THEORETICAL CHARACTERIZATION OF UNCERTAINTY IN HIGH-DIMENSIONAL MACHINE LEARNING

THIS IS A TEMPORARY TITLE PAGE
It will be replaced for the final print by a version
provided by the service academique.

Thèse n. ...
présenté le ...
à la Faculté des Sciences de Base
laboratoire SPOC
programme doctoral en informatique et communications
École Polytechnique Fédérale de Lausanne

pour l'obtention du grade de Docteur ès Sciences
par

Lucas Clarté

acceptée sur proposition du jury:

Prof Nicolas Flammarion, président du jury
Prof Lenka Zdeborová, directrice de thèse
Prof Michele Ceriotti, rapporteur
Prof Aymeric Dieuleveut, rapporteur
Prof Yoshiyuki Kabashima, rapporteur

Lausanne, EPFL, 2025

Abstract

Uncertainty quantification is a crucial aspect of modern machine learning : for some applications, for instance medical diagnosis, assessing the confidence in one's prediction is as important (if not important) as having good accuracy. Yet, theoretical understanding of uncertainty quantification is still limited, partly due to the complexity of the models used in practice and the numerous methods. This thesis contributes to the theoretical understanding of uncertainty quantification in the context of supervised learning, by analyzing some popular methods in the context of high-dimensional statistics.This manuscript is divided into three parts, each focusing on a different approach to uncertainty quantification :

- The first part, made of Chapters 2, 3 and 4 is devoted to classification : for classification tasks, uncertainty can be naturally quantified by the estimated probability of each class. The main problem is then to have a well-calibrated classifier, whose confidence matches the accuracy of the predictions. In this first part, we analyze the calibration of Bayesian and frequentist methods in the context of overparametrized models. In particular, we show that frequentist methods are as calibrated as Bayesian methods, for a fraction of the computation cost. In addition, the calibration of both methods is affected by the double-descent phenomenon. Lastly, we propose a new method to calibrate pre-trained neural networks, based on the commonly used temperature scaling algorithm, called Expectation Consistency. We evaluate our method on image classification tasks and show that it is more robust than temperature scaling to noisy labels.

- Then, in Chapters 5 and 6, we analyze resampling methods for uncertainty quantification. This analysis is motivated by the observation that the Boostrap seems to fail in high dimensions, despite being a cornerstone of classical statistics. We show that even with proper regularization, bootstrap seems to fail in high-dimensions.

- Motivated by the previous analysis of the bootstrap, we develop in chapter 7 an algorithm to produce prediction intervals for regression based on full conformal prediction. Conformal prediction is a framework that allows to build valid prediction intervals with very few assumptions on the estimator and the data used.

Key words: Uncertainty quantification, statistical physics, high-dimensional statistics, message passing, replica method

Résumé

La quantification de l'incertitude est un aspect crucial de l'apprentissage automatique moderne : dans certaines applications comme le diagnostic médical, évaluer la confiance dans une prédiction est aussi important (voire plus important) que d'obtenir une bonne précision. Cependant, la compréhension théorique de la quantification de l'incertitude reste limitée, en partie à cause de la complexité des modèles utilisés en pratique et de la diversité des méthodes existantes. Cette thèse contribue à son étude théorique dans le cadre de l'apprentissage supervisé, en analysant certaines méthodes populaires dans le régime des statistiques en haute dimension. Ce manuscrit est divisé en trois parties :

- La première partie, composée des chapitres 2, 3 et 4, est consacrée à la classification. Le principal défi est alors d'obtenir un classifieur calibré, dont le niveau de confiance reflète la précision des prédictions. Nous analysons la calibration de méthodes bayésiennes et fréquentistes dans le cadre de modèles surparamétrés. En particulier, nous montrons que les méthodes fréquentistes sont aussi calibrées que les méthodes bayésiennes, pour une fraction du coût computationnel. De plus, la calibration de ces deux types de méthodes est affectée par le phénomène de double descente. Enfin, nous proposons une nouvelle méthode pour calibrer des réseaux de neurones pré-entraînés, fondée sur l'algorithme de « temperature scaling » largement utilisé, que nous appelons Expectation Consistency. Nous évaluons notre méthode sur des tâches de classification d'images et montrons qu'elle est plus robuste aux étiquettes bruitées que le temperature scaling classique.

- Ensuite, dans les chapitres 5 et 6, nous analysons les méthodes de rééchantillonnage pour la quantification de l'incertitude. Cette analyse est motivée par le constat que le bootstrap semble échouer en grande dimension, malgré son rôle central en statistique classique. Nous montrons que même avec une régularisation adaptée, le bootstrap semble inefficace dans des régimes de grande dimension.

- Motivés par cette analyse du bootstrap, nous développons dans le chapitre 7 un algorithme pour produire des intervalles de confiance dans un contexte de régression, basé sur la méthode de *full conformal prediction*, qui est une méthode guarantissant des intervalles de confiance dit valides, avec très peu d'hypothèses sur l'estimateur ou les données utilisées.

Mots clefs : Quantification des incertitudes, physique statistique, statistiques en haute dimension, message passing, méthode des répliques

Contents

List of Figures

List of Tables

Acronyms

AMP Approximate Message Passing. xix, 5, 14–17, 19

BO Bayes-optimal. xix

BP Belief Propagation. xix, 13, 14

CP conformal prediction. xix, 11

EC expectation consistency. xix, 18

ECE expected calibration error. xix

ERM empirical risk minimization. xix

FCP full conformal prediction. xix, 19

GET Gaussian Equivalence Theorem. xix

GLM generalized linear model. xix

i.i.d Independent and identically distributed. xix, 14

MCMC Markov Chain Monte Carlo. xix, 5

r-BP relaxed-Belief Propagation. xix, 14

SE state-evolution. xix, 15

T-S teacher student. xix, 18, 19

TS temperature scaling. xix, 10, 18

UQ uncertainty quantification. xix, 1, 8, 9, 13

1 Introduction

Uncertainty quantification UQ in deep learning is a highly active and rapidly evolving field, with many coexisting metrics and methods proposed in the literature. We refer the reader to [2, 88] for two recent comprehensive reviews. In this thesis, we explore how tools from statistical physics can be leveraged to address challenges in uncertainty quantification. Our motivation stems from the fact that UQ plays a central role across a wide range of tasks—including supervised learning, reinforcement learning, and generative modeling—each presenting distinct challenges. For instance, in reinforcement learning, accurately estimating uncertainty over the reward function is crucial to efficiently balance exploration and exploitation. In generative modeling, large language models (LLMs) are known to suffer from hallucinations, where the model confidently generates false information. Furthermore, it remains unclear how to effectively quantify uncertainty in generated samples.

This manuscript focuses exclusively on supervised learning, covering both regression and classification tasks, where the goal is to learn a prediction function from a finite and noisy set of observations. The sources of uncertainty we consider are threefold: the finiteness of the training data, the aleatoric uncertainty inherent to the data-generating process, and the epistemic uncertainty arising from the model's inability to perfectly capture the true data distribution.

This chapter is organized as follows. We first establish the problem setting and introduce the mathematical formalism that will serve as the foundation for the rest of the manuscript. We begin by presenting the teacher-student model and discussing various estimators, ranging from the optimal but intractable Bayes posterior to simpler frequentist approaches. We then introduce the random features model, which provides a tractable framework to study overparametrized models; this model will play a central role in chapter 3 and chapter 6. Next, we define the different uncertainty quantification metrics that will be analyzed throughout the thesis. Finally, we introduce the two main mathematical tools employed in this work—the Approximate Message Passing algorithm and the replica method from statistical physics—and illustrate how they can be used to study uncertainty quantification methods.

1.1 Supervised learning in the teacher-student setting

1.1.1 General setting

Our general goal is to analyze the performance on unseen data of various estimators trained on some datasets. In this section, we present and motivate the assumptions on the data-generating process that will be used throughout this manuscript.

Data model

We consider a data-generating process defined by a latent *teacher* vector $\theta_\star \in \mathbb{R}^d$ that we aim to infer. The components of $\theta_\star$ are assumed to be independent and identically distributed (i.i.d.) according to a prior distribution $p_\star$, such that

$$p(\theta_\star) = \prod_{\mu=1}^{d} p_\star(\theta_{\star\mu}). \tag{1.1}$$

Given an input vector $\mathbf{x} \in \mathbb{R}^d$, the corresponding label $y \in \mathbb{R}$ is generated through a likelihood function of the form

$$p(y \mid \mathbf{x}, \theta_\star) = p(y \mid {\theta_\star}^\top \mathbf{x}), \tag{1.2}$$

i.e., the label depends on the input only through the scalar product ${\theta_\star}^\top \mathbf{x}$. In regression settings, for instance, the likelihood may correspond to a Gaussian noise model: $y = {\theta_\star}^\top \mathbf{x} + \sigma\varepsilon$, with $\varepsilon \sim \mathcal{N}(0,1)$.

We further assume that the input vectors $\mathbf{x}$ are drawn from a standard Gaussian distribution with isotropic covariance, i.e., $\mathbf{x} \sim \mathcal{N}(0, I_d/d)$. All data points $(\mathbf{x}_i, y_i)$ are considered to be independently sampled according to this generative process.

The objective is to approximate the conditional distribution $p(y \mid \mathbf{x}, \mathcal{D})$ of the label y given a test input $\mathbf{x}$ and a training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ consisting of n independent samples generated by the teacher model.

The high-dimensional limit Our theoretical analysis is conducted in the *high-dimensional proportional* regime, where both the number of samples n and the input dimension d grow to infinity while their ratio remains finite. More concretely, we consider the limit $n, d \to \infty$ with $n/d \to \alpha = O(1)$.

A central focus of our study will be the *overparametrized* regime, where the number of model parameters exceeds the number of training samples. This regime is particularly interesting as is it associated with new behavious at odds with classical statistics : a notable example is the *double descent* phenomenon [29, 90], which reveals a non-monotonic relationship between

model complexity and generalization error, thereby challenging the classical bias-variance trade-off [104]. Additonally, deep neural networks seem to circumvent the *curse of dimensionality*, according to which the number of samples required to learn a function grows exponentially with the input dimension. This motivates the study of simplified high-dimensional models that may help understand the mechanisms enabling such performance.

Limitation of this model While the assumption of Gaussian data will be convenient for our computations, it is clear that it does not hold at all in practice. Beyond the fact that real data that are obviously not Gaussian, [204] has shown that event a simple perceptron learns moments of order bigger than two of the data it is trained on. Moreover, note that our Gaussian assumption will imply that the training and test data come from the same distribution -which is not verified in the presence dataset drift or in the problem of out-of-distribution detection- and will exclude the existence of outliers in the training data. While the Gaussian teacher-student model is admittedly idealized and does not capture the full complexity of real-world machine learning scenarios, it allows for precise theoretical analysis and can lead to qualitative insights, for example to study self-attention [56] or image classifiers [152]. This setting will thus serve us as a valuable testing ground for different methods.

1.1.2 Estimators

Having set the distribution of the observed data, we will now introduce the different estimators that will be studied in this thesis. We will use the notation $\hat{f}(\mathbf{x})$ to denote the prediction function.

Bayes-optimal estimator If one has access to the prior on $\theta_\star$ (1.1) and the likelihood (1.2), then the optimal way to estimate $p\left(y|\mathbf{x},\mathscr{D}\right)$ is to use Bayes' theorem, and compute the *predictive posterior*

$$p\left(y|\mathbf{x},\mathscr{D}\right) = \int \mathrm{d}\theta_\star\, p\left(y|{\theta_\star}^\top\mathbf{x}\right) \times p\left(\theta_\star|\mathscr{D}\right) \tag{1.3}$$

where $p\left(\theta_\star|\mathscr{D}\right)$ is the posterior distribution of the teacher given the dataset $\mathscr{D}$. This distribution is

$$p\left(\theta_\star|\mathscr{D}\right) = \frac{1}{p\left(\mathscr{D}\right)} p(\theta_\star) \prod_{i=1}^{n} p\left(y_i|{\theta_\star}^\top\mathbf{x}_i\right) \tag{1.4}$$

the normalization factor $p\left(\mathscr{D}\right)$ is the marginal likelihood of the dataset, also called *evidence*. By definition, the posterior (1.4) allows to compute the optimal estimator given $\mathscr{D}$. For instance, the estimator that minimizes the square error

$$\varepsilon_g = \mathbb{E}_{\mathbf{x}}\left[|y - \hat{f}(\mathbf{x})|^2\right]$$

is the mean of the predictive posterior

$$\hat{f}(\mathbf{x}) = \mathbb{E}_y\left[y|\mathscr{D},\mathbf{x}\right]$$

Empirical-Bayes estimator However, the Bayes-optimal posterior is often intractable as it requires strong assumptions on the data that are often not satisfied in practice. For this reason, we will introduce *empirical-Bayes* estimator [164, 119] : in full generality, it consists of postulating a class of priors and likelihoods, defining a class of posterior distributions that aim to approximate the true posterior (1.4). However, the prior and likelihood used to compute the empirical-Bayes posterior may not match the true data-generating process, i.e. the prior and likelihood used to generate the dataset $\mathscr{D}$. A central question is to evaluate the impact of this mismatch on the performance of the empirical-Bayes estimators, both in terms of accuracy and uncertainty quantification. While he Bayes-optimal estimator $\hat{f}_{\text{bo}}$ will always refer to the estimator obtained using the true prior and likelihood, whereas the empirical-Bayes estimator $\hat{f}_{\text{eb}}$ will refer to an estimator obtained using a prior and likelihood that may not match the true data-generating process.

In the empirical-Bayes framework, we will denote by $p_{\text{eb}}\left(y|\theta^\top\mathbf{x}\right)$ and $p_{\star,\text{eb}}$ the prior and likelihood distributions used by the statistician. We will thus sample the weights from distribution

$$p_{\text{eb}}(\theta|\mathscr{D}) = \frac{1}{\mathscr{Z}_{\text{eb}}(\mathscr{D})} p_{\star,\text{eb}}(\theta) \prod_{i=1}^{n} p_{\text{eb}}\left(y_i|\theta^\top\mathbf{x}_i\right)$$

In all that followss, we will always use a Gaussian or Laplace prior, i.e

$$p_{\star,\text{eb}}(z) = e^{-\lambda z^2/2}, \quad \text{or } p_{\star,\text{eb}}(z) = e^{-\lambda z} \tag{1.5}$$

so that the distribution p_{eb} will be parametrized by the regularization parameter λ.

This distribution can be conveniently rewriten in the exponential form :

$$p_{\text{eb}}(\theta|\mathscr{D}) = \frac{1}{p_{\text{eb}}(\mathscr{D})} e^{-\mathscr{L}(\theta)} \tag{1.6}$$

where the negative log-posterior

$$\mathscr{L}(\theta) = -\left(\sum_{i=1}^{n} \log p_{\text{eb}}\left(y_i|\theta^\top\mathbf{x}_i\right) + \log p_{\star,\text{eb}}(\theta)\right) \tag{1.7}$$

is also called *empirical risk* in the context of frequentist estimators. We will add a scaling factor β, called the *inverse temperature* to the posterior so that its form is now

$$p_{\text{eb}}\left(\theta|\mathscr{D},\beta\right) = \frac{1}{\mathscr{Z}(\beta)} e^{-\beta\mathscr{L}(\theta)} \tag{1.8}$$

This distribution is reminiscent of the Gibbs distribution in statistical physics [247] : θ represents the degrees of freedom in a physical system, for instance the position of particles, whose energy is described by the function $\mathscr{L}$ and is at a fixed temperature $T = 1/\beta$. In the context of Bayesian neural networks, the choice $\beta > 1$ is called *called posterior* and has been observed to improve uncertainty quantification in some settings [239, 127].

Sampling from the posterior The first question is how to sample from the distribution (1.8). Markov Chain Monte-Carlo (MCMC) methods suchs as Metropolis-Hastings, Langevin dynamics [238] or Gibbs sampling [197] can be used to sample this posterior but all of them suffer from the curse of dimensionality or are hard to tune to obtain fast convergence to the posterior. In section 1.3, we will present a method called *Approximate Message Passing* (AMP) that allows to efficiently approximate the predictive posterior, by computing the first two moments of the distribution (1.8). We will show that AMP is asymptotically exact in high-dimensions if some assumptions on the data distribution are satisfied.

Laplace Approximation Another way to approximate the posterior (1.8) is to use the Laplace approximation [205, 61]. It was introduced by [155] in the context of Gaussian processes, and consists in approximating the posterior by a Gaussian distribution whose mean is minimizer of the empirical risk and whose covariance is the inverse of Hessian matrix $\nabla^2\mathscr{L}$ of the log-likelihood at its minimum $\hat{\theta}$:

$$\mathscr{L}(\theta) = \mathscr{L}(\hat{\theta}) + \frac{1}{2}(\theta - \hat{\theta})^\top \nabla^2\mathscr{L}(\hat{\theta})(\theta - \hat{\theta}) + \mathscr{O}\left((\theta - \hat{\theta})^3\right) \tag{1.9}$$

So that we'll sample from the distribution

$$p_{\text{Lap}} = \mathcal{N}\left(\hat{\theta}, \left(\nabla^2\mathscr{L}\right)^{-1}\right) \tag{1.10}$$

In the models considered here, the loss will always be convex in the parameters θ, so that the Hessian is positive-definite and its minimum is unique. In practice, the loss landscape of neural networks is non-convex, the minimum of the loss is not unique and the Hessian can be ill-defined. Moreover, due to the large number of parameters, the Hessian is not tractable and one must rely on approximations. Finally, we will show that in the context

Frequentist estimators Motivated by the initial task of sampling the Bayes-optimal estimator, in the previous sections we only considered Bayesian estimators. However, frequentists methods that learn a single estimator and output a point prediction, instead of sampling a distribution of estimators, are arguably more used in practice due to their relative simplicity and their lower computational cost. In the context of regression and classification, the simplest models are generalized linear models whose prediction functions are of the type

$$\hat{f}(\mathbf{x}) = \sigma(\theta^\top \mathbf{x}) \tag{1.11}$$

for some weights θ. A common way to learn these weights is by minimizing the empirical risk as defined in eq. (1.7). Note that the frequentist estimator can be framed as a special case of the distribution (1.8) in the limit $\beta \to \infty$: in this limit, the distribution p_{eb} is simply a Diract delta at the minimizer of the empirical risk. In the case where $\mathscr{L}$ is convex and differentiable, it is usually minimized by gradient descent, while other methods exist. For instance, the binary perceptron can be used to learn a binary classifier on linearly separable data.

In the context of neural networks, a common estimator is the following

$$\hat{f}(\mathbf{x}) = \mathbf{W}^L \sigma^{L-1}(\mathbf{W}^{L-1}\sigma^{(L-2)}(\cdots) + \mathbf{b}^{L-1}) + \mathbf{b}^L \tag{1.12}$$

for weight matrices $\mathbf{W}^l$, biases $\mathbf{b}^l$ and activation functions σ^l. The weights and biases are learned by minimizing the empirical risk. In this case, the risk is non-convex and may have several local minima.

While they will not be studied in this manuscript, plenty of other frequentists estimators can be used in practice : non-parametric methods include random forests or k-nearest neighbors to name a few.

Embedding the input in feature maps

In the (unrealistic case) that the statistician knows that the labels are generated by the model (1.2), then the optimal way to infer $\theta_\star$ is to use the original features $\mathbf{x}$. Most of the time, this is impossible in practice, and modern machine learning has shown that overparametrized models tend to generalize better, in spite of the usual bias-variance trade-off we learn in statistics. For instance, the *double-descent* phenomenon [29, 90] shows that the test error of a model can decrease after the interpolation threshold, where the model perfectly fits the training data.

Random features In order to model this overparametrization, we will consider the well-studied *random features* model [201], where the input $\mathbf{x}$ is embedded in a random feature space dimension p. We will consider the following model

$$\varphi(\mathbf{x}) = \sigma(F\mathbf{x}) / \sqrt{p} \tag{1.13}$$

where the entries of F are i.i.d sampled from a Gaussian distribution $\mathcal{N}(0, 1)$ and σ is a non-linear activation function. For the sake of simplicity, we will assume that the mean of $\sigma(z)$ is 0 if z is a standard Gaussian (this is the case e.g. for the hyperbolic tangent function). An exact characterization of the empirical risk minimizer in the random features model has been discussed in [168, 94, 99, 109, 66, 153].

Last-layer methods Numerous uncertainty quantification methods are based on ensembling the prediction of different neural networks [143]. However, due to the size of these models and the computational cost of training them, it is often impossible to train several of them. To circumvent this computational cost, one approach is to use *last-layer methods*, in which one pretrains a model and freezes all weights apart from the last-layer, also called the read-out layer. This freeze means fixing the feature map $\mathbf{x} \to \varphi$ and reducing the neural network to a generalized linear model

$$\hat{f}(\mathbf{x}) = \theta^\top \varphi(\mathbf{x}) \tag{1.14}$$

A possible way to generate an ensemble of estimators is the Laplace approximation, where we sample θ from a Gaussian distribution where the mean is the minimum of the empirical risk and the covariance is the inverse of the Hessian. [40] studied different methods for uncertainty quantification based on this last-layer approximation, and this approach has been used in practice e.g. in computational chemistry [48, 37]. Additionally, [136] has argued that a Bayesian treatment of the last layer of deep networks fixes overconfidence.

1.1.3 Learning an ensemble of estimators

Some uncertainty quantification schemes that will be analyzed here are based on ensembling the predictions of several estimators trained possibly on different datasets. We introduce in this section the different ways to generate these ensembles, and the different ways to quantify the uncertainty of the predictions.

