure affects the test error curves, and in particular the parameter-wise and sample-wise double descent phenomenon found for isotropic data.

We compare three cases. The first is the *isotropic* setup where $r_x = 1$ (blue curves in Fig. 3). In the two next setups, the data is anisotropic with a small subspace ($\phi_1 = 0.1$) of large variance and a large subspace ($\phi_2 = 0.9$) of small variance. The ratio of the variances $r_x = \sigma_{x,1}/\sigma_{x,2}$ is set to 10, and their values are chosen to keep the total variance of the inputs unchanged: $\frac{1}{D}\,\mathbb{E}\,\|\boldsymbol{x}\|^2 = \phi_1\sigma_{x,1} + \phi_2\sigma_{x,2} = 1$.

We study two cases for the outputs: (i) the *relevant salient features* scenario where the strong subspace is the most relevant ($r_\beta = \sigma_{\beta,1}/\sigma_{\beta,2} = 100$, green curves in Fig. 3); (ii) the *irrelevant salient features* scenario, where the strong subspace has low relevance ($r_\beta = 0.01$, orange curves in Fig. 3), i.e. contributes very little to the output. In both cases, we choose the $\sigma_{\beta,i}$ such that the total variance of the outputs is unchanged: $\mathbb{E}\,y^2 = \phi_1\sigma_{x,1}\sigma_{\beta,1} + \phi_2\sigma_{x,2}\sigma_{\beta,2} = 1$.

Validity at finite size First we validate our main result by comparing our analytical predictions with the outcomes of numerical experiments, both for test loss (Fig. 3) and the train loss (Fig. 6 of SM A.1). The agreement is already excellent for moderately large dimensions $D = 100$, both in the parameter-wise curves and the sample-wise curves[3]. Note that the replica method, which relies on solving a set of scalar fixed point equations, is also computationally efficient. It allows here to probe ratios of P/D and

[3]In the classification setup, the variance of the numerical experiments is increased by the presence of label flipping.

N/D far beyond what is tractable by the numerics (P, D and N only appear in the replica equations through the values of the ratios N/P and P/D). We also report the behavior of the order parameters M, Q and V in SM B.4.

Relevant salient features When the salient features are relevant, the anisotropy of the data makes the task easier. This can be seen both in the parameter-wise profile: the test loss drops at $P = \phi_1 D$ (instead of $P = D$), and in the sample-wise profile: the test loss drops at $N = \phi_1 D$ (instead of $N = D$). These observations are in line with the results of Ghorbani *et al.* [31] and show that relevant salient features make the problem low-dimensional with an effective dimension close to $\phi_1 D$. The remaining features act as additional noise in the data. As explained in the former reference, this setup is the most akin to real-world tasks. For example, the most salient Fourier features of an image are often the most relevant to its recognition. In this sense, the impressive performance of kernel methods such as the Convolutional NTK on real-world datasets [55] can be associated with the anisotropy of the data: feature learning is not required to beat the curse of dimensionality if the irrelevant features are weakly salient to begin with.

Irrelevant salient features We focus next on the case where the salient features are weakly relevant (orange curve in Fig. 3). In this case, these features carry very little signal and mainly act as a troublesome source of noise in input space. Hence, in the sample-wise curves, the test loss is always higher than the isotropic baseline (blue in Fig. 3), showing the detrimental effect of these features. In the regression setup, we even observe a second peak appear, forming a sample-wise triple descent curve: the salient irrelevant subspace creates a "linear peak" [56] at $N/D = \phi_1$, at which the system overfits the noise.

In the parameter-wise curves, an interesting crossover occurs: the irrelevant features are detrimental from small P, but become helpful at large P, as can be seen from the orange curve reaching a lower asymptotic value than the blue curve in the left panels of Fig. 3. We associate this to the phenomenon discovered for linear regression in Richards *et al.* [26], whereby adding noisy features acts as a form of implicit regularization.

4 More data or more parameters?

In this section, we investigate both analytically and empirically how the structure of data shapes the generalization phase space. We first address this issue in the random feature model then confirm our findings in numerical experiments on real data.

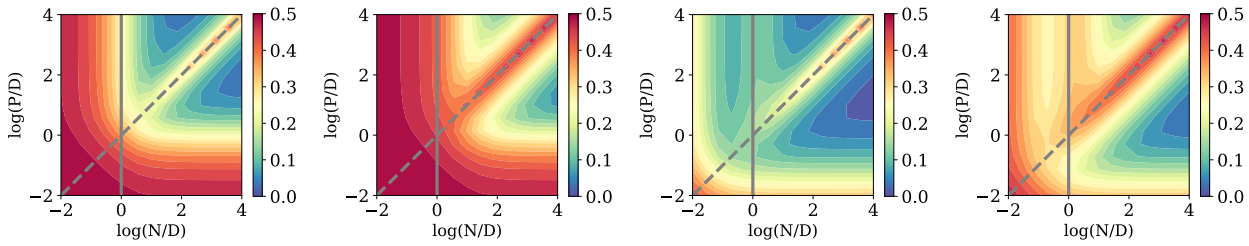
4.1 Random feature model

We display the test error of the random feature model in the (N, P)-phase space in Fig. 4, focusing on the classification setting. In the noiseless isotropic setup, we see that the phase space is quasi-symmetric around the line $N = P$ (panel (a), Fig. 4). We observe the usual overfitting peak of double descent along the interpolation threshold $N \sim P$ (dashed line, Fig. 4).

Noisy labels A clear asymmetry emerges when corrupting the labels (panel (b), Fig. 4). In this case, the test error becomes higher everywhere, but particularly in two regions: the small-data region at $N \sim D$, marked by the vertical solid line, and near the interpolation threshold $N \sim P$, along the diagonal dashed line. In this setting, many data and few parameters becomes better in terms of generalization than many parameter-few data.

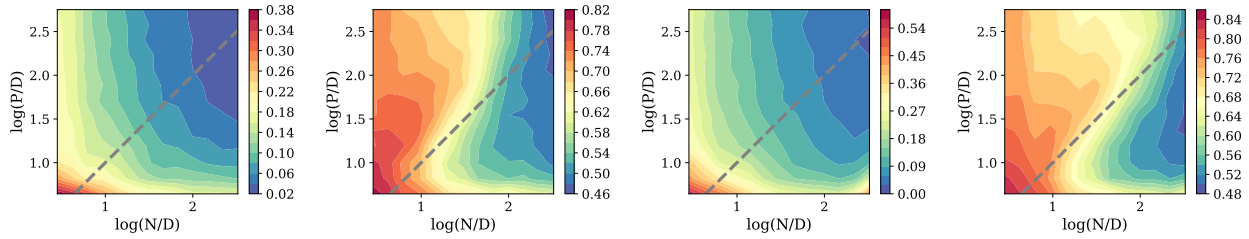
Structure of the inputs When we make the input data anisotropic, in the relevant salient features setup, asymmetry emerges again (panel (c), Fig. 4). In this case, the test error becomes lower everywhere: we saw already that the task becomes low-dimensional hence easier. However, despite the absence of noise in the labels, we see a vertical overfitting peak along $N \sim D$ (dashed line, Fig. 4). As we discussed in the previous section, the non-salient features induce an effective noise in the data, which give rise to a "triple descent" phase space, as was observed in [56] for the regression setup. Fig. 7 of the SM shows how the disymmetry of the phase-space builds as we increase the anisotropy r_x from 1, then vanishes when $r_x \to \infty$ as the non-salient features vanish, leaving an isotropic task of dimensionality ϕ_1.

Finally, the asymmetry in the generalization phase space is strongest when the task is both noisy and anisotropic (panel (d), Fig. 4). In this case, most of the overparametrized region $P > N$ is affected by overfitting: optimal performance is clearly achieved in the underparametrized region $P < N$.

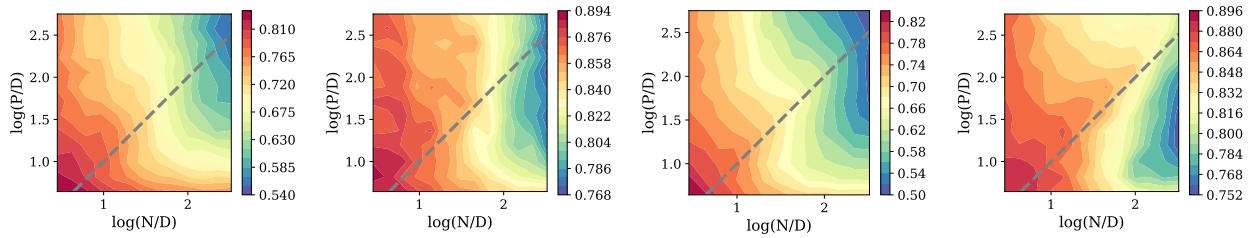


(a) Isotropic, noiseless (b) Isotropic, $\Delta = 0.4$ (c) Anisotropic, $\Delta = 0$ (d) Anisoptropic, $\Delta = 0.4$

Figure 4: **The symmetry of the generalization phase space vanishes when the data becomes noisy or anisotropic.** We studied the classification task for $\sigma = \mathrm{ReLU}$, $\lambda = 0.1$. In the anisotropic phase spaces we set $\phi_1 = 0.01$, $r_x = 100$, $r_\beta = 100$.



(a) MNIST, noiseless (b) MNIST, 50% corruption (c) MNIST PCA, noiseless (d) MNIST PCA, 50% corr.



(e) CIFAR10, noiseless (f) CIFAR10, 50% corruption (g) CIFAR10 PCA (h) CIFAR10 PCA, 50% corr.

Figure 5: **Noisy labels and PCA transformation break the symmetry of the phase space.** *Top:* MNIST, downsampled to $D = 10 \times 10$ images. *Bottom:* CIFAR10, downsampled to $D = 10 \times 10$ images. In all cases, we represented the test error at convergence of 2-layer networks trained via gradient descent. The results are averaged over 10 runs.

Sources of variance The roles of N and P can be understood in terms of a bias-variance decomposition [57, 58]. By overparametrizing, we average out the variance due to the randomness of the random features $\{\boldsymbol{F}_i\}_{i=1\dots P}$, whereas by increasing the number of training examples we average out the variance due to (1) the randomness of the inputs $\{\boldsymbol{x}_\mu\}_{\mu=1\dots N}$ and (2) the noise in the labels $\{\eta_\mu\}_{\mu=1\dots N}$.

