Generalisation error in learning with random features and the hidden manifold model

Federica Gerace, Bruno Loureiro, Florent Krzakala, Marc Mezard, Lenka Zdeborová

Generalisation error in learning with random features and the hidden manifold model

Federica Gerace[†], Bruno Loureiro[†],

Florent Krzakala[⋆], Marc Mézard [⋆], Lenka Zdeborová[†]

† *Institut de Physique Théorique*
CNRS & CEA & Université Paris-Saclay, Saclay, France
⋆ *LPENS, CNRS & Sorbonnes Universites,*
Ecole Normale Superieure, PSL University, Paris, France

Abstract

We study generalised linear regression and classification for a synthetically generated dataset encompassing different problems of interest, such as learning with random features, neural networks in the lazy training regime, and the hidden manifold model. We consider the high-dimensional regime and using the replica method from statistical physics, we provide a closed-form expression for the asymptotic generalisation performance in these problems, valid in both the under- and over-parametrised regimes and for a broad choice of generalised linear model loss functions. In particular, we show how to obtain analytically the so-called double descent behaviour for logistic regression with a peak at the interpolation threshold, we illustrate the superiority of orthogonal against random Gaussian projections in learning with random features, and discuss the role played by correlations in the data generated by the hidden manifold model. Beyond the interest in these particular problems, the theoretical formalism introduced in this manuscript provides a path to further extensions to more complex tasks.

Contents

1 Introduction

One of the most important goals of learning theory is to provide generalisation bounds describing the quality of learning a given task as a function of the number of samples. Existing results fall short of being directly relevant for the state-of-the-art deep learning methods [1, 2]. Consequently, providing tighter results on the generalisation error is currently a very active research subject. The traditional learning theory approach to generalisation follows for instance the Vapnik-Chervonenkis [3] or Rademacher [4] worst-case type bounds, and many of their more recent extensions [5]. An alternative approach, followed also in this paper, has been pursued for decades, notably in in statistical physics, where the generalisation ability of neural networks was analysed for a range of "typical-case" scenario *for synthetic data arising from a probabilistic model* [6, 7, 8, 9, 10, 11, 12, 13, 14]. While at this point it is not clear which approach will lead to a complete generalisation theory of deep learning, it is worth pursuing both directions.

The majority of works following the statistical physics approach study the generalisation error in the so-called teacher-student framework, where the input data are element-wise i.i.d. vectors, and the labels are generated by a teacher neural network. In contrast, in most of real scenarios the input data does not span uniformly the input space, but rather live close to a lower-dimensional manifold. The traditional focus onto i.i.d. Gaussian input vectors is an important limitation that has been recently stressed in [15, 14]. In [14], the authors proposed a model of synthetic data to mimic the latent structure of real data, named the *hidden manifold model*, and analysed the learning curve of one-pass stochastic gradient descent algorithm in a two-layer neural network with a small number of hidden units also known as committee machine.

Another key limitation of the majority of existing works stemming from statistical physics is that the learning curves were only computed for neural networks with a few hidden units. In particular, the input dimension is considered large, the number of samples is a constant times the input dimension and the number of hidden unit is of order one. Tight learning curves were only very recently analysed for two-layer neural network with more hidden units. These studies addressed in particular the case of networks that have a fixed first layer with random i.i.d. Gaussian weights [12, 13], or the lazy-training regime where the individual weights change only infinitesimally during training, thus not learning any specific features [16, 17, 18].

In this paper we compute the generalisation error and the corresponding learning curves, i.e. the test error as a function of the number of samples for a model of high-dimensional data that encompasses at least the following cases:

- generalised linear regression and classification for data generated by the hidden manifold model (HMM) of [14]. The HMM can be seen as a single-layer generative neural network with i.i.d. inputs and a rather generic feature matrix [19, 14].
- Learning data generated by the teacher-student model with a random-features neural network [20], with a very generic feature matrix, including deterministic ones. This model is also interesting because of its connection with the lazy regime, that is equivalent to the random features model with slightly more complicated features [16, 12, 13].

We give a closed-form expression for the generalisation error in the high-dimensional limit, obtained using a non-rigorous heuristic method from statistical physics known as the replica method [21], that has already shown its remarkable efficacy in many problems of machine learning [6, 22, 8, 23] and was proven rigorously in many cases see e.g. [24, 25]. While in the present model it remains an open problem to derive a rigorous proof for our results, we shall provide numerical support that the formula is indeed exact in the high-dimensional limit, and extremely accurate even for moderately small system sizes.

1.1 The model

We study high-dimensional regression and classification for a *synthetic* dataset $\mathcal{D} = \{(\boldsymbol{x}^\mu, y^\mu)\}_{\mu=1}^n$ where each sample μ is created in the following three steps:

(i) First, for each sample μ we create a "latent" vector $\boldsymbol{c}^\mu \in \mathbb{R}^d$ as

$$\boldsymbol{c}^\mu \sim \mathcal{N}(0, \mathrm{I}_d)\,, \tag{1.1}$$

(ii) We then draw $\boldsymbol{\theta}^0 \in \mathbb{R}^d$ from a separable distribution P_θ and draw independent labels $\{y^\mu\}_{\mu=1}^n$ from a (possibly probabilistic) rule f^0:

$$y^\mu = f^0\left(\frac{1}{\sqrt{d}}\boldsymbol{c}^\mu \cdot \boldsymbol{\theta}^0\right) \in \mathbb{R}\,. \tag{1.2}$$

(iii) The input data points $\boldsymbol{x}^\mu \in \mathbb{R}^d$ are created by a one-layer generative network with fixed weights $\mathrm{F} \in \mathbb{R}^{d\times p}$ and an activation function $\sigma : \mathbb{R} \to \mathbb{R}$:

$$\boldsymbol{x}^\mu = \sigma\left(\frac{1}{\sqrt{d}}\mathrm{F}^\top \boldsymbol{c}^\mu\right)\,. \tag{1.3}$$

We study the problem of supervised learning for the dataset $\mathcal{D}$ aiming at achieving a low generalisation error $\epsilon_{\rm g}$ on a new sample $\boldsymbol{x}^{\rm new}, y^{\rm new}$ drawn by the same rule as above, where:

$$\epsilon_{\rm g} = \frac{1}{4^k}\mathbb{E}_{\boldsymbol{x}^{\rm new},y^{\rm new}}\left[(\hat{y}_{\boldsymbol{w}}(\boldsymbol{x}^{\rm new}) - y^{\rm new})^2\right]. \tag{1.4}$$

with $k = 0$ for regression task and $k = 1$ for classification task. Here, $\hat{y}_{\boldsymbol{w}}$ is the prediction on the new label $y^{\rm new}$ of the form:

$$\hat{y}_{\boldsymbol{w}}(\boldsymbol{x}) = \hat{f}\left(\boldsymbol{x}\cdot\hat{\boldsymbol{w}}\right). \tag{1.5}$$

The weights $\hat{\boldsymbol{w}} \in \mathbb{R}^p$ are learned by minimising a loss function with a ridge regularisation term (for $\lambda \geq 0$) and defined as

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\rm argmin}\left[\sum_{\mu=1}^{n}\ell(y^\mu, \boldsymbol{x}^\mu\cdot\boldsymbol{w}) + \frac{\lambda}{2}||\boldsymbol{w}||_2^2\right], \tag{1.6}$$

where $\ell(.)$ can be, for instance, a logistic, hinge, or square loss. Note that although our formula is valid for any f^0 and $\hat{f}$, we take $f^0 = \hat{f} = {\rm sign}$, for the classification tasks and $f^0 = \hat{f} = {\rm id}$ for the regression tasks studied here. We now describe in more detail the above-discussed reasons why this model is of interest for machine learning.

Hidden manifold model: The dataset $\mathcal{D}$ can be seen as generated from the *hidden manifold model* introduced in [14]. From this perspective, although $\boldsymbol{x}^\mu$ lives in a p dimensional space, it is parametrised by a latent d-dimensional subspace spanned by the rows of the matrix F which are "hidden" by the application of a scalar non-linear function σ acting component-wise. The labels y^μ are drawn from a generalised linear rule defined on the latent d-dimensional subspace via eq. (1.2). In modern machine learning parlance, this can be seen as data generated by a one-layer generative neural network, such as those trained by generative adversarial networks or variational auto-encoders with random Gaussian inputs $\mathbf{c}^\mu$ and a rather generic weight matrix F [26, 27, 19, 28].

Random features: The model considered in this paper is also an instance of the random features learning discussed in [20] as a way to speed up kernel-ridge-regression. From this perspective, the $\boldsymbol{c}^\mu$s $\in \mathbb{R}^d$ are regarded as a set of d-dimensional i.i.d. Gaussian data points, which are projected by a feature matrix $\mathrm{F} = (\boldsymbol{f}_\rho)_{\rho=1}^p \in \mathbb{R}^{d\times p}$ into a higher dimensional space, followed by a non-linearity σ. In the $p \to \infty$ limit of infinite number of features , performing regression on $\mathcal{D}$ is equivalent to kernel regression on the $\mathbf{c}^\mu$s with a deterministic kernel $K(\mathbf{c}^{\mu_1}, \mathbf{c}^{\mu_2}) = \mathbb{E}_{\mathbf{f}}\left[\sigma(\mathbf{f}\cdot\mathbf{c}^{\mu_1}/d)\cdot\sigma(\mathbf{f}\cdot\mathbf{c}^{\mu_2}/d)\right]$ where $\mathbf{f} \in \mathbb{R}^d$ is sampled in the same way as the lines of F. Random features are also intimately linked with the lazy training regime, where the weights of a neural network stay close to their initial value during training. The training is lazy as opposed to a "rich" one where the weights change enough to learn useful features. In this regime, neural networks become equivalent to a random feature model with correlated features [16, 29, 30, 31, 17, 18].

1.2 Contributions and related work

The main contribution of this work is a closed-form expression for the generalisation error $\epsilon_{\rm g}$, eq. (2.1), that is valid in the high-dimensional limit where the number of samples n, and the two dimensions p and d are large, but their respective ratios are of order one, and for generic sequence of matrices F satisfying the following *balance conditions*:

$$\frac{1}{\sqrt{p}}\sum_{i=1}^{p} w_i^{a_1} w_i^{a_2} \cdots w_i^{a_s} \mathrm{F}_{i\rho_1} \mathrm{F}_{i\rho_2} \cdots \mathrm{F}_{i\rho_q} = O(1), \tag{1.7}$$

where $\{\boldsymbol{w}^a\}_{a=1}^r$ are r independent samples from the Gibbs measure (2.7), and $\rho_1, \rho_2, \cdots, \rho_q \in \{1, \cdots, d\}$, $a_1, a_2, \cdots, a_s \in \{1, \cdots, r\}$ are an arbitrary choice of subset of indices, with $s, q \in \mathbb{Z}_+$. This mild condition is satisfied for instance by random i.i.d matrices and weakly correlated ones, but also by deterministic orthogonal ones such as Fourier and Hadamard matrices, or the weights matrices used in Fast-food [32] and ACDC layers [33]. The non-linearities $f^0, \hat{f}, \sigma$ and the loss function ℓ can be arbitrary. Our result for the generalization error stems from the replica method and we conjecture it to be exact for convex loss functions ℓ. It can also be useful for non-convex loss functions but in those cases it is possible that the so-called replica symmetry breaking [21] needs to be taken into account to obtain an exact expression. In the present paper we hence focus on convex loss functions ℓ and leave the more general case for future work. The final formulas are simpler for nonlinearities σ that give zero when integrated over a centred Gaussian variable, and we hence focus of those cases.

A interesting application of our setting is Ridge regression, i.e. taking $\hat{f}(x) = x$ with square loss, and random i.i.d. Gaussian feature matrices. For this particular case [13] proved an equivalent expression. Indeed, in this case there is an explicit solution of eq. (1.6) that can be rigorously studied with random matrix theory. In a subsequent work [34] derived heuristically a formula for the special case of random i.i.d. Gaussian feature matrices for the maximum margin classification, corresponding to the hinge loss function in our setting, with the difference, however, that the labels y^μ are generated from the $\mathbf{x}^\mu$ instead of the latent variable $\mathbf{c}^\mu$ as in our case.

Our main technical contribution is thus to provide a generic formula for the model described in Section 1.1 for any loss function and for fairly generic features F, including for instance deterministic ones.

The authors of [14] analysed the learning dynamics of a neural network containing several hidden units using a one-pass stochastic gradient descent (SGD) for exactly the same model of data as here. In this online setting, the algorithm is never exposed to a sample twice, greatly simplifying the analysis as what has been learned at a given epoch can be considered independent of the randomness of a new sample. Another motivation of the present work is thus to study the sample complexity for this model (in our case only a bounded number of samples is available, and the one-pass SGD would be highly suboptimal).

An additional technical contribution of our work is to derive an extension of the equivalence between the considered data model and a model with Gaussian covariate, that has been observed and conjectured to hold rather generically in both [14, 34]. While we do not provide a rigorous proof for this equivalence, we show that it arises naturally using the replica method, giving further evidence for its validity.

Finally, the analysis of our formula for particular machine learning tasks of interest allows for an analytical investigation of a rich phenomenology that is also observed empirically in real-life scenarios. In particular

- The double descent behaviour, as termed in [35] and exemplified in [36], is exhibited for the non-regularized logistic regression loss. The peak of worst generalisation does not corresponds to $p = n$ as for the square loss [13], but rather corresponds to the threshold of linear separability of the dataset. We also characterise the location of this threshold, generalising the results of [11] to our model.

- When using projections to approximate kernels, it has been observed that orthogonal features F perform better than random i.i.d. [37]. We show that this behaviour arises from our analytical formula, illustrating the "unreasonable effectiveness of structured random orthogonal embeddings".

- We compute the phase diagram for the generalisation error for the hidden manifold model and discuss the dependence on the various parameters, in particular the ratio between the ambient and latent dimensions.

2 Main analytical results

We now state our two main analytical results. The replica computation used here is in spirit similar to the one performed in a number of tasks for linear and generalised linear models [38, 6, 39, 40], but requires a significant extension to account for the structure of the data. We refer the reader to the Appendix C for the detailed and lengthy derivation of the final formula. The resulting expression is conjectured to be exact and, as we shall see, observed to be accurate even for relatively small dimensions in simulations. Additionally, these formulas reproduce the rigorous results of [13], in the simplest particular case of a Gaussian projection matrix and ridge regression task. It remains a challenge to prove them rigorously in broader generality.

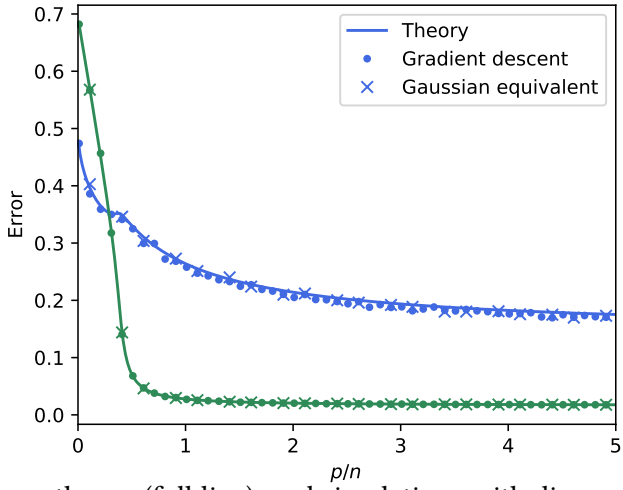

Figure 1: Comparison between theory (full line), and simulations with dimension $d = 200$ on the original model (dots), eq. (1.3), with $\sigma = \text{sign}$, and the Gaussian equivalent model (crosses), eq. (2.10), for logistic loss, regularisation $\lambda = 10^{-3}$, $n/d = 3$. Labels are generated as $y^\mu = \text{sign}\left(\boldsymbol{c}^\mu \cdot \boldsymbol{\theta}^0\right)$ and $\hat{f} = \text{sign}$. Both the training loss (green) and generalisation error (blue) are depicted. The theory and the equivalence with the Gaussian model are observed to be very accurate even at dimensions as small as $d = 200$.