Bootstrap and subsampling Bootstrap [60] is a cornerstone of statistics that allows to compute prediction intervals by generating new training sets from $\mathscr{D}$ by sampling with replacement. For a number B of bootstrap resamples, the dataset $\mathscr{D}^b$ (for $b = 1, \cdots, B$) is generated by sampling n samples from $\mathscr{D}$ with replacement. The goal of this procedure is to simulate the resampling of a new dataset, which is in general not possible in practice. Once the datasets $\mathscr{D}^b$ are obtained, one then fits estimators f^b for each dataset, giving us an empirical distribution of estimators. Alternatively, one can subsample the original dataset and generate $\mathscr{D}^b$ of size $m < n$, with $r = m/n$ called the *subsampling ratio*. Finally, the Jackknife[200] is a special case of subsampling, where one removes one sample from the dataset, hence generating n datasets of size $n-1$. An adaptation of Jackknife, called Jackknife+ [19], has been shown to produce prediction intervals with coverage guarantees under weak assumptions on the data generating process.

We will analyze the performance of bootstrap, subsampling and the jackknife in chapter 5. Our main interest will be the variance of the estimator induced by the resampling procedure. Considering a test sample $\mathbf{x}$, we will be interested in the variance of the prediction

$$\mathrm{Var}_b\left(\hat{f}^b(\mathbf{x})\right) = \frac{1}{B}\sum_{b=1}^{B}\left(\hat{f}^b(\mathbf{x}) - \bar{f}(\mathbf{x})\right)^2$$

where $\bar{f}(\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^b(\mathbf{x})$. Because the estimators will be $\hat{f}^b(\mathbf{x}) = \theta^{b\top}\mathbf{x}$, this variance will be

$$\mathrm{Var}_b\left(\hat{f}^b(\mathbf{x})\right) = \frac{1}{B}\sum_{b=1}^{B} \|\theta^b\|^2 - \|\frac{1}{B}\sum_{b=1}^{B} \theta^b\|^2$$

which requires to compute the scalar product $\theta^{b\top}\theta^{b'}$ between two estimators trained on different resampled datasets. The key in our computations will be to introduce *sampling weights* $\left(p_i^b\right)_{i=1}^n$ such that p_i^b indicates how many times the sample $(\mathbf{x}_i, y_i)$ is sampled in the dataset $\mathscr{D}^b$. As a consequence, the datasets will be rewritten as $\mathscr{D}^b = \left(\mathbf{x}_i, y_i, p_i^b\right)_{i=1}^n$. Also, the empirical risk (1.7) can be rewritten as

$$\mathscr{L}^b(\theta) = -\left(\sum_{i=1}^{n} p_i^b \log p_{\mathrm{eb}}\left(y_i|\theta^\top\mathbf{x}_i\right) + \log p_{\star,\mathrm{eb}}(\theta)\right) \tag{1.15}$$

In the case of bootstrap, $\left(p_i^b\right)_i$ is a random vector sampled from a multinomial distribution, which converges in probability to a Poisson distribution when $n \to \infty$. Finally, to compute the correlation between two estimators, we will consider the distribution

$$p\left(\theta^1, \theta^2\right) \propto e^{-\beta(\mathscr{L}^1(\theta^1)+\mathscr{L}^2(\theta^2))} \tag{1.16}$$

that is the joint posterior of two estimators trained on $\mathscr{D}^1, \mathscr{D}^2$.

Ensemble of random features One of our goals will be to analyze the performance of deep ensembles introduced earlier. These ensembles are made of several deep neural networks trained on the same data, but whose non-convexity induces different learnt features. To model this characteristic in a tractable way, we will use ensembles of random features models [152]. Consider a set of K random features $\left(\varphi^j(\mathbf{x})\right)_{i=1}^K$ as defined in eq. (1.13). Then, mimizing the empirical risk

$$\mathscr{L}^j\left(\theta^j\right) = \sum_{i=1}^{n} \ell\left(y_i, \theta^{j\top}\varphi^j(\mathbf{x}_i)\right) + r\left(\theta^j\right)$$

yields an ensemble of estimators whose variance can be used as UQ metric. We focus on this model in chapter 6.

1.2 Uncertainty quantification

1.2.1 Why quantify uncertainty ?

The question of uncertainty quantification in machine learning is a central one. In some applications such as financial forecast or medical diagnosis, having an accurate estimation of the uncertainty in our predictions is as important (if not more important) than having a good accuracy. In the context of adversarial attacks, having a good uncertainty estimate can be useful to

detect adversarial samples [81]. Likewise, uncertainty quantification is crucial for reinforcement learning and multiarmed bandist, where one needs to balance exploration and exploitation. One then needs to quantify the uncertainty in the reward of each arm, so that one can explore the arms with the highest uncertainty and exploit the arms with the highest expected reward.

Yet, there are a lot of ways to define "uncertainty" (aleatoric, epistemic, etc.), and many methods have been developed quantify it in modern ML : these methods range from Bayesian inference (variational inference [199], Monte-Carlo dropout [86]), to resampling method [60], to averaging the iterates of a learned model during the iterations of gradient descent [157]. However, theoretical guarantees for all these methods are often lacking, and their performance is not poerfectly understood. Moreover, in some applications such as generative models, it is unclear how to even quantify the uncertainty in generated predictions, for example with Large Language Models (LLM) [176, 111].

In this PhD thesis, we will analyse some methods used by practicioners by using toy models that are analytically tractable. Our goal is to analyse the performance of these UQ methods in different scenarios and derive insights that will guide their use in practice. In particular, we identify regimes where these methods are suboptimal and propose, in some cases, alternatives to improve UQ in high-dimensional statistics.

1.2.2 Confidence in classification

In the context of classification, a natural way to quantify uncertainty is through *confidence scores* : given K posssible classes, the estimator assigns a confidence score p_c to each class $c = 1, \cdots, K$ that represents the probability that the input $\mathbf{x}$ belongs to class c. In the context of neural networks, these confidence scores are obtained by applying the softmax function to the output of the last-layer :

$$p_c = \frac{e^{z_c}}{\sum_{c'} e^{z_{c'}}}, \mathbf{z} = W_L \varphi(\mathbf{x}) \tag{1.17}$$

where the z_c are the pre-activations logits of the last-layer, $\varphi(\mathbf{x})$ is the feature map learned by the model and W_L are the weights of the last-layer. The predicted class is the one with the highest confidence score, i.e. $\hat{y} = \text{argmax}_c p_c$, and the confidence is $\hat{f}(\mathbf{x}) = \max p_c$.

Calibration To evaluate the quality of the confidence score, a popular metric is the *calibration* : for $0 \leqslant p \leqslant 1$, the calibration Δ_p at level p compares the accuracy and confidence of our estimator when said confidence is equal to p.

$$\Delta_p = p - \mathbb{P}\left[y = y_\star | \hat{f}(\mathbf{x}) = y_\star\right] \tag{1.18}$$

where here $\hat{f}(\mathbf{x}) \in [0,1]$ is the confidence and y is the predicted class. Intuitively, one aims to achieve $\Delta_p = 0$ for all p, which means that the confidence score is equal to the accuracy of the model, in which case the model is said to be *calibrated*. Note that having a calibrated model is orthogonal to having good accuracy. For example, for binary classification, consider the completely random model that assigns $p_c = 0.5$ for all classes. This model is perfectly calibrated, but has an accuracy of 50%.

Temperature scaling A popular way to improve calibration in pre-trained models is temperature scaling (TS), introduced in [101] and that we study in chapter 3 and chapter 4. The principle is the following : in neural networks, the confidence scores p_c are defined by

$$p_c = \frac{e^{z_c}}{\sum_{c'} e^{z_{c'}}} \tag{1.19}$$

where the z_c are the logits of the model. The idea of TS is to rescale the logits by a temperature T that is learned on a validation set, so that the new confidence scores are given by

$$p_c = \frac{e^{z_c/T}}{\sum_{c'} e^{z_{c'}/T}} \tag{1.20}$$

so that the predicted class is the same but T modulates the confidence of the model. In addition to temperature scaling, other methods such as isotonic regression [245] or Platt scaling [198] have been proposed to improve the calibration of pretrained neural networks. Other approaches consist in calibrating the models during training [179, 150] or through data augmentation [231].

1.2.3 Prediction variance

Whereas calibration is restricted to classification tasks, prediction variance can be applied for both regression and clasification. The idea is to quantify the uncertainty in the prediction by computing the variance of the predictions of an ensemble of estimators, and the central question is how to generate this ensemble.

A popular way that has been used in the context of neural networks is *deep ensembles* [143] : because of their non-linearity, the empirical risk that is minimized by these models is non-convex in their parameters. Because of this, the empirical risk has many local minima, and one can generate an ensemble of estimators by training several models with different initializations. Alternatively, Monte-Carlo dropout [86] was introduced to generate various predictions using only one model. The idea is to randomly drop neurons during training and testing. By dropping neurons, one can generate an ensemble of predictions from a single neural network. This method is computationally efficient and has been shown to produce good uncertainty estimates in practice.

1.2.4 Prediction sets

Prediction sets are a way to quantify uncertainty in classification and regression tasks. It consists in producing a set $\mathscr{S}(\mathbf{x})$ of possible values for the prediction $\hat{f}(\mathbf{x})$, such that the true value y is contained in this set with a user-defined probability $1-\kappa$, called the coverage. A prediction set achieves marginal coverage if

$$\mathbb{P}_{\mathscr{D},\mathbf{x}}\left(y \in \mathscr{S}(\mathbf{x})\right) \geqslant 1-\kappa \tag{1.21}$$

The coverage is said *marginal* if it holds on average for a random sample $\mathbf{x}$. On the other hand, a harder condition is *conditional coverage*, which requires that the coverage holds for each sample $\mathbf{x}$, i.e.

$$\mathbb{P}_{\mathscr{D}}\left(y \in \mathscr{S}(\mathbf{x}) \,|\mathbf{x}\right) \geqslant 1-\kappa \tag{1.22}$$

In the context of regression, a common way to produce prediction sets is to use the empirical quantiles of the predictions of an ensemble of estimators : if one has B estimators $\hat{f}^b$ trained on different datasets, one can obtain a prediction interval with target coverage $1-\kappa$ by taking the empirical quantiles at level $\kappa/2$ and $1-\kappa/2$ of the predictions $\hat{f}^b(\mathbf{x})$.

Prediction sets with conformal prediction In general, prediction sets produced with an ensemble of estimators as described in the previous paragraph do not have any coverage guarantee. In this context, conformal prediction (CP) is a fairly recent framework [235] developed to produce valid prediction intervals with a user-defined coverage. This framework is agnostic and can be applied on any estimator (neural networks, random forests, etc.) under weak assumptions. In chapter 7, we will show how Approximate Message Passing can be used to produce valid prediction sets with conformal prediction. The two assumptions we will need are the following :

- The data is exchangeabe, meaning that the joint distribution of the data is invariant under permutations. This assumption is weaker than the i.i.d assumption
- The estimator we consider is symmetric, meaning that the estimator $\hat{f}(\mathbf{x})$ is invariant under permutations of the dataset $\mathscr{D}$.

The application of conformal prediction to scenarios outside of these two assumptions is a vast and active topic. For instance, it has been applied to time series [246],

1.3 The high-dimensional toolblox

In this section, we'll introduce the mathematical tools that are used throughout this thesis to analyze the methods presented in the previous sections : approximate-message passing and the

replica method. These two tools are ubiquitous in statistical physics and are used to study (among other topics) inference problems [188], error-correcting codes [124], matrix factorization [126] or generative models [57]. For the sake of concreteness, we will focus on the empirical-Bayes posterior (1.8) :

$$p_{\text{eb}}\left(\theta|\mathscr{D},\beta\right) = \frac{1}{\mathscr{Z}(\beta)} e^{-\beta\mathscr{L}(\theta)} \tag{1.23}$$

1.3.1 Characterizing the performance with overlaps

Throughout this manuscript, our analysis will be based on the crucial fact that the performance of the different methods will be characterized by the statistics of the local fields

$$\left(\theta_\star{}^\top\mathbf{x}, \theta^\top\mathbf{x}\right) \tag{1.24}$$

These sufficient statistics, called *overlaps* are low-dimensional and computing their limiting value in the high-dimensional regime will be at the heart of our work. In the case of eq. (1.23), the overlaps of interest are the following :

$$\rho = \frac{1}{d}\|\theta_\star{}^\top\|^2 \tag{1.25}$$

$$m = \frac{1}{d}\theta_\star{}^\top\mathbb{E}_{p_{\text{eb}}}[\theta] \tag{1.26}$$

$$q = \frac{1}{d}\|\mathbb{E}_{p_{\text{eb}}}[\theta]\|^2 \tag{1.27}$$

$$v = \frac{1}{d}\text{Tr}\left(\underset{p_{\text{eb}}}{\text{Cov}}(\theta)\right) \tag{1.28}$$

$$\tag{1.29}$$

A key assumption is that the $\mathbf{x}$ are to be i.i.d Gaussian with variance $1/d$. Indeed, the vector $\left(\theta_\star{}^\top\mathbf{x}, \hat{\theta}^\top\mathbf{x}\right)$ is Gaussian with 0 mean and covariance

$$\begin{pmatrix} \rho & m \\ m & q \end{pmatrix}$$

Alternatively, if one is interested in the prediction variance of the estimator, one computes

$$\begin{aligned}\text{Var}_{p_{\text{eb}}}(\hat{f}(\mathbf{x})) &= \text{Var}_{p_{\text{eb}}}(\theta^\top\mathbf{x}) = \mathbf{x}^\top \underset{p_{\text{eb}}}{\text{Cov}}[\theta](\theta)\,\mathbf{x} \\ &= \text{Tr}\left(\underset{p_{\text{eb}}}{\text{Cov}}[\theta]\mathbf{x}\mathbf{x}^\top\right) \to_\infty \text{Tr}\left(\underset{p_{\text{eb}}}{\text{Cov}}[\theta]\mathbb{E}_{\mathbf{x}}\mathbf{x}\mathbf{x}^\top\right) \to_\infty \text{Tr}\left(\underset{p_{\text{eb}}}{\text{Cov}}[\theta]\right) = v\end{aligned}$$

Hence, the overlap v completely characterizes the variance of the prediction variance in high-dimensions. Similarly, in the chapters 2, 3, 4 we will characterize the calibration (1.18) in terms of the overlaps m, q, v, and in Chapter 7, we will compute the size of prediction intervals in terms

of these same overlaps.

Note that in chapters 3 and 6, the local fields will instead be defined as

$$\left(\theta_\star{}^\top \mathbf{x}, \theta^\top \varphi(\mathbf{x})\right) \tag{1.30}$$

where the $\varphi(\mathbf{x})$ are the random features. Using the Gaussian Equivalence Theorem [98], studying these random features will be equivalent to studying the equivalent model

$$\left(\theta_\star{}^\top \mathbf{x}, \theta^\top \mathbf{v}\right) \tag{1.31}$$

where the input given to our estimator is a Gaussian vector $\mathbf{v}$ such that the vector $\mathbf{x}, \mathbf{v}$ has the same mean and covariance as $\mathbf{x}, \varphi(\mathbf{x})$.

To summarize, the core of our analysis will be to compute the relevant low-dimensional overlaps to compute exactly the performance of the different UQ methods. To do so, we will use two different formalisms to compute these overlaps : the Approximate Message Passing algorithm and the replica method. In our case, these two methods are equivalent in the sense that they will both yield the same state-evolution equations.

1.3.2 Approximate Message Passing

At its core, the inference problem we are interested in is a problem of inference on a graphical model, where the nodes are the parameters θ and the edges are the samples $(\mathbf{x}_i, y_i)$. Belief propagation (BP) is a well-known algorithm to compute the distribution on the nodes of a graphical model, and is exact in the case of trees [38]. BP consists in iterating a set of messages that represent the marginal distribution of the nodes θ_i. Putting our inference problem aside and considering a distribution with the general form

$$p(\theta) = \prod_a f_i(\theta_{\mu \in N_i}) \times \prod_{\mu=1}^{d} \Psi_\mu(\theta_\mu)$$

where f_i are the factors of the graph and represent the interactions between several nodes, N_i is the indices of the nodes interacting through f_i and Ψ_μ corresponds to the prior. For example, in the context of the distribution (1.23), the factors correspond to the samples i, and $N_i = \{1, \cdots, n\}$: $f_i(\theta) = e^{-\beta \ell(y_i, \theta^\top \mathbf{x}_i)}$, and the priors are $\Psi_\mu \propto e^{-\beta r(\theta_\mu)}$. The messages will read

$$\begin{cases} m_{\mu \to i}(\theta_\mu) & = \frac{1}{z_{\mu \to i}} \Psi_\mu(\theta_\mu) \prod_{j \neq i} m_{j \to \mu}(\theta_\mu) \\ m_{i \to \mu}(\theta_\mu) & = \frac{1}{z_{i \to \mu}} \int \prod_{\nu \neq \mu \in N_i} \mathrm{d}\theta_\mu m_{\nu \to i(\theta_\nu)} f_i(\theta_{\mu \in N_i}) \end{cases} \tag{1.32}$$

However, in the case of dense graphs such as the one that describes the empirical-Bayes posterior, BP is not exact, and is also computationally expensive in high-dimensions : computing $m_{i \to \mu}$

requires computing a $d-1$ dimensional integral. To circumvent these issues, Approximate Message Passing (AMP) is an algorithm that will approximate the computations of BP and compute the posterior distribution defined in eq. (1.8) in an efficient way.

The starting point of AMP is another algorithm called relaxed-Belief Propagation (r-BP). Under the assumptions that the observations $\left((x_{i\mu})_{i=1}^{n}\right)_{\mu=1}^{d}$ are Gaussian i.i.d with variance scaling as $1/d$, one can leverage the central limit theorem : the sum $\theta^\top \mathbf{x}_i$ appearing in the term $f_i(\theta_{\mu\in N_i})$ of eq. (1.32) will be asymptotically distributed as a Gaussian variable with mean and variance $\sum_\mu \hat{\theta}^\top_{\mu\to i} x_{i,\mu}$ and $\sum_\mu v^\top_{\mu\to i} x^2_{i,\mu}$ where $\hat{\theta}_{\cdot\to i}$ and $\mathbf{v}_{\cdot\to i}$ are respectively the mean and variance of the variable θ_μ under the distribution induced by $m_{\mu\to i}$. r-BP will compute these means and variances iteratively, and the computation of the posterior is reduced by computing $O(n\times d)$. In AMP, one can simplify the computations even further and compute an approximation $\hat{\theta}$ (respectively $\mathbf{v}$) of the mean $\hat{\theta}_{\mu\to i}$ (respectively $v_{\mu\to i}$). AMP and r-BP are equivalent in the high-dimensional limit $n,d\to\infty, n/d\to\alpha$, while AMP only iterates $O(V+D)$ quantities.

In the context of empirical risk minimization, the vector $\hat{\theta}$ is then an approximation of the minimizer of the empirical risk such that $\hat{y}=\hat{f}\left(\hat{\theta}^\top\mathbf{x}\right)$, while in the context of Bayesian inference, the mean and the variance $\hat{\theta}_{\text{amp}}, \hat{\mathbf{v}}_{\text{amp}}$ are such that the posterior distribution from eq. (1.8) is approximated by a Gaussian distribution :

$$p\left(y|\mathbf{x}\right)=\int \mathrm{d}z\, p\left(y|z\right)\times e^{-\frac{1}{2}(z-\hat{\theta}_{\text{amp}}{}^\top\mathbf{x})^2/\left(\hat{\mathbf{v}}^\top_{\text{amp}}(\mathbf{x}\otimes\mathbf{x})\right)} \tag{1.33}$$

where the components of $\mathbf{x}\otimes\mathbf{x}$ are the $\mathbf{x}_i^2$.