In the isotropic noiseless setup, the random features and the input data follow the same distribution and the roles of N and P are symmetric since the random feature projection is given by $\sigma(\boldsymbol{F}_i \cdot \boldsymbol{x}_\mu/\sqrt{D})$. However, in presence of corrupted labels or irrelevant inputs, increasing N also averages out the extra sources of randomness in the data, and hence has a stronger effect on the test error than increasing P.

4.2 Neural networks

We now turn to two-layer networks trained on MNIST and CIFAR10 to see if our analytical findings hold in a more realistic setting. To unify the

dimensions and explore the (N, P)-phase space consistently, we downscale the images to 10×10 pixels, such that $D = 100$. We train using full-batch gradient descent for 3000 epochs, with a learning rate of 0.05 and momentum 0.9. The width H of the hidden layer is varied from 4 to 512, and the number of examples N in the dataset is varied from 256 to 32k. Note that in this case, $P = (D + 1)H$ denotes the total number of parameters in both layers, therefore we only have access to the region $P > D$ of the landscape.

As can be seen in panels (a) and (e) of Fig. 5, the test error phase space is almost symmetric around the interpolation threshold both for MNIST and CIFAR10, despite the vastly different magnitudes of the test error (down to 2% for MNIST, against 54% for CIFAR10). The double descent phenomenon is absent because of the training algorithm: gradient descent with finite-learning rate and momentum exerts an implicit regularization [59]. Hence, the test error monotonically decays both as we increase N and P. In order to confirm the role played by the data in shaping the generalization phase space we now study the effect of corrupting the labels and modifying the input distribution of the datasets.

Noisy labels We corrupt the data by shuffling 50% of the labels of the training set, leaving the test set clean. This causes the lowest test error to rise to 46% for MNIST and 76% for CIFAR10. The effect on the corresponding phase spaces are shown in panels (b) and (f) of Fig. 5. As observed analytically, the increase in test error is higher in the overparametrized region $P > N$ due to overfitting, which causes the phase space to appear less symmetric.

Structure of the inputs In order to make the data anisotropic, we apply a PCA transformation to the original datasets, then retain the top $D = 100$ principal components as inputs. In this way, the first few features contain most of the relevant information, whereas the last features are essentially pure noise. Just like in the salient relevant features setup, generalization is improved by this preprocessing procedure: the minimum test error falls to 1% for MNIST and 50% on CIFAR10. Similarly to the case of noisy labels, we observe in panels (c) and (g) of Fig. 5 that the phase space becomes slightly less symmetric.

Finally, the asymmetry is strongest in the rightmost panels, where we applied the PCA transformation on top of corrupting the labels. As for the random features setup, the whole overparametrized region $P > N$ suffers from overfitting, particularly along the lines $N \sim P$ and near $N \sim D$. Again, having a lot of data is more valuable than having a lot of parameters.

5 Conclusion

Analytical explorations on isotropic data and empirical results on simple datasets tend to show that increasing model size is of same value as increasing sample size. However, this symmetry appears to be due to the "clean" nature of the data considered and vanishes as soon as the inputs or the labels are noisy. We have shown that in such scenarios, increasing sample size becomes more valuable than increasing model size since it allows to average out the noise. These contrasting conclusions illustrate the need for a thorough analysis of the impact of data structure on learning, which may ultimately lead to more interpretable and explainable systems. Bridging the gap between what is theoretically tractable and what works in practice is a challenging task that will also motivate our future work. In the current paper, we investigated independent input variables. Future directions include the investigation of the structure of data with correlations within or across subspaces, and the study of representations of data.

Acknowledgements We thank Berfin Simsek, Armand Joulin, Bruno Loureiro, Federica Gerace for helpful discussions. SD and GB acknowledge funding from the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute). MG acknowledges funding from the Flatiron Institute. We also thank Lenka Zdeborova and Florent Krzakala, and Les Houches Summer School for organizing a workshop on Statistical Physics and Machine Learning where this work was partially done.

A Further analytical results results

A.1 Train error

In Fig. 6, we provide the training error curves corresponding to the scenarii studied in Section 3 of the main text.

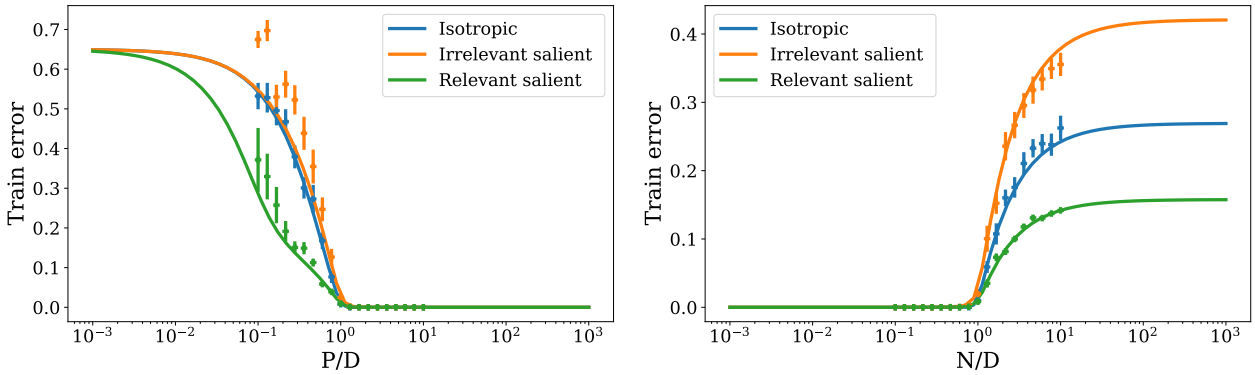
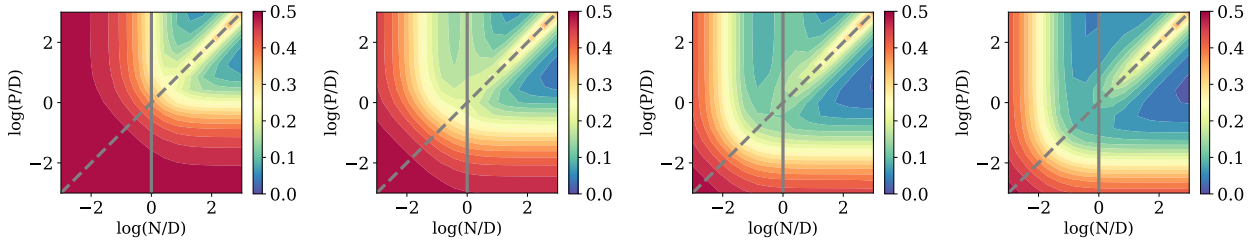


Figure 6: Train error parameter-wise and sample-wise profiles. Theoretical results (solid curves) and numerical results (dots with vertical bars denoting standard deviation over 10 runs) agree even at moderate size $D = 100$ in both cases. We study the regression task for $\sigma = \mathrm{Tanh}$, $\lambda = 10^{-3}$ and $\Delta = 0.3$. For the blue curve $r_x = r_\beta = 1$, for the orange curve $r_x = 10$ and $r_\beta = 0.01$, for the green curve $r_x = 10$ and $r_\beta = 100$. In all cases $\phi_1 = 0.1$. *Left:* parameter-wise profile at $N/D = 1$. *Right:* sample-wise profile at $P/D = 1$.

A.2 Phase space

Fig. 7 illustrates the modification of the phase space of the random features model trained on the strong and weak features model as the saliency r_x of the $\phi_1 D = 0.01D$ relevant features ($r_\beta = 1000$) is gradually increased. At $r_x = 1$ (panel (a)), the data is isotropic and the phase space is symmetric. When $r_x \to \infty$, all the variance goes in the salient subspace and we are left with an isotropic task of dimensionality $\phi_1 D$: the phase space is symmetric again (panel (d)). In between these two extreme scenarios, we see the asymmetry appear in the phase space under the form of an overfitting peak at $N = D$, as the irrelevant features come into play (panels (b) and (c)).



(a) $r_x = 1$ (b) $r_x = 10$ (c) $r_x = 100$ (d) $r_x = 1000$

Figure 7: Increasing the saliency of the first subspace from 1 to ∞, the asymmetry forms then vanishes as the data becomes effectively isotropic in smaller dimension. We study the classification task for $\sigma = \mathrm{ReLU}$, $\lambda = 0.1$, $\phi_1 = 0.01$, $r_\beta = 1000$ and $\Delta = 0$.

B Analytical derivations

The following sections present the analytical derivations of this work. First, we provide an outline of the computation of the generalization error. Second, we describe our extension of the Gaussian Equivalence Theorem to anisotropic data, a key ingredient to handle the strong and weak features model. The three following sections develop the steps of the replica computation, from the Gibbs formulation of the learning objective, to the set of self-consistent equations to obtain the order parameters. Lastly, we explain how to also obtain the asymptotic training loss from the output of the replica computation.

Setup and notations We recall notations: D is the input dimension, P is the number of random features and N is the number of training examples. We consider the high-dimensional limit where $D, P, N \to \infty$ and $\gamma = \frac{D}{P}$, $\alpha = \frac{N}{P}$ are of order one. The inputs $\boldsymbol{x} \in \mathbb{R}^D$ and the elements of the teacher vectors $\boldsymbol{\beta} \in \mathbb{R}^D$ are sampled from a block-structured covariance matrix :

$$\boldsymbol{x} \sim \mathcal{N}(0, \Sigma_x), \quad \Sigma_x = \begin{bmatrix} \sigma_{x,1}\mathbb{I}_{\phi_1 D} & 0 & 0 \\ 0 & \sigma_{x,2}\mathbb{I}_{\phi_2 D} & 0 \\ 0 & 0 & \ddots \end{bmatrix} \tag{16}$$

$$\boldsymbol{\beta} \sim \mathcal{N}(0, \Sigma_\beta), \quad \Sigma_\beta = \begin{bmatrix} \sigma_{\beta,1}\mathbb{I}_{\phi_1 D} & 0 & 0 \\ 0 & \sigma_{\beta,2}\mathbb{I}_{\phi_2 D} & 0 \\ 0 & 0 & \ddots \end{bmatrix} \tag{17}$$

$$\tag{18}$$

In the main text, we study only the strong and weak features scenario where we only have two blocks, but our derivation will be performed in full generality.
In the following, we denote as $\boldsymbol{x}|_i, \boldsymbol{\beta}|_i$ the orthogonal projection of $\boldsymbol{x}$ and $\boldsymbol{\beta}$ onto the subspace i of $\mathbb{R}^D$. For example, $\boldsymbol{x}|_1$ amounts to the first $\phi_1 D$ components of $\boldsymbol{x}$.