2.1 Generalisation error from replica method

Let F be a feature matrix satisfying the balance condition stated in eq. (1.7). Then, in the high-dimensional limit where $p, d, n \to \infty$ with $\alpha = n/p$, $\gamma = d/p$ fixed, the generalisation error, eq. (1.4), of the model introduced in Sec. (1.4) for σ such that its integral over a centered Gaussian variable is zero (so that $\kappa_0 = 0$ in eq. (2.10)) is given by the following easy-to-evaluate integral:

$$\lim_{n\to\infty} \epsilon_g \to \mathbb{E}_{\lambda,\nu}\left[\left(f^0(\nu) - \hat{f}(\lambda)\right)^2\right] , \tag{2.1}$$

where $f^0(.)$ is defined in (1.2), $\hat{f}(.)$ in (1.5) and (ν, λ) are jointly Gaussian random variables with zero mean and covariance matrix:

$$\Sigma = \begin{pmatrix} \rho & M^\star \\ M^\star & Q^\star \end{pmatrix} \in \mathbb{R}^2 \tag{2.2}$$

with $M^\star = \kappa_1 m_s^\star$, $Q^\star = \kappa_1^2 q_s^\star + \kappa_\star^2 q_w^\star$. The constants $\kappa_\star, \kappa_1$ depend on the nonlinearity σ via eq. (2.10), and $q_s^\star, q_w^\star, m_s^\star$, defined as:

$$\rho = \frac{1}{d}||\boldsymbol{\theta}^0||^2 \qquad q_s^\star = \frac{1}{d}\mathbb{E}||\mathrm{F}\hat{\boldsymbol{w}}||^2$$
$$q_w^\star = \frac{1}{p}\mathbb{E}||\hat{\boldsymbol{w}}||^2 \qquad m_s^\star = \frac{1}{d}\mathbb{E}\left[(\mathrm{F}\hat{\boldsymbol{w}}) \cdot \boldsymbol{\theta}^0\right] \tag{2.3}$$

The values of these parameters correspond to the solution of the optimisation problem in eq. (1.6), and can be obtained as the fixed point solutions of the following set of self-consistent *saddle-point equations*:

$$\begin{cases} \hat{V}_s = \frac{\alpha\kappa_1^2}{\gamma V}\mathbb{E}_\xi\left[\int_{\mathbb{R}} \mathrm{d}y\ \mathcal{Z}\left(y,\omega_0\right)\partial_\omega \eta\left(y,\omega_1\right)\right], \\ \hat{q}_s = \frac{\alpha\kappa_1^2}{\gamma V^2}\mathbb{E}_\xi\left[\int_{\mathbb{R}} \mathrm{d}y\ \mathcal{Z}\left(y,\omega_0\right)\left(\eta\left(y,\omega_1\right)-\omega_1\right)^2\right], \\ \hat{m}_s = \frac{\alpha\kappa_1}{\gamma V}\mathbb{E}_\xi\left[\int_{\mathbb{R}} \mathrm{d}y\ \partial_\omega\mathcal{Z}\left(y,\omega_0\right)\left(\eta\left(y,\omega_1\right)-\omega_1\right)\right], \\ \\ \hat{V}_w = \frac{\alpha\kappa_\star^2}{V}\mathbb{E}_\xi\left[\int_{\mathbb{R}} \mathrm{d}y\ \mathcal{Z}\left(y,\omega_0\right)\partial_\omega \eta\left(y,\omega_1\right)\right], \\ \hat{q}_w = \frac{\alpha\kappa_\star^2}{V^2}\mathbb{E}_\xi\left[\int_{\mathbb{R}} \mathrm{d}y\ \mathcal{Z}\left(y,\omega_0\right)\left(\eta\left(y,\omega_1\right)-\omega_1\right)^2\right], \end{cases} \quad \begin{cases} V_s = \frac{1}{\hat{V}_s}\left(1 - z\ g_\mu(-z)\right), \\ q_s = \frac{\hat{m}_s^2+\hat{q}_s}{\hat{V}_s}\left[1 - 2z g_\mu(-z) + z^2 g'_\mu(-z)\right] \\ \quad\quad -\frac{\hat{q}_w}{(\lambda+\hat{V}_w)\hat{V}_s}\left[-z g_\mu(-z) + z^2 g'_\mu(-z)\right], \\ m_s = \frac{\hat{m}_s}{\hat{V}_s}\left(1 - z\ g_\mu(-z)\right), \\ \\ V_w = \frac{\gamma}{\lambda+\hat{V}_w}\left[\frac{1}{\gamma} - 1 + z g_\mu(-z)\right], \\ q_w = \gamma\frac{\hat{q}_w}{(\lambda+\hat{V}_w)^2}\left[\frac{1}{\gamma} - 1 + z^2 g'_\mu(-z)\right], \\ \quad\quad +\frac{\hat{m}_s^2+\hat{q}_s}{(\lambda+\hat{V}_w)\hat{V}_s}\left[-z g_\mu(-z) + z^2 g'_\mu(-z)\right], \end{cases} \tag{2.4}$$

written in terms of the following auxiliary variables $\xi \sim \mathcal{N}(0,1)$, $z = \frac{\lambda+\hat{V}_w}{\hat{V}_s}$ and functions:

$$\begin{aligned} \eta(y,\omega) &= \underset{x\in\mathbb{R}}{\operatorname{argmin}}\left[\frac{(x-\omega)^2}{2V} + \ell(y,x)\right], \\ \mathcal{Z}(y,\omega) &= \int \frac{\mathrm{d}x}{\sqrt{2\pi V^0}} e^{-\frac{1}{2V^0}(x-\omega)^2}\delta\left(y - f^0(x)\right) \end{aligned} \tag{2.5}$$

where $V = \kappa_1^2 V_s + \kappa_\star^2 V_w$, $V^0 = \rho - \frac{M^2}{Q}$, $Q = \kappa_1^2 q_s + \kappa_\star^2 q_w$, $M = \kappa_1 m_s$, $\omega_0 = M/\sqrt{Q}\xi$ and $\omega_1 = \sqrt{Q}\xi$. In the above, we assume that the matrix $\mathrm{F}\mathrm{F}^\top \in \mathbb{R}^{d\times d}$ associated to the feature map F has a well behaved spectral density, and denote g_μ its Stieltjes transform.

The training loss on the dataset $\mathcal{D} = \{\boldsymbol{x}^\mu, y^\mu\}_{\mu=1}^n$ can also be obtained from the solution of the above equations as

$$\lim_{n\to\infty} \epsilon_t \to \frac{\lambda}{2\alpha} q_w^\star + \mathbb{E}_{\xi,y}\left[\mathcal{Z}\left(y,\omega_0^\star\right)\ell\left(y,\eta(y,\omega_1^\star)\right)\right] \tag{2.6}$$

where as before $\xi \sim \mathcal{N}(0,1)$, $y \sim \mathrm{Uni}(\mathbb{R})$ and $\mathcal{Z}, \eta$ are the same as in eq. (2.5), evaluated at the solution of the above saddle-point equations $\omega_0^\star = M^\star/\sqrt{Q^\star}\xi$, $\omega_1^\star = \sqrt{Q^\star}\xi$.

Sketch of derivation — We now sketch the derivation of the above result. A complete and detailed account can be found in Appendix C and D. The derivation is based on the key observation that in the high-dimensional limit the asymptotic generalisation error only depends on the solution $\hat{\boldsymbol{w}} \in \mathbb{R}^p$ of eq. (1.5) through the scalar parameters $(q_s^\star, q_w^\star, m_s^\star)$ defined in eq. (2.3). The idea is therefore to rewrite the high-dimensional optimisation problem in terms of only these scalar parameters.

The first step is to note that the solution of eq. (1.6) can be written as the average of the following Gibbs measure

$$\mu_\beta(\boldsymbol{w}|\{\boldsymbol{x}^\mu, y^\mu\}) = \frac{1}{\mathcal{Z}_\beta} e^{-\beta\left[\sum_{\mu=1}^n \ell(y^\mu, \boldsymbol{x}^\mu\cdot\boldsymbol{w}) + \frac{\lambda}{2}||\boldsymbol{w}||_2^2\right]}, \tag{2.7}$$

in the limit $\beta \to \infty$. Of course, we have not gained much, since an exact calculation of μ_β is intractable for large values of n, p and d. This is where the replica method comes in. It states that the distribution of the free energy density $f = -\log \mathcal{Z}_\beta$ (when seen as a random variable over different realisations of dataset $\mathcal{D}$) associated with the measure μ_β concentrates, in the high-dimensional limit, around a value f_β that depends only on the averaged *replicated partition function* $\mathcal{Z}_\beta^r$ obtained by taking $r > 0$ copies of $\mathcal{Z}_\beta$:

$$f_\beta = \lim_{r\to 0^+} \frac{\mathrm{d}}{\mathrm{d}r} \lim_{p\to\infty}\left[-\frac{1}{p}\left(\mathbb{E}_{\{\boldsymbol{x}^\mu, y^\mu\}}\mathcal{Z}_\beta^r\right)\right]. \tag{2.8}$$

Interestingly, $\mathbb{E}_{\{\boldsymbol{x},y\}}\mathcal{Z}_\beta^r$ can be computed explicitly (see Appendix C), the upshot being that it can be written as

$$\mathbb{E}_{\{\boldsymbol{x}^\mu,y^\mu\}}\mathcal{Z}_\beta^r \propto \int \mathrm{d}q_s\mathrm{d}q_w\mathrm{d}m_s\; e^{p\Phi_\beta^{(r)}(m_s,q_s,q_w)} \tag{2.9}$$

where Φ_β - known as the replica symmetric potential - is a concave function depending only on the following scalar parameters:

$$q_s = \frac{1}{d}||\mathrm{F}\boldsymbol{w}||^2, \qquad q_w = \frac{1}{p}||\boldsymbol{w}||^2, \qquad m_s = \frac{1}{d}(\mathrm{F}\boldsymbol{w})\cdot\boldsymbol{\theta}^0$$

for $\boldsymbol{w}\sim\mu_\beta$. In the limits of eq. (2.8), this integral concentrates around the extremum of the potential $\Phi_\beta^{(0)}$. Since the optimisation problem in eq. (1.5) is convex, by construction as $\beta\to\infty$ the overlap parameters $(q_s^\star, q_w^\star, m_s^\star)$ satisfying this optimisation problem are precisely the ones of eq. (2.3) corresponding to the solution $\hat{\boldsymbol{w}}\in\mathbb{R}^p$ of eq. (1.5).

In summa, the replica method allows to circumvent the hard-to-solve high-dimensional optimisation problem eq. (1.5) by directly computing the generalisation error in eq. (1.4) in terms of a simpler scalar optimisation. Doing gradient descent in $\Phi_\beta^{(0)}$ and taking $\beta\to\infty$ lead to the saddle-point eqs. (2.4).

2.2 Replicated Gaussian Equivalence

The backbone of the replica derivation sketched above is a central limit theorem type result coined as the *Gaussian equivalence theorem* (GET) from [14] used it in the context of the "replicated" Gibbs measure obtained by taking r copies of (2.7) - see Appendix B for more detail. In this approach, we need to assume that the "balance condition" (1.7) applies with probability one when the weights w are sampled from the replicated measure. We shall use this assumption in the following, checking its self-consistency via agreement with simulations (see Appendix E for more details on the simulations).

It is interesting to observe that, when applying the GET in the context of our replica calculation, the resulting asymptotic generalisation error stated in Sec. 2.1 is equivalent to the asymptotic generalisation error of the following linear model:

$$\boldsymbol{x}^\mu = \kappa_0\mathbf{1} + \kappa_1\frac{1}{\sqrt{d}}\mathrm{F}^\top\boldsymbol{c}^\mu + \kappa_\star\,\boldsymbol{z}^\mu\,, \tag{2.10}$$

with $\kappa_0 = \mathbb{E}\left[\sigma(z)\right]$, $\kappa_1 \equiv \mathbb{E}\left[z\sigma(z)\right]$, $\kappa_\star \equiv \mathbb{E}\left[\sigma(z)^2\right] - \kappa_0^2 - \kappa_1^2$, and $\boldsymbol{z}^\mu\sim\mathcal{N}(\mathbf{0},\mathrm{I}_p)$. We have for instance, $(\kappa_0,\kappa_1,\kappa_\star)\approx\left(0,\frac{2}{\sqrt{3\pi}},0.2003\right)$ for $\sigma=\mathrm{erf}$ and $(\kappa_0,\kappa_1,\kappa_\star)=\left(0,\sqrt{\frac{2}{\pi}},\sqrt{1-\frac{2}{\pi}}\right)$ for $\sigma=\mathrm{sign}$, two cases explored in the next section. This equivalence constitutes a result with an interest in its own, with applicability beyond the scope of the generalised linear task eq. (1.6) studied here.

Equation (2.10) is precisely the mapping obtained by [13], who proved its validity rigorously in the particular case of the square loss and Gaussian random matrix F using random matrix theory. The same equivalence arises in the analysis of kernel random matrices [41, 42] and in the study of online learning [14]. The replica method thus suggests that the equivalence actually holds in a much larger class of learning problem, as conjectured as well in [34], and numerically confirmed in all our numerical tests. It also potentially allows generalisation of the analysis in this paper for data coming from a learned generative adversarial network, along the lines of [43, 28].

Fig. 1 illustrates the remarkable agreement between the result of the generalisation formula, eq. (2.1) and simulations both on the data eq. (1.3) with $\sigma(x)=\mathrm{sign}(x)$ non-linearity, and on the Gaussian equivalent data eq. (2.10) where the non-linearity is replaced by rescaling by a constant plus noise. The agreement is flawless as implied by the theory in the high-dimensional limit, testifying that the used system size $d=200$ is sufficiently large for the asymptotic theory to be relevant. We observed similar good agreement between the theory and simulation in all the cases we tested, in particular in all those presented in the following.

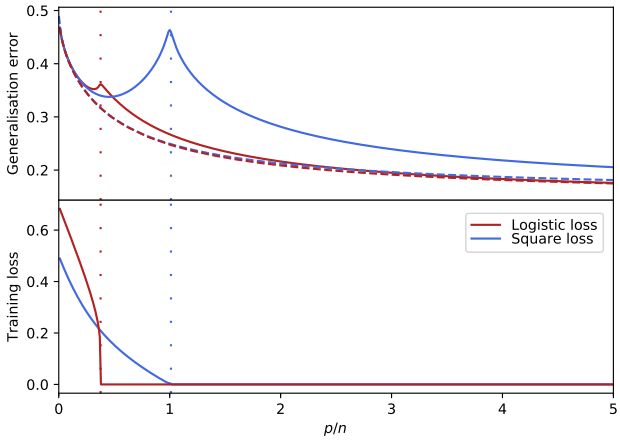

Figure 2: Upper panel: Generalisation error evaluated from eq. (2.1) plotted against the number of random Gaussian features per sample $p/n = 1/\alpha$ and fixed ratio between the number of samples and dimension $n/d = \alpha/\gamma = 3$ for logistic loss (red), square loss (blue). Labels are generated as $y^\mu = \text{sign}\left(\boldsymbol{c}^\mu \cdot \boldsymbol{\theta}^0\right)$, data as $\boldsymbol{x}^\mu = \text{sign}\left(\text{F}^\top \boldsymbol{c}^\mu\right)$ and $\hat{f} = \text{sign}$ for two different values of regularisation λ, a small penalty $\lambda = 10^{-4}$ (full line) and a value of lambda optimised for every p/n (dashed line). Lower pannel: The training loss corresponding to $\lambda = 10^{-4}$ is depicted.

3 Applications of the generalisation formula

3.1 Double descent for classification with logistic loss

Among the surprising observations in modern machine learning is the fact that one can use learning methods that achieve zero training error, yet their generalisation error does not deteriorate as more and more parameters are added into the neural network. The study of such "interpolators" have attracted a growing attention over the last few years [9, 36, 35, 44, 12, 13, 45, 46], as it violates basic intuition on the bias-variance trade-off [47]. Indeed classical learning theory suggests that generalisation should first improve then worsen when increasing model complexity, following a U-shape curve. Many methods, including neural networks, instead follow a so-called "double descent curve" [36, 35] that displays two regimes: the "classical" U-curve found at low number of parameters is followed at high number of parameters by an interpolation regime where the generalisation error decreases monotonically. Consequently neural networks do not drastically overfit even when using much more parameters than data samples [48], as actually observed already in the classical work [47]. Between the two regimes, a "peak" in the generalization is observed at the interpolation threshold [36, 35]. It should, however, be noted that existence of this "interpolation" peak is an independent phenomenon from the lack of overfitting in highly over-parametrized networks, and indeed in a number of the related works these two phenomena were observed separately [49, 22, 9, 47]. Scaling properties of the peak and its relation to the jamming phenomena in physics are studied in detail in [36, 45].

Among the simple models that allow to observe this behaviour, random projections —that are related to lazy training and kernel methods— are arguably the most natural one. The double descent has been analysed in detail in the present model in the specific case of a square loss on a regression task with random Gaussian features [13]. Our analysis allows to show the generality and the robustness of the phenomenon to other tasks, matrices and losses. In Fig. 2 we compare the double descent as present in the square loss (blue line) with the one of logistic loss (red line) for random Gaussian features. We plot the value of the generalisation error at small values of the regularisation λ (full line), and for optimal value of λ (dashed line) for a fixed ratio between the number of samples and the dimension n/d as a function of the number of random features per sample p/n. We also plot the value of the training error (lower pannel) for a small regularisation value, showing that the peaks indeed occur when the training loss goes to zero. For the square loss the peak appears at $1/\alpha = p/n = 1$ when the system of n linear equations with p parameters becomes solvable. For the logistic loss the peak instead appears at a value $1/\alpha^*$ where the data $\mathcal{D}$ become linearly separable and hence the logistic loss can

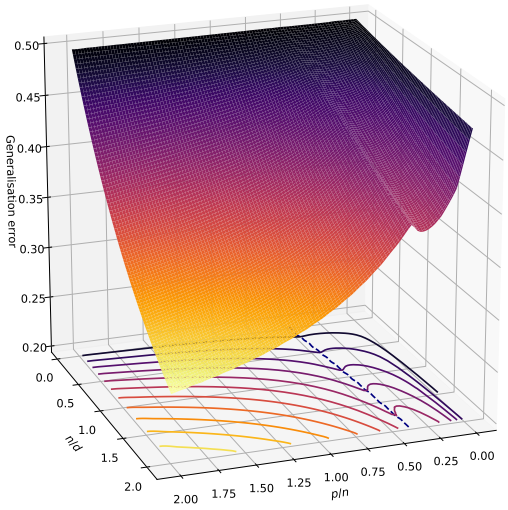

Figure 3: Generalisation error of the logistic loss at fixed very small regularisation $\lambda = 10^{-4}$, as a function of $n/d = \alpha/\gamma$ and $p/n = 1/\alpha$, for random Gaussian features. Labels are generated with $y^\mu = \text{sign}\left(\boldsymbol{c}^\mu \cdot \boldsymbol{\theta}^0\right)$, the data $\boldsymbol{x}^\mu = \text{sign}\left(\text{F}^\top \boldsymbol{c}^\mu\right)$ and $\hat{f} = \text{sign}$. The interpolation peak happening where data become linearly separable is clearly visible here.