Denoising and channel functions In our setting of inference task, the graph is dense and all the nodes interact through the likelihood function

$$f_i(\theta)=p(y_i|\theta^\top\mathbf{x}_i)$$

This likelihood induces a distribution

$$\mathcal{Z}_0(y,\omega,V)=\int \mathrm{d}z\, p(y|z)\times e^{-\frac{1}{2V}(z-\omega)^2}=\int \mathrm{d}e^{-\beta\ell(y,z)}\times e^{-\frac{1}{2V}(z-\omega)^2} \tag{1.34}$$

where the second equality is the special case of the empirical-Bayes estimator. This defines a *channel* function

$$g_0(y,\omega,V)=\frac{\partial\log\mathcal{Z}_0}{\partial\omega}(y,\omega,V) \tag{1.35}$$

For empirical risk minimization, taking the limit $\beta\to\infty$ in eq. (1.34) gives a channel of the form

$$g_0(y,\omega,V)=\text{argmin}_z\,\ell(y,z)+\frac{1}{2V}(z-\omega)^2 \tag{1.36}$$

Algorithm 1 AMP algorithm

Input: Data $\mathbf{X} \in \mathbb{R}^{n\times d}$, $\mathbf{y} \in \{-1,1\}^n$
Define $\mathbf{X}^2 = \mathbf{X} \odot \mathbf{X} \in \mathbb{R}^{n\times d}$ and Initialize $\hat{\mathbf{w}}^{t=0} = \mathcal{N}(\mathbf{0}, \sigma_w^2 \mathbf{I}_d)$, $\hat{\mathbf{c}}^{t=0} = \mathbf{1}_d$, $\mathbf{g}^{t=0} = \mathbf{0}_n$.
for $t \leq t_{\max}$ **do**
$\mathbf{V}^t = \mathbf{X}^2 \hat{\mathbf{v}}^t$; $\omega^t = \mathbf{X}\hat{\mathbf{w}}^t - \mathbf{V}^t \odot \mathbf{g}^{t-1}$
$\mathbf{g}^t = f_{\text{out}}(\mathbf{y}, \mathbf{w}^t, \mathbf{V}^t)$; $\partial \mathbf{g}^t = \partial_\omega f_{\text{out}}(\mathbf{y}, \mathbf{w}^t, \mathbf{V}^t)$
$A_\mu^t = -\sum_i x_{i,\mu}^{2\top} \partial (g_0)_\mu^t$; $b_\mu^t = x_{i,\mu}^\top (g_0)_i^t + \mathbf{A}^t \odot \hat{\mathbf{w}}^t$;
$\hat{\mathbf{w}}^{t+1} = f_a(\mathbf{b}^t, \mathbf{A}^t) := (\mathbf{I}_d + \mathbf{A}^t)^{-1}\mathbf{b}^t$; $\quad \hat{\mathbf{v}}^{t+1} = \partial_b f_w(\mathbf{b}^t, \mathbf{A}^t) := (\mathbf{I}_d + \mathbf{A}^t)^{-1}$
end for
Return: Estimators $\hat{\mathbf{w}}_{\text{amp}}, \hat{\mathbf{v}}_{\text{amp}} \in \mathbb{R}^d$

Likewise, the prior term (which is identical for all nodes) induces a *denoising function*

$$f_a(b, A) = \frac{\partial \log \mathcal{Z}_a}{\partial b}$$
$$\text{where} \mathcal{Z}_a = \int \mathrm{d}z \Psi(z) \times e^{-\frac{A}{2}(z - b/A)^2}$$

as for the channel, in the context of ERM where $\beta \to \infty$ with $\Psi(z) \propto e^{-\beta r(z)}$, the denoiser reads

$$f_a(b, A) = \arg\min_z r(z) + \frac{A}{2}(z - \frac{b}{A})^2 \tag{1.37}$$

Notice that, from eq. (1.36) and eq. (1.37), the minimizers in these equations are unique because we assume the loss and regularization terms to be convex.

In algorithm 1, we write down the AMP that computes the mean and variance for the distribution corresponding to the channel g_0 and denoiser f_a.

State-evolution equations of AMP An important property of AMP is its state-evolution (SE) equations, that allow to compute its performance in high-dimensions. These equations heavily leverage the assumption that the inputs are i.i.d Gaussian (with variance scaling as $1/d$) and that the labels are generated by a teacher. In this manuscript, we will compute the SE equations of AMP by computing those of SE that is equivalent in high-dimensions.

In particular, one can show that the asymptotic distribution of the estimator is

$$\hat{\theta}_i \sim f_a\left(\theta_{\star i}\hat{m} + \left(\hat{Q}\right)^{1/2}\xi, \hat{\mathbf{V}}\right) \quad \forall i \in \{1, \ldots, d\}, \tag{1.38}$$

where $\xi \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_B)$ and the overlaps $m, q, V, \hat{m}, \hat{q}, \hat{V}$ are the unique solutions of the equations

$$\begin{cases} m &= \mathbb{E}_{\theta_\star,\xi}\left[f_a\left(\hat{m}\theta_\star + \sqrt{\hat{q}}\xi, \hat{V}\right)\theta_\star\right] \\ Q &= \mathbb{E}_{\theta_\star,\xi}\left[f_a\left(\hat{m}\theta_\star + \sqrt{\hat{q}}\xi, \hat{V}\right)^2\right] \\ V &= \mathbb{E}_{\theta_\star,\xi}\left[\partial_{\mathbf{b}} f_a\left(\hat{m}\theta_\star + \sqrt{\hat{q}}\xi, \hat{V}\right)\right] \end{cases} \tag{1.39}$$

where $\xi \sim \mathcal{N}(0, 1)$, and

$$\begin{cases} \hat{m} &= \alpha \mathbb{E}_{(z,\omega)}\left[\partial_z g_0(\varphi_0(z), \omega, V)\right] \\ \hat{Q} &= \alpha \mathbb{E}_{(z,\omega)}\left[g_0(\varphi_0(z), \omega, V)^2\right] \\ \hat{V} &= -\alpha \mathbb{E}_{(z,\omega)}\left[\partial_\omega g_0(\varphi_0(z), \omega, V)\right] \end{cases}, \tag{1.40}$$

1.3.3 Replica method

In chapter 3, we will analyze the performance of different estimators in the context of random features. On top of the state-evolution equationsd of AMP, we will use the replica method [171, 247] to derive their performance in high-dimensions. The goal of this method coming from statistical physics is to compute the normalization constant $\mathcal{Z}(\beta)$ in the posterior distribution (1.8) :

$$\mathcal{Z}(\beta) = \int \mathrm{d}\theta e^{-\beta \mathscr{L}(\theta)} \tag{1.41}$$

Note that in the context of Bayesian inference, $\mathcal{Z}$ is related to the *evidence* that is used to compare different models : the evidence represents the probability that $\mathcal{D}$ was generated by a model $\mathcal{M}$, under the assumption that the model is correct. In chapter 3, we will analyze the performance of evidence maximization, which consists in selecting the model with highest evidence and is related to the Bayesian Information Criterion [104]. To compute $\mathcal{Z}$, it is convenient to introduce the cumulant-generating function, also called *free energy*, defined as

$$f = \frac{1}{\beta} \lim_{d\to\infty} \frac{1}{\beta d} \mathbb{E}_{\mathcal{D}}\left[\ln \mathcal{Z}\right] \tag{1.42}$$

where the expectation is over the probability distribution of the dataset $\mathcal{D}$. The replica trick consists in the equality

$$\mathbb{E}_{\mathcal{D}}\left[\ln \mathcal{Z}\right] = \lim_{r\to 0} \frac{\mathbb{E}_{\mathcal{D}}\left[\mathcal{Z}^r\right] - 1}{r} \tag{1.43}$$

We then only need to compute $\mathbb{E}_{\mathcal{D}}\left[\mathcal{Z}^r\right]$, which corresponds to the partition function of the distribution for r copies of the original estimator, trained on the same dataset. A subtelty of the replica method is that we will compute an expression of $\mathbb{E}_{\mathcal{D}}\left[\mathcal{Z}^r\right]$ for an integral number $r \in \mathbb{N}$ of copies $\theta^1, \cdots, \theta^r$ and take the limit $r \to 0$. In appendix B, we will see that some algebra yields an

expression of the form

$$\mathbb{E}_{\mathscr{D}}\left[\mathscr{Z}^r\right] = \int \mathrm{d}\rho \mathrm{d}\hat{\rho} \prod_{a<b} \mathrm{d}Q^{ab} \mathrm{d}\hat{Q}^{ab} e^{d\Phi^{(r)}(\rho,\hat{\rho},\mathbf{Q},\hat{\mathbf{Q}})}$$

where the matrix $Q^{ab} \in \mathbb{R}^{r\times r}$ describes the correlation between the different replicas, and the matrix $\hat{Q}^{ab}$ stems from the Fourier transform of the Diract delta enforcing the contraint $Q^{ab} = \frac{1}{d}\theta^{a\top}\theta^b$. In the limit $d \to \infty$, the value of $\mathbb{E}_{\mathscr{D}}\left[\mathscr{Z}^r\right]$ will be given by the maximum of the function $\Phi^{(r)}$, and will thus require to optimize over matrices of size $r \times r$. At this point, an important simplification will be to assume *replica symmetry*, meaning the matrices will have the form

$$Q_{aa} = q, Q_{ab} = q', \hat{Q}_{aa} = \hat{q}, \hat{Q}_{ab} = \hat{q}'$$

In the limit $r \to 0$, The replica symmetric ansatz will give us a free energy of the form [94, 13]

$$\begin{cases} f & = \mathbf{extr}_{m,q,v,\hat{m},\hat{q},\hat{v}} - m\hat{m} + \frac{1}{2}\left(q\hat{V} - \hat{q}V + \hat{V}V\right) + \Psi_w + \frac{n}{d}\Psi_y \\ \Psi_w & = \mathbb{E}_{\xi,w_\star} \log \mathscr{Z}_a(\hat{m}w_\star + \hat{q}\xi, \hat{V}) \\ \Psi_y & = \mathbb{E}_{\xi\sim\mathcal{N}(0,q)}\left[\int \mathrm{d}y \mathscr{Z}_\star(y, {}^{m}/_{q}\xi, \rho - {}^{m^2}/_{q}) \log \mathscr{Z}_0(y,\xi,V)\right] \end{cases} \tag{1.44}$$

Where the function

$$\mathscr{Z}_\star(y,\omega,V) = \int \mathrm{d}z p(y|z) e^{-\frac{1}{2V}(z-\omega)^2}$$

is the distribution induced by the teacher's likelihood. Note that the term Ψ_w (respectively Ψ_y) is closely related to the denoising function (respectively channel function) in AMP. In fact,in chapter 3 we will see that these two methods will be equivalent in and yield the same state-evolution equations to compute the overlaps m, q, V.

1.4 Contributions

This thesis results from the work of 5 published or pre-published work:

1. *Theoretical characterization of uncertainty in high-dimensional linear classification*, Clarté, Loureiro, Krzakala, Zdeborová
   Published in *Machine Learning, Science and Technology*, Volume 4, Number 2
   Summary: Existing work [13] has shown that, in classification tasks, logistic regression can approach very closely the accuracy of the Bayes posterior with proper regularization. This paper asks the question if logistic regression can have the same uncertainty estimation as Bayes. To answer this question, we use the state-evolution equations of AMP to compute the correlation between these two estimators, and show that even at optimal regularization, there is a significant discrepency in both uncertainty estimations.

Contribution: I derived Theorems 2.3.2 and 2.3.3, and run the simulations in the numerical experiments.
The paper corresponds to chapter 2.

2. *On double-descent in uncertainty quantification in overparametrized models*, Clarté, Loureiro, Krzakala, Zdeborová
Published in *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, 7089-7125, 2023
Summary: The *double-descent* is a well-known phenomenon where the test performance of neural networks improves by increasing their parameters count even interpolation, contradicting classical learning theory. This paper answers the question of whether similar behavior occurs for the calibration curve of neural networks. For this, we use a random feature model as a toy model of neural networks. First, we exhibit a double descent in the calibration curve for unregularized logistic regression. Second, we show that empirical Bayes methods perform no better than frequentist approaches, when the latter is combined with post-training calibration.
Contribution: I derived the state-evolution equations in section 3.3.1 to analyze the different estimators, and conducted the experiments.
The paper corresponds to chapter 3.

3. *Expectation consistency for calibration of neural networks*, Clarté, Loureiro, Krzakala, Zdeborová
Published in *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, 443-453, 2023
Summary: Temperature scaling (TS) is a popular algorithm to calibrate the predictions of neural networks after training in an efficient way. It consists in rescaling the confidence by a scalar parameter that is determined by cross-validation. In this paper, we propose an alternative to TS, called Expectation Consistency (EC), where the temperature is not chosen as the minimum of a validation loss but by matching the accuracy and average confidence on the validation set. We evaluate our method on image classification tasks and observe that our method perfoms similarly as TS and is more robust to perturbations in the labels. We analyze both methods in a T-S setting where
Contribution: I implemented EC and conducted the numerical experiments.
The paper corresponds to chapter 4.

4. *Analysis of Bootstrap and Subsampling in High-dimensional Regularized Regression*
, Clarté, Vandenbroucque, Dalle, Loureiro, Krzakala, Zdeborová
Published in *Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence*,787-819, 2024 (Spotlight).
Summary: Resampling techniques like the bootstrap or subsampling are a popular way to estimate uncertainty and the variability of estimators in classical statistics, but have been observed to fail in high-dimensions. In this paper, we analyze these resampling methods for empirical risk minimizers in the overparametrized regime and analyze the impact of

regularization on their performance. We observe that even at optimal regularization, these methods fail to capture the true variance and bias of the estimator, making them inadapted in high-dimensions.
Contribution: I derived the AMP algorithm and its state-evolution to analyze the resampling methods. With G. Dalle, I conducted the numerical experiments used to produce the plots.
The paper corresponds to chapter 5.

5. *Building Conformal Prediction Intervals with Approximate Message Passing*, Clarté, Zdeborová
Published in *arxiv preprint arxiv:2410.16493*
Summary: Full conformal prediction (FCP) is a framework that allows to build prediction sets with guaranteed coverage with minimal assumptions on the data, and applicable to a wide range of estimators. However, FCP is computationally expensive, making it hardly applicable in practice. A line or work is dedicated to accelerating the computations of FCP in an exact or approximate way. In this paper, we apply AMP to accelerate the computations of FCP and introduce a new algorithm, called Taylor-AMP . Our method provides guaranteed coverage under the standard assumptions of FCP and its state-evolution equations allow us to benchmark FCP in high-dimensions in the T-S setting.
Contribution: I implemented the algorithm, performed the experiments on synthetic and real data, and applied the state-evolution equations of AMP to analyze the behaviour of Taylor-AMP in high-dimensions.
The paper corresponds to chapter 7.

In addition, chapter 6 contains an extension of chapter 5 to overparametrized models by studying random features. We derive state-evolution equations that allow to analyze the impact of overparametrization on uncertainty quantification in the context of ensembling.

2 Bayesian and frequentist uncertainty in classification

Authors : Lucas Clarté, Bruno Loureiro, Florent Krzakala, Lenka Zdeborová

2.1 Introduction

In this chapter, we provide an exact characterisation of uncertainty for high-dimensional classification of data with Gaussian covariates and probit labels. There are two main sources of uncertainty in this model – the more explicit is the noise level parametrizing the probit function, then there is the uncertainty coming from the fact that learning is done from a limited number of samples. Uncertainty estimation in classification problems aims to compute the probability that a given new sample has one of the labels. The most likely label is then typically chosen for the prediction of the new labels, but the probability itself is of our interest here. We stress that we are interested in the uncertainty sample-wise, i.e. for every given sample, not on average. We address questions such as: a) How does the uncertainty of the logistic classifier compare with the actual Bayesian uncertainty when learning with a limited amount of data? b) How do these two uncertainty measures compare with the intrinsic model uncertainty due to the noise in the data-generating process?

The key player in our analysis will be the Bayesian estimator of uncertainty corresponding to the probabilities of labels for new samples computed by averaging over the posterior distribution. Although in general computing the Bayesian estimator from posterior sampling can be prohibitively computationally costly in high dimensions, we show that in the present model, it can be efficiently done using a tailored approximate message passing (GAMP) algorithm [27, 202]. Leveraging tools from the GAMP (Generalized Approximate Message Passing) literature and its state evolution, we provide an asymptotic characterisation of the joint statistics between the minimiser of the logistic loss, the optimal Bayesian estimator over the data and the oracle estimator. This allows us to provide quantitative answers to questions a) & b) above, and to study how uncertainty estimation depends on the parameters of the model, such as the regularisation, size of the training set and noise.

Main contributions – The main contributions in this paper are:

- It is well known that the optimal Bayesian classifier for a data model with Gaussian i.i.d. covariates and probit labels is well approximated by the generalized approximate message passing (AMP) algorithm [22, 115]. We extend these results by showing that GAMP also provides an exact sample-wise estimation of the Bayesian uncertainty when $d \to \infty$.
- We provide an exact asymptotic description of the joint statistics between the uncertainty of the oracle, and the one estimated by the Bayes-optimal and logistic classifiers for the aforementioned data model. This allows us to compare these uncertainties to each other. Comparing the oracle and Bayes optimal we quantify the uncertainty coming from the limited size of the dataset. Comparing Bayesian and logistic classifiers allows us to quantify the under- or overconfidence of the latter.
- We derive an asymptotic expression of the calibration for the Bayesian and logistic classifiers. In particular, we show that the Bayesian estimator is calibrated. For the logistic classifier, our expression allows us to characterize the influence of various parameters on under- or overconfidence of the logistic classifier.
- We quantify the role played by the ℓ_2-regularization on uncertainty estimation. In particular, we compare cross-validation with respect to the optimisation loss (logistic) with cross-validation with respect to the $0/1$ error.

Related work – Measures of uncertainty: Measuring uncertainty in neural networks is a challenging problem with a vast literature proposing both frequentist and Bayesian approaches [2]. On the frequentist side, various algorithms have been introduced to evaluate and improve the calibration of machine learning models. Some of them, such as isotonic regression [245], histogram binning [244], Platt scaling [198] or temperature scaling [101] are applied to previously trained models. Other approaches aim to calibrate models during training, using well-chosen metrics [179, 150], through data augmentation [231] or using the iterates of the optimiser [157]. Alternatively, different authors have proposed uncertainty measures based on Bayesian estimates [166, 240]. This includes popular methods such as Bayesian dropout [86, 133], deep ensembles [143, 160, 150] and variational inference [199], Laplace approximation [136, 61] and tempered posteriors [4, 5, 127] to cite a few. Finally, some works based on conformal inference [212] are concerned with providing non-asymptotic and distribution-free guarantees for the uncertainty [9, 102].

Exact asymptotics: Our theoretical analysis builds on series of developments on the study of exact asymptotics in high-dimensions. The generalised approximate message passing (GAMP) algorithm and the corresponding state evolution equations appeared in [202, 115]. Exact asymptotics for Bayesian estimation in generalised linear models was rigorously established in [22]. On the empirical risk minimisation side, exact asymptotics based on different techniques, such as Convex Gaussian Min-Max Theorem (GMMT) [220, 64, 222, 13, 172, 178, 153, 146], Random

Matrix Theory [158], GAMP [96, 154] and first order expansions [33] have been used to study high-dimensional logistic regression and max-margin estimation.

Uncertainty & exact asymptotics: An early discussion on the variance of high-dimensional Bayesian linear regression has been appeared in [165, 163, 41]. Calibration has been studied in the context of high-dimensional unregularised logistic regression in [16], where it was shown that the logistic classifier is systematically overconfident in the regime where number of samples is proportional to the dimension. An equivalent result for regression was discussed in [17], where it was shown that quantile regression suffers from an under-coverage bias in high-dimensions. While [16] is the closest to the present paper, we differ from their setting in three major ways. First, they consider the behavior of unpenalized logistic regression, while we study the effect of ℓ_2 regularization on uncertainty. Second, we compute the full joint distribution of the prediction for the oracle, the empirical risk minimizer and the Bayes optimal estimator, while [16] focus the discussion on the calibration of the empirical risk minimizer with respect to the oracle only. Lastly (and less importantly), [16] considers logit data, while we consider a probit data model.

Notation – Vectors are denoted in bold. $\mathcal{N}(\mathbf{x}|\mu,\Sigma)$ denotes the Gaussian density. $\odot$ denotes the (component-wise) Hadamard product. $\mathbf{1}(A)$ denotes the indicator on the set A.. For any differentiable function $f:\mathbb{R}\to\mathbb{R}$, f' will denote its first derivative.

2.2 Setting

The data model – Consider a binary classification problem where n samples $(\mathbf{x}^\mu, y^\mu)\in\mathbb{R}^d\times\{-1,1\}$, $\mu=1,\cdots,n$ are independently drawn from the following probit model:

$$f_\star(\mathbf{x}) := \mathbb{P}(y^\mu=1|\mathbf{x}^\mu) = \sigma_\star\left(\frac{\theta_\star{}^\top\mathbf{x}^\mu}{\tau}\right), \tag{2.1}$$

$$\mathbf{x}^\mu\sim\mathcal{N}(\mathbf{0},{}^1\!/_d\mathbf{I}_d),\quad \theta_\star\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d) \tag{2.2}$$

where $\sigma_\star(x)={}^1\!/_2\,\mathrm{erfc}(-{}^x\!/_{\sqrt{2}})$ and $\tau\geq 0$ parametrises the noise level. Note that the probit model is equivalent to generating the labels via $y^\mu=f_0(\theta_\star{}^\top\mathbf{x}^\mu+\tau\xi^\mu)$ with $\xi^\mu\sim\mathcal{N}(0,1)$ and $f_0(x):=\mathrm{sign}(x)$. In the following we will be referring to the function $f_\star(\mathbf{x})$ or to its parameters $\theta_\star$ as the *teacher*, having in mind the teacher-student setting from neural networks. We will refer to $f_\star(\mathbf{x})$ as the *oracle uncertainty* as it takes into account only the noise in the label-generating process, but it does not take into account uncertainty coming from the limited size of the training dataset.

Note that our discussion could be straighforwardly generalized to a generic prior distribution $\theta_\star\sim P_{\theta_\star}$. However, our goal in this work is to provide a fair comparison between Bayesian estimation and empirical risk minimization. Indeed, ERM does not assume any information on the components of $\theta_\star$, and a fair comparison is to consider the maximum entropy Gaussian prior.