Outline of the derivation of the generalization error. The key observation is that the test error can be rewritten, in the high-dimensional limit, in terms of so-called *order parameters*:

$$\epsilon_g = \frac{1}{2^k}\mathbb{E}_{\boldsymbol{x}^{\text{new}}, y^{\text{new}}}\left(y^{\text{new}} - \hat{f}\left(\frac{1}{\sqrt{P}}\sigma\left(\frac{\mathrm{F}^\top \boldsymbol{x}^{\text{new}}}{\sqrt{D}}\right)\cdot \hat{\boldsymbol{w}}\right)\right)^2 \tag{19}$$

$$= \frac{1}{2^k}\int \mathrm{d}\nu \int \mathrm{d}\lambda P(\nu,\lambda)(f^0(\nu) - \hat{f}(\lambda))^2 \tag{20}$$

where we defined

$$\lambda = \frac{1}{\sqrt{P}}\sigma\left(\frac{\mathrm{F}^\top \boldsymbol{x}^{\text{new}}}{\sqrt{D}}\right)\cdot \hat{\boldsymbol{w}} \in \mathbb{R}, \quad \nu = \frac{1}{\sqrt{D}}\boldsymbol{x}^{\text{new}}\cdot \boldsymbol{\theta}^0 \in \mathbb{R} \tag{21}$$

and the joint probability distribution

$$P(\nu,\lambda) = \mathbb{E}_{\boldsymbol{x}^{\text{new}}}\left[\delta\left(\nu - \frac{1}{\sqrt{D}}\boldsymbol{x}^{\text{new}}\cdot\boldsymbol{\beta}\right)\delta\left(\lambda - \frac{1}{\sqrt{P}}\sigma\left(\frac{\mathrm{F}^\top \boldsymbol{x}^{\text{new}}}{\sqrt{D}}\right)\cdot\hat{\boldsymbol{w}}\right)\right]. \tag{22}$$

The key objective to calculate the test error is therefore to obtain the joint distribution $P(\nu,\lambda)$. To do so, our derivation is organized as follows.

1. In Sec. B.1, we adapt the Gaussian equivalence theorem [24, 51, 52, 60] to the case of anisotropic data. The latter shows that $P(\nu,\lambda)$ is a joint gaussian distribution whose covariance depends only on the following *order parameters*:
$$m_{s,i} = \frac{1}{D}\boldsymbol{s}|_i\cdot\boldsymbol{\beta}|_i, \quad q_{s,i} = \frac{1}{D}\boldsymbol{s}|_i\cdot\boldsymbol{s}|_i, \quad q_w = \frac{1}{P}\hat{\boldsymbol{w}}\cdot\hat{\boldsymbol{w}},$$
where we denoted $\boldsymbol{s} = \frac{1}{\sqrt{P}}\boldsymbol{F}\hat{\boldsymbol{w}} \in \mathbb{R}^D$ and the index i refers to the subspace of $\mathbb{R}^D$ the vectors are projected onto.

2. In Sec. B.2, we recast the optimization problem as a Gibbs measure over the weights, from which one can sample the average value of the order parameters m_s, q_s, q_w. Thanks to the convexity of the problem, this measure concentrates around the solution of the optimization problem in the limit of high temperature, see [25].

3. In Sec. B.3, we leverage tools from random matrix theory to derive the saddle-point equations allowing the obtain the values of the order parameters

4. In Sec. B.3, we give the explicit expressions of the saddle-point equations for the two cases studied in the main text: square loss regression and classification

5. In Sec. B.5, we also show how to obtain the train error from the order parameters.

Once the order parameters are known, the joint law of λ, ν is determined as explained in the main text:

$$P(\lambda, \nu) = \mathcal{N}(0, \Sigma), \quad \Sigma = \begin{pmatrix} \rho & M \\ M & Q \end{pmatrix}. \tag{23}$$

It is then easy to perform the Gaussian integral giving the generalization error.
For the regression problem where $\hat{f} = f^0 = \mathrm{id}$, we have:

$$\epsilon_g = \frac{1}{2}\left(\rho + Q - 2M\right)$$

For the classification problem where $\hat{f} = f^0 = \mathrm{sgn}$, we have:

$$\epsilon_g = \frac{1}{\pi}\cos^{-1}\left(\frac{M}{\sqrt{\rho Q}}\right)$$

.

B.1 The anisotropic Gaussian Equivalence Theorem

We now present the derivation of the anisotropic Gaussian Equivalence Theorem. This is a key ingredient for the replica analysis presented in the next section.
Define, for a Gaussian input vector $\boldsymbol{x} \in \mathbb{R}^D$, the family of vectors indexed by $a = 1 \dots r$:

$$\lambda^a = \frac{1}{\sqrt{P}}\boldsymbol{w}^a \cdot \sigma\left(\frac{\boldsymbol{F}\boldsymbol{x}}{\sqrt{D}}\right) \in \mathbb{R}, \quad \nu = \frac{1}{\sqrt{D}}\boldsymbol{x} \cdot \boldsymbol{\beta} \in \mathbb{R} \tag{24}$$

Using the Gaussian equivalence principle, we can obtain expectancies for this family in the high-dimensional limit:

$$(\nu, \lambda^a) \sim \mathcal{N}(0, \Sigma), \quad \Sigma^{ab} = \begin{pmatrix} \rho & M^a \\ M^a & Q^{ab} \end{pmatrix} \in \mathbb{R}^{(r+1)\times(r+1)} \tag{25}$$

$$\rho = \sum_i \phi_i \beta|_i \sigma_{x,i}, \quad M^a = \kappa_1 \sum_i \sigma_{x,i} m^a_{s,i}, \quad Q^{ab} = \kappa_1^2 \sum_i \sigma_{x,i} q^{ab}_{s,i} + \kappa_\star^2 q^{ab}_w \tag{26}$$

$$m^a_{s,i} = \frac{1}{D}\boldsymbol{s}^a|_i \cdot \boldsymbol{\beta}|_i, \quad q^{ab}_{s,i} = \frac{1}{D}\boldsymbol{s}^a|_i \cdot \boldsymbol{s}^b|_i, \quad q^{ab}_w = \frac{1}{P}\boldsymbol{w}^a \cdot \boldsymbol{w}^b \tag{27}$$

where

$$\kappa_0 = \mathop{\mathbb{E}}_{z\sim\mathcal{N}(0,r)}[\sigma(z)], \quad \kappa_1 = \frac{1}{r}\mathop{\mathbb{E}}_{z\sim\mathcal{N}(0,r)}[z\sigma(z)]], \quad \kappa_\star = \sqrt{\mathop{\mathbb{E}}_{z\sim\mathcal{N}(0,\sum_i \sigma_{x,i}\phi_i)}[\sigma(z)^2] - \kappa_0^2 - r\kappa_1^2} \tag{28}$$

$$r = \sum_i \phi_i \sigma_{x,i}, \quad \boldsymbol{s}^a = \frac{1}{\sqrt{P}}\boldsymbol{F}\boldsymbol{w}^a \in \mathbb{R}^D, \quad \boldsymbol{s}^a_i = P_{E_i}\boldsymbol{s}^a, \quad \boldsymbol{\beta}|_i = P_{E_i}\boldsymbol{\beta} \tag{29}$$

$$\tag{30}$$

Isotropic setup Let us start in the setup where we only have one block, of unit variance.
By rotational invariance, we can write :

$$\mathbb{E}_{\boldsymbol{x}}\left[\sigma\left(\frac{\boldsymbol{x}\cdot F_\mu}{\sqrt{D}}\right)\sigma\left(\frac{\boldsymbol{x}\cdot F_\nu}{\sqrt{D}}\right)\right] = f\left(\frac{F_\mu\cdot F_\nu}{D}\right)\equiv f(q_{\mu\nu}) \tag{31}$$

$$= \delta_{\mu\nu}f(1) + (1-\delta_{\mu\nu})\left(f(0)+f'(0)q_{\mu\nu}\right) + o\left(q_{\mu\nu}\right) \tag{32}$$

$$= f(0) + f'(0)q_{\mu\nu} + \delta_{\mu\nu}(f(1)-f(0)-f'(0)) \tag{33}$$

Therefore, the expectancy is the same as the "Gaussian equivalent model" where we replace

$$\sigma\left(\frac{\boldsymbol{x}\cdot F_\mu}{\sqrt{D}}\right)\to\sqrt{f(0)}+\sqrt{f'(0)}\left(\frac{\boldsymbol{x}\cdot F_\mu}{\sqrt{D}}\right)+\sqrt{(f(1)-f(0)-f'(0)}\xi_\mu,\quad \xi_\mu\sim\mathcal{N}(0,1) \tag{34}$$

What remains is to find the expression of $f(0), f(1)$ and $f'(0)$. If $q_{\mu\nu}=1$, then F_μ and F_ν are the same vector and clearly $f(1)=\mathbb{E}_z\left[\sigma(z)^2\right]$.
Otherwise, we use the fact that $x_\mu=\frac{\boldsymbol{x}\cdot F_\mu}{\sqrt{D}}$ and $x_\nu=\frac{\boldsymbol{x}\cdot F_\nu}{\sqrt{D}}$ are correlated gaussian variables:

$$\mathbb{E}\left[x_\mu^2\right]=1,\qquad \mathbb{E}\left[x_\mu x_\nu\right]=\frac{F_\mu\cdot F_\nu}{D}\equiv q_{\mu\nu}\sim O\left(\frac{1}{\sqrt{D}}\right) \tag{35}$$