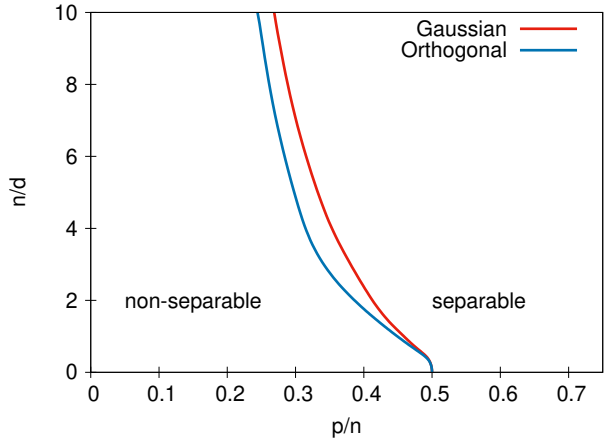

Figure 4: The position of the interpolation peak in logistic regression with $\lambda = 10^{-4}$, where data become linearly separable, as a function of the ratio between the number of samples n and the dimension d. Labels are generated with $y^\mu = \text{sign}\left(\boldsymbol{c}^\mu \cdot \boldsymbol{\theta}^0\right)$, the data $\boldsymbol{x}^\mu = \text{sign}\left(\text{F}^\top \boldsymbol{c}^\mu\right)$ and $\hat{f} = \text{sign}$. The red line is with Gaussian random features, the blue line with orthogonal features. We see that for linear separability we need smaller number of projections p with orthogonal random features than with Gaussian.

be optimised down to zero. These values $1/\alpha^*$ depends on the value n/d, and this dependence is plotted in Fig. 4. For very large dimension d, i.e. $n/d \to 0$ the data matrix X is close to iid random matrix and hence the $\alpha^*(n/d = 0) = 2$ as famously derived in classical work by Cover [50]. For $n/d > 0$ the α^* is growing ($1/\alpha^*$ decreasing) as correlations make data easier to linearly separate, similarly as in [11].

Fig. 2 also shows that better error can be achieved with the logistic loss compared to the square loss, both for small and optimal regularisations, except in a small region around the logistic interpolation peak. In the Kernel limit, the generalization error at optimal regularisation saturates at $\epsilon_g(p/n \to \infty) \simeq 0.17$ for square loss and at $\epsilon_g(p/n \to \infty) \simeq 0.16$ for logistic loss. Fig. 3 then depicts a 3D plot of the generalisation error also illustrating the position of the interpolation peak.

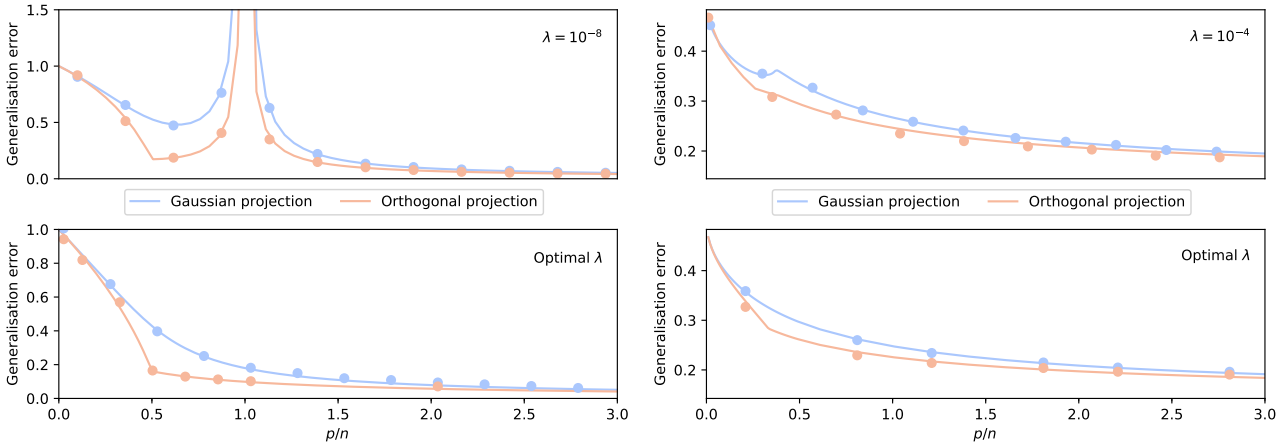

Figure 5: Generalisation error against the number of features per sample p/n, for a regression problem (left) and a classification one (right). **Left (ridge regression):** We used $n/d = 2$ and generated labels as $y^\mu = \boldsymbol{c}^\mu \cdot \boldsymbol{\theta}^0$, data as $\boldsymbol{x}^\mu = \text{sign}\left(\text{F}^\top \boldsymbol{c}^\mu\right)$ and $\hat{f}(x) = x$. The two curves correspond to Ridge regression with Gaussian (blue) versus orthogonal (red) projection matrix F for both $\lambda = 10^{-8}$ (top) and optimal regularisation λ (bottom). **Right (logistic classification)**: We used $n/d = 3$ and generated labels as $y^\mu = \text{sign}\left(\boldsymbol{c}^\mu \cdot \boldsymbol{\theta}^0\right)$, data as $\boldsymbol{x}^\mu = \text{sign}\left(\text{F}^\top \boldsymbol{c}^\mu\right)$ and $\hat{f} = \text{sign}$. The two curves correspond to a logistic classification with again Gaussian (blue) versus orthogonal (red) projection matrix F for both $\lambda = 10^{-4}$ and optimal regularisation λ. In all cases, full lines is the theoretical prediction, and points correspond to gradient-descent simulations with $d = 200$.

3.2 Random features: Gaussian versus orthogonal

Kernel methods are a very popular class of machine learning techniques, achieving state-of-the-art performance on a variety of tasks with theoretical guarantees [51, 52, 53]. In the context of neural network, they are the subject of a renewal of interest in the context of the Neural Tangent Kernel [17]. Applying kernel methods to large-scale "big data" problems, however, poses many computational challenges, and this has motivated a variety of contributions to develop them at scale, see, e.g., [52, 54, 55, 56]. Random features [20] are among the most popular techniques to do so.

Here, we want to compare the performance of random projection with respect to structured ones, and in particular orthogonal random projections [37] or deterministic matrices such as the Fourier and Hadamard ones used in fast projection methods [32, 57, 58]. In our computation, the effect of orthogonality appears in the spectrum of the matrix $\mathbf{F} = \text{U}^\top \mathbf{D} \text{V}$ for $\text{U} \in \mathbb{R}^{d\times d}$, $\text{V} \in \mathbb{R}^{p\times p}$ two orthogonal matrices drawn from the Haar measure, and $\text{D} \in \mathbb{R}^{d\times p}$ a diagonal matrix of rank $\min(d, p)$. In order to compare the results with the Gaussian case, we fix the diagonal entries $d_k = \max(\sqrt{\gamma}, 1)$ of D such that an arbitrary projected vector has the same norm, on average, to the Gaussian case.

Fig. 5 shows that random orthogonal embeddings always outperform Gaussian random projections, in line with empirical observations, and that they allow to reach the kernel limit with fewer number of projections. Their behaviour is, however, qualitatively similar to the one of random i.i.d. projections. We also show in Fig. 4 that orthogonal projections allow to separate the data more easily than the Gaussian ones, as the phase transition curve delimiting the linear separability of the logistic loss get shifted to the left.

3.3 The hidden manifold model phase diagram

In this subsection we consider the hidden manifold model where p-dimensional x data lie on a d-dimensional manifold, we have mainly in mind $d < p$. The labels y are generated using the coordinates on the manifold, eq. (1.2).

In Fig. 6 we plot the generalisation error of classification with the square loss for various values of the regularisation λ. We fix the ratio between the dimension of the sub-manifold and the dimensionality of the input data to $d/p = 0.1$ and plot the learning curve, i.e. the error as a function of the number of samples per

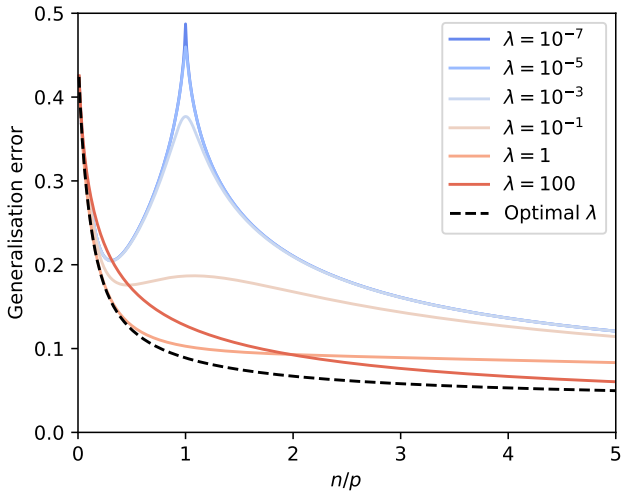

Figure 6: Generalisation error against the number of samples per dimension, $\alpha = n/p$, and fixed ratio between the latent and data dimension, $d/p = 0.1$, for a classification task with square loss on labels generated as $y^\mu = \text{sign}\left(\boldsymbol{c}^\mu \cdot \boldsymbol{\theta}^0\right)$, data $\boldsymbol{x}^\mu = \text{erf}\left(\text{F}^\top \boldsymbol{c}^\mu\right)$ and $\hat{f} = \text{sign}$, for different values of the regularisation λ (full lines), including the optimal regularisation value (dashed).

dimension. Depending on the value of the regularisation, we observe that the interpolation peak, which is at $\alpha = 1$ at very small regularisation (here the over-parametrised regime is on the left hand side), decreases for larger regularisation λ. A similar behaviour has been observed for other models in the past, see e.g. [49]. Finally Fig. 6 depicts the error for optimised regularisation parameter in the black dashed line. For large number of samples we observe the generalisation error at optimal regularisation to saturate in this case at $\epsilon_g(\alpha \to \infty) \to 0.0325$. A challenge for future work is to see whether better performance can be achieved on this model by including hidden variables into the neural network.

Fig. 7 then shows the generalisation error for the optimised regularisation λ with square loss as a function of the ratio between the latent and the data dimensions d/p. In the limit $d/p \gg 1$ the data matrix becomes close to a random i.i.d. matrix and the labels are effectively random, thus only bad generalisation can be reached. Interestingly, as d/p decreases to small values even the simple classification with regularised square loss is able to "disentangle" the hidden manifold structure in the data and to reach a rather low generalisation error. The figure quantifies how the error deteriorates when the ratio between the two dimensions d/p increases. Rather remarkably, for a low d/p a good generalisation error is achieved even in the over-parametrised regime, where the dimension is larger than the number of samples, $p > n$. In a sense, the square loss linear classification is able to locate the low-dimensional subspace and find good generalisation even in the over-parametrised regime as long as roughly $d \lesssim n$. The observed results are in qualitative agreement with the results of learning with stochastic gradient descent in [14] where for very low d/p good generalisation error was observed in the hidden manifold model, but a rather bad one for $d/p = 0.5$.

Acknowledgements

This work is supported by the ERC under the European Union's Horizon 2020 Research and Innovation Program 714608-SMiLe, as well as by the French Agence Nationale de la Recherche under grant ANR-17-CE23-0023-01 PAIL and ANR-19-P3IA-0001 PRAIRIE. We also acknowledge support from the chaire CFM-ENS "Science desdonnées". We thank Google Cloud for providing us access to their platform through the Research Credits Application program. BL was partially financed by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

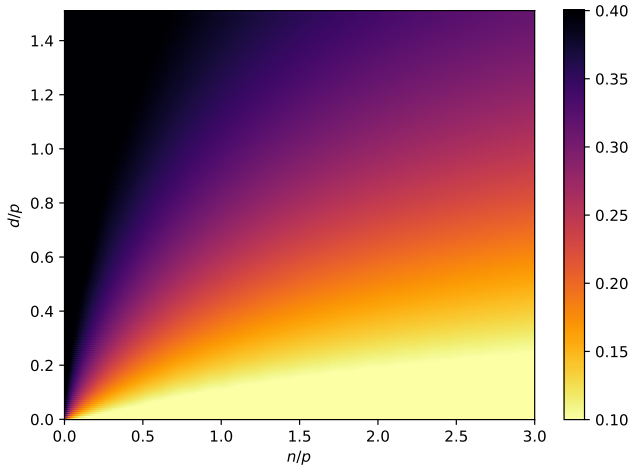

Figure 7: Heat-map of the generalisation error as a function of the number of samples per data dimension n/p against the ratio of the latent and data dimension d/p, for a classification task with square loss on labels $y^\mu = \text{sign}\left(\boldsymbol{c}^\mu \cdot \boldsymbol{\theta}^0\right)$ and data $\boldsymbol{x}^\mu = \text{erf}\left(\text{F}^\top \boldsymbol{c}^\mu\right)$ for the optimal values of the regularisation λ.

Appendix

A Definitions and notations

In this section we recall the models introduced in the main body of the article, and introduce the notations used throughout the appendices.

A.1 The dataset

In this work we study a series of regression and classification tasks for a dataset $\{\boldsymbol{x}^\mu, y^\mu\}_{\mu=1}^n$ with labels $y^\mu \in \mathbb{R}$ sampled identically from a *generalised linear model*:

$$y^\mu \sim P_y^0\left(y^\mu \middle| \frac{\boldsymbol{c}^\mu \cdot \boldsymbol{\theta}^0}{\sqrt{d}}\right), \tag{A.1}$$

where the output-channel $P_y^0\left(\cdot\right)$ is defined as:

$$P_y^0\left(y^\mu \middle| \frac{\boldsymbol{c}^\mu \cdot \boldsymbol{\theta}^0}{\sqrt{d}}\right) = \int d\xi^\mu P\left(\xi^\mu\right) \delta\left(y^\mu - f^0\left(\frac{\boldsymbol{c}^\mu \cdot \boldsymbol{\theta}^0}{\sqrt{d}}; \xi^\mu\right)\right) \tag{A.2}$$

for some noise ξ^μ and for data points $\boldsymbol{x}^\mu \in \mathbb{R}^p$ given by:

$$\boldsymbol{x}^\mu = \sigma\left(\frac{1}{\sqrt{d}} \sum_{\rho=1}^{d} c_\rho^\mu \boldsymbol{f}_\rho\right). \tag{A.3}$$

The vectors $\boldsymbol{c}^\mu \in \mathbb{R}^d$ is assumed to be identically drawn from $\mathcal{N}(0, \mathrm{I}_d)$, and $\boldsymbol{\theta}^0 \in \mathbb{R}^d$ from a separable distribution P_θ. The family of vectors $\boldsymbol{f}_\rho \in \mathbb{R}^p$ and the scalar function $\sigma : \mathbb{R} \to \mathbb{R}$ can be arbitrary.

Although our results are valid for the general model introduced above, the two examples we will be exploring in this work are the noisy linear channel (for regression tasks) and the deterministic sign channel (for classification tasks):

$$y^\mu = \frac{\boldsymbol{c}^\mu \cdot \boldsymbol{\theta}^0}{\sqrt{d}} + \sqrt{\Delta}\, \xi^\mu \qquad \Leftrightarrow \qquad P_y^0\left(\boldsymbol{y} \middle| \frac{\boldsymbol{c}^\mu \cdot \boldsymbol{\theta}^0}{\sqrt{d}}\right) = \prod_{\mu=1}^{n} \mathcal{N}\left(y^\mu; \frac{\boldsymbol{c}^\mu \cdot \boldsymbol{\theta}^0}{\sqrt{d}}, \Delta\right) \tag{A.4}$$

$$y^\mu = \text{sign}\left(\frac{\boldsymbol{c}^\mu \cdot \boldsymbol{\theta}^0}{\sqrt{d}}\right) \qquad \Leftrightarrow \qquad P_y^0\left(\boldsymbol{y} \middle| \frac{\boldsymbol{c}^\mu \cdot \boldsymbol{\theta}^0}{\sqrt{d}}\right) = \prod_{\mu=1}^{n} \delta\left(y^\mu - \text{sign}\left(\frac{\boldsymbol{c}^\mu \cdot \boldsymbol{\theta}^0}{\sqrt{d}}\right)\right) \tag{A.5}$$

where $\xi^\mu \sim \mathcal{N}(0,1)$ and $\Delta > 0$.

This dataset can be regarded from two different perspectives.

Hidden manifold model: The dataset $\{\boldsymbol{x}^\mu, y^\mu\}_{\mu=1,\cdots,n}$ is precisely the *hidden manifold model* introduced in [14] to study the dynamics of online learning in a synthetic but structured dataset. From this perspective, although $\boldsymbol{x}^\mu$ lives in a p dimensional space, it is parametrised by a latent $d < p$-dimensional subspace spanned by the basis $\{\boldsymbol{f}_\rho\}_{\rho=1,\cdots,d}$ which is "hidden" by the application of a scalar nonlinear function σ acting component-wise. The labels $\boldsymbol{y}^\mu$ are then drawn from a generalised linear rule defined on the latent d-dimensional space.

Random features model: The dataset $\{\boldsymbol{x}^\mu, y^\mu\}_{\mu=1,\cdots,n}$ is tightly related to the Random Features model studied in [20] as a random approximation for kernel ridge regression. In this perspective, $\boldsymbol{c}^\mu \in \mathbb{R}^d$ is regarded as a collection of d-dimensional data points which are projected by a random feature matrix $\mathrm{F} = (\boldsymbol{f}_\rho)_{\rho=1}^p \in \mathbb{R}^{d\times p}$ into a higher dimensional space, followed by a non-linearity σ. In the limit of infinite number of features $d, p \to \infty$ with fixed ratio d/p, performing ridge regression of $\boldsymbol{x}^\mu$ is equivalent to kernel ridge regression with a limiting kernel depending on the distribution of the feature matrix F and on the non-linearity σ.