Given the training data $\mathcal{D}=\{(\mathbf{x}^\mu,y^\mu)\}_{\mu=1}^n$ and a test sample $\mathbf{x}\sim\mathcal{N}(\mathbf{0},{}^1\!/_d\mathbf{I}_d)$, the goal is to find a

(probabilistic) classifier $\mathbf{x} \mapsto \hat{y}(\mathbf{x})$ minimizing the 0/1 test error

$$\varepsilon_g = \mathbb{E}_{(\mathbf{x},y)}\mathbb{P}\left(\hat{y}(\mathbf{x}) \neq y\right). \tag{2.3}$$

Considered classifiers – We will focus on comparing two probabilistic classifiers $\hat{f}(\mathbf{x}) = \mathbb{P}(y = 1|\mathbf{x})$. The first is the widely used logistic classifier: $\hat{f}_{\text{erm}}(\mathbf{x}) = \sigma(\hat{\mathbf{w}}_{\text{erm}}^\top \mathbf{x})$ where $\sigma(x) = (1+e^{-x})^{-1}$ is the logistic function and the weights $\hat{\mathbf{w}} \in \mathbb{R}^d$ are obtained by minimising the following (regularised) empirical risk:

$$\hat{\mathscr{R}}_n(\mathbf{w}) = \frac{1}{n}\sum_{\mu=1}^{n} \log\left(1 + e^{-y^\mu \mathbf{w}^\top \mathbf{x}^\mu}\right) + \frac{\lambda}{2}||\mathbf{w}||_2^2. \tag{2.4}$$

Using $\hat{f}_{\text{erm}}(\mathbf{x})$ as a measure of uncertainty is not considered very principled. Nevertheless, it is arguably the most commonly used measure to give a rough idea of how confident is the neural network prediction for a given sample.

The second estimator we investigate is the statistically optimal Bayesian estimator for the problem, which is given by:

$$\begin{aligned}\hat{f}_{\text{bo}}(\mathbf{x}) = \mathbb{P}_{\text{BO}}(y = 1|\mathbf{x}) &= \int_{\mathbb{R}^d} \mathrm{d}\mathbf{w}\; P(y = 1|\mathbf{x}^\top \mathbf{w}) P(\mathbf{w}|\mathscr{D}) \\ &= \int_{\mathbb{R}^d} \mathrm{d}\mathbf{w}\; \sigma_\star\left(\frac{\mathbf{w}^\top \mathbf{x}}{\tau}\right) P(\mathbf{w}|\mathscr{D}),\end{aligned} \tag{2.5}$$

where the posterior distribution $P(\mathbf{w}|\mathscr{D})$ given the training data $\mathscr{D} = \{(\mathbf{x}^\mu, y^\mu)\}_{\mu=1}^n$ is explicitly given by:

$$P(\mathbf{w}|\mathscr{D}) = \frac{1}{\mathscr{Z}(\tau)} \prod_{\mu=1}^{n} \sigma_\star\left(y^\mu \frac{\mathbf{w}^\top \mathbf{x}^\mu}{\tau}\right) \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{I}_d), \tag{2.6}$$

for a normalisation constant $\mathscr{Z}(\tau) \in \mathbb{R}$. The Bayes-optimal (BO) estimator $\hat{f}_{\text{bo}}(\mathbf{x})$ provides the perfect measure of uncertainty that takes into account both the noise in the data generation and the finite number of samples in the training set. The traditional drawback of course is that it assumes the knowledge of the value τ and other details of the data-generating model.

Uncertainty and calibration – The main purpose of this manuscript is to characterise how the intrinsic uncertainty of the probit model compares to both the Bayesian and logistic confidences/uncertainties in the high-dimensional setting where the number of samples n is comparable to the dimension d. In this case, the limited number of samples is a source of uncertainty comparable in magnitude to the noise level τ. To define what is uncertainty in our context, note that the *confidence functions* $\hat{f}(\mathbf{x}) = \mathbb{P}(y = 1|\mathbf{x})$ defined above give the probability that the label is $y = 1$ (with the label prediction commonly given by thresholding this function). In mathematical terms, we aim at characterising the correlation between the oracle, Bayesian and logistic confidences, as

parametrised by the joint probability density:

$$\rho(a,b,c) = \mathbb{P}_{\mathscr{D},\mathbf{x}}\big(f_\star(\mathbf{x}) = a, \hat{f}_{\rm bo}(\mathbf{x}) = b, \hat{f}_{\rm erm}(\mathbf{x}) = c\big). \tag{2.7}$$

Similarly, we will note $\rho_{\star,\rm erm}(a,c) = \mathbb{P}(f_\star = a, \hat{f}_{\rm erm} = c)$, $\rho_{\rm bo,erm}(b,c) = \mathbb{P}(\hat{f}_{\rm bo} = b, \hat{f}_{\rm erm} = c)$ and $\rho_{\star,\rm bo}(a,b) = \mathbb{P}(f_\star = a, \hat{f}_{\rm bo} = b)$. These densities correspond to ρ summed over $\hat{f}_{\rm bo}$, $f_\star$ and $\hat{f}_{\rm erm}$ respectively. Here the sample $\mathbf{x}$ is understood as any sample from the test set, on which the confidence/uncertainty is evaluated. It is important that Eq. (2.7) is defined for the same sample $\mathbf{x}$ in all the 3 arguments. Note that $\rho_{*,erm}$ allows to compare the ERM uncertainty with the oracle uncertainty (the best we could do if we had infinite data), while $\rho_{bo,erm}$ quantifies the ERM uncertainty with respect to the best statistical estimate under a finite amount of data.

In the next Section, we provide a characterisation of this joint density in the high-dimensional limit where $n, d \to \infty$ with fixed sample complexity $\alpha = {}^{n}/_{d}$, as a function of the noise level τ and regularization λ. To obtain this result we leverage recent works on approximate message passing algorithms and their state evolution.

Some of our results will be conveniently formulated in terms of so-called calibration of a probabilistic classifier $\hat{f} : \mathbb{R}^d \to [0,1]$ defined as:

$$\Delta_p(\hat{f}) := p - \mathbb{E}_{\mathbf{x},y^*}(f_\star(\mathbf{x}) | \hat{f}(\mathbf{x}) = p) \tag{2.8}$$

where $\hat{f}$ can be the logistic classifier or the Bayes-optimal one. Intuitively, the calibration quantifies how well the predictor assigns probabilities to events. If $\Delta_p = 0$ the predictor is said to be *calibrated at level p*. Instead, if for $p > {}^{1}/_{2}, \Delta_p > 0$ (respectively $\Delta_p < 0$), then the predictor is said to be *overconfident* (respectively *underconfident*) Note, however, that the calibration is an average notion, while the above joint probability distribution (2.7) captures more detailed information about the point-wise confidence and its reliability. In this work, we will also consider the calibration of ERM with respect to Bayes

$$\tilde{\Delta}_p := p - \mathbb{E}_{\mathbf{x},y^*}(\hat{f}_{\rm bo}(\mathbf{x}) | \hat{f}_{\rm erm}(\mathbf{x}) = p) \tag{2.9}$$

Finally, while our discussion focuses in the calibration for concreteness, note that many other uncertainty metrics could be studied from the joint density eq. (2.7).

2.3 Technical theorems

Our first technical result is the existence of an efficient algorithm (Algorithm 2), called Generalized Approximate Message Passing (GAMP)) [202, 115] that is able to accurately estimate $\hat{f}_{\rm bo}(\mathbf{x})$ in high-dimensions. The asymptotic accuracy of GAMP for the Bayes-optimal average (over the samples) test error is know from [22]. In order to formulate our results we also need to prove that the probabilities estimated by GAMP are also accurate *sample-wise*, this relatively

Algorithm 2 GAMP

Input: Data $\mathbf{X} \in \mathbb{R}^{n \times d}$, $\mathbf{y} \in \{-1, 1\}^n$
Define $\mathbf{X}^2 = \mathbf{X} \odot \mathbf{X} \in \mathbb{R}^{n \times d}$ and Initialize $\hat{\mathbf{w}}^{t=0} = \mathcal{N}(\mathbf{0}, \sigma_w^2 \mathbf{I}_d)$, $\hat{\mathbf{c}}^{t=0} = \mathbf{1}_d$, $\mathbf{g}^{t=0} = \mathbf{0}_n$.
for $t \leq t_{\max}$ **do**
$\mathbf{V}^t = \mathbf{X}^2 \hat{\mathbf{c}}^t$; $\omega^t = \mathbf{X}\hat{\mathbf{w}}^t - \mathbf{V}^t \odot \mathbf{g}^{t-1}$; /* *Update channel mean and variance*
$\mathbf{g}^t = f_{\text{out}}(\mathbf{y}, \mathbf{w}^t, \mathbf{V}^t)$; $\partial \mathbf{g}^t = \partial_\omega f_{\text{out}}(\mathbf{y}, \mathbf{w}^t, \mathbf{V}^t)$; /* *Update channel*
$\mathbf{A}^t = -\mathbf{X}^{2^\top} \partial \mathbf{g}^t$; $\mathbf{b}^t = \mathbf{X}^\top \mathbf{g}^t + \mathbf{A}^t \odot \hat{\mathbf{w}}^t$; /* *Update prior mean and variance*
/* *Update marginals* */
$\hat{\mathbf{w}}^{t+1} = f_w(\mathbf{b}^t, \mathbf{A}^t) := (\mathbf{I}_d + \mathbf{A}^t)^{-1} \mathbf{b}^t$; $\hat{\mathbf{c}}^{t+1} = \partial_b f_w(\mathbf{b}^t, \mathbf{A}^t) := (\mathbf{I}_d + \mathbf{A}^t)^{-1}$
end for
Return: Estimators $\hat{\mathbf{w}}_{\text{amp}}, \hat{\mathbf{c}}_{\text{amp}} \in \mathbb{R}^d$

straightforward extension of the results of [22] is covered by the following lemma:

Lemma 2.3.1 (Sample-wise GAMP-Optimality)**.** *For a sequence of problems given by eq.* (2.2)*, and given the estimator* $\hat{\mathbf{w}}_{\text{amp}}$ *from Algorithm 1, the predictor*

$$\hat{f}_{\text{amp}}(\mathbf{x}) = \mathbb{P}(y = 1|\mathbf{x}) = \sigma_\star \left(\frac{\hat{\mathbf{w}}_{\text{amp}}^\top \mathbf{x}}{\sqrt{\tau^2 + \hat{\mathbf{c}}_{\text{amp}}^\top (\mathbf{x} \odot \mathbf{x})}} \right) \tag{2.10}$$

is such that, with high probability over a new sample $\mathbf{x}$ *the classifier above is asymptotically equal to the Bayesian estimator* $\hat{f}_{\text{bo}}(\mathbf{x}) = \mathbb{P}(y = 1|\mathbf{x}) = \hat{f}_{\text{amp}}(\mathbf{x})$ *in eq.* (2.5)*. More precisely:*

$$\forall \varepsilon > 0, \lim_{d \to \infty} \mathbb{P}_{\mathbf{x}, \mathscr{D}} \left(|\hat{f}_{\text{amp}}(\mathbf{x}) - \hat{f}_{\text{bo}}(\mathbf{x})|^2 \leqslant \varepsilon \right) \to 1 \tag{2.11}$$

In particular, the predictor $\hat{f}_{\text{amp}}$ *asymptotically achieves the best possible test performance (the one achieved by the Bayes-optimal estimator)*

The proof of Lemma 2.3.1 is provided in Appendix A.2. As mentioned above, the lemma does not require the prior on $\theta_\star$ to be Gaussian. Changing the prior of $\theta_\star$ amount to changing the denoising functions $(f_w, \partial_b f_w)$ in Algorithm 2. Similarly, the probit likelihood defined in equation (2.2) is not required for our analysis. In fact, the equations hold for any probabilistic generalized linear model, and in particular for the logit data model studied in [16], reproduced in Appendix A.4. This choice of likelihood function only changes the denoising *channel* functions $(f_{out}, \partial_\omega f_{out})$. The motivation behind the use of the GAMP Algorithm is twofold. First, it allows us to characterize the posterior mean needed to express the probability $\hat{f}_{\text{amp}}(\mathbf{x})$ for a given new sample $\mathbf{x}$ in polynomial time in d. Indeed, each iteration of the loop in Algorithm 2 is $O(d^2)$. Second, the asymptotic performance of GAMP is conveniently tracked by low-dimensional *state evolution* equations which can be easily solved in a computer.

Our second technical result is a formula for the joint distribution of the teacher label, its Bayes estimate, and the estimate from empirical risk minimisation defined in eq. (2.7), described in the

following theorem:

Theorem 2.3.2. *Consider training data* $\mathscr{D} = \{(\mathbf{x}^\mu, y^\mu)\}_{\mu=1}^n$ *sampled from the model defined in eq.* (2.2)*. Let* $\hat{\mathbf{w}}_{\text{erm}} \in \mathbb{R}^d$ *be the solution of the empirical risk minimisation* (2.4) *and* $\hat{\mathbf{w}}_{\text{amp}}$ *denote the estimator returned by running algorithm 2 on the data* $\mathscr{D}$*. Then in the high-dimensional limit where* $n, d \to \infty$ *with* $\alpha = {}^n/_d$ *fixed, the asymptotic joint density* (2.7) *is given by:*

$$\rho(a,b,c) = \tau'\tau \frac{\mathcal{N}\left(\begin{bmatrix} \tau \cdot \sigma_\star^{-1}(a) \\ \tau' \cdot \sigma_\star^{-1}(b) \\ \sigma^{-1}(c) \end{bmatrix} \middle| \mathbf{0}_3, \Sigma\right)}{|\sigma_\star'(\sigma_\star^{-1}(a))||\sigma_\star'(\sigma_\star^{-1}(b))||\sigma'(\sigma^{-1}(c))|} \tag{2.12}$$

where we noted

$$\tau'^2 = \tau^2 + 1 - q_{\text{bo}}, \qquad \Sigma = \begin{bmatrix} 1 & q_{\text{bo}} & m \\ q_{\text{bo}} & q_{\text{bo}} & m \\ m & m & q_{\text{erm}} \end{bmatrix} \tag{2.13}$$

and the so-called overlaps:

$$q_{\text{bo}} = \frac{1}{d}\hat{\mathbf{w}}_{\text{amp}}^\top \theta_\star = \frac{1}{d}||\hat{\mathbf{w}}_{\text{amp}}||_2^2 \tag{2.14}$$

$$m = \frac{1}{d}\hat{\mathbf{w}}_{\text{erm}}^\top \theta_\star, \quad q_{\text{erm}} = \frac{1}{d}||\hat{\mathbf{w}}_{\text{erm}}||_2^2 \tag{2.15}$$

solve the following set of self-consistent equations:

$$\frac{1}{q_{\text{bo}}} = 1 + \alpha\, \mathbb{E}_{(z,\eta),\xi}\left[f_{\text{out}}(f_0(z + \tau\xi), \eta, 1 - q_{\text{bo}})^2\right], \tag{2.16}$$

and

$$V = \frac{1}{\lambda + \hat{V}}, \qquad q_{\text{erm}} = \frac{\hat{m}^2 + \hat{q}}{(\lambda + \hat{V})^2}, \qquad m = \frac{\hat{m}}{\lambda + \hat{V}}. \tag{2.17}$$

$$\begin{cases} \hat{V} &= -\alpha \mathbb{E}_{(z,\omega),\xi}\left[\partial_\omega f_{\text{erm}}(f_0(z + \tau\xi), \omega, V)\right] \\ \hat{q} &= \alpha \mathbb{E}_{(z,\omega),\xi}\left[f_{\text{erm}}(f_0(z + \tau\xi), \omega, V)^2\right] \\ \hat{m} &= \alpha \mathbb{E}_{(z,\omega),\xi}\left[f_{\text{erm}}(f_0(z + \tau\xi), \omega, V)\right] \end{cases} \tag{2.18}$$

where $(z, \eta, \omega) \sim \mathcal{N}(0_3, \Sigma)$*,* $\xi \sim \mathcal{N}(0,1)$ *and the thresholding functions are defined as*

$$f_{\text{out}}(y, \omega, V) = \frac{2y\, \mathcal{N}(\omega y | 0, V + \tau^2)}{\text{erfc}\left(-\frac{y\omega}{\sqrt{2(\tau^2+V)}}\right)}$$

$$f_{\text{erm}}(y, w, V) = V^{-1}\left(\text{prox}_{Vl(y,.)}(w) - w\right) \tag{2.19}$$

with $\text{prox}_{\tau f}(x) = \text{argmin}_z\left({}^1/_{2\tau}\|z - x\|_2^2 + f(z)\right)$ *being the proximal operator.*

In Appendix A.1 we show how this result can be deduced directly from the heuristic cavity method, and the analysis of the GAMP state evolution to compute the overlaps of ERM and BO estimators. To compute the correlation between the ERM and BO estimators, we use the Nishimori identity [112, 247]. More details, as well as the formal proof, are given in Appendix A.2.

Our third theorem is an asymptotic expression for the calibration error.

Theorem 2.3.3. *The analytical expression of the joint density ρ yields the following expression for the calibration Δ_p:*

$$\Delta_p(\hat{f}_{\text{erm}}) = p - \sigma_\star\left(\frac{{}^{m}/_{q_{\text{erm}}} \times \sigma^{-1}(p)}{\sqrt{1 - {}^{m^2}/_{q_{\text{erm}}} + \tau^2}}\right). \tag{2.20}$$

Moreover, the Bayesian classifier is always well calibrated with respect to the teacher, meaning:

$$\forall\, p \in [0,1], \quad \Delta_p(\hat{f}_{\text{bo}}) = 0\,. \tag{2.21}$$

Additionally, the calibration of ERM with respect to the Bayesian classifier and the oracle are equal:

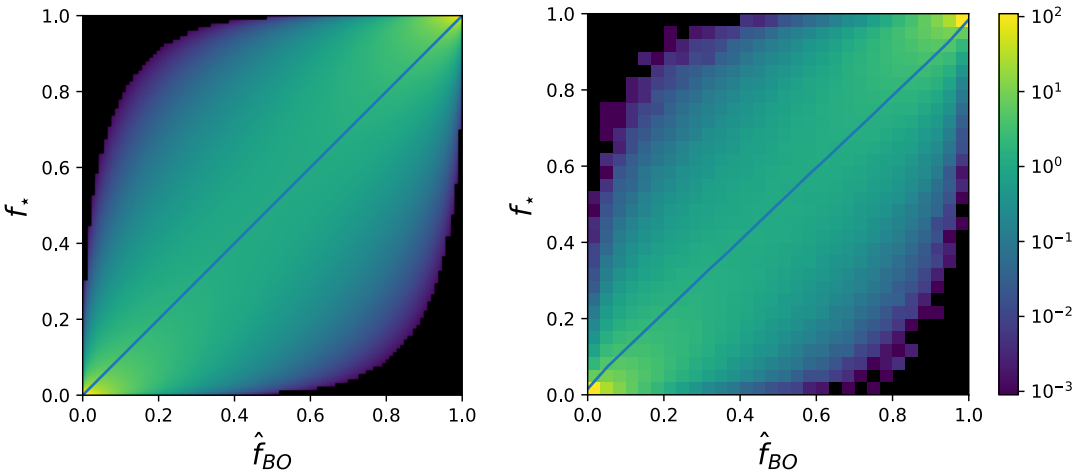
$$\forall\, p \in [0,1], \quad \Delta_p(\hat{f}_{\text{erm}}) = \tilde{\Delta}_p\,. \tag{2.22}$$

The proof of Theorem 2.3.3 is provided in appendix A.2.3. Equation (2.20) shows the different factors that influence Δ_p: the aleatoric uncertainty represented by the noise τ^2, the finiteness of data that appears through ${}^{m}/_{q_{\text{erm}}}$ and ${}^{m^2}/_{q_{\text{erm}}}$, and the mismatch in the model with the activations $\sigma_\star, \sigma$. Moreover, Equation (2.22) provides a recipe to compute the calibration Δ_p in the high-dimensional limit from the knowledge of the data model (2.2) only, but without knowing the specific realisation of the weights $\theta_\star$. This is because the quantities q_{bo}, q_{erm} and m self-average as $n, d \to \infty$, we then obtain the calibration Δ_p without knowing the realisation of $\theta_\star$.

2.4 Results for uncertainty estimation

2.4.1 Bayes versus oracle uncertainty

We now discuss the consequences of the theorems from Section 2.3. Figure 2.1 left panel depicts the theoretical prediction of the joint density $\rho_{\text{bo},\star}$, between the Bayes posterior confidence/uncertainty $\hat{f}_{\text{bo}}$ (x-axes) and the oracle confidence/uncertainty $f_\star$ (y-axes). The theoretically derived density (Figure 2.1 left panel) is compared to its numerical estimation in Figure 2.1 right panel, computed numerically using the GAMP algorithm. To estimate the numerical density in the right panel, we proceed as follows: after fixing the dimension d and the number of training samples $n = \alpha d$, GAMP is run on the training set. Once GAMP estimators have been obtained, n_{test} test samples are drawn and for each of them, we compute the confidence of the oracle/teacher $f_\star(\mathbf{x})$



(a) (b)

Figure 2.1: Theoretical prediction (left panel) and numerical estimation (right panel) of the joint density $\rho_{\mathrm{bo},\star}$ at $\alpha = 10$ and noise level $\tau = 0.5$. Numerical plot was done by running GAMP at dimension $d = 1000$, computing $(f_\star, \hat{f}_{\mathrm{bo}})$ on $n_{\mathrm{test}} = 10^7$ test samples. The blue curve is the mean of $f_\star$ given $\hat{f}_{\mathrm{bo}}$. For these parameters, the test error of Bayes is $\varepsilon_g^{\mathrm{bo}} = 0.173$, the oracle test error $\varepsilon^\star = 0.148$.

from eq. (2.2), and the Bayesian confidence $\hat{f}_{\mathrm{bo}}(\mathbf{x}) = \hat{f}_{\mathrm{amp}}(\mathbf{x})$ from Theorem 2.3.1. Finally, we plot the histogram of the thus obtained joint density $\rho_{\mathrm{bo},\star}$ over the test samples. As the figure shows, there is a good agreement between theory and finite instance simulations.