Therefore we can parametrize as follows,

$$x_\mu=\eta_\mu\sqrt{r-q_{\mu\nu}}+z\sqrt{q_{\mu\nu}}\sim\eta_\mu(1-\frac{1}{2}q_{\mu\nu})+z\sqrt{q_{\mu\nu}} \tag{36}$$

$$x_\nu=\eta_\nu\sqrt{r-q_{\mu\nu}}+z\sqrt{q_{\mu\nu}}\sim\eta_\mu(1-\frac{1}{2}q_{\mu\nu})+z\sqrt{q_{\mu\nu}} \tag{37}$$

$$\tag{38}$$

where $\eta_\mu,\eta_\nu,z\sim\mathcal{N}(0,1)$. To calculate $f(0), f'(0)$, we can expand the nonlinearity,

$$f(q_{\mu\nu})\equiv f(0)+f'(0)q_{\mu\nu}+o(q_{\mu\nu}^2) \tag{39}$$

$$= \mathbb{E}\left[\left(\sigma(\eta_\mu)+\sigma'(\eta_\mu)\left(-\frac{q}{2}\eta_\mu+z\sqrt{q}_{\mu\nu}\right)+\frac{\sigma''(\eta_\mu)}{2}\left(z^2q_{\mu\nu}\right)\right)\right. \tag{40}$$

$$\left.\left(\sigma(\eta_\nu)+\sigma'(\eta_\nu)\left(-\frac{q}{2}\eta_\nu+z\sqrt{q}_{\mu\nu}\right)+\frac{\sigma''(\eta_\nu)}{2}\left(z^2q_{\mu\nu}\right)\right)\right] \tag{41}$$

$$= \mathbb{E}_\eta\left[\sigma(\eta)\right]^2+q_{\mu\nu}\left(\mathbb{E}[\sigma'(\eta)]^2+\mathbb{E}[\sigma''(\eta)]\mathbb{E}[\sigma(\eta)]-\mathbb{E}[\sigma'(\eta)\eta]\mathbb{E}[\sigma(\eta)]\right)+o(q_{\mu\nu}^2) \tag{42}$$

$$\tag{43}$$

But since $\mathbb{E}[\sigma(\eta)\eta]=\mathbb{E}[\sigma'(\eta)]$, the two last terms cancel out and we are left with

$$f(0)=\mathbb{E}[\sigma(z)]^2,\qquad f'(0)=\mathbb{E}[\sigma(z)\eta]^2 \tag{44}$$

$$\mathbb{E}_{\boldsymbol{x}}\left[\sigma\left(\frac{\boldsymbol{x}\cdot F_\mu}{\sqrt{D}}\right)\sigma\left(\frac{\boldsymbol{x}\cdot F_\nu}{\sqrt{D}}\right)\right]=\kappa_0^2+\kappa_1^2\frac{F_\mu\cdot F_\nu}{D}+\kappa_\star^2\delta_{\mu\nu} \tag{45}$$

Anisotropic setup Now in the block setup,

$$\mathbb{E}\left[x_\mu^2\right]=\sum_i\sigma_{x,i}\phi_i\equiv r,\qquad \mathbb{E}\left[x_\mu x_\nu\right]=\sum_i\sigma_{x,i}\frac{F_\mu^i\cdot F_\nu^i}{D}=\sum_i\sigma_{x,i}q_{\mu\nu}^i\equiv q_{\mu\nu}\sim O\left(\frac{1}{\sqrt{D}}\right) \tag{46}$$

Therefore we can parametrize as follows,

$$x_\mu = \eta_\mu \sqrt{1 - \frac{q_{\mu\nu}}{r}} + z\sqrt{\frac{q_{\mu\nu}}{r}} \sim \eta_\mu(1 - \frac{1}{2}\frac{q_{\mu\nu}}{r}) + z\sqrt{\frac{q_{\mu\nu}}{r}} \tag{47}$$

$$x_\nu = \eta_\nu \sqrt{1 - \frac{q_{\mu\nu}}{r}} + z\sqrt{\frac{q_{\mu\nu}}{r}} \sim \eta_\mu(1 - \frac{1}{2}\frac{q_{\mu\nu}}{r}) + z\sqrt{\frac{q_{\mu\nu}}{r}} \tag{48}$$

$$\tag{49}$$

where $\eta_\mu, \eta_\nu, z \sim \mathcal{N}(0, r)$. As before,

$$f(q_{\mu\nu}) = \mathbb{E}_\eta \left[\sigma(\eta)\right]^2 + \frac{q_{\mu\nu}}{r}\left(\mathbb{E}[\sigma'(\eta)]^2\,\mathbb{E}[z^2] + \mathbb{E}[\sigma''(\eta)]\mathbb{E}[\sigma(\eta)]\,\mathbb{E}[z^2] - \mathbb{E}[\sigma'(\eta)\eta]\mathbb{E}[\sigma(\eta)]\right) + o(q_{\mu\nu}^2) \tag{50}$$

$$\tag{51}$$

Now since $\mathbb{E}[z^2] = r$ and $\mathbb{E}[\sigma(\eta)\eta] = r\mathbb{E}[\sigma'(\eta)]$, the two last terms do not cancel out like before and we have :

$$f(0) = \mathbb{E}[\sigma(z)]^2, \qquad f'(0) = \frac{1}{r}\mathbb{E}[\sigma(z)\eta]^2\mathbb{E}[z^2] = \frac{1}{r^2}\mathbb{E}[\sigma(z)z]^2 \tag{52}$$

Finally we have

$$\mathbb{E}_{\boldsymbol{x}}\left[\sigma\left(\frac{\boldsymbol{x}\cdot F_\mu}{\sqrt{D}}\right)\sigma\left(\frac{\boldsymbol{x}\cdot F_\nu}{\sqrt{D}}\right)\right] = \kappa_0^2 + \kappa_1^2\sum_i \sigma_{x,i}\frac{F_\mu^i \cdot F_\nu^i}{D} + \kappa_\star^2\delta_{\mu\nu} \tag{53}$$

$$\kappa_0 = \mathop{\mathbb{E}}_{z\sim\mathcal{N}(0,r)}[\sigma(z)], \quad \kappa_1 = \frac{1}{r}\mathop{\mathbb{E}}_{z\sim\mathcal{N}(0,r)}[z\sigma(z)]], \quad \kappa_\star = \sqrt{\mathop{\mathbb{E}}_{z\sim\mathcal{N}(0,r)}[\sigma(z)^2] - \kappa_0^2 - r\kappa_1^2} \tag{54}$$

$$\tag{55}$$

B.2 Gibbs formulation of the problem

To obtain the test error, we need to find the typical value of the order parameters m_s, q_s, q_w. To do so, we formulate the optimization problem for the second layer weights as a Gibbs measure over the weights as in [25]:

$$\mu_\beta\left(\boldsymbol{w} \mid \{\boldsymbol{x}^\mu, y^\mu\}\right) = \frac{1}{\mathcal{Z}_\beta}e^{-\beta\left[\sum_{\mu=1}^N \ell(y^\mu, \boldsymbol{x}^\mu\cdot\boldsymbol{w}) + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2\right]} = \frac{1}{\mathcal{Z}_\beta}\underbrace{\prod_{\mu=1}^N e^{-\beta\ell(y^\mu,\boldsymbol{x}^\mu\cdot\boldsymbol{w})}}_{\equiv P_y(\boldsymbol{y}\mid\boldsymbol{w}\cdot\boldsymbol{x}^\mu)}\underbrace{\prod_{i=1}^P e^{-\frac{\beta\lambda}{2}w_i^2}}_{\equiv P_w(\boldsymbol{w})} \tag{56}$$

Of key interest is the behavior of the free energy density, which is self-averaging in the high-dimensional limit and whose minimization gives the optimal value of the overlaps:

$$f_\beta = -\lim_{P\to\infty}\frac{1}{P}\,\mathbb{E}_{\{\boldsymbol{x}^\mu, y^\mu\}}\log\mathcal{Z}_\beta \tag{57}$$

To calculate the latter, we use the Replica Trick from statistical physics:

$$\mathbb{E}_{\{\boldsymbol{x}^\mu, y^\mu\}}\log\mathcal{Z}_\beta = \lim_{r\to 0}\frac{D}{Dr}\mathbb{E}_{\{\boldsymbol{x}^\mu, y^\mu\}}\mathcal{Z}_\beta^r \tag{58}$$

The right hand side can be written in terms of the *order parameters* :

$$\mathbb{E}_{\{\boldsymbol{x}^\mu, y^\mu\}}\,\mathcal{Z}_\beta^r = \int\frac{\mathrm{d}\rho\mathrm{d}\hat{\rho}}{2\pi}\int\prod_{a=1}^r\frac{\mathrm{d}m_s^a\mathrm{d}\hat{m}_s^a}{2\pi}\int\prod_{1\le a\le b\le r}\frac{\mathrm{d}q_s^{ab}\mathrm{d}\hat{q}_s^{ab}}{2\pi}\frac{\mathrm{d}q_w^{ab}\mathrm{d}\hat{q}_w^{ab}}{2\pi}e^{p\Phi^{(r)}} \tag{59}$$

$$\Phi^{(r)} = -\gamma\rho\hat{\rho} - \gamma\sum_{a=1}^r\sum_i m_{s,i}^a\hat{m}_{s,i}^a - \sum_{1\le a\le b\le r}\left(\gamma\sum_i q_{s,i}^{ab}\hat{q}_{s,i}^{ab} + q_w\hat{q}_w\right) \tag{60}$$

$$+\,\alpha\Psi_y^{(r)}\left(\rho, m_s^a, q_s^{ab}, q_w^{ab}\right) + \Psi_w^{(r)}\left(\hat{\rho}, \hat{m}_s^a, \hat{q}_s^{ab}, \hat{q}_w^{ab}\right) \tag{61}$$