A.2 The task

In this work, we study the problem of learning the rule from eq. (A.1) from the dataset $\{(\boldsymbol{x}^\mu, y^\mu)\}_{\mu=1,\cdots,n}$ introduced above with a *generalised linear model*:

$$\hat{y}^\mu = \hat{f}\left(\boldsymbol{x}^\mu \cdot \hat{\boldsymbol{w}}\right) \tag{A.6}$$

where the weights $\boldsymbol{w} \in \mathbb{R}^p$ are learned by minimising a loss function with a ridge regularisation term:

$$\hat{\boldsymbol{w}} = \min_{\boldsymbol{w}} \left[\sum_{\mu=1}^{n} \ell(y^\mu, \boldsymbol{x}^\mu \cdot \boldsymbol{w}) + \frac{\lambda}{2}||\boldsymbol{w}||_2^2\right] . \tag{A.7}$$

for $\lambda > 0$.

It is worth stressing that our results hold for general ℓ, $\hat{f}$ and f^0 - including non-convex loss functions. However, for the purpose of the applications explored in this manuscript, we will be mostly interested in the cases $\hat{f}(x) = f^0(x) = x$ for regression and $\hat{f}(x) = f^0(x) = \mathrm{sign}(x)$ for classification, and we will focus on the following two loss functions:

$$\ell(y^\mu, \boldsymbol{x}^\mu \cdot \boldsymbol{w}) = \begin{cases} \frac{1}{2}(y^\mu - \boldsymbol{x}^\mu \cdot \boldsymbol{w})^2, & \text{square loss} \\ \log\left(1 + e^{-y^\mu(\boldsymbol{x}^\mu \cdot \boldsymbol{w})}\right), & \text{logistic loss} \end{cases} \tag{A.8}$$

Note that these loss functions are strictly convex. Therefore, for these losses, the regularised optimisation problem in (A.7) has a unique solution.

Given a new pair $(\boldsymbol{x}^{\text{new}}, y^{\text{new}})$ drawn independently from the same distribution as $\{(\boldsymbol{x}^\mu, y^\mu)\}_{\mu=1}^n$, we define the success of our fit through the generalisation error, defined as:

$$\epsilon_g = \frac{1}{4^k}\mathbb{E}_{\boldsymbol{x}^{\text{new}}, y^{\text{new}}}\left(y^{\text{new}} - \hat{y}^{\text{new}}\right)^2 \tag{A.9}$$

where $\hat{y}^{\text{new}} = \hat{f}(\boldsymbol{x}^{\text{new}} \cdot \hat{\boldsymbol{w}})$, and for convenience we choose $k = 0$ for the regression tasks and $k = 1$ for the classification task, such that the generalisation error in this case counts misclassification. Note that for a classification problem, the generalisation error is just one minus the classification error.

Similarly, we define the *training loss* on the dataset $\{\boldsymbol{x}^\mu, y^\mu\}_{\mu=1}^n$ as:

$$\epsilon_t = \frac{1}{n}\mathbb{E}_{\{\boldsymbol{x}^\mu, y^\mu\}}\left[\sum_{\mu=1}^{n} \ell\left(y^\mu, \boldsymbol{x}^\mu \cdot \hat{\boldsymbol{w}}\right) + \frac{\lambda}{2}\|\hat{\boldsymbol{w}}\|_2^2\right] . \tag{A.10}$$

Finally, all the results of this manuscript are derived in the *high-dimensional limit*, also known as *thermodynamic limit* in the physics literature, in which we take $p, d, n \to \infty$ while keeping the ratios $\alpha = n/p$, $\gamma = d/p$ fixed.

B Gaussian equivalence theorem

In this section we introduce the *replicated Gaussian equivalence* (rGE), a central result we will need for our replica calculation of the generalisation error in Sec. 2.1 of the main body. The rGET is a stronger version of the Gaussian equivalence theorem (GET) that was introduced and proved in [14]. Previously, particular cases of the GET were derived in the context of random matrix theory [59, 41, 60, 42]. The gaussian equivalence has also been stated and used in [13, 34].

B.1 Gaussian equivalence theorem

Let $\mathrm{F} \in \mathbb{R}^{d\times p}$ be a fixed matrix, $\boldsymbol{w}^a \in \mathbb{R}^p$, $1 \le a \le r$ be a family of vectors, $\boldsymbol{\theta}^0 \in \mathbb{R}^d$ be a fixed vector and $\sigma : \mathbb{R} \to \mathbb{R}$ be a scalar function acting component-wise on vectors.

Let $\boldsymbol{c} \in \mathbb{R}^d$ be a Gaussian vector $\mathcal{N}(0, \mathrm{I}_d)$. The GET is a statement about the (joint) statistics of the following $r+1$ random variables

$$\lambda^a = \frac{1}{\sqrt{p}} \boldsymbol{w}^a \cdot \sigma(\boldsymbol{u}) \in \mathbb{R}, \qquad \nu = \frac{1}{\sqrt{d}} \boldsymbol{c} \cdot \boldsymbol{\theta}^0 \in \mathbb{R}. \tag{B.1}$$

in the asymptotic limit where $d, p \to \infty$ with fixed p/d and fixed r. For simplicity, assume that $\sigma(x) = -\sigma(-x)$ is an odd function. Further, suppose that in the previously introduced limit the following two balance conditions hold:

Condition 1:

$$\frac{1}{\sqrt{p}} \sum_{i=1}^{p} F_{i\rho} F_{j\rho} = O(1), \tag{B.2}$$

for any ρ.

Condition 2:

$$S^{a_1,\dots,a_k}_{\rho_1,\dots,\rho_q} = \frac{1}{\sqrt{p}} \sum_{i=1}^{p} w_i^{a_1} w_i^{a_2} \cdots w_i^{a_k} F_{i\rho_1} F_{i\rho_2} \cdots F_{i\rho_q} = O(1), \tag{B.3}$$

for any integers $k \ge 0$, $q > 0$, for any choice of indices $\rho_1, \rho_2, \cdots, \rho_q \in \{1, \cdots, d\}$ all distinct from each other, and for any choice of indices $a_1, a_2, \cdots, a_k \in \{1, \cdots, r\}$. Under the aforementioned conditions, the following theorem holds:

Theorem 1. *In the limit $d, p \to \infty$ with fixed p/d, the random variables $\{\boldsymbol{u}, \lambda^a, u\}$ are jointly normal, with zero mean and covariances:*

$$\mathbb{E}\left[\lambda^a \lambda^b\right] = \frac{\kappa_\star^2}{p} \boldsymbol{w}^a \cdot \boldsymbol{w}^b + \frac{\kappa_1^2}{d} \boldsymbol{s}^a \cdot \boldsymbol{s}^b, \qquad \mathbb{E}\left[\nu^2\right] = \frac{1}{d} ||\boldsymbol{\theta}^0||^2$$

$$\mathbb{E}\left[\lambda^a \nu\right] = \frac{\kappa_1}{d} \boldsymbol{s}^a \cdot \boldsymbol{\theta}^0 \tag{B.4}$$

where:

$$\boldsymbol{s}^a = \frac{1}{\sqrt{p}} \mathrm{F} \boldsymbol{w}^a \in \mathbb{R}^d, \qquad a = 1, \cdots, r \tag{B.5}$$

and

$$\kappa_0 = \mathbb{E}_z\left[\sigma(z)\right], \qquad \kappa_1 = \mathbb{E}_z\left[z\sigma(z)\right], \qquad \kappa_\star = \mathbb{E}_z\left[\sigma(z)^2\right] - \kappa_0^2 - \kappa_1^2 \tag{B.6}$$

where $z \sim \mathcal{N}(0, 1)$.

B.2 Replicated Gaussian equivalence

Note that the GET holds for a fixed family $\{\boldsymbol{w}^a\}_{a=1}^r$ and matrix $\mathrm{F} \in \mathbb{R}^{d\times p}$ satisfying the balance condition from eq. (B.3). In the replica setting, we will need to apply the GET under an average over r samples (refered here as *replicas*) of the Gibbs distribution μ_β, introduced in eq. 2.7 on the main. We therefore shall require the assumption that the balance condition eq. (B.3) holds for any sample of μ_β. We refer to this stronger version of the GET as the *replicated Gaussian equivalence* (rGE). Although proving this result is out of the scope of the present work, we check its self-consistency extensively with numerical simulations.

C Replica analysis

In this section we give a full derivation of the result in Sec. 2.1 in the main manuscript for the generalisation error of the problem defined in Sec. A. Our derivation follows from a Gibbs formulation of the optimisation problem in eq. (A.7) followed by a replica analysis inspired by the toolbox of the statistical physics of disordered systems.

C.1 Gibbs formulation of problem

Given the dataset $\{\boldsymbol{x}^\mu, y^\mu\}_{\mu=1}^n$ defined in Section A.1, we define the following Gibbs measure over $\mathbb{R}^p$:

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

for $\beta > 0$. When $\beta \to \infty$, the Gibbs measure peaks at the solution of the optimisation problem in eq. (A.7) - which, in the particular case of a strictly convex loss, is unique. Note that in the second equality we defined the factorised distributions P_y and P_w, showing that μ_β can be interpreted as a posterior distribution of $\boldsymbol{w}$ given the dataset $\{\boldsymbol{x}^\mu, y^\mu\}$, with P_y and P_w being the likelihood and prior distributions respectively.

An exact calculation of μ_β is intractable for large values of n, p and d. However, the interest in μ_β is that in the limit $n, p, d \to \infty$ with d/p and n/p fixed, the free energy density associated to the Gibbs measure:

$$f_\beta = -\lim_{p\to\infty} \frac{1}{p}\mathbb{E}_{\{\boldsymbol{x}^\mu, y^\mu\}} \log \mathcal{Z}_\beta \tag{C.2}$$

can be computed exactly using the replica method, and at $\beta \to \infty$ give us the optimal overlaps:

$$q_w = \frac{1}{p}\mathbb{E}||\hat{\boldsymbol{w}}||^2 \qquad q_x = \frac{1}{d}\mathbb{E}||\mathrm{F}\hat{\boldsymbol{w}}||^2 \qquad m_x = \frac{1}{d}\mathbb{E}\left[\boldsymbol{\theta}^0 \cdot \mathrm{F}\hat{\boldsymbol{w}}\right] \tag{C.3}$$

that - as we will see - fully characterise the generalisation error defined in eq. (A.9).

C.2 Replica computation of the free energy density

The replica calculation of f_β is based on a large deviation principle for the free energy density. Let

$$f_\beta(\{\boldsymbol{x}^\mu, y^\mu\}) = -\frac{1}{p}\log \mathcal{Z}_\beta \tag{C.4}$$

be the free energy density for one given sample of the problem, i.e. a fixed dataset $\{\boldsymbol{x}^\mu, y^\mu\}_{\mu=1}^n$. We assume that the distribution $P(f)$ of the free energy density, seen as a random variable over different samples of the problem, satisfies a large deviation principle, in the sense that, in the thermodynamic limit:

$$P(f) \simeq e^{p\Phi(f)}\,, \tag{C.5}$$

with Φ a concave function reaching its maximum at the free energy density $f = f_\beta$, with $\Phi(f_\beta) = 0$. This hypothesis includes the notion of *self-averageness* which states that the free-energy density is the same for almost all samples in the thermodynamic limit.

The value of f_β can be computed by computing the *replicated partition function*

$$\mathbb{E}_{\{\boldsymbol{x}^\mu, y^\mu\}} \mathcal{Z}_\beta^r = \int df \; e^{p[\Phi(f) - rf]} \,, \tag{C.6}$$

and taking the limit

$$f_\beta = \lim_{r \to 0^+} \frac{\mathrm{d}}{\mathrm{d}r} \lim_{p \to \infty} \left[-\frac{1}{p} \left(\mathbb{E}_{\{\boldsymbol{x}^\mu, y^\mu\}} \mathcal{Z}_\beta^r \right) \right] \tag{C.7}$$

Although this procedure is not fully rigorous, experience from the statistical physics of disordered systems shows that it gives exact results, and in fact the resulting expression can be verified to match the numerical simulations.

Using the replica method we need to evaluate:

$$\begin{aligned} \mathbb{E}_{\{\boldsymbol{x}^\mu, y^\mu\}} \mathcal{Z}_\beta^r = \int \mathrm{d}\boldsymbol{\theta}^0 \, P_\theta(\boldsymbol{\theta}^0) \int \prod_{a=1}^r \mathrm{d}\boldsymbol{w} \, P_w\left(\boldsymbol{w}^a\right) \times \\ \times \prod_{\mu=1}^n \int \mathrm{d}y^\mu \underbrace{\mathbb{E}_{\boldsymbol{c}^\mu} \left[P_y^0 \left(y^\mu \middle| \frac{\boldsymbol{c}^\mu \cdot \boldsymbol{\theta}^0}{\sqrt{d}} \right) \prod_{a=1}^r P_y \left(y^\mu | \boldsymbol{w}^a \cdot \sigma \left(\frac{1}{\sqrt{d}} \mathrm{F}^\top \boldsymbol{c}^\mu \right) \right) \right]}_{\text{(I)}} \end{aligned} \tag{C.8}$$

where P_w and P_y have been defined in (C.1). In order to compute this quantity, we introduce, for each point μ in the database, the $r+1$ variables

$$\nu_\mu = \frac{1}{\sqrt{d}} \boldsymbol{c}^\mu \cdot \boldsymbol{\theta}^0 \,, \tag{C.9}$$

$$\lambda_\mu^a = \boldsymbol{w}^a \cdot \sigma \left(\frac{1}{\sqrt{d}} \mathrm{F}^\top \boldsymbol{c}^\mu \right) \,. \tag{C.10}$$

Choosing $\boldsymbol{c}^\mu$ at random induces a joint distribution $P(\nu_\mu, \lambda_\mu^a)$. In the thermodynamic limit $p, d \to \infty$ with fixed p/n, and for matrices F satisfying the balance condition in eq. (B.3), the *replicated Gaussian equivalence* introduced in Section B.2 tells us that, for a given μ, the $r+1$ variables $\{\nu_\mu, \lambda_\mu^a\}_{a=1}^r$ are Gaussian random values with zero mean and covariance given by:

$$\Sigma^{ab} = \begin{pmatrix} \rho & M^a \\ M^a & Q^{ab} \end{pmatrix} \in \mathbb{R}^{(r+1)\times(r+1)} \tag{C.11}$$

The elements of the covariance matrix M^a and Q^{ab} are the rescaled version of the so-called *overlap parameters*:

$$\rho = \frac{1}{d} ||\boldsymbol{\theta}^0||^2, \qquad m_s^a = \frac{1}{d} \boldsymbol{s}^a \cdot \boldsymbol{\theta}^0, \qquad q_s^{ab} = \frac{1}{d} \boldsymbol{s}^a \cdot \boldsymbol{s}^b, \qquad q_w^{ab} = \frac{1}{p} \boldsymbol{w}^a \cdot \boldsymbol{w}^b, \tag{C.12}$$

where $\boldsymbol{s}^a = \frac{1}{\sqrt{p}} \mathrm{F} \boldsymbol{w}^a$. They are thus given by:

$$M^a = \kappa_1 m_s^a, \qquad Q^{ab} = \kappa_\star^2 q_w^{ab} + \kappa_1^2 q_s^{ab}. \tag{C.13}$$

where $\kappa_1 = \mathbb{E}_z \left[z \sigma(z) \right]$ and $\kappa_\star = \mathbb{E}_z \left[\sigma(z)^2 \right] - \kappa_1^2$ as in eq. (B.6). With this notation, the asymptotic joint probability is simply written as:

$$P(\nu_\mu, \{\lambda_\mu^a\}_{a=1}^r) = \frac{1}{\sqrt{\det\left(2\pi\Sigma\right)}} e^{-\frac{1}{2} \sum_{a,b=0}^r z_\mu^a \left(\Sigma^{-1}\right)^{ab} z_\mu^b} \tag{C.14}$$

with $z_\mu^0 = \nu_\mu$ and $z_\mu^a = \lambda_\mu^a$ for $a = 1, \cdots, r$. The average over the replicated partition function (C.8) therefore reads:

$$\mathbb{E}_{\{\boldsymbol{x}^\mu, y^\mu\}} \mathcal{Z}_\beta^r = \int \mathrm{d}\boldsymbol{\theta}^0 \; P_\theta(\boldsymbol{\theta}^0) \int \prod_{a=1}^{r} \mathrm{d}\boldsymbol{w} \; P_w\left(\boldsymbol{w}^a\right) \prod_{\mu=1}^{n} \int \mathrm{d}y^\mu \times$$
$$\times \int \mathrm{d}\nu_\mu \; P_y^0\left(y^\mu | \nu_\mu\right) \int \prod_{a=1}^{r} \mathrm{d}\lambda_\mu^a \; P(\nu_\mu, \{\lambda_\mu^a\}) \prod_{a=1}^{r} P_y\left(y^\mu | \{\lambda_\mu^a\}\right). \tag{C.15}$$