We see that the density is positive on the whole support, it peaks around $(0,0)$ and $(1,1)$, but has a notable weight around the diagonal as well. The relatively large spread of the joint density is a consequence of the fact that on top of the intrinsic uncertainty of the teacher, the learning is only done with $n = \alpha d$ samples which brings an additional source of uncertainty captured in the Bayes estimator. Fig. 2.1 thus quantifies this additional uncertainty due to finite α. We are not aware of something like this being done analytically in previous literature.

The blue curve is the mean of $f_\star$ conditioned on the values of $\hat{f}_{\mathrm{bo}}$. The difference between this and the diagonal is the calibration Δ_p defined in Equation (2.8). We see that the figure illustrates $\Delta_p(\hat{f}_{\mathrm{bo}}) = 0$, i.e. the Bayesian prediction is well calibrated, as predicted by Theorem 2.3.3.

Figure 2.2 then depicts the same densities as Figure 2.1 for several different values of the sample complexity α and noise τ. The corresponding test error is given for information. We see, for instance, that at small α the BO confidence is low, close to 0.5, because not much can be learned from very few samples. The oracle confidence does not depend on α, and is low for growing τ. At large α, on the other hand, the BO confidence is getting well correlated with the oracle one. At larger α and small noise the BO test error is getting smaller and the corresponding confidence is close to 1 or 0 (depending on the label). The trends seen in this figure are expected, but again

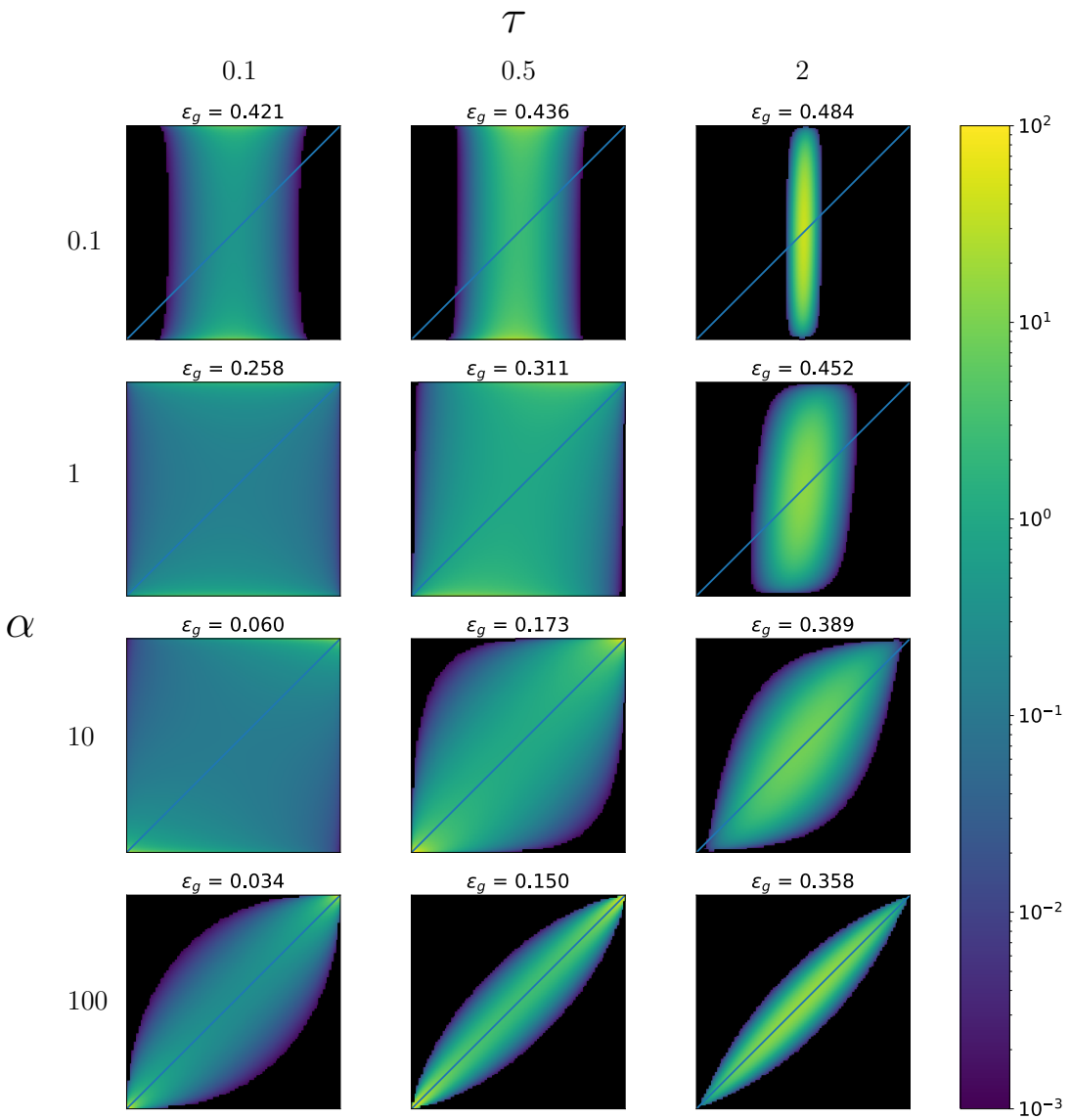
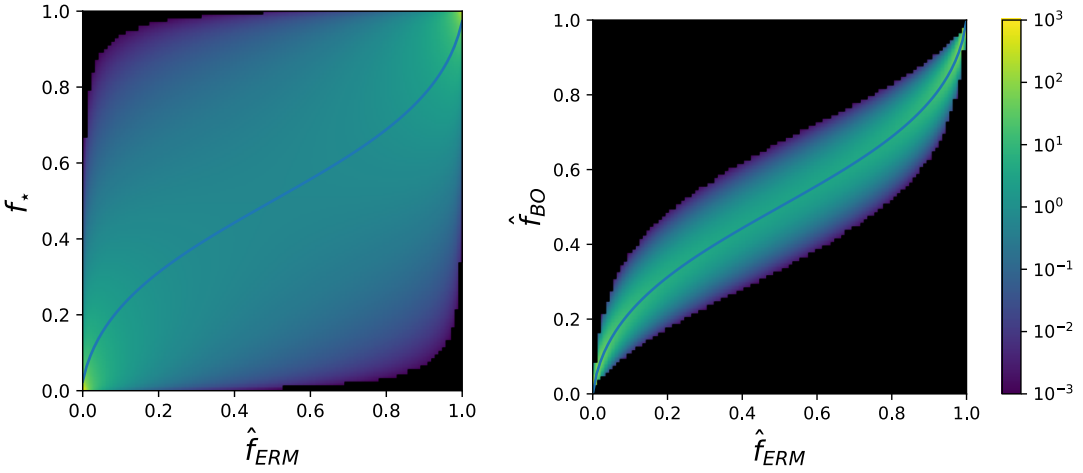


Figure 2.2: Density between Bayes confidence $\hat{f}_{\text{bo}}$ (x-axis) and the oracle confidence $f_\star$ (y-axis) for multiple values of α, τ: the rows correspond respectively to $\alpha = 0.1, 1, 10, 100$ from top to bottom, and the columns correspond respectively to $\tau = 0.1, 0.5, 2$. The blue curve is the mean of $f_\star$ given $\hat{f}_{\text{bo}}$. Generalisation errors of the Bayes estimator are in written on top of the corresponding plot. The best possible generalisation errors, achieved if the teacher weights are known, for $\tau = 0.1, 0.5, 2$ are respectively $\varepsilon_g^\star = 0.032, 0.148, 0.352$.

here we quantify them in an analytic form of eq. (2.12) which as far as we know has not been done previously.



(a) (b)

Figure 2.3: The probability density $\rho_{\mathrm{erm},\star}$ (left panel) and $\rho_{\mathrm{erm,bo}}$ (right panel), at $\alpha = 10$, $\tau = 0.5$ and $\lambda = 0^+$. The blue curves are the mean of the marginal distribution of $f_\star$ and $\hat{f}_{\mathrm{bo}}$ respectively under fixed $\hat{f}_{\mathrm{erm}}$, which are equal to $p - \Delta_p$ and $p - \tilde{\Delta}_p$. We observe overconfidence of the logistic classifier for these parameters. Test error of ERM is here $\varepsilon_g^{\mathrm{erm}} = 0.174$, very close to the one of BO $\varepsilon_g^{\mathrm{bo}} = 0.173$.

2.4.2 Logistic regression uncertainty and calibration

Having explicit access to the Bayesian confidence/uncertainty in a high-dimensional setting is a unique occasion to quantify the quality of the logistic classifier, which has its own natural measure of confidence induced by the logit. How accurate is this measure? We start with the logistic classifier at zero regularization and then move to the regularised case in the next section.

Figure 2.3 compares the joint density of $(\hat{f}_{\mathrm{erm}}, f_\star)$ (left panel), and $(\hat{f}_{\mathrm{erm}}, \hat{f}_{\mathrm{bo}})$ (right panel) with the same noise and number of samples as used in figure 2.1. The blue curves are the means of $f_\star$ (respectively $\hat{f}_{\mathrm{bo}}$) conditioned on $\hat{f}_{\mathrm{erm}}$, their shape is demonstrating that the (non-regularized) logistic classifier is on average overconfident, as is well known in practice.

The equality between these two blue curves illustrates Theorem 2.3.3, Equation (2.22): $\Delta_p(\hat{f}_{\mathrm{erm}}) = \tilde{\Delta}_p$. Note, however, that while the calibrations of the ERM with respect to the oracle or the BO are equal, the conditional variances of $f_\star$ and $\hat{f}_{\mathrm{bo}}$ are very different. This shows how the calibration is only a very partial fix of the confidence estimation for ERM: when $\hat{f}_{\mathrm{erm}} = p$, both Bayes and the oracle's predictions will be $p - \Delta_p$ *on average*, but for the considered parameters the predictions of the oracle are much more spread around this value than those of Bayes estimator. This means that the ERM still captures rather well some part of the uncertainty coming from the limited number of samples. Figure A.1 in the Appendix A.3 complements Figure 2.3 by showing other values of α and τ.

We now investigate the calibration as a function of the sample complexity α. The plot (a) of

Figure 2.4 shows the curve Δ_p at $\lambda = 0^+$ computed using the analytical expression (2.20). The curve is compared to the numerical estimation of Δ_p (green crosses) and $\tilde{\Delta}_p$ (orange crosses). For a small $\mathrm{d}p$, If we define $I_{p,\mathrm{d}p} = \{1 \leqslant i \leqslant n_{\text{test}} | \hat{f}_{\text{erm}}(x_i) \in [p, p+\mathrm{d}p]\}$, Δ_p and $\tilde{\Delta}_p$ are estimated experimentally with the formulas

$$\Delta_p \simeq p - \frac{\sum_{i \in I_{p,\mathrm{d}p}} f_\star(x_i)}{|I_{p,\mathrm{d}p}|}, \tilde{\Delta}_p \simeq p - \frac{\sum_{i \in I_{p,\mathrm{d}p}} \hat{f}_{\text{bo}}(x_i)}{|I_{p,\mathrm{d}p}|} \tag{2.23}$$

The calibrations Δ_p and $\tilde{\Delta}_p$ are both equal to the theoretical curve, further confirming the results of Equation (2.22). Note the transition at $\alpha_c \sim 2.4$: for $\alpha < \alpha_c$, the training data is linearly separable. Since $\lambda = 0^+$, the empirical risk has no minimum and the estimator $\mathbf{w}_{\text{erm}}$ diverges in norm. As a consequence, $\Delta_p \to p - 0.5$, as we observe on the plot.In the inset of Fig. 2.4 (left) we depict the theoretical curve evaluated up to larger values of α. We see a saturation at about $\Delta_p \simeq 0.0011 \neq 0$. We note that in the work of [16] (partly reproduced in Appendix A.4) the calibration was observed to go to 0 as $1/\alpha$. This difference is due to the mismatch between the function producing the data (probit) and the estimator (logit) in our case (whereas [16] used logit for both) which will generically be present in real data and thus the decay to zero observed in [16] is not expected to be seen generically.

Right panel of Figure 2.4 displays the variance of $f_\star$ and $\hat{f}_{\text{bo}}$ at fixed $\hat{f}_{\text{erm}}$ as a function of α. This plot illustrates that the conditional variance of $f_\star$ is significantly higher than that of $\hat{f}_{\text{bo}}$, as was previously noted in figure 2.3. This shows that the (non-regularized) logistic uncertainty captures rather decently the uncertainty due to the limited number of samples.

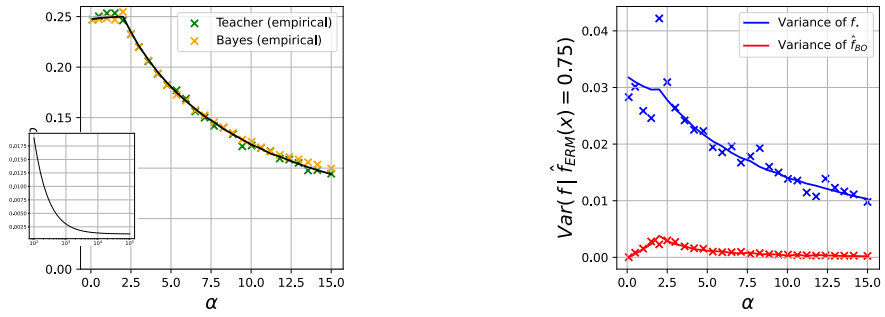
2.4.3 Effect of regularization on uncertainty and calibration

Logistic regression is rarely used in practice without regularization. In Figs. A.4 and A.2 in appendix A.3 we depict the role of regularization on the density $\rho_{\text{erm,bo}}$. As one would anticipate as the regularization strength grows the overconfidence of the logistic classifier at small λ becomes under-confidence at large λ.

One usually optimizes the strength λ of the ℓ_2 penalty through cross-validation. Ideally, we would choose λ that gives a low validation error but also that yields a well-calibrated estimator. The two main ways to choose λ is to minimize the validation $0/1$ classification error or the validation logistic loss. In our teacher-student setting, the classification error and logistic loss on test data can be computed exactly in the high-dimensional limit, using our state-evolution equations. We will thus define λ_{error} (respectively λ_{loss}) as the minimizer of the expected $0/1$ classification error (respectively the logistic loss) for a new test sample. More precisely :

$$\begin{cases} \lambda_{\text{error}} &= \operatorname{argmin}_\lambda \mathbb{P}_{\mathbf{x},y} \left[y \neq \operatorname{sign}(\hat{\mathbf{w}}(\lambda)^T \mathbf{x}) \right] \\ \lambda_{\text{loss}} &= \operatorname{argmin}_\lambda \mathbb{E}_{\mathbf{x},y} \left[-\log \sigma \left(y \times \hat{\mathbf{w}}(\lambda)^T \mathbf{x} \right) \right] \end{cases} \tag{2.24}$$

where $\hat{\mathbf{w}}(\lambda)$ minimizes the empirical risk with regularization strength λ. Note that cross-validating



(a) (b)

Figure 2.4: (a) Calibration of the logistic regression with $\lambda = 0^+, \tau = 2, p = 0.75$. Orange (respectively green) crosses are the numerical estimations of $\tilde{\Delta}_p$ (respectively Δ_p). Numerical values are obtained by averaging the calibration over 10 test sets of size $n_{\text{test}} = 10^5$, at $d = 300$. Inset depicts the larger α behaviour. (b) Variance of $f_\star$ and $\hat{f}_{\text{bo}}$ conditioned on $\hat{f}_{\text{ERM}} = p = 0.75$. Crosses are numerical values with the same parameters as figure (a). Though both $f_\star$ and $\hat{f}_{BO}$ have the same mean, their variance is significantly different. The high variability of experimental values at low α is due to the fact that the training data is linearly separable and $\lambda = 0$, hence $\hat{w}_{\text{erm}}$ diverges in norm. Thus, $\hat{f}_{\text{erm}}$ is very close to 0 or 1 for most test inputs, thus we estimate the variance of the teacher on $\hat{f}_{\text{erm}} = 0.75$ with few test inputs, and the estimator of the variance has a higher variability.

λ on a validation set would induce fluctuations due to the finiteness of validation data. These fluctuations are not present when defining λ with Equations (2.24). In the setting of the present paper, these two values of regularisation lead to a very close test error/loss. In other words, choosing one or another of these λ seems to have little effect on the test performance of logistic regression.

Figure 2.5 plots the calibration Δ_p in the noiseless (left panel) and noisy (right panel) settings. We observe that for most parameters ERM with λ_{loss} is significantly less overconfident than with λ_{error}. However, for larger values of α and τ we observe the opposite.

We also note that for small α the logistic regression at λ_{loss} even gets mildly underconfident, $\Delta_p < 0$. The bottom panels of the figure depict the corresponding variance. Interestingly we see that in both cases, despite a better calibration, λ_{loss} yields a higher variance than λ_{error} hence its point-wise estimates of uncertainty are not necessarily better.

Figure 2.6 shows $\rho_{\text{bo,erm}}$ evaluated at λ_{error} and λ_{loss}. Comparing the upper panels to Figure 2.3 (at $\lambda = 0$), it is clear that choosing λ to optimize the error (and the loss) improves calibration. In the lower panels of Figure 2.6 we can also see that the calibration at λ_{loss} (right panel) is better, i.e. the blue line is closer to $y = x$, than the one at λ_{error} (left panel). We conclude that using optimal regularization is clearly advantageous to obtain better-calibrated classification. However, we also note that the interplay between the mean of the distribution (the calibration) and its variance is

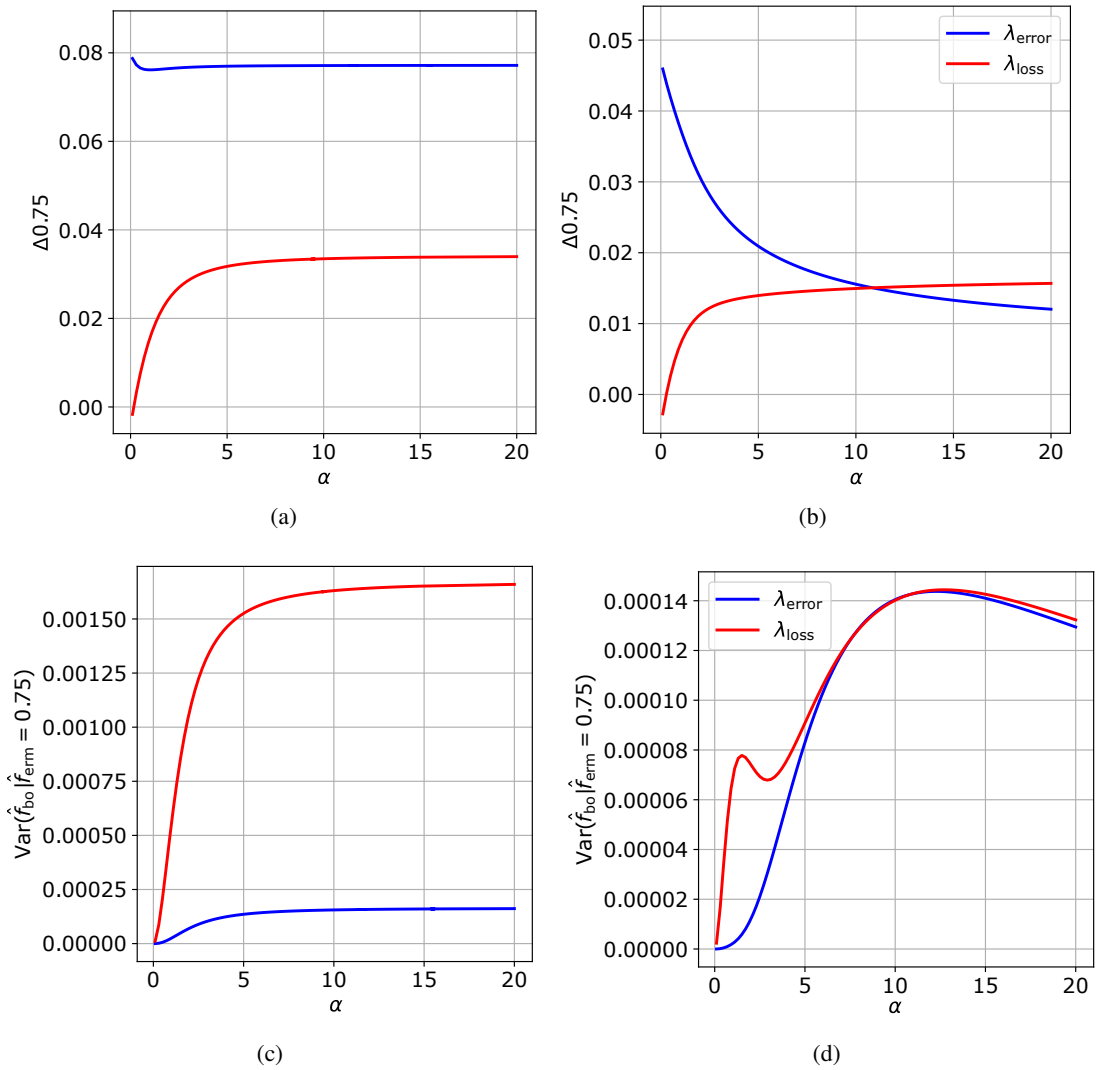
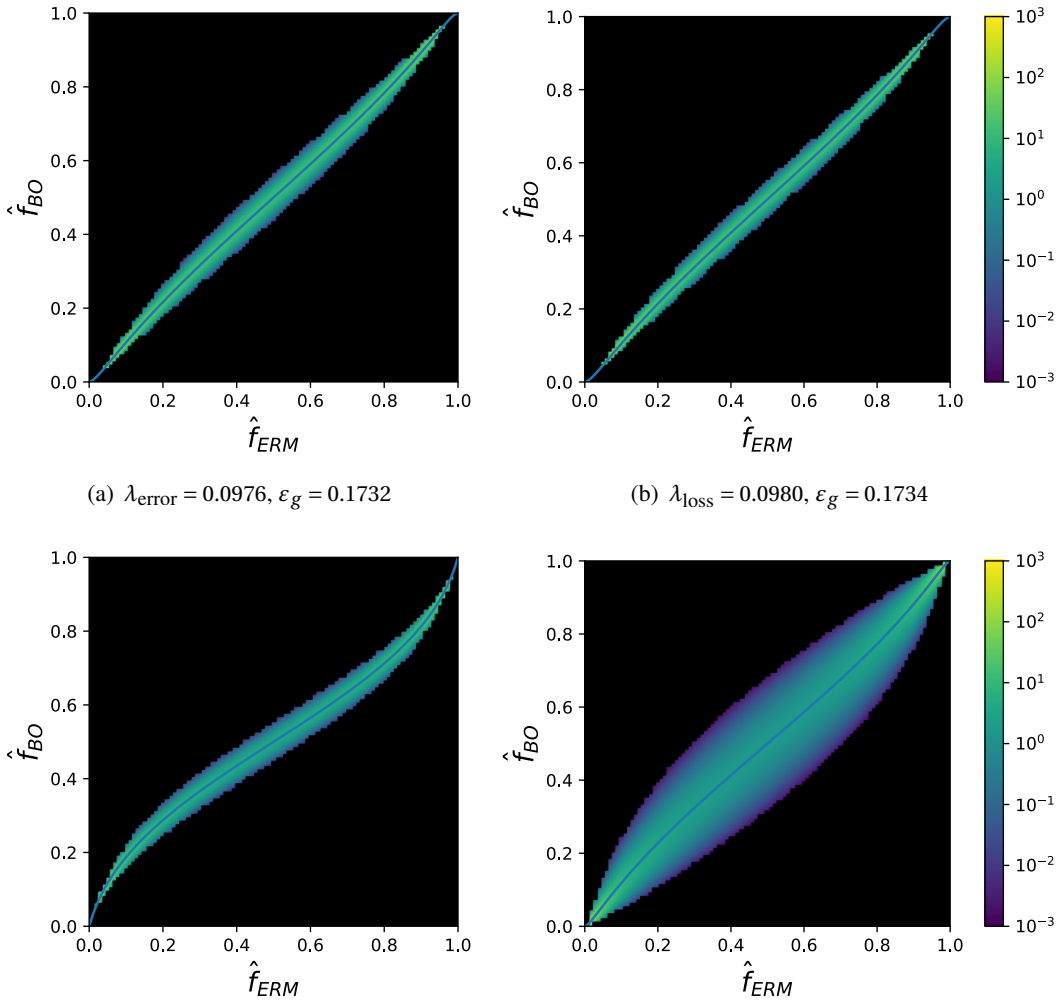