Where we introduced and *energetic* part Ψ_y which corresponds to the likelihood of the order parameters taking a certain value based on the data, and an *entropic* part Ψ_w which corresponds to the volume compatible with those order parameters :

$$\begin{aligned}\Psi_y^{(r)} &= \log \int \mathrm{d}y \int \mathrm{d}\nu P_y^0(y \mid \nu) \int \prod_{a=1}^r \left[\mathrm{d}\lambda^a P_y\left(y \mid \lambda^a\right)\right] P\left(\nu, \{\lambda^a\}\right) \\ \Psi_w^{(r)} &= \tfrac{1}{P} \log \int \mathrm{d}\boldsymbol{\theta}^0 P_\theta\left(\boldsymbol{\theta}^0\right) e^{-\hat{\rho}\left\|\boldsymbol{\theta}^0\right\|^2} \int \prod_{a=1}^r \mathrm{d}\boldsymbol{w}^a P_w\left(\boldsymbol{w}^a\right) e^{\sum_{1\le a\le b\le r}\left[\hat{q}_w^{ab}\boldsymbol{w}^a\cdot\boldsymbol{w}^b+\sum_i \hat{q}_{s,i}^{ab}\boldsymbol{s}^a|_i\cdot\boldsymbol{s}^b|_i\right]-\sum_{a=1}^r\sum_i \hat{m}_{s,i}^a\boldsymbol{s}^a|_i\cdot\boldsymbol{\theta}^0|_i}\end{aligned} \tag{62}$$

The calculation of Ψ_y is exactly the same as in [25] : we defer the reader to the latter for details. As for Ψ_w, considering that $P_w(\boldsymbol{w}) = e^{-\frac{\lambda}{2}\|\boldsymbol{w}\|^2}$ we have (denoting the block indices as i):

$$\Psi_w = \lim_{P\to\infty} \mathbb{E}_{\theta^0,\xi,\eta}\left[\frac{1}{2}\frac{\eta^2\hat{q}_w}{\beta\lambda+\hat{V}_w} - \frac{1}{2}\log\left(\beta\lambda+\hat{V}_w\right) - \frac{1}{2P}\operatorname{tr}\log\left(\mathrm{I}_D+\hat{V}_s\Sigma\right) - \frac{1}{2P}\boldsymbol{\mu}^\top\Sigma^{-1}\boldsymbol{\mu}\right. \tag{63}$$

$$\left. +\frac{1}{2P}\left(\boldsymbol{b}+\Sigma^{-1}\boldsymbol{\mu}\right)^\top\hat{V}_s^{-1}\left(\boldsymbol{b}+\Sigma^{-1}\boldsymbol{\mu}\right) - \frac{1}{2P}\left(b+\Sigma^{-1}\mu\right)^\top\hat{V}_s^{-1}\left(\mathrm{I}_D+\hat{V}_s\Sigma\right)^{-1}\left(b+\Sigma^{-1}\mu\right)\right] \tag{64}$$

where $\hat{V}_s$ is a block diagonal matrix where the block diagonals read $(\hat{V}_{s,1}, \hat{V}_{s,2})$, and

$$\begin{aligned} b &= \left(\sqrt{\hat{q}_{s,1}}\xi_1\mathbf{1}_{\phi_1 D} + \hat{m}_{s,1}\boldsymbol{\theta}_1^0, \sqrt{\hat{q}_{s,2}}\xi_2\mathbf{1}_{\phi_2 D} + \hat{m}_{s,2}\boldsymbol{\theta}_2^0\right) \\ \boldsymbol{\mu} &= \frac{\sqrt{\hat{q}_w}\eta}{\beta\lambda+\hat{W}_w}\frac{\mathrm{F}\mathbf{1}_P}{\sqrt{P}} \\ \Sigma &= \frac{1}{\beta\lambda+\hat{V}_w}\frac{\mathrm{F}\mathrm{F}^\top}{P}\end{aligned}$$

and λ here is the coefficient in the ℓ_2 regularization. Then, we have

$$\begin{aligned} \mathbb{E}_\eta\left[\boldsymbol{\mu}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right] &= \tfrac{1}{P}\tfrac{\hat{q}_w}{\left(\beta\lambda+\hat{V}_w\right)^2}\left(\mathrm{F}\mathbf{1}_P\right)^\top\boldsymbol{\Sigma}^{-1}\left(\mathrm{F}\mathbf{1}_P\right) = D\tfrac{\hat{q}_w}{\beta\lambda+\hat{V}_w} \\ \mathbb{E}_{\eta,\xi,\theta^0}\left\|b+\Sigma^{-1}\boldsymbol{\mu}\right\|^2 &= \textstyle\sum_i \phi_i D\left(\hat{m}_{s,i}^2+\hat{q}_{s,i}\right) + \tfrac{1}{P}\hat{q}_w\operatorname{tr}(\mathrm{F}\mathrm{F}^\top)^{-1} \\ \mathbb{E}_{\eta,\xi,\theta^0}\left(b+\Sigma^{-1}\boldsymbol{\mu}\right)^\top\left(\mathrm{I}_D+\hat{V}_s\Sigma\right)^{-1}\left(b+\Sigma^{-1}\boldsymbol{\mu}\right) &= \tfrac{1}{P}\hat{q}_w\operatorname{tr}\left[\mathrm{F}\mathrm{F}^\top\left(\mathrm{I}_D+\hat{V}_s\Sigma\right)^{-1}\right] \\ &\quad + \textstyle\sum_i\left(\hat{m}_{s,i}^2+\hat{q}_{s,i}\right)\operatorname{tr}\left(\mathrm{I}_D+\hat{V}_{s_i}\Sigma_i\right)^{-1}\end{aligned} \tag{65}$$

$$\Psi_w = -\frac{1}{2}\log\left(\beta\lambda+\hat{V}_w\right) - \frac{1}{2}\lim_{P\to\infty}\frac{1}{P}\operatorname{tr}\log\left(\mathrm{I}_D+\frac{\hat{V}_s}{\beta\lambda+\hat{V}_w}\frac{\mathrm{F}\mathrm{F}^\top}{P}\right) \tag{66}$$

$$+\sum_i\frac{\hat{m}_{s,i}^2+\hat{q}_{s,i}}{2\hat{V}_{s,i}}\left[\phi_i\gamma\right] - \lim_{P\to\infty}\frac{1}{P}\operatorname{tr}\left(\left(\hat{m}_s^2+\hat{q}_s\right)\left(\mathrm{I}_D+\frac{\hat{V}_s}{\beta\lambda+\hat{V}_w}\frac{\mathrm{F}\mathrm{F}^\top}{P}\right)^{-1}\right) \tag{67}$$

$$+\frac{1}{2}\frac{\hat{q}_w}{\beta\lambda+\hat{V}_w}\left[1-\gamma\right] + \frac{\hat{q}_w}{2}\lim_{P\to\infty}\frac{1}{P}\left[\operatorname{tr}\left(\hat{V}_s^{-1}(\mathrm{F}\mathrm{F}^\top)^{-1}\right) - \operatorname{tr}\left((\hat{V}_s\mathrm{F}\mathrm{F}^\top)^{-1}\left(\mathrm{I}_D+\frac{\hat{V}_s}{\beta\lambda+\hat{V}_w}\frac{\mathrm{F}\mathrm{F}^\top}{P}\right)^{-1}\right)\right] \tag{68}$$

where $(\hat{m}_s^2+\hat{q}_s)$ and $\hat{V}_s$ in the trace operator are to be understood as diagonal matrices here. Using the

following simplification

$$\operatorname{tr}\left(V_s^{-1}(\mathrm{F}\mathrm{F}^\top)^{-1}\right) - \operatorname{tr}\left((V_s\mathrm{F}\mathrm{F}^\top)^{-1}\left(\mathrm{I}_D + \frac{\hat{V}_s}{\beta\lambda+\hat{V}_w}\frac{\mathrm{F}\mathrm{F}^\top}{P}\right)^{-1}\right) \tag{69}$$

$$= \frac{1}{P(\beta\lambda+\hat{V}_w)}\operatorname{tr}\left(\mathrm{I}_d + \frac{\hat{V}_s}{\beta\lambda+\hat{V}_w}\frac{\mathrm{F}\mathrm{F}^\top}{P}\right), \tag{70}$$

we finally obtain

$$\Psi_w = -\frac{1}{2}\log\left(\beta\lambda+\hat{V}_w\right) - \frac{1}{2}\lim_{P\to\infty}\frac{1}{P}\operatorname{tr}\log\left(\mathrm{I}_D + \frac{\hat{V}_s}{\beta\lambda+\hat{V}_w}\frac{\mathrm{F}\mathrm{F}^\top}{P}\right) \tag{71}$$

$$+\sum_i \frac{\hat{m}_{s,i}^2+\hat{q}_{s,i}}{2\hat{V}_{s,i}}\left[\phi_i\gamma\right] - \lim_{P\to\infty}\frac{1}{P}\operatorname{tr}\left((\hat{m}_s^2+\hat{q}_s)\left(\mathrm{I}_D + \frac{\hat{V}_s}{\beta\lambda+\hat{V}_w}\frac{\mathrm{F}\mathrm{F}^\top}{P}\right)^{-1}\right) \tag{72}$$

$$+\frac{1}{2}\frac{\hat{q}_w}{\beta\lambda+\hat{V}_w}\left[1-\gamma+\lim_{P\to\infty}\frac{1}{P}\operatorname{tr}\left(\left(\mathrm{I}_D + \frac{\hat{V}_s}{\beta\lambda+\hat{V}_w}\frac{\mathrm{F}\mathrm{F}^\top}{P}\right)^{-1}\right)\right] \tag{73}$$

.
Define $M = \frac{1}{P}\hat{V}_s F F^\top$. Ψ_w involves the following term :

$$\frac{1}{P}\operatorname{tr}\left(\mathrm{I}_D + \frac{\hat{V}_s}{\beta\lambda+\hat{V}_w}M\right)^{-1} = \gamma(\beta\lambda+\hat{V}_w)g(-\beta\lambda+\hat{V}_w) \tag{74}$$

Where g is the Stieljes transform of M : $g(z) = \frac{1}{D}\operatorname{Tr}(z-M)^{-1}$.

B.3 Some random matix theory for block matrices

To calculate the desired Stieljes transform, we specialize to the case of Gaussian random feature matrices and use again tools from Statistical Physics.