Rewriting as a saddle-point problem

Note that after taking the average over $\boldsymbol{x}$, the integrals involved in the replicated partition function only couple through the overlap parameters. It is therefore useful to introduce the following Dirac δ-functions to unconstrain them, introducing the decomposition:

$$\begin{aligned}
1 &= d^{-(r+1)^2} \int \mathrm{d}\rho \; \delta\left(d\rho - ||\boldsymbol{\theta}^0||^2\right) \int \prod_{a=1}^{r} \mathrm{d}m_s^a \; \delta\left(dm_s^a - \boldsymbol{s}^a \cdot \boldsymbol{\theta^0}\right) \times \\
&\times \int \prod_{1 \le a \le b \le r} \mathrm{d}q_s^{ab} \delta\left(dq_s^{ab} - \boldsymbol{s}^a \cdot \boldsymbol{s}^b\right) \int \prod_{1 \le a \le b \le r} \mathrm{d}q_w^{ab} \; \delta\left(pq_w^{ab} - \boldsymbol{w}^a \cdot \boldsymbol{w}^b\right) \\
&= d^{-(r+1)^2} \int \frac{\mathrm{d}\rho \mathrm{d}\hat{\rho}}{2\pi} \; e^{-i\hat{\rho}\left(d\rho - ||\boldsymbol{\theta}^0||^2\right)} \int \prod_{a=1}^{r} \frac{\mathrm{d}m_s^a \mathrm{d}\hat{m}_s^a}{2\pi} \; e^{-i \sum\limits_{a=1}^{r} \hat{m}_s^a \left(dm_s^a - \boldsymbol{s}^a \cdot \boldsymbol{\theta^0}\right)} \times \\
&\times \int \prod_{1 \le a \le b \le r} \frac{\mathrm{d}q_s^{ab} \mathrm{d}\hat{q}_s^{ab}}{2\pi} e^{-i \sum\limits_{1 \le a \le b \le r} \hat{q}_s^{ab} \left(dq_s^{ab} - \boldsymbol{s}^a \cdot \boldsymbol{s}^b\right)} \int \prod_{1 \le a \le b \le r} \frac{\mathrm{d}q_w^{ab} \hat{q}_w^{ab}}{2\pi} \; e^{-i \sum\limits_{1 \le a \le b \le r} \hat{q}_w^{ab} \left(pq_w^{ab} - \boldsymbol{w}^a \cdot \boldsymbol{w}^b\right)}.
\end{aligned} \tag{C.16}$$

Introducing the above in eq. (C.15) and exchanging the integration order allows to factorise the integrals over the d, p, n dimensions and rewrite:

$$\mathbb{E}_{\{\boldsymbol{x}^\mu, y^\mu\}} \mathcal{Z}_\beta^r = \int \frac{\mathrm{d}\rho \mathrm{d}\hat{\rho}}{2\pi} \int \prod_{a=1}^{r} \frac{\mathrm{d}m_s^a \mathrm{d}\hat{m}_s^a}{2\pi} \int \prod_{1 \le a \le b \le r} \frac{\mathrm{d}q_s^{ab} \mathrm{d}\hat{q}_s^{ab}}{2\pi} \frac{\mathrm{d}q_w^{ab} \mathrm{d}\hat{q}_w^{ab}}{2\pi} e^{p\Phi^{(r)}} \tag{C.17}$$

where the integrals over the variables m_s^a, q_s^{ab} and q_w^{ab} run over $\mathbb{R}$, while those over $\hat{m}_s^a$, $\hat{q}_s^{ab}$ and $\hat{q}_w^{ab}$ run over $i\mathbb{R}$. The function $\Phi^{(r)}$, a function of all the overlap parameters, is given by:

$$\begin{aligned}
\Phi^{(r)} = &-\gamma\rho\hat{\rho} - \gamma \sum_{a=1}^{r} m_s^a \hat{m}_s^a - \sum_{1 \le a \le b \le r} \left(\gamma q_s^{ab} \hat{q}_s^{ab} + q_w \hat{q}_w\right) + \alpha \Psi_y^{(r)}\left(\rho, m_s^a, q_s^{ab}, q_w^{ab}\right) \\
&+ \Psi_w^{(r)}\left(\hat{\rho}, \hat{m}_s^a, \hat{q}_s^{ab}, \hat{q}_w^{ab}\right)
\end{aligned} \tag{C.18}$$

where we recall that $\alpha = n/p$, $\gamma = d/p$, and we have introduced:

$$\begin{aligned}
\Psi_y^{(r)} &= \log \int \mathrm{d}y \int \mathrm{d}\nu \; P_y^0\left(y|\nu\right) \int \prod_{a=1}^{r} \left[\mathrm{d}\lambda^a P_y\left(y|\lambda^a\right)\right] P(\nu, \{\lambda^a\}) \\
\Psi_w^{(r)} &= \frac{1}{p} \log \int \mathrm{d}\boldsymbol{\theta}^0 P_\theta(\boldsymbol{\theta}^0) e^{-\hat{\rho}||\boldsymbol{\theta}^0||^2} \int \prod_{a=1}^{r} \mathrm{d}\boldsymbol{w}^a \; P_w(\boldsymbol{w}^a) e^{1 \le a \le b \le r \left[\hat{q}_w^{ab} \boldsymbol{w}^a \cdot \boldsymbol{w}^b + \hat{q}_s^{ab} \boldsymbol{s}^a \cdot \boldsymbol{s}^b\right] - \sum\limits_{a=1}^{r} \hat{m}_s^a \boldsymbol{s}^a \cdot \boldsymbol{\theta}^0}
\end{aligned} \tag{C.19}$$

Note that $\boldsymbol{s}^a = \frac{1}{\sqrt{p}} \mathrm{F} \boldsymbol{w}^a$ is a function of $\boldsymbol{w}^a$, and must be kept under the $\boldsymbol{w}^a$ integral. In the thermodynamic limit where $p \to \infty$ with n/p and d/p fixed, the integral in eq. (C.17) concentrates around the values of the overlap parameters that extremize $\Phi^{(r)}$, and therefore

$$f = -\lim_{r \to 0^+} \frac{1}{r} \underset{\substack{\{\rho, \hat{\rho}, m_s^a, \hat{m}_s^a\} \\ \{q_s^{ab}, \hat{q}_s^{ab}, q_w^{ab}, \hat{q}_w^{ab}\}}}{\mathbf{extr}} \Phi^{(r)}. \tag{C.20}$$

Replica symmetric Ansatz

In order to proceed with the $r \to 0^+$ limit, we restrict the extremization above to the following replica symmetric Ansatz:

$$\begin{aligned} m_s^a &= m_s & \hat{m}^a &= \hat{m}_s & &\text{for } a = 1, \cdots, r \\ q_{s/w}^{aa} &= r_{s/w} & \hat{q}_{s/w}^{aa} &= -\frac{1}{2}\hat{r}_{s/w} & &\text{for } a = 1, \cdots, r \\ q_{s/w}^{ab} &= q_{s/w} & \hat{q}_{s/w}^{ab} &= \hat{q}_{s/w} & &\text{for } 1 \leq a < b \leq r \end{aligned} \tag{C.21}$$

Note that, in the particular case of a convex loss function with $\lambda > 0$, the replica symmetric Ansatz is justified: the problem only admitting one solution, it *a fortiori* coincides with the replica symmetric one. For non-convex losses, solutions that are not replica symmetric (also known as *replica symmetry breaking*) are possible, and the energy landscape of the free energy needs to be carefully analysed. In the practical applications explored in this manuscript, we focus on convex losses with ridge regularisation, and therefore the replica symmetric assumption is fully justified.

Before proceeding with the limit in eq. (C.20), we need to verify that the above Ansatz is well defined - in other words, that we have not introduced a spurious order one term in Φ that would diverge. This means we need to check that $\lim_{r\to 0^+} \Phi = 0$.

First, with a bit of algebra one can check that, within our replica symmetric Ansatz:

$$\lim_{r\to 0^+} \Psi_y^{(r)} = 0. \tag{C.22}$$

Therefore,

$$\lim_{r\to 0^+} \Phi^{(r)} = -\gamma\rho\hat{\rho} + \gamma \log \int_{\mathbb{R}} \mathrm{d}\theta^0 \; P_\theta\left(\theta^0\right) e^{\hat{\rho}\theta^{0\,2}} \tag{C.23}$$

where we have used the fact that P_θ is a factorised distribution to take the $p \to \infty$ limit. In order for this limit to be 0, we need that $\hat{\rho} = 0$, which also fixes ρ to be a constant given by the second moment of θ^0:

$$\rho = \mathbb{E}_{\theta^0}\left[\theta^{0\,2}\right] \tag{C.24}$$

We now proceed with the limit in eq. (C.20). Let's look first at Ψ_y. The non-trivial limit comes from the fact that $\det \Sigma$ and Σ^{-1} are non-trivial functions of r. It is not hard to see, however, that Σ^{-1} itself has replica symmetric structure, with components given by:

$$\begin{aligned} \left(\Sigma^{-1}\right)^{00} &= \tilde{\rho} = \frac{R + (r-1)Q}{\rho(R + (r-1)Q) - rM^2}, & \left(\Sigma^{-1}\right)^{aa} &= \tilde{R} = \frac{\rho R + (r-2)\rho\, Q - (r-1)M^2}{(R-Q)(\rho\, R + (r-1)\rho\, Q - r\, M^2)} \\ \left(\Sigma^{-1}\right)^{a0} &= \tilde{M} = \frac{M}{r\, M^2 - \rho\, R - (r-1)\rho\, Q}, & \left(\Sigma^{-1}\right)^{ab} &= \tilde{Q} = \frac{M^2 - \rho\, Q}{(R-Q)(\rho\, R + (r-1)\rho\, Q - r\, M^2)} \end{aligned} \tag{C.25}$$

where M, Q and R are the rescaled overlap parameters in the replica symmetric Ansatz, that is:

$$M = \kappa_1 m_s, \qquad Q = \kappa_\star^2 q_w + \kappa_1^2 q_s, \qquad R = \kappa_\star^2 r_w + \kappa_1^2 r_s. \tag{C.26}$$

This allows us to write:

$$\begin{aligned} \Psi_y^{(r)} = \log \int \mathrm{d}y \int \mathrm{d}\nu \; P_y^0\left(y|\nu\right) e^{-\frac{\tilde{\rho}}{2}\nu^2} \int \prod_{a=1}^{r} \mathrm{d}\lambda^a P_y\left(y|\lambda^a\right) e^{-\frac{\tilde{Q}}{2}\sum\limits_{a,b=1}^{n}\lambda^a\lambda^b - \frac{\tilde{R}-\tilde{Q}}{2}\sum\limits_{a=1}^{r}(\lambda^a)^2 - \tilde{M}\nu\sum\limits_{a=1}^{n}\lambda^a} \\ - \frac{1}{2}\log\det\left(2\pi\Sigma\right). \end{aligned} \tag{C.27}$$

In order to completely factor the integral above in the replica space, we use the *Hubbard-Stratonovich transformation*:

$$e^{-\frac{\tilde{Q}}{2}\sum\limits_{a,b=1}^{r}\lambda^a\lambda^b} = \mathbb{E}_\xi e^{\sqrt{-\tilde{Q}}\xi\sum\limits_{a=1}^{r}\lambda^a} \tag{C.28}$$

for $\xi \sim \mathcal{N}(0,1)$, such that

$$\begin{aligned}\Psi_y^{(r)} = \log \int \mathrm{d}y \int \mathrm{d}\nu\; P_y^0\left(y|\nu\right) e^{-\frac{\tilde{\rho}}{2}\nu^2} \left[\int \mathrm{d}\lambda P_y\left(y|\lambda\right) e^{-\frac{\tilde{R}-\tilde{Q}}{2}\lambda^2 + \left(\sqrt{-\tilde{Q}}\xi - \tilde{M}\nu\right)\lambda}\right]^r \\ - \frac{1}{2}\log\det\left(2\pi\Sigma\right).\end{aligned} \tag{C.29}$$

Taking into account the r dependence of the inverse elements and of the determinant, we can take the limit to get:

$$\begin{aligned}\lim_{r\to 0^+}\frac{1}{r}\Psi_y^{(r)} = \mathbb{E}_\xi \int_{\mathbb{R}} \mathrm{d}y \int \frac{\mathrm{d}\nu}{\sqrt{2\pi\rho}} P_y^0\left(y|\nu\right) e^{-\frac{1}{2\rho}\nu^2} \log \int \frac{\mathrm{d}\lambda}{\sqrt{2\pi}}\; P_y\left(y|\lambda\right) e^{-\frac{1}{2}\frac{\lambda^2}{R-Q} + \left(\frac{Q-M^2/\rho}{(R-Q)^2}\xi + \frac{M/\rho}{R-Q}\nu\right)\lambda} \\ - \frac{1}{2}\log\left(R-Q\right) - \frac{Q}{R-Q}\end{aligned} \tag{C.30}$$

Finally, making a change of variables and defining:

$$\mathcal{Z}_y^{\cdot/0}(y;\omega,V) = \int \frac{\mathrm{d}x}{\sqrt{2\pi V}} e^{-\frac{1}{2V}(x-\omega)^2} P_y^{\cdot/0}\left(y|x\right) \tag{C.31}$$

allows us to rewrite the limit of Ψ_y - which abusing notation we still denote Ψ_y - as:

$$\Psi_y = \mathbb{E}_\xi\left[\int_{\mathbb{R}} \mathrm{d}y\; \mathcal{Z}_y^0\left(y; \frac{M}{\sqrt{Q}}\xi, \rho\; - \frac{M^2}{Q}\right) \log \mathcal{Z}_y\left(y; \sqrt{Q}\xi, R-Q\right)\right]. \tag{C.32}$$

One can follow a very similar approach for the limit of Ψ_w, although in this case the limit is much simpler, since there is no r dependence on the hat variables. The limit can be written as:

$$\Psi_w = \lim_{p\to\infty}\frac{1}{p}\mathbb{E}_{\xi,\eta,\theta^0}\log\int_{\mathbb{R}^d}\mathrm{d}\boldsymbol{s}\; P_s(\boldsymbol{s};\eta) e^{-\frac{\hat{V}_s}{2}||\boldsymbol{s}||^2 + \left(\sqrt{\hat{q}_s}\xi\mathbf{1}_d + \hat{m}_s\boldsymbol{\theta}^0\right)^\top\boldsymbol{s}} \tag{C.33}$$

for $\xi, \eta \sim \mathcal{N}(0,1)$, and we have defined:

$$P_s(\boldsymbol{s};\eta) = \int_{\mathbb{R}^p}\mathrm{d}\boldsymbol{w}\; P_w(\boldsymbol{w}) e^{-\frac{\hat{r}_w+\hat{q}_w}{2}||\boldsymbol{w}||^2 + \sqrt{\hat{q}_w}\mathbf{1}_p^\top\boldsymbol{w}}\delta\left(\boldsymbol{s} - \frac{1}{\sqrt{p}}\mathrm{F}\boldsymbol{w}\right) \tag{C.34}$$

Summary of the replica symmetric free energy density

Summarising the calculation above, the replica symmetric free energy density reads:

$$\begin{aligned}f = \mathbf{extr}\Big\{ - \frac{\gamma}{2}r_s\hat{r}_s - \frac{\gamma}{2}q_s\hat{q}_s + \gamma m_s\hat{m}_s - \frac{1}{2}r_w\hat{r}_w - \frac{1}{2}q_w\hat{q}_w \\ - \alpha\Psi_y(R,Q,M) - \Psi_w\left(\hat{r}_s,\hat{q}_s,\hat{m}_s,\hat{r}_w,\hat{q}_w\right)\Big\}\end{aligned} \tag{C.35}$$

with $\alpha = \frac{n}{p}$, $\gamma = \frac{d}{p}$, and:

$$Q = \kappa_1^2 q_s + \kappa_\star^2 q_w, \qquad R = \kappa_1^2 r_s + \kappa_\star^2 r_w \qquad M = \kappa_1 m_s. \tag{C.36}$$

The so-called potentials (Ψ_y, Ψ_w) are given by:

$$\Psi_w = \lim_{p\to\infty} \frac{1}{p}\mathbb{E}_{\xi,\eta,\theta^0} \log \int_{\mathbb{R}^d} \mathrm{d}\boldsymbol{s} P_s(\boldsymbol{s};\eta) e^{-\frac{\hat{V}_s}{2}||\boldsymbol{s}||^2 + \left(\sqrt{\hat{q}_s}\xi \mathbf{1}_d + \hat{m}_s\boldsymbol{\theta}^0\right)^\top \boldsymbol{s}} \tag{C.37}$$

$$\Psi_y = \mathbb{E}_\xi \left[\int_{\mathbb{R}} \mathrm{d}y \; \mathcal{Z}_y^0\left(y; \frac{M}{\sqrt{Q}}\xi, \rho - \frac{M^2}{Q}\right) \log \mathcal{Z}_y\left(y; \sqrt{Q}\xi, R-Q\right)\right]. \tag{C.38}$$

where:

$$P_s(\boldsymbol{s};\eta) = \int_{\mathbb{R}^p} \mathrm{d}\boldsymbol{w} \; P_w(\boldsymbol{w}) e^{-\frac{\hat{r}_w+\hat{q}_w}{2}||\boldsymbol{w}||^2 + \sqrt{\hat{q}_w}\eta \mathbf{1}_p^\top \boldsymbol{w}} \delta\left(\boldsymbol{s} - \frac{1}{\sqrt{p}}\mathrm{F}\boldsymbol{w}\right)$$

$$\mathcal{Z}_y^{\prime/0}(y;\omega,V) = \int \frac{\mathrm{d}x}{\sqrt{2\pi V}} e^{-\frac{1}{2V}(x-\omega)^2} P_y^{\prime/0}(y|x) \tag{C.39}$$