Figure 2.5: (Top) The calibration $\Delta_{0.75}(\hat{f}_{\text{erm}})$ as a function of α with $\lambda = \lambda_{\text{error}}(\alpha, \tau)$ (blue curve) and $\lambda = \lambda_{\text{loss}}(\alpha, \tau)$ (red curve)s. (Bottom) Variance of $\hat{f}_{\text{bo}}$ conditioned on $\hat{f}_{\text{erm}}(\mathbf{x}) = 0.75$ with λ_{error} and λ_{loss}. In (a) and (c), $\tau = 0$; in (b) and (d), $\tau = 0.5$.

subtle and more investigation is needed into designing a model-agnostic method where both are optimal simultaneously.

2.5 Discussion

This paper leverages the properties of the GAMP algorithm and associated closed-form control of the posterior marginals to provide a detailed theoretical analysis of uncertainty in a simple probit model. We investigate the relations between the respective uncertainties of the oracle, Bayes and regularized logistic regression. We see this as a grounding step for a line of future



(a) $\lambda_{\text{error}} = 0.0976$, $\varepsilon_g = 0.1732$

(b) $\lambda_{\text{loss}} = 0.0980$, $\varepsilon_g = 0.1734$

(c) $\lambda_{\text{error}} = 0.0039$, $\varepsilon_g = 0.0843$

(d) $\lambda_{\text{loss}} = 0.0096$, $\varepsilon_g = 0.0847$

Figure 2.6: Density $\rho_{\text{erm,bo}}$ for different α, τ. Top row: $\alpha = 10, \tau = 0.5$. Bayes test error is $\varepsilon_g^{\text{bo}} = 0.1731$. Plot (a) (respectively (b)) is done at $\lambda = \lambda_{\text{error}}$ (respectively $\lambda = \lambda_{\text{loss}}$). Bottom row: $\alpha = 5, \tau = 0$, $\varepsilon_g^{\text{bo}} = 0.0839$. Plot (c) (respectively (d)) is done at $\lambda = \lambda_{\text{error}}$ (respectively $\lambda = \lambda_{\text{loss}}$). On the bottom row, we can clearly see that the calibration is better for λ_{loss}. Generalization errors of ERM as well as the values of the regularizations are indicated below the plots.

work that will leverage recent extensions of the GAMP algorithm and its associated analysis to multi-layer neural networks [14, 97], learning with random features and kernels [168, 94, 66], estimation under generative priors [15, 12], classification on more realistic models of data [98, 99, 210], etc. The present methodology is not restricted to classification and can be used for a more thorough study of confidence intervals in high-dimensional regression, extending [17]. This is left for further studies. The code of this project is available at https://github.com/SPOC-group/high-dimensional-uncertainty.

3 Double descent in uncertainty quantification in random features

Authors : Lucas Clarté, Bruno Loureiro, Florent Krzakala, Lenka Zdeborová

3.1 Introduction

Uncertainty estimation is the cornerstone of reliable data processing. A large body of literature in classical statistical theory is dedicated to providing solid mathematical guarantees on a model's uncertainty, such as confidence scores for classification and confidence intervals for regression [237]. Yet, when it comes to modern machine learning methods such as deep neural networks our mathematical understanding of the uncertainty associated with prediction falls short. A key aspect in current machine learning practice is that, in contrast to classical wisdom, models often operate in a regime where the complexity of the hypothesis class (e.g. as measured by the number of parameters in the model) is comparable or larger than the quantity of data available for training. This modern, overparametrized regime defies the common intuition rooted on classical statistics, therefore posing interesting challenges to their mathematical treatments. sol For example, deep neural networks are able to achieve optimal generalization performance even when the training data are perfectly interpolated [91, 89, 181], a behaviour at odds with the bias-variance intuition. This *benign overfitting* property was recently shown to be common among overparametrized convex methods, such as linear regression [25, 105], random features regression [168] and classification [94].

While much of the theoretical effort has focused on the generalization properties of point estimates from overparametrized models, less is understood about their confidence. Indeed, a popular method to estimate uncertainty in neural networks consists of interpreting the last layer pre-activations as class probabilities. Numerical experiments suggest that deep neural networks tend to suffer from *overconfidence* with respect this notion [101], a problem which has motivated many empirical calibration methods in the literature [107, 136, 179, 150]. Recently, it has been shown that actually overconfidence is a common problem in high-dimensional classification [16], although it can be considerably mitigated by properly regularising the risk

[53]. An alternative to the pre-activation scores consists in applying a Bayesian treatment to neural networks, for instance by averaging the last layer weights over the measure induced by the empirical risk. In some contexts, these techniques were shown to provide better calibrated uncertainty measures than pre-activation score. A priori, Bayesian techniques require sampling from a high-dimensional measure, and therefore can be computationally demanding [7]. Despite the success and widespread use of these uncertainty measures, mathematical guarantees relating these notions to intrinsic uncertainty measures such as the true class probabilities or the best uncertainty estimation given the available data (i.e. the true posterior uncertainty given the features) are scarce. In this work, we provide a sharp mathematical comparison between these different uncertainty notions in the context of a simple, solvable model for binary classification on structured features - such as the ones given by the first layers of neural networks. To the best of our knowledge, our work is the first to provide a sharp asymptotic analysis of uncertainty in overparametrized high-dimensional models.

Related work – Uncertainty quantification in deep learning is an active and rapidly evolving field, with many coexisting metrics and methods in the literature, see e.g. [2, 88] for two recent reviews. [185, 101] empirically observed that different from "small" networks [186], modern deep neural networks tend to give overconfident predictions. [101] proposed *temperature scaling*, a simple post-processing variant of Platt scaling [198] consisting of rescaling & cross-validating the norm of the last-layer weights, and showed it can effectively calibrate them. Alternatively, [136] has argued that a Bayesian treatment of the last layer of deep networks fixes overconfidence. Bayesian methods typically involve sampling from a high-dimensional posterior [166], and different methods have been proposed to compute them efficiently [100, 86, 143, 157]. Of particular interest to our work is the Laplace approximation introduced in [155] for Gaussian process classification and adapted to Bayesian deep learning in [205, 136, 61]. An asymptotic discussion of evidence maximization in Bayesian ridge regression appeared in [164, 41, 163]. [16] has shown that the logit model is overconfident in high-dimensions, and [53] discussed how to mitigate it by properly regularizing. An exact asymptotic characterization of the empirical risk minimizer for random features model has been derived and discussed in [168, 94, 99, 109, 66, 153]. Particularly relevant to our technical results is the recent progress in approximate message-passing schemes for structured matrices [97, 152]. Finally, exact asymptotics for Bayes-optimal estimation has been discussed in the context of generalized linear models in [22, 85].

Notation – We denote vectors with bold letters, and matrices with capital letters. For $n \in \mathbb{N}$, we let $[n] := \{1, \cdots, n\}$. $\mathcal{N}(\mu, \Sigma)$ denotes the Gaussian density, $\sigma(t) := (1 + e^{-t})^{-1}$ denotes the sigmoid function. We define

$$\sigma_v(x) := \int \sigma(z)\mathcal{N}(z|x, v)\mathrm{d}z \tag{3.1}$$

the averaged sigmoid with a Gaussian noise of variance v.

3.2 Setting

3.2.1 Probabilistic classifiers and uncertainty

Consider a supervised binary classification task given by n independent samples $\mathscr{D} = (\mathbf{x}^\mu, y^\mu)_{\mu\in[n]} \in \mathscr{X} \times \{-1, +1\}$ from a joint distribution ν, and denote by $f_\star(\mathbf{x}) = \nu(y = 1|\mathbf{x})$ the oracle class probability obtained by conditioning ν over an input. In this work we are interested in studying the uncertainty associated to probabilistic classifiers $\hat{f}(\mathbf{x}) = \mathbb{P}(y = 1|\mathbf{x}) \in [0, 1]$ obtained by fitting the data[1], and how they compare with the true class probability $f_\star$. A key motivation is the recent stream of works on uncertainty quantification for neural networks, and in particular the line of works proposing uncertainty measures based on classifiers defined by sampling over the last layer of neural networks [40, 136]. To set notation, let $\varphi : \mathscr{X} \to \mathbb{R}^p$ denote a *feature map*, for instance the features learned by the first layers of a trained neural network. We shall be interested in the following classifiers:

Empirical risk classifier – The empirical risk classifier is the one obtained by naively interpreting the scores in the last layer as probability distributions. Mathematically, it is defined as $\hat{f}_{\text{erm}}(\mathbf{x}) = \sigma(\hat{\theta}_{\text{erm}}^\top \varphi(\mathbf{x}))$, where $\sigma : \mathbb{R} \to (0, 1)$ is a non-linearity. For concreteness, we will focus on the popular case where $\sigma(z) = (1 + e^{-z})^{-1}$ is the sigmoid function, and $\hat{\theta}_{\text{erm}} \in \mathbb{R}^p$ is the minimizer of the associated (regularized) logistic or cross-entropy risk:

$$\hat{\mathscr{R}}_n(\theta) = \frac{1}{n}\sum_{\mu=1}^{n} \log\left(1 + e^{-y^\mu \theta^\top \varphi(\mathbf{x}^\mu)}\right) + \frac{\lambda}{2}||\theta||_2^2. \tag{3.2}$$

This is also commonly referred to as the *logit classifier*.

Bayes-optimal classifier – Denoting the training features $\mathscr{D}_\varphi := \{(\varphi(\mathbf{x}^\mu), y^\mu)\}_{\mu\in[n]}$, the optimal Bayesian classifier for the last layer is given by:

$$\hat{f}_{\text{bo}}(\mathbf{x}) = \int \mathrm{d}\theta \; p\left(y = 1|\theta, \{\varphi\left(\mathbf{x}^\mu\right)\}_{\mu\in[n]}\right) p\left(\theta|\mathscr{D}_\varphi\right) \tag{3.3}$$

where $p\left(y = 1|\theta, \{\varphi(\mathbf{x}^\mu)\}_{\mu\in[n]}\right)$ is the likelihood over the labels and $p(\theta|\mathscr{D}_\varphi)$ is the posterior distribution over the weights given the training features and labels. In practice, the Bayes-optimal classifier f_{bo} is not accessible to the statistician, since she doesn't have access to the distribution ν that has generated the data - and even if she had, sampling from the high-dimensional posterior distribution would be computationally cumbersome. However, as we will discuss in Sec. 3.3.1, for the data generative model considered here, the Bayes-optimal classifier can be asymptotically characterized, and its marginals can be computed by a polynomial-time message passing algorithm.

[1] In the following, we consistently denote with a hat classifiers which are a function of the training data.

Bayesian classifiers – Since the optimal Bayesian classifier is not accessible in practice, different classifiers inspired by Bayesian methods have been proposed in the literature. In this manuscript, we will consider two popular choices.

The first is the *empirical Bayes* classifier $\hat{f}_{\text{eb}}$ [164, 119]. In full generality, the empirical Bayes method consists of postulating a class of plausible likelihoods and priors and doing model selection from the training data via evidence maximization. In the context of Bayesian neural networks, the likelihood and priors are defined by the network architecture and regularization, which are normalized to define proper probability distributions.

In our setting, the empirical Bayes classifier is explicitly given by:

$$\begin{aligned} \hat{f}_{\text{eb}}(\mathbf{x}) &= \int_{\mathbb{R}^p} \mathrm{d}\theta\ \sigma(\beta\theta^\top\varphi(\mathbf{x}))\, p_{\text{eb}}(\theta|\mathscr{D},\beta,\lambda), \\ p_{\text{eb}}(\theta|\mathscr{D},\beta,\lambda) &= \frac{\prod_\mu \sigma(\beta y^\mu \theta^\top\varphi(\mathbf{x}^\mu))\,\mathscr{N}(\theta|{}^{\mathbf{I}_p}/_{\beta\lambda})}{p(\mathscr{D}|\beta,\lambda)} \end{aligned} \tag{3.4}$$

The normalisation constant $p(\mathscr{D}|\beta,\lambda)$ is known as the *marginal likelihood* or the *evidence*. In the empirical Bayes method the evidence is maximized in order to select the most likely hyperparameters (β,λ) explaining the training data [156]. In our specific model, we note that the evidence is actually only a function of the ratio ${}^{\lambda}/_{\beta}$ (this can be seen from the change of variables $\theta \leftarrow \beta\theta$). Therefore, without loss of generality we take $\beta = 1$ and optimize only over λ. It is important to stress that the postulated prior and likelihood in $\hat{f}_{\text{eb}}$ may not correspond to the ones that generated the data in general.

Note that, differently from the Bayes-optimal estimator, the empirical Bayes classifier can be a priori computed using only the training data. However, it can be computationally demanding to sample from the posterior distribution above, specially in large dimensions $p, n \gg 1$. To avoid this computational bottleneck, a common approximation consists of expanding the posterior around the $\hat{\theta}_{\text{erm}}$ to second order, known as the *Laplace approximation* [136, 205, 61]:

$$\hat{f}_{\text{Lap}}(\mathbf{x}) = \int \mathrm{d}\theta\ \sigma\left(\hat{\theta}_{\text{erm}}^\top\varphi(\mathbf{x})\right)\mathscr{N}(\theta|\hat{\theta}_{\text{erm}},\mathscr{H}^{-1}) \tag{3.5}$$

where $\mathscr{H} := \nabla^2_\theta\hat{\mathscr{R}}_n(\hat{\theta}_{\text{erm}})$ is the Hessian of the empirical risk evaluated at the minimum. Therefore, in the Laplace approximation the posterior is effectively approximated by a Gaussian distribution centred at $\hat{\theta}_{\text{erm}}$ and with covariance given by the inverse curvature around the minimum. The "sharper" the minimum, the lower the variance and the more confident the Laplace classifier is. Note that the generalization errors associated to the Laplace classifier coincide exactly with the empirical risk classifier. Finally, in the model considered here, the Laplace approximation $\hat{f}_{\text{Lap}}$ will always be less confident than the ERM estimator using $\hat{\theta}_{\text{erm}}$. This is due to the concavity of the logit function σ on $[0,\infty)$.

Performance and uncertainty – Given a probabilistic classifier $\hat{f}$, the most common measure for the generalization performance is the *misclassification test error* (also known as *0/1 error*) :

$$\mathscr{E}_{\text{gen.}}(\hat{f}) = \mathbb{E}_{(\mathbf{x},y)\sim\nu}\mathbb{P}\left(\text{sign}(\hat{f}(\mathbf{x})) \neq y\right). \tag{3.6}$$

For $\hat{f}_{\text{erm}}$, another commonly used metric is the *test loss* :

$$\mathscr{L}_{\text{gen.}}(\hat{f}) = -\mathbb{E}_{(\mathbf{x},y)\sim\nu}\log(\sigma(y\hat{f}(\mathbf{x}))). \tag{3.7}$$

However, our key goal in this manuscript is to mathematically characterize the uncertainty associated to the prediction of the different classifiers above, and in particular how they correlate with the true class uncertainty as measured by $f_\star$. Mathematically, this can be measured by the following joint density:

$$\rho_{\star,t}(a,b) \coloneqq \mathbb{E}_{\mathscr{D}}\mathbb{P}_{\mathbf{x}}\left(f_\star(\mathbf{x}) = a, \hat{f}_t(\mathbf{x}) = b\right) \tag{3.8}$$

where $(a,b) \in [0,1]^2$ and $\hat{f}_t$, $t \in \{\text{bo}, \text{erm}, \text{Lap}, \text{eb}\}$ can be any of the classifiers defined above, and the expectation is taken both over the training data $\mathscr{D} = \{(\mathbf{x}^\mu, y^\mu)\}_{\mu\in[n]}$. In particular, this joint density gives access to different notions used in the literature to quantify uncertainty. For instance, a widely studied notion is the *calibration at level* $\ell \in [0,1]$ of a classifier $\hat{f}$:

$$\Delta_\ell(\hat{f}) \coloneqq \ell - \mathbb{E}_{\mathbf{x},\mathscr{D}}\left[f_\star(\mathbf{x}) | \hat{f}(\mathbf{x}) = \ell\right]. \tag{3.9}$$

A related metric is the *Expected Calibration Error* (ECE):

$$\text{ECE}(\hat{f}) \coloneqq \mathbb{E}_{\mathbf{x}}\left[|\Delta_{\hat{f}(\mathbf{x})}|\right]. \tag{3.10}$$

Note that in this work we focus on the calibration. Other uncertainty quantification metrics exist in the literature, e.g. the *Brier Score* and the *Maximum Calibration Error*, and although the theoretical methods presented here can be readily adapted to characterize their asymptotics, this is outside of the scope of this work.

3.2.2 The random features model

Following our aim to investigate the interplay between overparametrization and uncertainty, we will focus on one of the simplest settings of feature maps defined by two-layer neural networks $\varphi : \mathbf{x} \in \mathscr{X} \subset \mathbb{R}^p \mapsto \phi(F\mathbf{x})/\sqrt{p}$ with weights $F \in \mathbb{R}^{p\times d}$ and component-wise activation ϕ. We will consider *random features* [201], where the first layer weights $F \in \mathbb{R}^{p\times d}$ are fixed at initialization, typically taken to be i.i.d. standard Gaussian. Random features have been widely studied as a convex proxy for investigating the impact of overparametrization in generalization, since they were shown to display the characteristic non-monotonic *double descent* behaviour of the

generalization error [29, 217], with optimal generalization achieved beyond interpolation of the data [168, 94, 58], also known as *benign overfitting* [25].

We will assume Gaussian input data $\mathbf{x}^\mu \sim \mathcal{N}(\mathbf{0}, {}^1\!/_d \mathbf{I}_d)$ with labels drawn from a logit model:

$$f_\star(\mathbf{x}) = \int_{\mathbb{R}} \sigma\left(\theta_\star^\top \mathbf{x} + \tau z\right) \mathcal{N}(z|0,1)\mathrm{d}z \tag{3.11}$$

with random weights $\theta_\star \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ and $\tau \geq 0$ defines a tunable label noise level. This completely specifies the data distribution ν. In the following, we will be interested in the *proportional high-dimensional limit* defined by $n, p, d \to \infty$ with fixed ratios $\alpha := {}^n\!/_p$ and $\gamma := {}^p\!/_d$. While this *teacher-student* setup is quite common in high-dimensional statistics, we could make it more realistic by assuming a general covariance Ψ for the input $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, {}^1\!/_d \Psi)$.