Isotropic setup We first consider the isotropic setup where $\hat{V}_s$ is a scalar. Then we have

$$g(z) = \frac{1}{D}\operatorname{Tr}(z-M)^{-1} \tag{75}$$

$$= -\frac{1}{D}\frac{d}{dz}\operatorname{Tr}\log(z-M) \tag{76}$$

$$= -\frac{1}{D}\frac{d}{dz}\log\det(z-M) \tag{77}$$

$$= -\frac{2}{D}\frac{d}{dz}\left\langle \log\int dy e^{-\frac{1}{2}y(z-M)y^\top}\right\rangle \tag{78}$$

where $\langle . \rangle$ stands for the average over disorder, here the matrix F. Then we use the replica trick,

$$\langle \log Z\rangle \to_{n\to 0} \frac{1}{n}\log\langle Z^n\rangle \tag{79}$$

Therefore we need to calculate Z^n:

$$\langle Z^n\rangle = \int\prod_{a=1}^n d\vec{y}_a e^{-\frac{1}{2}z\sum_a \vec{y}_a^2}\int dF e^{-\frac{1}{2}\mathrm{F}\mathrm{L}\mathrm{F}^\top} \tag{80}$$

$$= \int\prod_{a=1}^n d\vec{y}_a e^{-\frac{1}{2}z\sum_a \vec{y}_a^2}(\det L)^{-P/2} \tag{81}$$

where $L = \mathbb{I}_d - \frac{\gamma \hat{V}_s}{D} \sum_a y_a \vec{y}_a^\top$. Here we decompose the vectors $\vec{y}$ into two parts. Then we use that $\det L = \det \tilde{L}$, where $\tilde{L}_{ab} = \delta_{ab} - \frac{\gamma \hat{V}_s}{D} \vec{y}_a \cdot \vec{y}_b$. Then,

$$\langle Z^n \rangle = \int \prod_a d\vec{y}_a e^{\frac{1}{2}\sum_a \vec{y}_a \cdot \vec{y}_a} \det \left[\mathbb{I}_d - \frac{\gamma \hat{V}_s}{D} Y^\top Y \right]^{-\frac{P}{2}} \tag{82}$$

where $Y \in \mathbb{R}^{d \times n}$ with $Y_{ia} = y_i^a$. We introduce $1 = \int dQ_{ab} \delta(dQ_{ab} - \vec{y}_a \cdot \vec{y}_b)$, and use the Fourier representation of the delta function, yielding

$$\langle Z^n \rangle = \int dQ e^{-\frac{D}{2} z \operatorname{Tr} Q} \left(\det \left[\mathbb{I} - \hat{V}_s Q \right] \right)^{-\frac{P}{2}} \int d\hat{Q}_{ab} e^{\sum_{ab} dQ_{ab} \hat{Q}_{ab} - \hat{Q}_{ab} \vec{y}_a \vec{y}_b} \tag{83}$$

$$= \int dQ d\hat{Q} e^{-dS[Q,\hat{Q}]} \tag{84}$$

where

$$S[Q, \hat{Q}] = \frac{1}{2} z \operatorname{Tr} Q + \frac{1}{2\gamma} \log\det \left(1 - \hat{V}_s Q \right) + \frac{1}{2} \log\det \left(2\hat{Q} \right) - \operatorname{Tr} \left(Q\hat{Q} \right) \tag{85}$$

A saddle-point on $\hat{Q}$ gives $Q = (2\hat{Q})^{-1}$. Therefore we can replace in S,

$$S[Q] = \frac{1}{2} z \operatorname{Tr} Q + \frac{1}{2\gamma} \log\det \left(1 - \hat{V}_s Q \right) - \frac{1}{2} \log\det \left(Q \right) \tag{86}$$

In the RS ansatz $Q_{ab} = q\delta_{ab}$, this yields

$$S[q]/n = \frac{1}{2} zq + \frac{1}{2\gamma} \log \left(1 - \hat{V}_s q \right) - \frac{1}{2} \log q \tag{87}$$

Now we may apply a saddle point method to write

$$\langle Z^n \rangle = e^{-dS[q^\star]} \tag{88}$$

where $q^\star$ minimizes the action, i.e. $\frac{dS}{dq}|_{q^\star} = 0$:

$$z - \frac{\hat{V}_s}{1 - \hat{V}_s q^\star} - \frac{1}{q^\star} = 0 \tag{89}$$

Therefore,

$$g(z) = -\frac{2}{D} \frac{d}{dz} (-DS[q^\star]) = q^\star(z) \tag{90}$$

Anisotropic setup Now we consider that V is a black diagonal matrix, with blocks of size $\phi_i d$ with values $\hat{V}_{s,i}$. We need to adapt the calculation by decomposing the auxiliary fields y along the different blocks, then define separately the overlaps of the blocks q_i. Then we obtain the following action:

$$S[\{q_i\}]/n = \frac{1}{2} z \sum_i \phi_i q_i + \frac{1}{2\gamma} \log \left(1 - \sum_i \phi_i \hat{V}_{s,i} q_i \right) - \sum_i \frac{\phi_i}{2} \log(q_i) \tag{91}$$

To obtain the Stieljes transform, we need to solve a system of coupled equations :

$$g(z) = \sum_i \phi_i q_i^\star \tag{92}$$

$$\phi_i z \Omega q_i^\star - \phi_i \hat{V}_i q_i^\star - \phi_i \Omega = 0 \tag{93}$$

$$\Omega = 1 - \sum_i \phi_i \hat{V}_{s,i} q_i^\star \tag{94}$$

We therefore conclude that

$$\lim_{P\to\infty}\frac{1}{P}\operatorname{tr}\left(\left(\mathrm{I}_D+\frac{\hat{V}_s}{\beta\lambda+\hat{V}_w}\frac{\mathrm{F}\mathrm{F}^\top}{P}\right)^{-1}\right)=\gamma(\beta\lambda+\hat{V}_w)\sum_i\phi_i q_i^\star \tag{95}$$

$$\lim_{P\to\infty}\frac{1}{P}\operatorname{tr}\left((\hat{m}_s^2+\hat{q}_s)\left(\mathrm{I}_D+\frac{\hat{V}_s}{\beta\lambda+\hat{V}_w}\frac{\mathrm{F}\mathrm{F}^\top}{P}\right)^{-1}\right)=\gamma(\beta\lambda+\hat{V}_w)\sum_i(\hat{m}_{s,i}^2+\hat{q}_{s,i})\phi_i q_i^\star \tag{96}$$

B.4 Obtaining the saddle-point equations

The saddle-point equations of [25] become the following in the anisotropic setup :

$$\begin{cases}\hat{r}_{s,i}=-2\sigma_{x,i}\frac{\alpha}{\gamma}\partial_{r_{s,i}}\Psi_y(R,Q,M) & r_{s,i}=-\frac{2}{\gamma}\partial_{\hat{r}_{s,i}}\Psi_w\left(\hat{r}_{s,i},\hat{q}_{s,i},\hat{m}_{s,i},\hat{r}_w,\hat{q}_w\right)\\ \hat{q}_{s,i}=-2\sigma_{x,i}\frac{\alpha}{\gamma}\partial_{q_{s,i}}\Psi_y(R,Q,M) & q_{s,i}=-\frac{2}{\gamma}\partial_{\hat{q}_{s,i}}\Psi_w\left(\hat{r}_{s,i},\hat{q}_{s,i},\hat{m}_{s,i},\hat{r}_w,\hat{q}_w\right)\\ \hat{m}_{s,i}=\sigma_{x,i}\frac{\alpha}{\gamma}\partial_{m_{s,i}}\Psi_y(R,Q,M) & m_{s,i}=\frac{1}{\gamma}\partial_{\hat{m}_{s,i}}\Psi_w\left(\hat{r}_{s,i},\hat{q}_{s,i},\hat{m}_{s,i},\hat{r}_w,\hat{q}_w\right)\\ \hat{r}_w=-2\alpha\partial_{r_w}\Psi_y(R,Q,M) & r_w=-2\partial_{\hat{r}_w}\Psi_w\left(\hat{r}_{s,i},\hat{q}_{s,i},\hat{m}_{s,i},\hat{r}_w,\hat{q}_w\right)\\ \hat{q}_w=-2\alpha\partial_{q_w}\Psi_y(R,Q,M) & q_w=-2\partial_{\hat{q}_w}\Psi_w\left(\hat{r}_{s,i},\hat{q}_{s,i},\hat{m}_{s,i},\hat{r}_w,\hat{q}_w\right)\end{cases} \tag{97}$$

The saddle point equations corresponding to Ψ_w do not depend on the learning task, and can be simplified in full generality to the following set of equations :

$$\begin{cases}V_{s,i}=\frac{1}{\hat{V}_{s,i}}\left(\phi_i-z_i g_\mu(-z_i)\right)\\ q_{s,i}=\frac{\sigma_{\beta,i}\hat{m}_{s,i}^2+\hat{q}_{s,i}}{\hat{V}_{s,i}^2}\left[\phi_i-2z_i g_\mu(-z_i)+z_i^2 g'_\mu(-z_i)\right]-\frac{\hat{q}_w}{(\beta\lambda+\hat{V}_w)\hat{V}_{s,i}}\left[-z_i g_\mu(-z_i)+z_i^2 g'_\mu(-z_i)\right]\\ m_{s,i}=\frac{\sigma_{\beta,i}\hat{m}_{s,i}}{\hat{V}_{s,i}}\left(\phi_i-z_i g_\mu(-z_i)\right)\\ V_w=\sum_i\frac{\gamma}{\beta\lambda+\hat{V}_w}\left[\frac{1}{\gamma}-1+z_i g_\mu(-z_i)\right]\\ q_w=\sum_i\frac{\sigma_{\beta,i}\hat{m}_s^2+\hat{q}_{s,i}}{(\beta\lambda+\hat{V}_w)\hat{V}_{s,i}}\left[-z_i g_\mu(-z_i)+z_i^2 g'_\mu(-z_i)\right]+\gamma\frac{\hat{q}_w}{(\beta\lambda+\hat{V}_w)^2}\left[\frac{1}{\gamma}-1+z_i^2 g'_\mu(-z_i)\right]\end{cases} \tag{98}$$

As for the saddle point equations corresponding to Ψ_y, they depend on the learning task. We can solve analytically for the two setups below.
The solution of the saddle-point equations allow to obtain the order parameter values that determine the covariance of ν and λ (for one given replica, say $a=1$) and hence to compute the test error as explained at the beginning of the appendix.