C.3 Evaluating Ψ_w for ridge regularisation and Gaussian prior

Note that as long as the limit in Ψ_w is well defined, the eq. (C.35) holds for any P_θ and P_w. However, as discussed in Sec. A.1, we are interested in $\boldsymbol{\theta}^0 \sim \mathcal{N}(0, \mathrm{I}_d)$ and ridge regularisation so that $P_w = \exp\left(-\frac{\beta\lambda}{2}||\boldsymbol{w}||^2\right)$. In this case, we simply have:

$$P(\boldsymbol{s};\eta) = \frac{e^{\frac{p}{2}\frac{\eta^2\hat{q}_w}{\beta\lambda+\hat{V}_w}}}{(\beta\lambda+\hat{V}_w)^{p/2}} \mathcal{N}(\boldsymbol{s};\boldsymbol{\mu},\Sigma) \tag{C.40}$$

with:

$$\boldsymbol{\mu} = \frac{\sqrt{\hat{q}_w}\eta}{\beta\lambda+\hat{V}_w}\frac{\mathrm{F}\mathbf{1}_p}{\sqrt{p}} \in \mathbb{R}^d, \qquad \Sigma = \frac{1}{\beta\lambda+\hat{V}_w}\frac{\mathrm{F}\mathrm{F}^\top}{p} \in \mathbb{R}^{d\times d} \tag{C.41}$$

and we have defined the shorthand $\hat{V}_w = \hat{r}_w + \hat{q}_w$. Therefore the argument of the logarithm in Ψ_w is just another Gaussian integral we can do explicitly:

$$\mathbb{E}_s e^{-\frac{\hat{V}_s}{2}||\boldsymbol{s}||^2 + \boldsymbol{b}^\top\boldsymbol{s}} = \frac{e^{\frac{p}{2}\frac{\eta^2\hat{q}_w}{\beta\lambda+\hat{V}_w}}}{\left(\beta\lambda+\hat{V}_w\right)^{p/2}} \frac{e^{-\frac{1}{2}\boldsymbol{\mu}^\top\Sigma^{-1}\boldsymbol{\mu} + \frac{1}{2\hat{V}_s}||\boldsymbol{b}+\Sigma^{-1}||^2}}{\sqrt{\det\left(\mathrm{I}_d + \hat{V}_s\Sigma\right)}} e^{-\frac{1}{2\hat{V}_s}\left(b+\Sigma^{-1}\mu\right)^\top\left(\mathrm{I}_d+\hat{V}_s\Sigma\right)^{-1}\left(b+\Sigma^{-1}\mu\right)} \tag{C.42}$$

where we have defined the shorthands $\hat{V}_s = \hat{r}_s + \hat{q}_s$ and $\boldsymbol{b} = \left(\sqrt{\hat{q}_s}\xi\mathbf{1}_d + \hat{m}_s\boldsymbol{\theta}^0\right) \in \mathbb{R}^d$. Inserting back in eq. (C.37) and taking the log,

$$\Psi_w = \lim_{p\to\infty} \mathbb{E}_{\theta^0,\xi,\eta}\left[\frac{1}{2}\frac{\eta^2\hat{q}_w}{\beta\lambda+\hat{V}_w} - \frac{1}{2}\log\left(\beta\lambda+\hat{V}_w\right) - \frac{1}{2p}\operatorname{tr}\log\left(\mathrm{I}_d+\hat{V}_s\Sigma\right) - \frac{1}{2p}\boldsymbol{\mu}^\top\Sigma^{-1}\boldsymbol{\mu}\right.$$
$$\left. + \frac{1}{2p\hat{V}_s}||\boldsymbol{b}+\Sigma^{-1}||^2 - \frac{1}{2p\hat{V}_s}\left(b+\Sigma^{-1}\mu\right)^\top\left(\mathrm{I}_d+\hat{V}_s\Sigma\right)^{-1}\left(b+\Sigma^{-1}\mu\right)\right] \tag{C.43}$$

The averages over $\eta, \xi, \boldsymbol{\theta}^0$ simplify this expression considerably:

$$\mathbb{E}_\eta\left[\boldsymbol{\mu}^\top\Sigma\boldsymbol{\mu}\right] = \frac{1}{p}\frac{\hat{q}_w}{\left(\beta\lambda+\hat{V}_w\right)^2}\left(\mathrm{F}\mathbf{1}_p\right)^\top\Sigma^{-1}\left(\mathrm{F}\mathbf{1}_p\right) = d\frac{\hat{q}_w}{\beta\lambda+\hat{V}_w}$$
$$\mathbb{E}_{\eta,\xi,\theta^0}||b+\Sigma^{-1}\boldsymbol{\mu}||^2 = d(\hat{m}_s^2+\hat{q}_s) + \frac{1}{p}\hat{q}_w\operatorname{tr}\left(\mathrm{F}\mathrm{F}\right)^{-1}$$

$$\mathbb{E}_{\eta,\xi,\theta^0}\left(b+\Sigma^{-1}\boldsymbol{\mu}\right)^\top\left(\mathrm{I}_d+\hat{V}_s\Sigma\right)^{-1}\left(b+\Sigma^{-1}\boldsymbol{\mu}\right) = \frac{1}{p}\hat{q}_w\operatorname{tr}\left[\mathrm{F}\mathrm{F}^\top\left(\mathrm{I}_d+\hat{V}_s\Sigma\right)^{-1}\right]$$
$$+ (\hat{m}_s^2+\hat{q}_s)\operatorname{tr}\left(\mathrm{I}_d+\hat{V}_s\Sigma\right)^{-1} \tag{C.44}$$

Finally, we can combine the two terms:

$$\operatorname{tr}\frac{\mathrm{F}\mathrm{F}^\top}{p}+\operatorname{tr}\left[\frac{\mathrm{F}\mathrm{F}^\top}{p}\left(\mathrm{I}_d+\hat{V}_s\Sigma\right)^{-1}\right]=\frac{\hat{V}_s}{\beta\lambda+\hat{V}_w}\operatorname{tr}\left(\mathrm{I}_d+\hat{V}_s\Sigma\right)^{-1}, \tag{C.45}$$

and write:

$$\begin{aligned}\Psi_w=&-\frac{1}{2}\log\left(\beta\lambda+\hat{V}_w\right)-\frac{1}{2}\lim_{p\to\infty}\frac{1}{p}\operatorname{tr}\log\left(\mathrm{I}_d+\frac{\hat{V}_s}{\beta\lambda+\hat{V}_w}\frac{\mathrm{F}\mathrm{F}^\top}{p}\right)\\&+\frac{\hat{m}_s^2+\hat{q}_s}{2\hat{V}_s}\left[\gamma-\lim_{p\to\infty}\frac{1}{p}\operatorname{tr}\left(\mathrm{I}_d+\frac{\hat{V}_s}{\beta\lambda+\hat{V}_w}\frac{\mathrm{F}\mathrm{F}^\top}{p}\right)^{-1}\right]\\&+\frac{1}{2}\frac{\hat{q}_w}{\beta\lambda+\hat{V}_w}\left[1-\gamma+\lim_{p\to\infty}\frac{1}{p}\operatorname{tr}\left(\mathrm{I}_d+\frac{\hat{V}_s}{\beta\lambda+\hat{V}_w}\frac{\mathrm{F}\mathrm{F}^\top}{p}\right)^{-1}\right]\end{aligned} \tag{C.46}$$

Note that Ψ only depends on the spectral properties of the matrix $\frac{1}{p}\mathrm{F}\mathrm{F}^\top\in\mathbb{R}^{p\times p}$, and more specifically on its resolvent in the asymptotic limit. A case of particular interest is when $\mathrm{F}\mathrm{F}^\top$ has a well defined spectral measure μ on the $p,d\to\infty$ limit with $\gamma=d/p$ fixed. In that case, we can write:

$$\lim_{p\to\infty}\frac{1}{p}\operatorname{tr}\left(\mathrm{I}_d+\frac{\hat{V}_s}{\beta\lambda+\hat{V}_w}\frac{\mathrm{F}\mathrm{F}^\top}{p}\right)^{-1}=\gamma\frac{\beta\lambda+\hat{V}_w}{\hat{V}_s}g_\mu\left(-\frac{\beta\lambda+\hat{V}_w}{\hat{V}_s}\right) \tag{C.47}$$

$$\tag{C.48}$$

where g_μ is the Stieltjes transform of μ, defined by:

$$g_\mu(z)=\int\frac{\mathrm{d}\mu(t)}{t-z}. \tag{C.49}$$

Similarly, the logarithm term can be expressed as the logarithm potential of μ - although for the purpose of evaluating the generalisation error we will only need the derivative of these terms, and therefore only the Stieltjes transforms and its derivative.

In what follows, we will mostly focus on two kinds of projection matrices F:

Gaussian projections: For $\mathrm{F}\in\mathbb{R}^{d\times p}$ a random matrix with i.i.d. Gaussian entries with zero mean and variance 1, μ is given by the well-known Marchenko-Pastur law, and the corresponding Stieltjes transform is given by:

$$g_\mu(z)=\frac{1-z-\gamma-\sqrt{(z-1-\gamma)^2-4\gamma}}{2z\gamma},\qquad z<0 \tag{C.50}$$

Orthogonally invariant projection: For $\mathrm{F}=\mathrm{U}^\top\mathrm{D}\mathrm{V}$ with $\mathrm{U}\in\mathbb{R}^{d\times d}$ and $\mathrm{V}\in\mathbb{R}^{p\times p}$ two orthogonal matrices and $\mathrm{D}\in\mathbb{R}^{d\times p}$ a rectangular diagonal matrix of rank $\min(d,p)$ and diagonal entries d_k, the empirical spectral density μ_p is given by:

$$\mu_d(\lambda)=\frac{1}{d}\sum_{k=1}^{\min(r,p)}\delta(\lambda-\lambda_k)=\left(1-\min\left(1,\frac{1}{\gamma}\right)\right)\delta(\lambda)+\frac{1}{p}\sum_{k=1}^{\min(d,p)}\delta(\lambda-d_k^2) \tag{C.51}$$

Therefore the choice of diagonal elements d_k fully characterise the spectrum of $\mathrm{F}\mathrm{F}^\top$. In order for the orthogonally invariant case to be comparable to the Gaussian case, we fix d_k in such a way that the projected vector $\mathrm{F}\boldsymbol{w}$ is of the same order in both cases, i.e.

$$d_k^2=\begin{cases}\gamma & \text{for } \gamma>1\\ 1 & \text{for } \gamma\leq 1\end{cases} \tag{C.52}$$

With this choice, the Stieltjes transform of μ reads:

$$g_\mu(z) = \begin{cases} -(1-\frac{1}{\gamma})\frac{1}{z} + \frac{1}{\gamma}\frac{1}{\gamma - z} & \text{for } \gamma > 1 \\ \frac{1}{1-z} & \text{for } \gamma \leq 1 \end{cases} \tag{C.53}$$

C.4 Gaussian equivalent model

It is interesting to note that the average over the dataset $\{\boldsymbol{x}^\mu, y^\mu\}_{\mu=1}^n$ of the replicated partition function $\mathcal{Z}_\beta^r$ in eq. (C.15), obtained after the application of the GET, is identical to the replicated partition function of the same task over the following dual dataset $\{\tilde{\boldsymbol{x}}^\mu, y^\mu\}_{\mu=1}^n$, where:

$$\tilde{\boldsymbol{x}}^\mu = \kappa_0 \mathbf{1}_p + \kappa_1 \frac{1}{\sqrt{d}} \mathrm{F}^\top \boldsymbol{c}^\mu + \kappa_\star \boldsymbol{z}^\mu \tag{C.54}$$

where $\boldsymbol{z}^\mu \sim \mathcal{N}(\mathbf{0}, \mathrm{I}_p)$, and the labels $y^\mu \sim P_y$ are the same. Indeed, calling $\tilde{\mathcal{Z}}_\beta^r$ the replicated partition function for this equivalent dataset, and considering κ_0 we have:

$$\mathbb{E}_{\{\tilde{\boldsymbol{x}}^\mu, y^\mu\}} \tilde{\mathcal{Z}}_\beta^r = \int \mathrm{d}\boldsymbol{\theta}^0 \, P_\theta(\boldsymbol{\theta}^0) \int \prod_{a=1}^r \mathrm{d}\boldsymbol{w} \, P_w\left(\boldsymbol{w}^a\right) \times \\ \times \prod_{\mu=1}^n \int \mathrm{d}y^\mu \underbrace{\mathbb{E}_{\boldsymbol{c}^\mu, \boldsymbol{z}^\mu} \left[P_y^0 \left(y^\mu \middle| \frac{\boldsymbol{c}^\mu \cdot \boldsymbol{\theta}^0}{\sqrt{d}} \right) \prod_{a=1}^r P_y \left(y^\mu | \boldsymbol{w}^a \cdot \left(\frac{\kappa_1}{\sqrt{d}} \mathrm{F}^\top \boldsymbol{c}^\mu + \kappa_\star \boldsymbol{z}^\mu \right) \right) \right]}_{\text{(I)}} . \tag{C.55}$$

Rewriting (I):

$$\text{(I)} = \int \mathrm{d}\nu_\mu \, P_y^0\left(y^\mu | \nu_\mu\right) \int \prod_{a=1}^r \mathrm{d}\lambda_\mu^a \, P_y\left(y^\mu | \lambda_\mu^a\right) \times \\ \times \underbrace{\mathbb{E}_{\boldsymbol{c}^\mu, \boldsymbol{z}^\mu} \left[\delta\left(\nu_\mu - \frac{1}{\sqrt{d}} \boldsymbol{c}^\mu \cdot \boldsymbol{\theta}^0 \right) \prod_{a=1}^r \delta\left(\lambda_\mu^a - \frac{\kappa_1}{\sqrt{d}} \boldsymbol{w}^a \cdot \mathrm{F}^\top \boldsymbol{c}^\mu + \kappa_\star \boldsymbol{w}^a \cdot \boldsymbol{z}^\mu \right) \right]}_{\equiv P(\nu, \lambda)} . \tag{C.56}$$

It is easy to show that taking (κ_0, κ_1) to match those from eq. (B.6), the variables $\left(\nu_\mu, \{\lambda_\mu^a\}\right)$ are jointly Gaussian variables with correlation matrix given by Σ exactly as in eq. (C.11). This establishes the equivalence

$$\tilde{\mathcal{Z}}_\beta^r = \mathcal{Z}_\beta^r \tag{C.57}$$

from which follows the equivalence between the asymptotic generalisation and test error of these two models.

D Saddle-point equations and the generalisation error

The upshot of the replica analysis is to exchange the p-dimensional minimisation problem for $\boldsymbol{w} \in \mathbb{R}^p$ in eq. (A.7) for a one-dimensional minimisation problem for the parameters $\{r_s, q_s, m_s, r_w, q_w\}$ and their conjugate in eq. (C.35). In particular, note that by construction at the limit $\beta \to \infty$ the solution $\{q_s^\star, m_s^\star, q_w^\star\}$ of eq. (C.35) corresponds to:

$$q_w^\star = \frac{1}{p} ||\hat{\boldsymbol{w}}||^2 \qquad q_s^\star = \frac{1}{d} ||\mathrm{F}\hat{\boldsymbol{w}}||^2 \qquad m_s^\star = \frac{1}{d} \left(\mathrm{F}\hat{\boldsymbol{w}}\right) \cdot \boldsymbol{\theta}^0 \tag{D.1}$$

where $\hat{\boldsymbol{w}}$ is the solution of the solution of eq. (A.7). As we will see, both the generalisation error defined in eq. (A.9) and the training loss can be expressed entirely in terms of these overlap parameters.

D.1 Generalisation error as a function of the overlaps

Let $\{\boldsymbol{x}^{\text{new}}, y^{\text{new}}\}$ be a new sample independently drawn from the same distribution of our data $\{\boldsymbol{x}^{\mu}, y^{\mu}\}_{\mu=1}^{n}$. The generalisation error can then be written as:

$$\begin{aligned}\epsilon_g &= \frac{1}{4^k}\mathbb{E}_{\boldsymbol{x}^{\text{new}},y^{\text{new}}}\left(y^{\text{new}} - \hat{f}\left(\sigma\left(\mathrm{F}^{\top}\boldsymbol{c}^{\text{new}}\right)\cdot\hat{\boldsymbol{w}}\right)\right)^2 \\ &= \frac{1}{4^k}\int \mathrm{d}y \int \mathrm{d}\nu\, P_y^0(y|\nu)\int \mathrm{d}\lambda\,(y-\hat{f}(\lambda))^2\mathbb{E}_{\boldsymbol{c}^{\text{new}}}\left[\delta\left(\nu - \boldsymbol{c}^{\text{new}}\cdot\boldsymbol{\theta}^0\right)\delta\left(\lambda - \sigma\left(\mathrm{F}^{\top}\boldsymbol{c}^{\text{new}}\right)\cdot\hat{\boldsymbol{w}}\right)\right]. \end{aligned} \tag{D.2}$$

where for convenience, we normalise $k = 0$ for the regression task and $k = 1$ for the classification task. Again, we apply the GET from Sec. B to write the joint distribution over $\{\nu, \lambda\}$:

$$P(\nu,\lambda) = \frac{1}{\sqrt{\det\left(2\pi\Sigma\right)}}e^{-\frac{1}{2}\boldsymbol{z}^{\top}\Sigma^{-1}\boldsymbol{z}}, \tag{D.3}$$

where $\boldsymbol{z} = (\nu,\lambda)^{\top} \in \mathbb{R}^2$ and Σ is given by

$$\Sigma = \begin{pmatrix}\rho & M^{\star}\\ M^{\star} & Q^{\star}\end{pmatrix}, \qquad \rho = \frac{1}{d}||\boldsymbol{\theta^0}||^2 \qquad M^{\star} = \frac{\mu_1}{d}\left(\mathrm{F}\hat{\boldsymbol{w}}\right)\cdot\boldsymbol{\theta}^0, \qquad Q^{\star} = \frac{\mu_1^2}{d}||\mathrm{F}\hat{\boldsymbol{w}}||^2 + \frac{\mu_{\star}^2}{p}||\hat{\boldsymbol{w}}||^2. \tag{D.4}$$

Inserting in eq. (D.2) gives the desired representation of the generalisation error in terms of the optimal overlap parameters:

$$\epsilon_g = \frac{1}{4^k}\int \mathrm{d}y \int \mathrm{d}\nu\, P_y^0(y|\nu)\int \mathrm{d}\lambda\, P(\nu,\lambda)(y-\hat{f}(\lambda))^2 \tag{D.5}$$

For linear labels $y = \boldsymbol{c}\cdot\boldsymbol{\theta}^0$ in the regression problem, we simply have:

$$\epsilon_g = \rho + Q^{\star} - 2M^{\star} \tag{D.6}$$

while for the corresponding classification problem with $y = \text{sign}\left(\boldsymbol{c}\cdot\boldsymbol{\theta}^0\right)$:

$$\epsilon_g = \frac{1}{\pi}\cos^{-1}\left(\frac{M^{\star}}{\sqrt{Q^{\star}}}\right) \tag{D.7}$$

which, as expected, only depend on the angle between $\mathrm{F}\hat{\boldsymbol{w}}$ and $\boldsymbol{\theta}^0$.