An asymptotic characterization of the generalization and training errors of empirical risk minimization for the random features model in the proportional limit was derived for ridge regression in [168] and generalized to convex losses in [94]. A key ingredient in this analysis is a *Gaussian equivalence principle* [98, 99] proven in [109, 177] stating that the statistics of the empirical risk minimizer is asymptotically equal to the one of an equivalent Gaussian problem with matching moments. More recently, Gaussian equivalence has been proven for two-layer neural tangent features in [177] and features coming from mixture models in [93], and was conjectured to hold for a broader class including features coming from trained neural networks [153]. Although the discussion in this manuscript focus in the random features case, our analysis can be readily extended to all cases in which Gaussian equivalence holds. We provide in Appendix B.2 an extension of our main theoretical result to a general Gaussian covariate model with convex loss encompassing all these cases.

3.3 Results

3.3.1 Technical results

Let $\hat{\mu}_p$ denote the empirical spectral distribution of the matrix $\mathbf{F}\mathbf{F}^\top \in \mathbb{R}^{p\times p}$. In the following, we assume that in the proportional high-dimensional limit defined above, $\hat{\mu}_p$ weakly converges to an asymptotic spectral distribution μ on $\mathbb{R}_+$ with normalized second moment $\int \mu(\mathrm{d}x)x^2 = 1$. Further, assume $\kappa_0 = \mathbb{E}[\phi(z)]$, $\kappa_1 = \mathbb{E}[z\phi(z)]$ and $\kappa_\star^2 = \mathbb{E}[\phi(z)^2] - \kappa_1^2 - \kappa_0^2$ are all finite for $z \sim \mathcal{N}(0,1)$. Note that this assumption simply implies some mild regularity in the activation (e.g. that is does not grow too fast). All the commonly considered activations functions, e.g. ReLU, tanh, sigmoid, etc., satisfy these assumptions. Also, for simplicity of exposition, in the following we assume $\kappa_0 = 0$, which can always be obtained by letting $\phi \to \phi - \kappa_0$.

We also define the effective noise $\tau_{\text{add}}^2 = 1 - \mathbb{E}_{x\sim\mu}\left[\frac{\kappa_1^2 x}{\kappa_1^2 x + \kappa_\star^2}\right]$.

The first step is to characterize the density $\rho_{\star,t}$ with $t \in \{\text{bo}, \text{erm}, \text{Lap}, \text{eb}\}$ defined in eq. (3.8).

All relevant quantities depend on this density. In the asymptotic regime, the estimator $\hat{f}(\mathbf{x})$ is characterized by six quantities $(m, q, v, \hat{m}, \hat{q}, \hat{v})$ that are solutions of self-consistent equations.

Classifier	$g_t(y, \omega, v)$	$\hat{\pi}_t(x)$	$\hat{\tau}_t$
$\hat{f}_{\text{erm}}$	$\text{prox}_{\log\sigma(y\times\cdot)}(\omega)$	λ	0
$\hat{f}_{\text{Lap}}$	$\text{prox}_{\log\sigma(y\times\cdot)}(\omega)$	λ	$\mathbb{E}_{x\sim\mu}\left[\frac{\kappa_1^2 x+\kappa_\star}{\lambda+\hat{v}_\star(\kappa_1^2 x+\kappa_\star)}\right]$
$\hat{f}_{\text{eb}}$	$\partial\omega \log\int \sigma(\beta y\times z)\mathcal{N}(z\|\omega, v)\mathrm{d}z$	λ	$v^\star$
$\hat{f}_{\text{bo}}$	$\partial\omega \log\int \sigma_{\tau^2+\tau^2_{\text{add}}}(y\times z)\mathcal{N}(z\|\omega, v)\mathrm{d}z$	$\frac{\kappa_1^2 x}{(\kappa_1^2 x+\kappa_\star^2)^2}$	$v^\star+\tau^2+\tau^2_{\text{add}}$

Table 3.1: Auxiliary functions and value of $\hat{\tau}_t$ for the different classifiers defined in Sec. 3.2.1.

Theorem 3.3.1 (Joint density)**.** *Let $\mathcal{D} = \{(\mathbf{x}^\mu, y^\mu)\}_{\mu=1}^n$ denote data independently drawn from the model defined in Equation* (3.11)*. Consider $\hat{f}_t$, $t \in \{\text{bo}, \text{erm}, \text{Lap}, \text{eb}\}$ one of the classifiers defined in Sec. 3.2.1. Then, in the proportional high-dimensional limit where $n, d, p \to \infty$ with fixed $\alpha = {}^n/_p, \gamma = {}^p/_d$, the asymptotic joint density $\rho_{\star,t}$ defined in Equation* (3.8) *is given by $\rho^{\lim}_{\star,t}(a,b) = \lim_{p\to\infty} \rho_{\star,t}(a,b)$:*

$$\rho^{\lim}_{\star,t}(a,b) = \frac{\mathcal{N}\left(\begin{bmatrix}\sigma^{-1}_{\tau^2+\tau^2_{\text{add}}}(a)\\ \sigma^{-1}_{\hat{\tau}_t^2}(b)\end{bmatrix}\middle|\mathbf{0}_2, \Sigma_t\right)}{|\sigma'_{\tau^2+\tau^2_{\text{add}}}(\sigma^{-1}_{\tau^2+\tau^2_{\text{add}}}(a))||\sigma'_{\hat{\tau}_t^2}(\sigma^{-1}_{\hat{\tau}_t^2}(b))|} \tag{3.12}$$

where

$$\Sigma_t = \begin{bmatrix} 1 & m_t^\star \\ m_t^\star & q_t^\star \end{bmatrix} \tag{3.13}$$

and the sufficient statistics $(m_t^\star, q_t^\star, v_t^\star) \in \mathbb{R}^3$ are the unique fixed points of the following system of equations:

$$\begin{cases} v = 2\times\partial_{\hat{q}}\Psi_w(\hat{m}, \hat{q}, \hat{v}; \hat{\pi}_t) \\ q = 2\times(\partial_{\hat{q}}\Psi_w - \partial_{\hat{v}}\Psi_w)(\hat{m}, \hat{q}, \hat{v}; \hat{\pi}_t) \\ m = \sqrt{\gamma}\partial_{\hat{m}}\Psi_w(\hat{m}, \hat{q}, \hat{v}; \hat{\pi}_t) \end{cases} \tag{3.14}$$

$$\begin{cases} \hat{v} &= -\alpha\mathbb{E}_{\xi\sim\mathcal{N}(0,q)}\left[\sum_y \mathcal{Z}_0\left(y, {}^m/_q\xi, v_\star\right)\partial_\omega g_t\left(y,\xi,v\right)\right] \\ \hat{q} &= \alpha\mathbb{E}_{\xi\sim\mathcal{N}(0,q)}\left[\sum_y \mathcal{Z}_0\left(y, {}^m/_q\xi, v_\star\right) g_t\left(y,\xi,v\right)^2\right] \\ \hat{m} &= \sqrt{\gamma}\alpha\mathbb{E}_{\xi\sim\mathcal{N}(0,q)}\left[\sum_y \partial_\omega\mathcal{Z}_0\left(y, {}^m/_q\xi, v_\star\right) g_t\left(y,\xi,v\right)\right] \end{cases}$$

where $\mathcal{Z}_0(y,\omega,v) = \sigma_{v+\tau^2+\tau^2_{\text{add}}}(y\omega)$, $v_\star = 1 - {}^{m^2}/_q - \tau^2_{\text{add}}$. The functions g_t and $\hat{\pi}_t$ and the scalar

$\hat{\tau}_t$ depend on the estimator and the sufficient statistics, and are given in Table 3.1. Also :

$$\Psi_w(\hat{m},\hat{q},\hat{v};\hat{\pi}) = \frac{1}{2}\mathbb{E}_{x\sim\mu}\left[\frac{\hat{m}\kappa_1^2 x+\hat{q}(\kappa_1^2 x+\kappa_\star^2)}{\hat{\pi}(x)+\hat{v}(\kappa_1^2 x+\kappa_\star^2)}\right] - \frac{1}{2}\log(\hat{\pi}(x)+\hat{v}(\kappa_1^2 x+\kappa_\star^2))\,. \tag{3.15}$$

Proof idea: Let $\mathbf{x}\sim\mathcal{N}(\mathbf{0},{}^1\!/_d\mathbf{I}_d)$. For any of the classifiers $t\in\{\text{bo},\text{erm},\text{Lap},\text{eb}\}$ from Sec. 3.2.1, the 2-dimensional vector $(f_\star(\mathbf{x}),\hat{f}_t(\mathbf{x}))$ is asymptotically distributed as $(\sigma(z),\sigma_{\tilde{v}}(z_t'))$ for some $\tilde{v}$ that depends on the estimator, where $(z,z_t')\sim\mathcal{N}(\mathbf{0}_2,\Sigma_t)$, and

$$\Sigma_t = \frac{1}{d}\begin{pmatrix} ||\theta_\star||_2^2 & \hat{\theta}_t^\top\Phi\theta_\star \\ \theta_\star^\top\Phi^\top\hat{\theta}_t & \hat{\theta}_t^\top\Omega\hat{\theta}_t\end{pmatrix}$$

where we defined the shorthand $\Phi=\kappa_1\mathbf{F}\in\mathbb{R}^{p\times d}$ and $\Omega=\kappa_1^2\mathbf{F}\mathbf{F}^\top+\kappa_\star^2\mathbf{I}_p$ and $\hat{\theta}_t$ is either the unique minimizer of the empirical risk in eq. (3.2) for $t\in\{\text{erm},\text{Lap}\}$ or the mean over the respective posterior distribution for $t\in\{\text{bo},\text{eb}\}$. The computation of $\rho_{\star,t}$ thus boils down to computing the sufficient statistics $(m^\star,q^\star):=(\hat{\theta}_t^\top\Phi\theta_\star,\hat{\theta}_t^\top\Omega\hat{\theta}_t)$. For $\hat{f}_{\text{erm}}$ on the random features model, the theorem can be proven using recent work in high-dimensional statistics [168, 66, 153], where $(m^\star,q^\star)$ is proven to asymptotically obey a set of self-consistent "state-evolution" equations [97, 26, 69], mathematically equivalent to eqs. (3.14). A similar strategy was used in [53] for the simpler vanilla logistic model. This is discussed in Appendix B.2.3 where we show how to derive analogous results for $t\in\{\text{bo},\text{Lap},\text{eb}\}$.

Corollary 3.3.2 (Test error and calibration)**.** *Under the conditions of Theorem 3.3.1, the asymptotic generalization error and calibration are given by:*

$$\mathscr{E}_{\text{gen.}}^{\text{lim}} = \iint_{b<0.5,a} a\times\rho_{\star,t}^{\text{lim}}(a,b)\mathrm{d}a\mathrm{d}b + \iint_{b>0.5,a}(1-a)\times\rho_{\star,t}^{\text{lim}}(a,b)\mathrm{d}a\mathrm{d}b \tag{3.16}$$

$$\Delta_p^{\text{lim}} = p - \frac{\int a\times\rho_{\star,t}^{\text{lim}}(a,p)\mathrm{d}a}{\int\rho_{\star,t}^{\text{lim}}(a,p)\mathrm{d}a}\,. \tag{3.17}$$

Intuition of the technical results – The key intuition behind Theorem 3.3.1 is the fact that for the models considered here all the statistics of interest depend only on low-dimensional projections of the estimators and the features, i.e. the joint distribution of $\hat{\theta}_t^\top\varphi(\mathbf{x})$ and $\theta_\star^\top\mathbf{x}$. Even if the input data $\mathbf{x}$ is assumed Gaussian, the distribution of the features $\varphi(\mathbf{x})$ can be complicated. However, thanks to recent universality results in the high-dimensional statistics literature [99, 109,

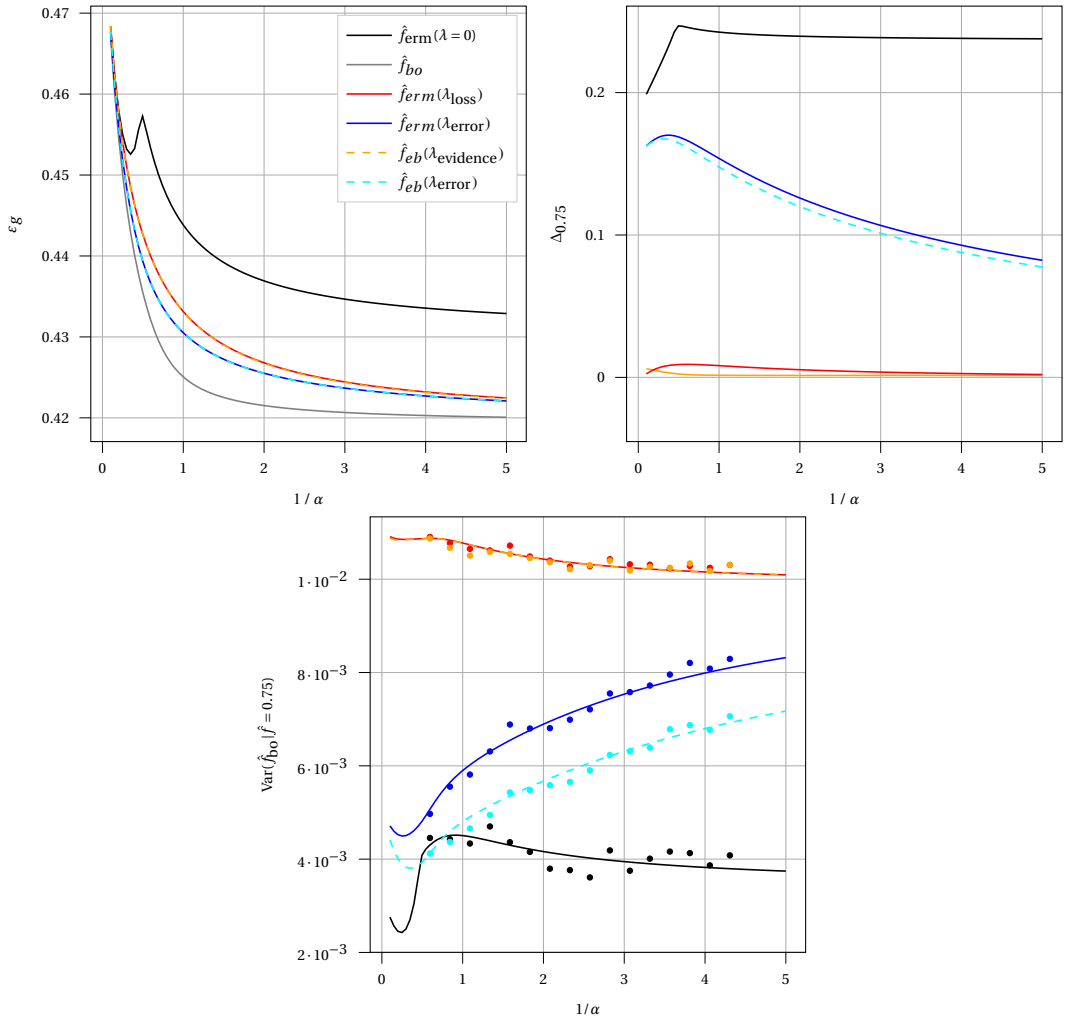


Figure 3.1: (**Left**) Test errors of the different methods as a function of the number of parameter per sample p/n. ERM, and Empirical Bayes (EB) are used with different penalizations. Here we use a logit teacher with $n/d = 2.0, \tau = 1/2$ and `erf` activation. The curves $\hat{f}_{\text{eb}}(\lambda_{\text{error}})$ and $\hat{f}_{\text{erm}}(\lambda_{\text{error}})$ are very close and indistinguishable on the plot, as well as the curves $\hat{f}_{\text{eb}}(\lambda_{\text{evidence}})$ and $\hat{f}_{\text{erm}}(\lambda_{\text{loss}})$. Due to the intrinsic noise in the model the oracle error is $\mathscr{E}^{\star}_{gen.} \simeq 0.332$. (**Center**) Calibration at a level $\ell = 0.75$. (**Right**) Variance of $\hat{f}_{\text{bo}}$ conditioned on the different other estimators. Points are experimental values obtained on Gaussian data at $d = 200$, averaged over 30 trials.

177, 59], in the high-dimensional limit of interest here the joint distribution of these projections are asymptotically captured by a Gaussian model with matching second moments $(m_t^{\star}, q_t^{\star})$, see Appendix B.1 for a detailed discussion. Moreover, these moments (which are the sufficient statistics for the quantities of interest) can be explicitly computed from the state evolution (3.14) of a tailored message passing scheme for each of the estimators $\hat{\theta}_t$, see Appendix B.2 for the technical details. This allow us to fully characterize all the quantities of interest asymptotically.

3.3.2 Trade-off between performance and uncertainty

In sensitive applications of machine learning having a reliable estimation of the model's uncertainty can be as important as having accurate predictions. Therefore, a key question is *"can my model achieve good generalization while being calibrated?"*.

Comparing the performances: In Figure 3.1 (left) we compare the misclassification test error eq. (3.6) of the different classifiers defined in Sec. 3.2.1[2] as a function of the overparametrization ratio p/n at fixed sample complexity $n/d = 2$ for different choices of the hyperparameters (β, λ). First, note the characteristic double descent behaviour of the empirical risk minimizer with $\lambda \to 0^+$, with the peak at the interpolating threshold corresponding in our setting to the existence of linear separator [207]. As discussed in e.g. [182] for neural networks and shown in e.g. [94] for random features classification, this peak is mitigated by cross-validation on the ℓ_2 regularization $\lambda > 0$, which is shown in Fig. 3.1 with the blue and red full lines, corresponding to optimally tuning λ to minimize the misclassification error eq. (3.6) and the test loss respectively eq. (3.7).

It is interesting to contrast these ERM estimators to the empirical Bayes classifier, which averages over different classifiers. We see that, evaluating the empirical Bayes with a Gaussian prior of variance given by the cross-validated λ_{error} achieves almost identical performance to the ERM estimator, with a difference of the order of 10^{-5}.

An often quoted strength of the Bayesian approach is that model selection can be performed directly on the training data by evidence maximization over the model hyperparameters [156]. Curiously, in our setting this yields a very close performance to ERM cross-validated with respect to the test loss, as shown in Fig. 3.1 (left) in dashed yellow line. Despite achieving similar performances in our setting, it is important to stress that these two classifiers are computationally radically different, as the empirical Bayes classifier requires sampling from a high-dimensional distribution which can be prohibitive in practice. These should be contrasted with the Bayes-optimal classifier, shown in solid grey, which by definition gives the best achievable performance at fixed data availability.

To summarise, from the point-of-view of the performance we observe no significant difference between Bayesian and ERM estimators, with (not surprisingly) best performance achieved by cross-validating over the misclassification error.

Calibration: Despite the relatively small difference in performance, the discussed classifiers are rather different in terms of calibration. Figure 3.1 (center) shows the calibration at fixed level $\ell = 0.75$ for the same classifiers. Note that the max-margin interpolator $\lambda \to 0^+$ produces consistently overconfident predictions. Indeed, we observe a maximum in the calibration curve around the interpolation threshold reminiscent of the double descent behaviour, with worst

[2]Note that by construction $\mathscr{E}_{\text{gen.}}(\hat{f}_{\text{Lap}}) = \mathscr{E}_{\text{gen.}}(\hat{f}_{\text{erm}})$.

possible calibration $\Delta_\ell = \ell - 1/2$ corresponding to a confidence completely uncorrelated with the true class probabilities achieved at the interpolation transition. As noted in [16], overconfidence is inherent for unregularized logistic regression in high-dimensions, as it is present even when data is abundant with respect to the number of parameters. However, in their simpler setting of matched linear classifiers the number of parameters is equal to the input dimension $p = d$, and therefore overparametrization cannot be distinguished from high-dimensionality. Indeed, they observe an asymptotic scaling of the calibration $\Delta_\ell \sim d/n$, which suggests that overconfidence increases with the number of parameters. Our setting allow us to decouple the number of parameters p from the data dimension d, suggesting instead that overparametrization can improve calibration at fixed number of samples.

More strikingly, we observe that optimal regularization does not mitigate this double descent-like behaviour in the calibration, which is in contrast with what happens with the error itself that becomes monotonic when optimally regularized. Indeed, while cross-validating with respect to the misclassification error achieves the best accuracy, it produces consistently overconfident predictions for both the empirical risk minimizer and the empirical Bayes classifiers. On the other hand, cross-validation with respect to the loss produces better calibrated estimates, with an interesting non-monotonic behaviour crossing from over- to underconfidence as a function of overparametrization. In contrast, maximising the evidence yields better calibrated estimation with a monotonic calibration curve very close to zero.

To summarise, we observe a fundamental trade-off between optimising the accuracy of classification and obtaining calibrated classifiers. A similar discussion holds for other calibration levels and for the expected calibration error eq. (3.10), as shown in Appendix B.5.