Square loss regression Now we specialize to our mean-square regression case study where the teacher is a Gaussian additive channel :

$$\mathcal{P}(x|y)=\frac{1}{\sqrt{2\pi\Delta}}e^{-\frac{(x-y)^2}{2\Delta}} \tag{99}$$

$$\ell(y,x)=\frac{1}{2}(x-y)^2 \tag{100}$$

In this case, the saddle-point equations simplify to :

$$\begin{cases}\hat{V}_{s,i}^\infty=\sigma_{x,i}\frac{\alpha}{\gamma}\frac{\kappa_{1,i}^2}{1+V^\infty}\\ \hat{q}_{s,i}^0=\sigma_{x,i}\alpha\kappa_{1,i}^2\frac{\phi_i\sigma_{\beta,i}+\Delta+Q^\infty-2M^\infty}{\gamma(1+V^\infty)^2}\\ \hat{m}_{s,i}=\sigma_{x,i}\frac{\alpha}{\gamma}\frac{\kappa_{1,i}}{1+V^\infty}\\ \hat{V}_w^\infty=\frac{\alpha\sum_i\phi_i\kappa_{\star,i}^2}{1+V^\infty}\\ \hat{q}_w^\infty=\alpha\sum_i\phi_i\kappa_{\star,i}^2\frac{1+\Delta+Q^\infty-2M^\infty}{\left(\phi_i\sigma_{\beta,i}+V^\infty\right)^2}\end{cases} \tag{101}$$

Square loss classification Next we examine the classification setup where the teacher gives binary labels with a sign flip probability of Δ, and the student learns them through the mean-square loss:

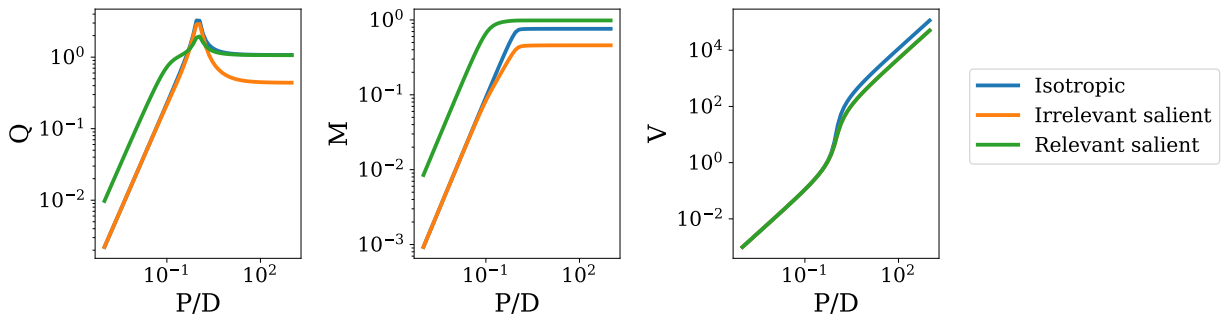
$$\mathcal{P}(x|y) = (1-\Delta)\delta(x - \text{sign}(y)) + \Delta\delta(x + \text{sign}(y)), \quad \Delta \in [0, 1] \tag{102}$$

$$\ell(y, x) = \frac{1}{2}(x - y)^2 \tag{103}$$

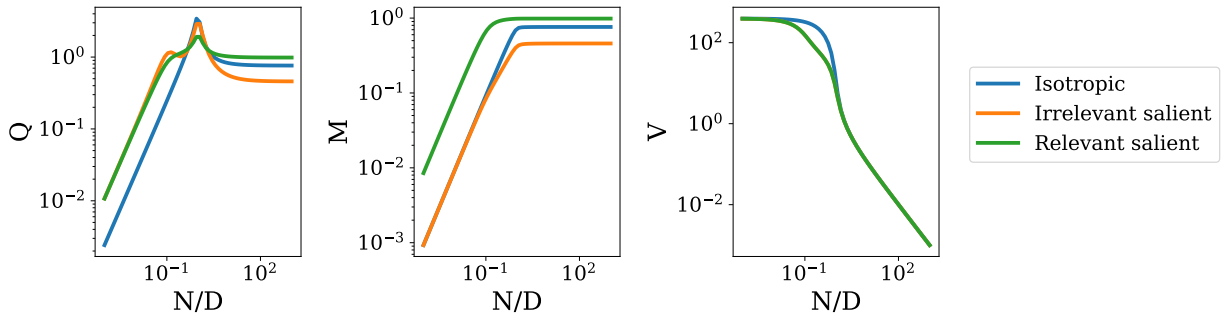
In this case, the equations simplify to :

$$\begin{cases} \hat{V}^{\infty}_{s,i} = \sigma_{x,i}\frac{\alpha}{\gamma}\frac{\kappa^2_{1,i}}{1+V^{\infty}} \\ \hat{q}^{0}_{s,i} = \sigma_{x,i}\alpha\kappa^2_{1,i}\frac{\phi_i\sigma_{\beta,i}+\Delta+Q^{\infty}-2\frac{(1-\Delta)\sqrt{2}}{\sqrt{\pi}}M^{\infty}}{\gamma(1+V^{\infty})^2} \\ \hat{m}_{s,i} = \sigma_{x,i}\frac{\alpha}{\gamma}\frac{\kappa_{1,i}}{1+V^{\infty}} \\ \hat{V}^{\infty}_{w} = \frac{\alpha\sum_i \phi_i\kappa^2_{\star,i}}{1+V^{\infty}} \\ \hat{q}^{\infty}_{w} = \alpha\sum_i \phi_i\kappa^2_{\star,i}\frac{1+\Delta+Q^{\infty}-2\frac{(1-\Delta)}{\sqrt{\pi}}M^{\infty}}{\left(\phi_i\sigma_{\beta,i}+V^{\infty}\right)^2} \end{cases} \tag{104}$$

Cross-entropy loss classification For general loss functions such as the cross-entropy loss, $\ell(x, y) = \log(1 + e^{-xy})$, the saddle-point equations for Ψ_y do not simplify and one needs to evaluate the integral over ξ numerically. This case is outside the scope of this paper.



(a) Parameter-wise



(b) Sample-wise

Figure 8: Order parameters. We considered the regression setting for $r_\phi = 10, \sigma = \text{Tanh}, \lambda = 10^{-3}, \Delta = 0.3$.

B.5 Training loss

To calculate the training loss, we remove the regularization term:

$$\epsilon_t = \frac{1}{N}\mathbb{E}_{\{\boldsymbol{x}^\mu, y^\mu\}}\left[\sum_{\mu=1}^{N}\ell\left(y^\mu, \boldsymbol{x}^\mu\cdot\hat{\boldsymbol{w}}\right)\right] \tag{105}$$

As explained in [25], the latter can be written as

$$\epsilon_t = \mathbb{E}_\xi \left[\int_{\mathbb{R}} \mathrm{d}y \mathcal{Z}_y^0 \left(y, \omega_0, V_0^\star\right) \ell\left(y, \eta\left(y, \omega_1\right)\right) \right], \quad \xi \sim \mathcal{N}(0,1) \tag{106}$$

where $\mathcal{P}$ is the teacher channel defined in (2), and we have:

$$\omega_0 = M/\sqrt{Q}\xi \tag{107}$$
$$V_0 = \rho - M^2/Q \tag{108}$$
$$\omega_1 = \sqrt{Q}\xi \tag{109}$$
$$\eta(y,\omega) = \arg\min_x \frac{(x-\omega)^2}{2V} + \ell(y,x) \tag{110}$$
$$\mathcal{Z}_y^0\left(y,\omega\right) = \int \frac{\mathrm{d}x}{\sqrt{2\pi V_0}} e^{-\frac{1}{2V_0}(y-\omega)^2} \mathcal{P}(y|x) \tag{111}$$

Square loss regression Now we specialize to our regression case study where the teacher is a Gaussian additive channel $\mathcal{P}(x|y) = \frac{1}{\sqrt{2\pi\Delta}} e^{-\frac{(x-y)^2}{2\Delta}}$ and the loss is $\ell(y,x) = \frac{1}{2}(x-y)^2$. These assumptions imply

$$\mathcal{Z}_y^0\left(y,\omega\right) = \frac{1}{\sqrt{2\pi(V_0+\Delta)}} e^{-\frac{(y-\omega)^2}{2(V_0+\Delta)}}, \tag{112}$$
$$\eta(y,\omega) = \frac{\omega + Vy}{1+V}, \tag{113}$$

which yields the simple formula for the training error

$$\epsilon_t = \frac{1}{2(1+V)^2} \mathbb{E}_\xi \left[\int \mathrm{d}y \mathcal{Z}_y^0\left(y,\omega\right)\left(y-\omega_1\right)^2 \right] \tag{114}$$
$$= \frac{1}{2(1+V)^2} \mathbb{E}_\xi \left[V_0 + \Delta + (\omega_0 - \omega_1)^2 \right] \tag{115}$$
$$= \frac{1}{2(1+V)^2} \left(\rho + Q - 2M + \Delta\right) \tag{116}$$
$$= \frac{\epsilon_g + \Delta}{(1+V)^2}. \tag{117}$$

Square loss classification Next we specialize to our classification case study where the teacher is a Gaussian additive channel $\mathcal{P}(x|y) = (1-\Delta)\delta(x - \mathrm{sign}(y)) + \Delta\delta(x + \mathrm{sign}(y))$ and the loss is $\ell(y,x) = \frac{1}{2}(x-y)^2$. These assumptions imply

$$\mathcal{Z}_y^0\left(y,\omega\right) = \frac{1}{\sqrt{2\pi V_0}} \left((1-\Delta) e^{-\frac{(y-\omega)^2}{2V_0}} + \Delta e^{-\frac{(-y-\omega)^2}{2V_0}} \right), \tag{118}$$
$$\eta(y,\omega) = \frac{\omega + Vy}{1+V}, \tag{119}$$

which yields the simple formula for the training error

$$\epsilon_t = \frac{1}{2(1+V)^2} \mathbb{E}_\xi \left[V_0 + (1-\Delta)(\omega_0 - \omega_1)^2 + \Delta(\omega_0 + \omega_1)^2 \right] \tag{120}$$
$$= \frac{1}{2(1+V)^2} \left(\rho + Q - 2(1-2\Delta)M\right) \tag{121}$$
$$\tag{122}$$