D.2 Training loss

Similarly to the generalisation error, the asymptotic of the training loss, defined for the training data $\{\boldsymbol{x}^{\mu}, y^{\mu}\}_{\mu=1}^{n}$ as:

$$\epsilon_t = \frac{1}{n}\mathbb{E}_{\{\boldsymbol{x}^{\mu},y^{\mu}\}}\left[\sum_{\mu=1}^{n}\ell\left(y^{\mu}, \boldsymbol{x}^{\mu}\cdot\hat{\boldsymbol{w}}\right) + \frac{\lambda}{2}\|\hat{\boldsymbol{w}}\|_2^2\right], \tag{D.8}$$

can also be written only in terms of the overlap parameters. Indeed, it is closely related to the free energy density defined in eq. (C.2). A close inspection on this definition tells us that:

$$\lim_{n\to\infty}\epsilon_t = \lim_{\beta\to\infty}\partial_{\beta}f_{\beta}. \tag{D.9}$$

Taking the derivative of the free energy with respect to the parameter β and recalling that $p = \alpha n$, we can then get:

$$\lim_{n\to\infty}\epsilon_t = \frac{\lambda}{2\alpha}\lim_{p\to\infty}\mathbb{E}_{\{\boldsymbol{x}^{\mu},y^{\mu}\}}\left[\frac{\|\hat{\boldsymbol{w}}\|_2^2}{p}\right] - \lim_{\beta\to\infty}\partial_{\beta}\Psi_y. \tag{D.10}$$

For what concerns the contribution of the regulariser, we simply note that in the limit of $p \to \infty$, the average concentrates around the overlap parameter $q_w^\star$. Instead, for what concerns the contribution of the loss function, we can start by explicitly taking the derivative with respect to β of Ψ_y in eq. (C.32), i.e.:

$$\partial_\beta \Psi_y = -\mathbb{E}_\xi \left[\int_{\mathbb{R}} \mathrm{d}y \, \frac{\mathcal{Z}_y^0(y, \omega_0^\star)}{\mathcal{Z}_y(y, \omega_1^\star)} \int \frac{\mathrm{d}x}{\sqrt{2\pi V_1^\star}} e^{-\frac{1}{2V_1^\star}(x-\omega_1^\star)^2 - \beta \ell(y,x)} \ell(y,x) \right], \tag{D.11}$$

with $\mathcal{Z}_y^{:/0}$ defined in eq. (C.31). At this point, as explained more in details in section D.4, we can notice that in the limit of $\beta \to \infty$, it holds:

$$\lim_{\beta\to\infty} \partial_\beta \Psi_y = -\mathbb{E}_\xi \left[\int_{\mathbb{R}} \mathrm{d}y \, \mathcal{Z}_y^0(y, \omega_0^\star) \, \ell(y, \eta(y, \omega_1^\star)) \right], \tag{D.12}$$

with $\eta(y, \omega_1^\star)$ given in eq. (D.21). Combining the two results together we then finally get:

$$\lim_{n\to\infty} \epsilon_t \to \frac{\lambda}{2\alpha} q_w^\star + \mathbb{E}_\xi \left[\int_{\mathbb{R}} \mathrm{d}y \, \mathcal{Z}_y^0(y, \omega_0^\star) \, \ell(y, \eta(y, \omega_1^\star)) \right]. \tag{D.13}$$

D.3 Solving for the overlaps

As we showed above, both the generalisation error and the training loss are completely determined by the $\beta \to \infty$ solution of the extremization problem in eq. (C.35). For strictly convex losses ℓ, there is a unique solution to this problem, that can be found by considering the derivatives of the replica potential. This leads to a set of self-consistent saddle-point equations that can be solved iteratively:

$$\begin{cases} \hat{r}_s = -2\frac{\alpha}{\gamma}\partial_{r_s}\Psi_y(R,Q,M) \\ \hat{q}_s = -2\frac{\alpha}{\gamma}\partial_{q_s}\Psi_y(R,Q,M) \\ \hat{m}_s = \frac{\alpha}{\gamma}\partial_{m_s}\Psi_y(R,Q,M) \\ \\ \hat{r}_w = -2\alpha\partial_{r_w}\Psi_y(R,Q,M) \\ \hat{q}_w = -2\alpha\partial_{q_w}\Psi_y(R,Q,M) \end{cases} \qquad \begin{cases} r_s = -\frac{2}{\gamma}\partial_{\hat{r}_s}\Psi_w(\hat{r}_s,\hat{q}_s,\hat{m}_s,\hat{r}_w,\hat{q}_w) \\ q_s = -\frac{2}{\gamma}\partial_{\hat{q}_s}\Psi_w(\hat{r}_s,\hat{q}_s,\hat{m}_s,\hat{r}_w,\hat{q}_w) \\ m_s = \frac{1}{\gamma}\partial_{\hat{m}_s}\Psi_w(\hat{r}_s,\hat{q}_s,\hat{m}_s,\hat{r}_w,\hat{q}_w) \\ \\ r_w = -2\partial_{\hat{r}_w}\Psi_w(\hat{r}_s,\hat{q}_s,\hat{m}_s,\hat{r}_w,\hat{q}_w) \\ q_w = -2\partial_{\hat{q}_w}\Psi_w(\hat{r}_s,\hat{q}_s,\hat{m}_s,\hat{r}_w,\hat{q}_w) \end{cases} \tag{D.14}$$

In the case of a F with well-defined spectral density μ, we can be more explicit and write:

$$\begin{cases} V_s = \frac{1}{\hat{V}_s}(1 - z\, g_\mu(-z)) \\ q_s = \frac{\hat{m}_s^2+\hat{q}_s}{\hat{V}_s^2}\left[1 - 2z g_\mu(-z) + z^2 g'_\mu(-z)\right] - \frac{\hat{q}_w}{(\beta\lambda+\hat{V}_w)\hat{V}_s}\left[-z g_\mu(-z) + z^2 g'_\mu(-z)\right] \\ m_s = \frac{\hat{m}_s}{\hat{V}_s}(1 - z\, g_\mu(-z)) \\ \\ V_w = \frac{\gamma}{\beta\lambda+\hat{V}_w}\left[\frac{1}{\gamma} - 1 + z g_\mu(-z)\right] \\ q_w = \gamma\frac{\hat{q}_w}{(\beta\lambda+\hat{V}_w)^2}\left[\frac{1}{\gamma} - 1 + z^2 g'_\mu(-z)\right] + \frac{\hat{m}_s^2+\hat{q}_s}{(\beta\lambda+\hat{V}_w)\hat{V}_s}\left[-z g_\mu(-z) + z^2 g'_\mu(-z)\right] \end{cases} \tag{D.15}$$

where:

$$V_{s/w} = r_{s/w} - q_{r/w} \qquad \hat{V}_{s/w} = \hat{r}_{s/w} + \hat{q}_{r/w} \qquad z = \gamma\frac{\beta\lambda + \hat{V}_w}{\hat{V}_s} \tag{D.16}$$

We can also simplify slightly the derivatives of Ψ_y without loosing generality by applying Stein's lemma, yielding:

$$\begin{cases} \hat{V}_s = \frac{\alpha\mu_1^2}{\gamma}\mathbb{E}_\xi\left[\int_{\mathbb{R}}\mathrm{d}y\ \mathcal{Z}_y^0\left(y;\frac{M}{\sqrt{Q}}\xi,\rho-\frac{M^2}{Q}\right)\partial_\omega f_y\left(y;\sqrt{Q}\xi,R-Q\right)\right] \\ \hat{q}_s = \frac{\alpha\mu_1^2}{\gamma}\mathbb{E}_\xi\left[\int_{\mathbb{R}}\mathrm{d}y\ \mathcal{Z}_y^0\left(y;\frac{M}{\sqrt{Q}}\xi,\rho-\frac{M^2}{Q}\right) f_y\left(y;\sqrt{Q}\xi,R-Q\right)^2\right] \\ \hat{m}_s = \frac{\alpha\mu_1}{\gamma}\mathbb{E}_\xi\left[\int_{\mathbb{R}}\mathrm{d}y\ \mathcal{Z}_y^0\left(y;\frac{M}{\sqrt{Q}}\xi,\rho-\frac{M^2}{Q}\right) f_y^0\left(y;\frac{M}{\sqrt{Q}}\xi,\rho-\frac{M^2}{Q}\right) f_y\left(y;\sqrt{Q}\xi,R-Q\right)\right] \\ \\ \hat{V}_w = \alpha\mu_\star^2\mathbb{E}_\xi\left[\int_{\mathbb{R}}\mathrm{d}y\ \mathcal{Z}_y^0\left(y;\frac{M}{\sqrt{Q}}\xi,\rho-\frac{M^2}{Q}\right)\partial_\omega f_y\left(y;\sqrt{Q}\xi,R-Q\right)\right] \\ \hat{q}_w = \alpha\mu_\star^2\mathbb{E}_\xi\left[\int_{\mathbb{R}}\mathrm{d}y\ \mathcal{Z}_y^0\left(y;\frac{M}{\sqrt{Q}}\xi,\rho-\frac{M^2}{Q}\right) f_y\left(y;\sqrt{Q}\xi,R-Q\right)^2\right] \end{cases} \tag{D.17}$$

with $f_y^{\cdot/0}(y;\omega,V) = \partial_\omega \log \mathcal{Z}_y^{\cdot/0}$. For a given choice of spectral density μ (corresponding to a choice of projection F), label rule P_y^0 and loss function ℓ, the auxiliary functions $(\mathcal{Z}^0,\mathcal{Z})$ can be computed, and from them the right-hand side of the update equations above. The equations can then be iterated until the convergence to the fixed point minimising the free energy at fixed (α,γ,β). For convex losses and $\beta\to\infty$, the fixed point of these equations gives the overlap corresponding to the estimator solving eq. (A.7).

D.4 Taking $\beta\to\infty$ explicitly

Although the saddle-point equations above can be iterated explicitly for any $\beta>0$, it is envisageable to take the limit $\beta\to\infty$ explicitly, since β is an auxiliary parameter we introduced, and that was not present in the original problem defined in eq. (A.7).

Since the overlap parameters depend on β only implicitly through $\mathcal{Z}_y$ and its derivatives, we proceed with the following ansatz for their $\beta\to\infty$ scaling:

$$\begin{aligned} V_{s/w}^\infty &= \beta V_{s/w} & q_{s/w}^\infty &= q_{s/w} & m_s^\infty &= m_s \\ \hat{V}_{s/w}^\infty &= \frac{1}{\beta}\hat{V}_{s/w} & \hat{q}_{s/w}^\infty &= \frac{1}{\beta^2}\hat{q}_{s/w} & \hat{m}_s^\infty &= \hat{m}_s. \end{aligned} \tag{D.18}$$

This ansatz can be motivated as follows. Recall that:

$$\mathcal{Z}_y(y;\omega,V) = \int\frac{\mathrm{d}x}{\sqrt{2\pi V}}e^{-\frac{1}{2V}(x-\omega)^2-\beta\ell(x,y)} = \int\frac{\mathrm{d}x}{\sqrt{2\pi V}}e^{-\beta\left[\frac{(x-\omega)^2}{2\beta V}+\ell(x,y)\right]}. \tag{D.19}$$

Therefore, letting $V=\mu_1^2V_s+\mu_\star^2V_w$ scale as $V^\infty=\beta V$, at $\beta\to\infty$:

$$\mathcal{Z}_y(y;\omega,V) \underset{\beta\to\infty}{=} \delta\left(x-\eta\right) \tag{D.20}$$

where:

$$\eta(y;\omega,V) = \operatorname*{argmin}_{x\in\mathbb{R}}\left[\frac{(x-\omega)^2}{2V^\infty}+\ell(x,y)\right]. \tag{D.21}$$

For convex losses ℓ with $\lambda>0$, this one-dimensional minimisation problem has a unique solution that can be easily evaluated. Intuitively, this ansatz translates the fact the variance of our estimator goes to zero as a power law at $\beta\to\infty$, meaning the Gibbs measure concentrates around the solution of the optimisation problem eq. (A.7). The other scalings in eq. (D.19) follow from analysing the dependence of the saddle-point equations in V.

The ansatz in eq. (D.18) allow us to take the $\beta \to \infty$ and rewrite the saddle-point equations as:

$$\begin{cases} \hat{V}_s^\infty = \frac{\alpha\mu_1^2}{\gamma}\mathbb{E}_\xi\left[\int_{\mathbb{R}} \mathrm{d}y\ \mathcal{Z}_y^0 \frac{\partial_\omega \eta}{V^\infty}\right] \\ \hat{q}_s^\infty = \frac{\alpha\mu_1^2}{\gamma}\mathbb{E}_\xi\left[\int_{\mathbb{R}} \mathrm{d}y\ \mathcal{Z}_y^0 \left(\frac{\eta-\omega}{V^\infty}\right)^2\right] \\ \hat{m}_s^\infty = \frac{\alpha\mu_1}{\gamma}\mathbb{E}_\xi\left[\int_{\mathbb{R}} \mathrm{d}y\ \partial_\omega \mathcal{Z}_y^0 \left(\frac{\eta-\omega}{V^\infty}\right)\right] \\ \\ \hat{V}_w^\infty = \alpha\mu_\star^2\mathbb{E}_\xi\left[\int_{\mathbb{R}} \mathrm{d}y\ \mathcal{Z}_y^0 \frac{\partial_\omega \eta}{V^\infty}\right] \\ \hat{q}_w^\infty = \alpha\mu_\star^2\mathbb{E}_\xi\left[\int_{\mathbb{R}} \mathrm{d}y\ \mathcal{Z}_y^0 \left(\frac{\eta-\omega}{V^\infty}\right)^2\right] \end{cases} \tag{D.22}$$

$$\begin{cases} V_s^\infty = \frac{1}{\hat{V}_s^\infty}\left(1 - z\ g_\mu(-z)\right) \\ q_s^\infty = \frac{(\hat{m}_s^\infty)^2+\hat{q}_s^\infty}{\left(\hat{V}_s^\infty\right)^2}\left[1 - 2z g_\mu(-z) + z^2 g'_\mu(-z)\right] - \frac{\hat{q}_w^\infty}{(\beta\lambda+\hat{V}_w)\hat{V}_s}\left[-z g_\mu(-z) + z^2 g'_\mu(-z)\right] \\ m_s^\infty = \frac{\hat{m}_s^\infty}{\hat{V}_s^\infty}\left(1 - z\ g_\mu(-z)\right) \\ \\ V_w^\infty = \frac{\gamma}{\lambda+\hat{V}_w^\infty}\left[\frac{1}{\gamma} - 1 + z g_\mu(-z)\right] \\ q_w^\infty = \gamma\frac{\hat{q}_w^\infty}{(\lambda+\hat{V}_w^\infty)^2}\left[\frac{1}{\gamma} - 1 + z^2 g'_\mu(-z)\right] + \frac{(\hat{m}_s^\infty)^2+\hat{q}_s^\infty}{(\lambda+\hat{V}_w^\infty)\hat{V}_s^\infty}\left[-z g_\mu(-z) + z^2 g'_\mu(-z)\right] \end{cases} \tag{D.23}$$

where $\mathcal{Z}_y^0(y;\omega,V)$ is always evaluated at $(\omega,V) = \left(\frac{M^\infty}{\sqrt{Q^\infty}}\xi, \rho - \frac{M^{\infty\,2}}{Q^\infty}\right)$, $\eta(y;\omega,V)$ at $(\omega,V) = \left(\sqrt{Q^\infty}\xi, V^\infty\right)$ and $z = \frac{\lambda+\hat{V}_w^\infty}{\hat{V}_s^\infty}$.

D.5 Examples

In this section we exemplify our general result in two particular cases for which the integrals in the right-hand side of eq. (D.22) can be analytically performed: the ridge regression task with linear labels and a classification problem with square loss and ridge regularisation term. The former example appears in Fig. 5 (left) and the later in Figs. 2 (blue curve), 6, 7 of the main.

Ridge regression with linear labels: Consider the task of doing ridge regression $\ell(y,x) = \frac{1}{2}(y-x)^2$, $\lambda > 0$ on the linear patterns $\boldsymbol{y} = \frac{1}{\sqrt{d}}\mathrm{C}\boldsymbol{\theta}^0 + \sqrt{\Delta}\boldsymbol{z}$, with $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathrm{I}_n)$ and $\boldsymbol{\theta}^\star \sim \mathcal{N}(\mathbf{0}, \mathrm{I}_d)$. In this case, we have:

$$\eta(y;\omega,V) = \frac{\omega + yV}{1+V} \tag{D.24}$$

and the saddle-point equations for the hat overlap read:

$$\hat{V}_s^\infty = \frac{\alpha}{\gamma}\frac{\mu_1^2}{1+V^\infty} \qquad \hat{q}_s^0 = \frac{\alpha\mu_1^2}{\gamma}\frac{1+\Delta+Q^\infty-2M^\infty}{(1+V^\infty)^2} \qquad \hat{m}_s = \frac{\alpha}{\gamma}\frac{\mu_1}{1+V^\infty}$$

$$\hat{V}_w^\infty = \frac{\alpha\mu_\star^2}{1+V^\infty} \qquad \hat{q}_w^\infty = \alpha\mu_\star^2\frac{1+\Delta+Q^\infty-2M^\infty}{(1+V^\infty)^2} \tag{D.25}$$

This particular example corresponds precisely to the setting studied in [34].

Classification with square loss and ridge regularisation: Consider a classification task with square loss $\ell(y,x) = \frac{1}{2}(y-x)^2$ and labels generated as $\boldsymbol{y} = \mathrm{sign}\left(\frac{1}{\sqrt{d}}\mathrm{C}\boldsymbol{\theta}^0\right)$, with $\boldsymbol{\theta}^0 \sim \mathcal{N}(\mathbf{0}, \mathrm{I}_d)$. Then the saddle-point

equations are simply:

$$\hat{V}_s^\infty = \frac{\alpha}{\gamma}\frac{\mu_1^2}{1+V^\infty} \qquad \hat{q}_s^\infty = \frac{\alpha}{\gamma}\mu_1^2\frac{1+Q^\infty-\frac{2M^\infty}{\sqrt{\pi}}}{(1+V^\infty)^2} \qquad \hat{m}_s = \frac{\alpha}{\gamma}\sqrt{\frac{2}{\pi}}\frac{\mu_1}{1+V^\infty}$$

$$\hat{V}_w^\infty = \frac{\alpha\mu_\star^2}{1+V^\infty} \qquad \hat{q}_w^\infty = \alpha\mu_\star^2\frac{1+Q^\infty-\frac{2M^\infty}{\sqrt{\pi}}}{(1+V^\infty)^2} \tag{D.26}$$

E Numerical Simulations

In this section, we provide more details on how the numerical simulations in the main manuscript have been performed.

First, the dataset $\{\boldsymbol{x}^\mu, y^\mu\}_{\mu=1}^n$ is generated according to the procedure described in Section 1.1 of the main, which we summarise here for convenience in algorithm 1:

Algorithm 1 Generating dataset $\{\boldsymbol{x}^\mu, y^\mu\}_{\mu=1}^n$

Input: Integer d, parameters $\alpha, \gamma \in \mathbb{R}_+$, matrix $\mathrm{F} \in \mathbb{R}^{d\times p}$, vector $\boldsymbol{\theta}^0 \in \mathbb{R}^d$ non-linear functions $\sigma, f^0 : \mathbb{R} \to \mathbb{R}$.
Assign $p \leftarrow \lfloor d/\gamma \rfloor$, $n \leftarrow \lfloor \alpha p \rfloor$
Draw $\mathrm{C} \in \mathbb{R}^{n\times d}$ with entries $c_\rho^\mu \sim \mathcal{N}(0,1)$ i.i.d.
Assign $\boldsymbol{y} \leftarrow f^0\left(\mathrm{C}\boldsymbol{\theta}^0\right) \in \mathbb{R}^n$ component-wise.
Assign $\mathrm{X} \leftarrow \sigma\left(\mathrm{CF}\right) \in \mathbb{R}^{n\times p}$ component-wise.
Return: $\mathrm{X}, \boldsymbol{y}$

In all the examples from the main, we have drawn $\boldsymbol{\theta}^0 \sim \mathcal{N}(0, \mathrm{I}_d)$. For the regression task in Fig. 5 we have taken $f^0(x) = x$, while in the remaining classification tasks $f^0(x) = \mathrm{sign}(x)$. For Gaussian projections, the components of F are drawn from $\mathcal{N}(0,1)$ i.i.d., and in for the random orthogonal projections we draw two orthogonal matrices $\mathrm{U} \in \mathbb{R}^{d\times d}$, $\mathrm{V} \in \mathbb{R}^{p\times p}$ from the Haar measure and we let $\mathrm{F} = \mathrm{U}^\top \mathrm{D}\mathrm{V}$ with $\mathrm{D} \in \mathbb{R}^{d\times p}$ a diagonal matrix with diagonal entries $d_k = \max(\sqrt{\gamma}, 1)$, $k = 1, \cdots, \min(n,p)$.

Given this dataset, the aim is to infer the configuration $\hat{\boldsymbol{w}}$, minimising a given loss function with a ridge regularisation term. In the following, we describe how to accomplish this task for both square and logistic loss.

Square Loss: In this case, the goal is to solve the following optimisation problem:

$$\hat{\boldsymbol{w}} = \min_{\boldsymbol{w}} \left[\frac{1}{2}\sum_{\mu=1}^n \left(y^\mu - \boldsymbol{x}^\mu\cdot\boldsymbol{w}\right)^2 + \frac{\lambda}{2}||\boldsymbol{w}||_2^2\right] . \tag{E.1}$$

which has a simple closed-form solution given in terms of the Moore-Penrose inverse:

$$\hat{\boldsymbol{w}} = \begin{cases} \left(\mathrm{X}^\top\mathrm{X} + \lambda \mathrm{I}_p\right)^{-1}\mathrm{X}^\top \boldsymbol{y}, & \text{if } n > p \\ \\ \mathrm{X}^\top\left(\mathrm{X}\mathrm{X}^T + \lambda \mathrm{I}_n\right)^{-1}\boldsymbol{y}, & \text{if } p > n \end{cases} \tag{E.2}$$

Logistic Loss: In this case, the goal is to solve the following optimisation problem:

$$\hat{\boldsymbol{w}} = \min_{\boldsymbol{w}} \left[\sum_{\mu=1}^n \log\left(1 + e^{y^\mu(\boldsymbol{x}^\mu\cdot\boldsymbol{w})}\right) + \frac{\lambda}{2}||\boldsymbol{w}||_2^2\right] . \tag{E.3}$$

To solve the above, we use the *Gradient Descent* (GD) on the regularised loss. In our simulations, we took advantage of Scikit-learn 0.22.1, an out-of-the-box open source library for machine learning tasks in Python

[61, 62]. The library provides the class *sklearn.linear_model.LogisticRegression*, which implements GD with logistic loss and a further ℓ_2-regularisation, if the parameter 'penalty' is set to 'l2'. GD stops either if the following condition is satisfied:

$$\max\{(\nabla \boldsymbol{w})_i \,|i = 1, ..., p\} \leqslant \text{tol}, \tag{E.4}$$

with $\nabla \boldsymbol{w}$ being the gradient, or if a maximum number of iterations is reached. We set tol to 10^{-4} and the maximum number of iterations to 10^4.

In both cases described above, the algorithm returns the estimator $\hat{\boldsymbol{w}} \in \mathbb{R}^p$, from which all the quantities of interest can be evaluated. For instance, the generalisation error can be simply computed by drawing a new and independent sample $\{\mathrm{X}^{\text{new}}, \boldsymbol{y}^{\text{new}}\}$ using algorithm 1 with the same inputs F, σ, f^0 and $\boldsymbol{\theta}^0$ and computing:

$$\epsilon_g(n, p, d) = \frac{1}{4^k n} ||\boldsymbol{y}^{\text{new}} - \hat{f}\left(\mathrm{X}^{\text{new}} \hat{\boldsymbol{w}}\right)||_2^2 \tag{E.5}$$

with $\hat{f}(x) = x$ for the regression task and $\hat{f}(x) = \text{sign}(x)$ for the classification task.

The procedure outlined above is repeated n_{seeds} times, for different and independent draws of the random quantities F, $\boldsymbol{\theta}^0$, and a simple mean is taken in order to obtain the ensemble average of the different quantities. In most of the examples from the main, we found that $n_{\text{seeds}} = 30$ was enough to obtain a very good agreement with the analytical prediction from the replica analysis. The full pipeline for computing the averaged generalisation error is exemplified in algorithm 2.

Algorithm 2 Averaged generalisation error.

Input: Integer d, parameters $\alpha, \gamma, \lambda \in \mathbb{R}_+$, non-linear functions σ, f^0, $\hat{f}$ and integer n_{seeds}.
Assign $p \leftarrow \lfloor d/\gamma \rfloor$, $n \leftarrow \lfloor \alpha p \rfloor$
Initialise $E_g = 0$.
for $i = 1$ **to** n_{seeds} **do**
  Draw F, $\boldsymbol{\theta}^0$.
  Assign X, $\boldsymbol{y} \leftarrow$ Alg. 1.
  Compute $\hat{\boldsymbol{w}}$ from eq. (E.1) or (E.3) with X, $\boldsymbol{y}$ and λ.
  Generate new dataset X^{new}, $\boldsymbol{y}^{\text{new}}$ from Alg. 1.
  Assign $E_g \leftarrow E_g + \frac{1}{4^k n} ||\boldsymbol{y}^{\text{new}} - \hat{f}\left(\mathrm{X}^{\text{new}} \hat{\boldsymbol{w}}\right)||_2^2$
end for
Return: $\epsilon_g = \frac{E_g}{n_{\text{seeds}}}$

References

[1] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *ICLR 2017, arXiv preprint arXiv:1611.03530*, 2016.

[2] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pages 5947–5956, 2017.

[3] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Wiley, New York, 1st edition, September 1998.

[4] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[5] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[6] Hyunjune Sebastian Seung, Haim Sompolinsky, and Naftali Tishby. Statistical mechanics of learning from examples. *Physical review A*, 45(8):6056, 1992.

[7] Timothy LH Watkin, Albrecht Rau, and Michael Biehl. The statistical mechanics of learning a rule. *Reviews of Modern Physics*, 65(2):499, 1993.

[8] Madhu Advani, Subhaneil Lahiri, and Surya Ganguli. Statistical mechanics of complex neural systems and high dimensional data. *Journal of Statistical Mechanics: Theory and Experiment*, 2013(03):P03014, 2013.

[9] Madhu S Advani and Andrew M Saxe. High-dimensional dynamics of generalization error in neural networks. *arXiv preprint arXiv:1710.03667*, 2017.

[10] Benjamin Aubin, Antoine Maillard, Florent Krzakala, Nicolas Macris, Lenka Zdeborová, et al. The committee machine: Computational to statistical gaps in learning a two-layers neural network. In *Advances in Neural Information Processing Systems*, pages 3223–3234, 2018.

[11] Emmanuel J Candès and Pragya Sur. The phase transition for the existence of the maximum likelihood estimate in high-dimensional logistic regression. *arXiv preprint arXiv:1804.09753*, 2018.

[12] Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *arXiv preprint arXiv:1903.08560*, 2019.

[13] Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and double descent curve. *arXiv preprint arXiv:1908.05355*, 2019.

[14] Sebastian Goldt, Marc Mézard, Florent Krzakala, and Lenka Zdeborová. Modelling the influence of data structure on learning in neural networks. *arXiv preprint arXiv:1909.11500*, 2019.

[15] Marc Mézard. Mean-field message-passing equations in the hopfield model and its generalizations. *Physical Review E*, 95(2):022117, 2017.

[16] Lénaïc Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems 32*, pages 2933–2943. 2019.

[17] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

[18] Mario Geiger, Stefano Spigler, Arthur Jacot, and Matthieu Wyart. Disentangling feature and lazy learning in deep neural networks: an empirical study. *arXiv preprint arXiv:1906.08034*, 2019.

[19] Cosme Louart, Zhenyu Liao, Romain Couillet, et al. A random matrix approach to neural networks. *The Annals of Applied Probability*, 28(2):1190–1248, 2018.

[20] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems 20*, pages 1177–1184. 2008.

[21] 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.

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

[23] Lenka Zdeborová and Florent Krzakala. Statistical physics of inference: Thresholds and algorithms. *Advances in Physics*, 65(5):453–552, 2016.

[24] Michel Talagrand. The Parisi formula. *Annals of mathematics*, pages 221–263, 2006.

[25] Jean Barbier, Florent Krzakala, Nicolas Macris, Léo Miolane, and Lenka Zdeborová. Optimal errors and phase transitions in high-dimensional generalized linear models. *Proceedings of the National Academy of Sciences*, 116(12):5451–5460, 2019.

[26] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[27] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[28] Mohamed El Amine Seddik, Cosme Louart, Mohamed Tamaazousti, and Romain Couillet. Random matrix theory proves that deep learning representations of gan-data behave as gaussian mixtures. *arXiv preprint arXiv:2001.08370*, 2020.

[29] Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *ICLR 2019, arXiv preprint arXiv:1810.02054*, 2018.

[30] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252, 2019.

[31] Blake Woodworth, Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Kernel and deep regimes in overparametrized models. *arXiv preprint arXiv:1906.05827*, 2019.

[32] Quoc Le, Tamás Sarlós, and Alex Smola. Fastfood-approximating kernel expansions in loglinear time. In *Proceedings of the international conference on machine learning*, volume 85, 2013.

[33] Marcin Moczulski, Misha Denil, Jeremy Appleyard, and Nando de Freitas. ACDC: A structured efficient linear layer. *arXiv preprint arXiv:1511.05946*, 2015.

[34] Andrea Montanari, Feng Ruan, Youngtak Sohn, and Jun Yan. The generalization error of max-margin linear classifiers: High-dimensional asymptotics in the overparametrized regime. *arXiv preprint arXiv:1911.01544*, 2019.

[35] 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.

[36] 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, 2019.

[37] Krzysztof M Choromanski, Mark Rowland, and Adrian Weller. The unreasonable effectiveness of structured random orthogonal embeddings. In *Advances in Neural Information Processing Systems*, pages 219–228, 2017.

[38] Elizabeth Gardner and Bernard Derrida. Three unfinished works on the optimal storage capacity of networks. *Journal of Physics A: Mathematical and General*, 22(12):1983, 1989.

[39] Yoshiyuki Kabashima, Tadashi Wadayama, and Toshiyuki Tanaka. A typical reconstruction limit for compressed sensing based on lp-norm minimization. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(09):L09003, 2009.

[40] Florent Krzakala, Marc Mézard, François Sausset, YF Sun, and Lenka Zdeborová. Statistical-physics-based reconstruction in compressed sensing. *Physical Review X*, 2(2):021005, 2012.

[41] Xiuyuan Cheng and Amit Singer. The spectrum of random inner-product kernel matrices. *Random Matrices: Theory and Applications*, 02(04):1350010, 2013.

[42] Jeffrey Pennington and Pratik Worah. Nonlinear random matrix theory for deep learning. In *Advances in Neural Information Processing Systems 30*, pages 2637–2646. 2017.

[43] Mohamed El Amine Seddik, Mohamed Tamaazousti, and Romain Couillet. Kernel random matrices of large concentrated data: the example of gan-generated images. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7480–7484. IEEE, 2019.

[44] Brady Neal, Sarthak Mittal, Aristide Baratin, Vinayak Tantia, Matthew Scicluna, Simon Lacoste-Julien, and Ioannis Mitliagkas. A modern take on the bias-variance tradeoff in neural networks. *arXiv preprint arXiv:1810.08591*, 2018.

[45] 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. *arXiv preprint arXiv:1901.01608*, 2019.

[46] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *ICLR 2020, arXiv preprint arXiv:1912.02292*, 2019.

[47] Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992.

[48] Leo Breiman. Reflections after refereeing papers for nips. *The Mathematics of Generalization*, pages 11–15, 1995.

[49] Manfred Opper and Wolfgang Kinzel. Statistical mechanics of generalization. In *Models of neural networks III*, pages 151–209. Springer, 1996.

[50] Thomas M Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers*, (3):326–334, 1965.

[51] Bernhard Schölkopf, Alexander J Smola, Francis Bach, et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[52] Alessandro Rudi, Luigi Carratino, and Lorenzo Rosasco. Falkon: An optimal large scale kernel method. In *Advances in Neural Information Processing Systems*, pages 3888–3898, 2017.

[53] Andrea Caponnetto and Ernesto De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

[54] Yuchen Zhang, John Duchi, and Martin Wainwright. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *The Journal of Machine Learning Research*, 16(1):3299–3340, 2015.

[55] Alaa Saade, Francesco Caltagirone, Igor Carron, Laurent Daudet, Angélique Drémeau, Sylvain Gigan, and Florent Krzakala. Random projections through multiple optical scattering: Approximating kernels at the speed of light. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6215–6219. IEEE, 2016.

[56] Ruben Ohana, Jonas Wacker, Jonathan Dong, Sébastien Marmin, Florent Krzakala, Maurizio Filippone, and Laurent Daudet. Kernel computations from large-scale random features obtained by optical processing units. *arXiv preprint arXiv:1910.09880*, 2019.

[57] Alexandr Andoni, Piotr Indyk, Thijs Laarhoven, Ilya Razenshteyn, and Ludwig Schmidt. Practical and optimal lsh for angular distance. In *Advances in neural information processing systems*, pages 1225–1233, 2015.

[58] Mariusz Bojarski, Anna Choromanska, Krzysztof Choromanski, Francois Fagan, Cedric Gouy-Pailler, Anne Morvan, Nourhan Sakr, Tamas Sarlos, and Jamal Atif. Structured adaptive and random spinners for fast machine learning computations. *arXiv preprint arXiv:1610.06209*, 2016.

[59] Walid Hachem, Philippe Loubaton, and Jamal Najim. Deterministic equivalents for certain functionals of large random matrices. *Ann. Appl. Probab.*, 17(3):875–930, 06 2007.

[60] Zhou Fan and Andrea Montanari. The spectral norm of random inner-product kernel matrices. *Probability Theory and Related Fields*, 173(1):27–85, Feb 2019.

[61] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[62] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.