References

1. Krizhevsky, A., Sutskever, I. & Hinton, G. E. *Imagenet classification with deep convolutional neural networks* in *Advances in neural information processing systems* (2012), 1097–1105.
2. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *nature* **521,** 436–444 (2015).
3. Hinton, G. *et al.* Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine* **29,** 82–97 (2012).
4. Sutskever, I., Vinyals, O. & Le, Q. V. *Sequence to sequence learning with neural networks* in *Advances in neural information processing systems* (2014), 3104–3112.
5. Zhang, C., Bengio, S., Hardt, M., Recht, B. & Vinyals, O. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530* (2016).
6. Neyshabur, B., Li, Z., Bhojanapalli, S., LeCun, Y. & Srebro, N. Towards understanding the role of over-parametrization in generalization of neural networks. *arXiv preprint arXiv:1805.12076* (2018).
7. Advani, M. S., Saxe, A. M. & Sompolinsky, H. High-dimensional dynamics of generalization error in neural networks. *Neural Networks* **132,** 428–446 (2020).
8. Zagoruyko, S. & Komodakis, N. Wide residual networks. *arXiv preprint arXiv:1605.07146* (2016).
9. Brown, T. B. *et al.* Language models are few-shot learners. *arXiv preprint arXiv:2005.14165* (2020).
10. Spigler, S. *et al.* A jamming transition from under-to over-parametrization affects generalization in deep learning. *Journal of Physics A: Mathematical and Theoretical* **52,** 474001 (2019).
11. Belkin, M., Hsu, D., Ma, S. & Mandal, S. Reconciling modern machine learning and the bias-variance trade-off. *arXiv preprint arXiv:1812.11118* (2018).
12. Geiger, M. *et al.* Scaling description of generalization with number of parameters in deep learning. *Journal of Statistical Mechanics: Theory and Experiment* **2020,** 023401 (2020).
13. Opper, M. & Kinzel, W. in *Models of neural networks III* 151–209 (Springer, 1996).
14. Engel, A. & Van den Broeck, C. *Statistical mechanics of learning* (Cambridge University Press, 2001).
15. Krzakala, F. & Kurchan, J. Landscape analysis of constraint satisfaction problems. *Physical Review E* **76,** 021122 (2007).
16. Franz, S. & Parisi, G. The simplest model of jamming. *Journal of Physics A: Mathematical and Theoretical* **49,** 145001 (2016).
17. Geiger, M. *et al.* Jamming transition as a paradigm to understand the loss landscape of deep neural networks. *Physical Review E* **100,** 012115 (2019).
18. Nakkiran, P. *et al.* Deep double descent: Where bigger models and more data hurt. *arXiv preprint arXiv:1912.02292* (2019).
19. Nakkiran, P. More Data Can Hurt for Linear Regression: Sample-wise Double Descent. *arXiv preprint arXiv:1912.07242* (2019).
20. Krogh, A. & Hertz, J. A. Generalization in a linear perceptron in the presence of noise. *Journal of Physics A: Mathematical and General* **25,** 1135 (1992).
21. Hastie, T., Montanari, A., Rosset, S. & Tibshirani, R. J. Surprises in high-dimensional ridgeless least squares interpolation. *arXiv preprint arXiv:1903.08560* (2019).
22. Nakkiran, P., Venkat, P., Kakade, S. & Ma, T. Optimal regularization can mitigate double descent. *arXiv preprint arXiv:2003.01897* (2020).
23. Rahimi, A. & Recht, B. *Random features for large-scale kernel machines* in *Advances in neural information processing systems* (2008), 1177–1184.
24. Mei, S. & Montanari, A. The generalization error of random features regression: Precise asymptotics and double descent curve. *arXiv preprint arXiv:1908.05355* (2019).

25. Gerace, F., Loureiro, B., Krzakala, F., Mézard, M. & Zdeborová, L. Generalisation error in learning with random features and the hidden manifold model. *arXiv preprint arXiv:2002.09339* (2020).

26. Richards, D., Mourtada, J. & Rosasco, L. Asymptotics of Ridge (less) Regression under General Source Condition. *arXiv preprint arXiv:2006.06386* (2020).

27. Dobriban, E. & Wager, S. High-dimensional asymptotics of prediction: Ridge regression and classification. *arXiv preprint arXiv:1507.03003* (2015).

28. Wu, D. & Xu, J. On the Optimal Weighted l2 Regularization in Overparameterized Linear Regression. *arXiv preprint arXiv:2006.05800* (2020).

29. Chen, L., Min, Y., Belkin, M. & Karbasi, A. Multiple descent: Design your own generalization curve. *arXiv preprint arXiv:2008.01036* (2020).

30. Kobak, D., Lomond, J. & Sanchez, B. Optimal ridge penalty for real-world high-dimensional data can be zero or negative due to the implicit ridge regularization. *liver* **1,** 1–2 (2019).

31. Ghorbani, B., Mei, S., Misiakiewicz, T. & Montanari, A. When do neural networks outperform kernel methods? *Advances in Neural Information Processing Systems* **33** (2020).

32. Liao, Z., Couillet, R. & Mahoney, M. W. A random matrix analysis of random Fourier features: beyond the Gaussian kernel, a precise phase transition, and the corresponding double descent. *arXiv preprint arXiv:2006.05013* (2020).

33. Canatar, A., Bordelon, B. & Pehlevan, C. Statistical Mechanics of Generalization in Kernel Regression. *arXiv preprint arXiv:2006.13198* (2020).

34. Montanari, A., Ruan, F., Sohn, Y. & Yan, J. The generalization error of max-margin linear classifiers: High-dimensional asymptotics in the overparametrized regime. *arXiv preprint arXiv:1911.01544* (2019).

35. Chatterji, N. S. & Long, P. M. Finite-sample analysis of interpolating linear classifiers in the overparameterized regime. *arXiv preprint arXiv:2004.12019* (2020).

36. Liang, T. & Sur, P. A precise high-dimensional asymptotic theory for boosting and min-l1-norm interpolated classifiers. *arXiv preprint arXiv:2002.01586* (2020).

37. Deng, Z., Kammoun, A. & Thrampoulidis, C. A Model of Double Descent for High-dimensional Binary Linear Classification. *arXiv preprint arXiv:1911.05822* (2019).

38. Mignacco, F., Krzakala, F., Lu, Y., Urbani, P. & Zdeborova, L. *The role of regularization in classification of high-dimensional noisy Gaussian mixture* in *International Conference on Machine Learning* (2020), 6874–6883.

39. Kini, G. & Thrampoulidis, C. Analytic Study of Double Descent in Binary Classification: The Impact of Loss. *arXiv preprint arXiv:2001.11572* (2020).

40. Muthukumar, V. *et al.* Classification vs regression in overparameterized regimes: Does the loss function matter? *arXiv preprint arXiv:2005.08054* (2020).

41. Li, Y., Wei, C. & Ma, T. *Towards explaining the regularization effect of initial large learning rate in training neural networks* in *Advances in Neural Information Processing Systems* (2019), 11674–11685.

42. Jacot, A., Gabriel, F. & Hongler, C. *Neural tangent kernel: Convergence and generalization in neural networks* in *Advances in neural information processing systems* (2018), 8571–8580.

43. Chizat, L., Oyallon, E. & Bach, F. in *Advances in Neural Information Processing Systems 32* (eds Wallach, H. *et al.*) 2933–2943 (Curran Associates, Inc., 2019).

44. Gabrié, M. Mean-field inference methods for neural networks. *Journal of Physics A: Mathematical and Theoretical* **53** (2020).

45. Bahri, Y. *et al.* Statistical Mechanics of Deep Learning. *Annual Review of Condensed Matter Physics* **11,** 501–528 (2020).

46. Barbier, J., Krzakala, F., Macris, N., Miolane, L. & Zdeborová, L. Optimal errors and phase transitions in high-dimensional generalized linear models. *Proceedings of the National Academy of Sciences of the United States of America* **116,** 5451–5460 (2019).

47. Aubin, B. *et al. The committee machine: Computational to statistical gaps in learning a two-layers neural network* in *Neural Information Processing Systems 2018* (2018), 1–44.

48. Gabrié, M. *et al. Entropy and mutual information in models of deep neural networks* in (Curran Associates, Inc., 2018), 1821–1831.

49. Goldt, S., Reeves, G., Mézard, M., Krzakala, F. & Zdeborová, L. The Gaussian equivalence of generative models for learning with two-layer neural networks. *arXiv preprint arXiv:2006.14709* (2020).

50. Mézard, M., Parisi, G. & Virasoro, M. *Spin glass theory and beyond: An Introduction to the Replica Method and Its Applications* (World Scientific Publishing Company, 1987).

51. El Karoui, N. *et al.* The spectrum of kernel random matrices. *Annals of statistics* **38,** 1–50 (2010).

52. Hu, H. & Lu, Y. M. Universality laws for high-dimensional learning with random features. *arXiv preprint arXiv:2009.07669* (2020).

53. Hui, L. & Belkin, M. Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks. *arXiv preprint arXiv:2006.07322* (2020).

54. Demirkaya, A., Chen, J. & Oymak, S. *Exploring the role of loss functions in multiclass classification* in *2020 54th Annual Conference on Information Sciences and Systems (CISS)* (2020), 1–5.

55. Arora, S. *et al.* Harnessing the Power of Infinitely Wide Deep Nets on Small-data Tasks. *arXiv preprint arXiv:1910.01663* (2019).

56. d'Ascoli, S., Sagun, L. & Biroli, G. *Triple descent and the two kinds of overfitting: where and why do they appear?* in *Advances in Neural Information Processing Systems* **33** (Curran Associates, Inc., 2020), 3058–3069.

57. d'Ascoli, S., Refinetti, M., Biroli, G. & Krzakala, F. *Double trouble in double descent: Bias and variance (s) in the lazy regime* in *International Conference on Machine Learning* (2020), 2280–2290.

58. Adlam, B. & Pennington, J. Understanding Double Descent Requires a Fine-Grained Bias-Variance Decomposition. *arXiv preprint arXiv:2011.03321* (2020).

59. Gunasekar, S., Woodworth, B. E., Bhojanapalli, S., Neyshabur, B. & Srebro, N. *Implicit regularization in matrix factorization* in *Advances in Neural Information Processing Systems* (2017), 6151–6159.

60. Goldt, S., Mézard, M., Krzakala, F. & Zdeborová, L. Modelling the influence of data structure on learning in neural networks. *arXiv preprint arXiv:1909.11500* (2019).