Conditional variance: Theorem 3.3.1 gives us access to a rich set of uncertainty measures, of which the calibration is a particular example. For instance, we have access to the full distribution of the Bayes-optimal classifier $\hat{f}_{\text{bo}}$ conditioned on the predictors defined in Sec. 3.2.1. Note that since $\mathbb{E}(\hat{f}_{\text{bo}}|\hat{f} = \ell) = \mathbb{E}(f_\star|\hat{f} = \ell) = \ell - \Delta_\ell(\hat{f})$, the mean of this conditional distribution is equal to the calibration up to a constant. A natural measure of uncertainty beyond the calibration is the variance of this conditional distribution $\text{Var}(\hat{f}_{\text{bo}}|\hat{f} = \ell)$, which quantifies how much the prediction $\hat{f}(\mathbf{x}) = \ell$ inform us on $\hat{f}_{\text{bo}}(\mathbf{x})$, which is by definition the best achievable classifier at finite availability of data. An explicit expression for this variance can be derived from Theorem 3.3.1 for any of the classifiers $t \in \{\text{erm}, \text{bo}, \text{Lap}, \text{eb}\}$:

$$\text{Var}(\hat{f}_{\text{bo}}(\mathbf{x})|\hat{f}_t(\mathbf{x}) = \ell) = \int \mathrm{d}a\, \sigma_{\hat{v}^\star_{\text{bo}}+\tau^2+\tau^2_{\text{add}}}(a)^2 \times \\ \times \mathcal{N}\left(a|^{m^\star_t}/_{q^\star_t}\sigma^{-1}_{\hat{\tau}_t}(\ell), q^\star_{\text{bo}} - {}^{m^{\star 2}_t}/_{q^\star_t}\right) - (\ell - \Delta_\ell)^2$$

where $(m^\star_t, q^\star_t, q^\star_{\text{bo}})$ are solutions to the self-consistent equations eq. (3.14) and Δ_ℓ is the asymptotic calibration eq. (3.17). The detailed derivations are shown in Appendix B.4.

Figure 3.1 (right) shows this conditional variance as a function of the overparametrization. Note

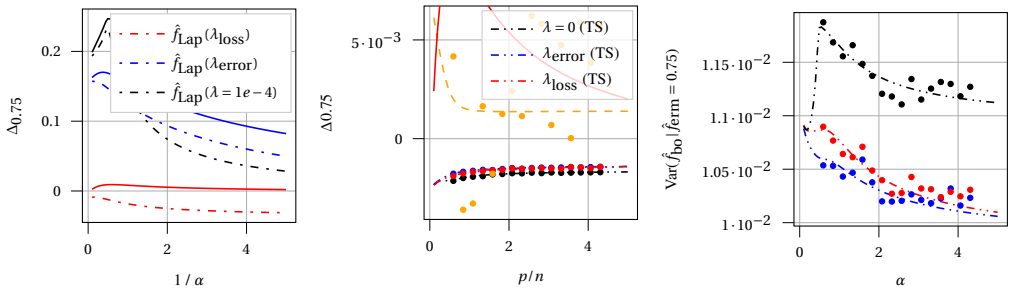


Figure 3.2: (**Left**) Calibration at level $\ell = 0.75$ of $\hat{f}_{\text{erm}}$ (solid lines, refer to Figure 3.1 for the legend) and $\hat{f}_{\text{Lap}}$ with the three different regularizations. (**Center**) Calibration at level $\ell = 0.75$ of $\hat{f}_{\text{erm}}$ after temperature scaling (TS), compared to $\hat{f}_{\text{eb}}$ (dashed yellow) and $\hat{f}_{\text{loss}}$ (full red) for reference. (**Right**) Variance of $\hat{f}_{\text{bo}}$ conditioned on $\hat{f}_{\text{erm}} = 0.75$ after temperature scaling, compared to variance at $\hat{f}_{\text{loss}}$ (full red) and $\hat{f}_{\text{eb}}$ (dashed yellow). Points are experimental values obtained on Gaussian data at $d = 200$, averaged over 30 trials.

that in this setting worse calibration is correlated with a lower conditional variance, and we observe a trade-off between these two metrics. We also observe a behaviour reminiscent of double descent in the value of the conditional variance that does not go away with optimal regularization.

3.3.3 Temperature scaling

Temperature scaling is a calibration method introduced in [101] to mitigate overconfidence in trained neural networks. It is applied after training, and consists in introducing a "temperature" scaling parameter on the last layer pre-activations $\hat{f}_{\text{erm}}(\mathbf{x}) = \sigma(\hat{\theta}_{\text{erm}}^\top \varphi(\mathbf{x})/T)$. It is then tuned to minimize the validation loss. In our analysis, this corresponds to simply re-scaling the predictor $\hat{\theta}_{\text{erm}} \to \hat{\theta}_{\text{erm}}/T$, & Thm. 3.3.1 thus applies mutatis mutandis.

Figure 3.2 (center) compares the calibration at level $\ell = 0.75$ of the regularized empirical risk minimizers with λ_{loss} and λ_{error} after temperature scaling with the empirical Bayes classifier with $\lambda_{\text{evidence}}$ and ERM classifier at λ_{loss}, the best calibrated in our setting so far. We observe that the temperature scaling yields very similar calibrations for λ_{loss} and λ_{error}. While empirical Bayes remains the best calibrated estimator, temperature scaling has a calibration around 0.1%, which would be satisfying in most practical scenarios. We also observe that the maximum around the interpolation threshold is not present in the calibration curves after temperature scaling.

Looking at the variance of $\hat{f}_{\text{bo}}$ conditioned on $\hat{f}_{\text{erm}}$ after temperature scaling we see that it is lower for λ_{error} than for λ_{loss}, see Fig. 3.2 (right). We see that again the variance has an increase in the vicinity of the interpolation threshold, reminiscent of the double descent behaviour. As discussed in the previous section, we aim to have the lowest variance possible to ensure that the uncertainty estimation is accurate not only on average but also point-wise. It appears that cross-validating the empirical risk minimizer on the misclassification error and then applying

temperature scaling gives an estimator that both has the best test error and is very well calibrated, both on average and point-wise.

3.3.4 The calibration of the Laplace approximation

Estimating any of the Bayesian classifiers in Sec. 3.2.1 is computationally demanding, since they involve a sampling over a high-dimensional distribution. This has motivated practitioners to develop different approximations for making Bayesian methods more efficient. These include Bayesian dropout [86], deep ensembles [143], stochastic gradient Langevin dynamics [238] and the Laplace approximation [205, 61], among others. The Laplace approximation was introduced by [155] in the context of Gaussian processes, and consists of approximating the posterior distribution by a Gaussian density centred around the empirical risk minimizer - or equivalently to a low-temperature expansion of the posterior - see eq. (3.5). By construction, the Laplace classifier has the same misclassification error as the empirical risk minimizer, and hence can be effectively seen as endowing this point-estimator with a covariance given by the inverse of the Hessian evaluated at the minimum. Although computing the Hessian of the empirical risk for a deep neural network can be costly, an approximate scheme has been recently proposed [205, 61], making Laplace a viable uncertainty estimation technique for deep learning.

On the theory side, sharp results have been limited to the Gaussian process and ridge regression setting, where the Laplace approximation is exact [215, 214]. While exact asymptotic results characterizing the statistics of the logit estimator in high-dimensions abound [219, 94, 13, 64], to our best knowledge the asymptotic spectral distribution of the Hessian at the minimum is missing. Recently, [148] has computed the asymptotic spectral distribution of the Hessian in a matched logit model under the assumption that the weights are uncorrelated with the input data. Hence, their results do not apply for the empirical risk minimizer, and cannot be used to characterize the uncertainty of the Laplace classifier. Characterizing the Hessian of the logistic risk eq. (3.2) at the minimizer is a challenging technical result that we believe is of independent interest to the scope of the discussion in this manuscript.

Claim 3.3.3 (Hessian of logit, informal). Let

$$\mathscr{H}(\theta) := \sum_{\mu\in[n]} (\sigma'(y^\mu\theta^\top\varphi(\mathbf{x}^\mu)) - 1)\varphi(\mathbf{x}^\mu)\varphi(\mathbf{x}^\mu)^\top + \lambda\mathbf{I}_p$$

denote the Hessian of the logistic empirical risk eq. (3.2), and denote $\hat{\theta}_{\text{erm}} = \text{argmin}_\theta \hat{\mathscr{R}}_n(\theta)$ its minimizer. Then, under the same conditions of Thm. 3.3.1 and additional technical assumptions, the following asymptotic characterization holds in the sense of deterministic equivalent:

$$\mathscr{H}^{-1}(\hat{\theta}_{\text{erm}}) \underset{p\to\infty}{\asymp} (\hat{v}^\star\left(\kappa_1^2\mathbf{F}\mathbf{F}^\top + \kappa_\star^2\mathbf{I}_p\right) + \lambda I_p)^{-1} \tag{3.18}$$

where $\hat{v}^\star \in \mathbb{R}$ is the solution of the self-consistent eq. (3.14).

A heuristic derivation of this result is provided in App. B.3 in the general context of the Gaussian

covariate model. With Claim. 3.3.3 in hands, we can characterize the asymptotic calibration of the Laplace classifier for our model.

Figure 3.2 (left) shows the calibration curve at level $\ell = 0.75$, at sample complexity $n/d = 2$ and noise variance $\tau_0 = 0.5$ as a function of the number of parameters. As mentioned in the introduction, we observe here that $\hat{f}_{\text{Lap}}$ is always less confident than $\hat{f}_{\text{erm}}$, due to the concavity of σ. While this might seem desirable in the scenarios where ERM is very overconfident, e.g. for $\lambda \to 0^+$ or λ_{error}, it hurts calibration when the classifier is well-calibrated as for λ_{loss}. Moreover, it highly depends on the sample complexity and noise variance, see Appendix B.5 in the supplementary material where we show a setting in which the Laplace approximation yields an underconfident classifier even in the $\lambda \to 0^+$ at mild overparametrization. Then, the Laplace approximation seems to be an unreliable way to control the calibration of the estimators, contrary to temperature scaling.

3.4 Verification on real data

In this section, we provide qualitative results on real data using a subset the MNIST dataset. We draw inspiration from the work done in [89] and train a fully connected neural network of depth $L = 5$ to classify 0 and 1 digits. Instead of using the original pixels as features, we compute the principal component analysis (PCA) of the training set and use the first 25 components as features. This choice of architecture and features is made to have a setting close to [89]. However, it would be interesting to investigage other architectures, as it has been observed in [101, 174] that the miscalibration of neural networks depends on the architecture. The models are trained with early stopping on the validation set, and the Adam optimizer is used with no weight decay. The shared width of the hidden layers is varied from 10 to 10^4 and observe in fig. 3.3 the error and calibration curve as a function of the width. We observe the interpolation threshold around $h = 2 \times 10^3$ where the training error reaches zero. Around this point, we observe a maximum in the expected calibration error (ECE), despite the fact that there is no double-descent in the test error. This observation is qualitatively consistent with our results from previous sections, where we noted that the calibration can exhibit a double-descent-like behaviour, even when the test error does not.

Failure of Laplace approximation We also compute the Laplace approximation by computing the Hessian of the empirical risk at the minimum, and then inverting it. Due to computational constraints, we only compute the Hessian with respect to the weights of the last layer, similarly to what is done in [40]. In the right-hand side of fig. 3.3, we observe that the ECE globally increases with the width, and then the Laplace approximation is a poor way to calibrate the model, as we observed in the previous sections.

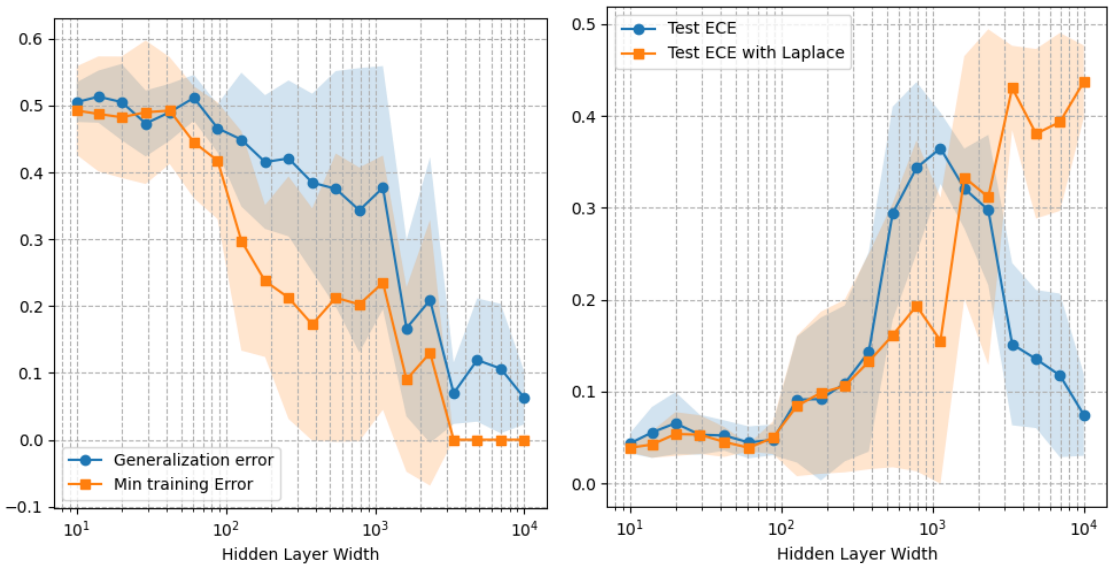


Figure 3.3: Test error (Left) and expected calibration error (Right) of a fully connected neural network trained on the MNIST dataset, as a function of the width of the hidden layers. The network has depth $L = 5$ and uses the first 25 principal components of the training set as features. Curves and shaded areas correspond to the mean and standard deviation over 10 runs.

3.5 Conclusion

In this paper, we studied the performance of different frequentist and Bayesian classifiers for random features classification. In the high-dimensional limit, the asymptotic behaviour of these algorithms can be precisely characterized. Our first contribution is the derivation of the Bayes-optimal estimator. By definition it is the estimator with the best possible performance, and although it is inaccessible in practice, it provides a baseline to compare the classifiers. Then, we compared the generalization error of frequentist and Bayesian approaches, showing they yield very similar test error. We then focused on uncertainty quantification, and showed there is a trade-off between generalization and calibration in our model. Moreover, we observed a non-monotonic behaviour of the calibration curve for certain estimators, akin to the famous *double-descent* phenomenon for the test error. Finally, we compared two popular approaches for post-training calibration: temperature scaling and the Laplace approximation, benchmarking them against the baseline classifiers. In our model, we observe that temperature scaling on top of cross-validating the empirical risk classifier on the accuracy achieves the best result : it has both the lowest test error and best calibration. Moreover, despite requiring a validation set, in practice it is a computationally more efficient method than the Bayesian approach, which requires sampling from a high-dimensional distribution. The code used in this project will be made available at github.com/SPOC-group/double_descent_uncertainty.

Limitations: It is worth pointing some limitation of our results. The first resides in the (nevertheless classical) Gaussian assumption for the data. We note, however, that there are good reasons to

believe that this can be a very good model in high-dimensions [109, 177]. A second limitation of is the lack of feature learning. While many of the uncertainty quantification methods discussed here apply directly to the last layer of trained neural networks, other methods considered in the literature apply to the full architecture [2]. Since the performance of deep neural networks can be largely attributed to feature learning, it shall be important to take it into account in theoretical studies of uncertainty. We hope that our work can offer a starting point towards this more ambitious goal.

Acknowledgements

We thank Pierre-Alexandre Mattei, Yevgeny Seldin, Anshuk Uppal, Kristoffer Stensbo-Smidt, Simon Bartels and Melih Kandemir for valuable discussions. We acknowledge funding from the ERC under the European Union's Horizon 2020 Research and Innovation Program Grant Agreement 714608-SMiLe, the Swiss National Science Foundation grant SNFS OperaGOST, 200021_200390 and the *Choose France - CNRS AI Rising Talents* program. This research was supported by the NCCR MARVEL, a National Centre of Competence in Research, funded by the Swiss National Science Foundation (grant number 205602).

4 Expectation consistency for calibration of neural networks

Authors : Lucas Clarté, Bruno Loureiro, Florent Krzakala, Lenka Zdeborová

4.1 Introduction

As deep learning models become more widely employed in all aspects of human society, there is an increasing necessity to develop reliable methods to properly assess the trustworthiness of their predictions. Indeed, different uncertainty quantification procedures have been proposed to measure the confidence associated with trained neural network predictions [2, 88]. Despite their popularity in practice, it is well known that some of these metrics, such as interpreting the last-layer softmax scores as confidence scores, lead to an overestimation of the true class probability [101]. As a consequence, various methods have been proposed to calibrate neural networks [86, 101, 157, 174].

In this work, we propose a novel method for the post-training calibration of neural networks named *expectation consistency* (EC). It consists of fixing the scale of the last-layer weights by enforcing the average confidence to coincide with the average classification accuracy on the validation set. This procedure is inspired by optimality conditions steaming from the Bayesian inference literature. Therefore, it provides a mathematically principled alternative to similar calibration techniques such as temperature scaling, besides being simple to implement and computationally efficient. Our goal in this work is to introduce the expectation consistency calibration method, illustrate its performance across different deep learning tasks and provide theoretical guarantees in a controlled setting. More specifically, our **main contributions** are:

• We introduce a novel method, *Expectation Consistency* (EC) to calibrate the post-training predictions of neural networks. The method is based on rescaling the last-layer weights so that the average confidence matches the average accuracy on the validation set. We provide a Bayesian inference perspective on expectation consistency that grounds it mathematically.
• While calibration methods abound in the uncertainty quantification literature, we compare EC to a close and widely employed method in the deep learning practice: *temperature scaling*

(TS). Our experiments with different network architectures and real data sets show that the two methods yield very similar results in practice.
• We provide a theoretical analysis of EC in a high-dimensional logistic regression exhibiting overconfidence issues akin to deep neural networks. We show that in this setting EC consistently outperforms temperature scaling in different uncertainty metrics. The theoretical analysis also elucidates the origin of the similarities between the two methods.

The manuscript is structured as follows : after a review of the literature and an exposition of the EC method (Section 3), we compare the performance of EC with TS on real data (Section 4) and show that the two methods behave similarly. In complement to Section 4, we provide in Section 5 a theoretical analysis of both methods and describe a synthetic setting in which EC outperforms TS.

The code used in this project is available at the repository https://github.com/SPOC-group/expectation-consistency

Algorithm 3 Expectation consistency (EC)

Input: Validation set $(\mathbf{x}_i, y_i)_{i=1}^{n_{val}}$, classifier $\hat{f} : \mathscr{X} \to \mathbb{R}^K$
Compute the logits $\mathbf{z}_i = \hat{f}(\mathbf{x}_i) \in \mathbb{R}^K$ and output $\hat{y}_i = \arg\max_k \mathbf{z}_{i\,k}$
Compute the accuracy on validation set $\mathscr{A}_{val} = \frac{1}{n_{val}} \sum_i \delta(y_i = \hat{y}_i)$
Determine T_{EC} such that $\frac{1}{n_{val}} \sum_i \max_k \sigma^{(k)}(\mathbf{z}_i/T) = \mathscr{A}_{val}$
Output: Temperature T_{EC}, and probabilities on new samples $\max_k \sigma^{(k)}(\mathbf{z}^{\mathrm{new}}/T_{\mathrm{EC}})$,

4.1.1 Related work

Calibration of neural networks — The calibration of predictive models, in particular neural networks, has been extensively studied, see [2, 88] for two reviews. In particular, modern neural network architectures have been observed to return overconfident predictions [101, 174]. While their overconfidence could be partly attributed to their over-parametrization, some theoretical works [16, 53, 53] have shown that even simple regression models in the under-parametrized regime can exhibit overconfidence.

There exists a range of methods that guarantee calibration asymptotically (i.e. when the number of samples is sufficiently large) without assuming anything about the data distribution, see e.g. [102]. However, for a limited number of samples, it is less clear which of the proposed methods provides the most accurate calibration.

Temperature scaling — [101] proposed *Temperature Scaling* (TS), a simple post-processing method consisting of rescaling & cross-validating the norm of the last-layer weights. Due to its simplicity and efficiency compared to other methods such as Platt scaling [198] or histogram binning [244], TS is widely used in practice to calibrate the output of neural networks [2]. Moreover, [53] has shown that in some settings, TS is competitive with much more costly Bayesian

approaches in terms of uncertainty quantification. While [102] has shown that without any assumption on the data model, injective calibration methods such as TS cannot be calibrated in general, [101] conclude that: "*Temperature scaling is the simplest, fastest, and most straightforward of the methods, and surprisingly is often the most effective.*" This justifies why TS is used so widely in practice.

Bayesian methods — Bayesian methods such as Gaussian processes allow estimating the uncertainty out of the box for a limited number of samples under (at least implicit) data distribution assumptions. When the data-generating process is known, the best way to estimate the uncertainty of a model is to use the predictive posterior. However, Bayesian inference is often intractable, and several approximate Bayesian methods have been adapted to neural networks, such as deep ensembles [143] or weight averaging [157]. On the other hand, the strength of posthoc methods like temperature scaling is that it applies directly to the unnormalized output of the network, and does not require additional training. A comparable Bayesian approach has been developed in [136], where a Gaussian distribution is applied to the last-layer weights. Bayesian methods typically involve sampling from a high-dimensional posterior [166], and different methods have been proposed to compute them efficiently [100, 86, 143, 157].

Notation — We denote $[n] := \{1, \cdots, n\}$; $\mathbf{1}(A)$ the indicator function of the set A; $\mathcal{N}(\mathbf{x}|\mu, \Sigma)$ the multi-variate Gaussian p.d.f. with mean μ and covariance Σ.

4.2 Setting

Consider a K-class classification problem where a neural network classifier is trained on a data set $(\mathbf{x}_i, y_i)_{i\in[n]} \in \mathbb{R}^d \times [K]$. Without loss of generality, for a given input $\mathbf{x} \in \mathbb{R}^d$ we can write the output of the classifier as a K-dimensional vector $\mathbf{z}(\mathbf{x}) = W\varphi(\mathbf{x}) \in \mathbb{R}^K$, where we have denoted the last-layer features by $\varphi : \mathbb{R}^d \to \mathbb{R}^p$ and the read-out weights $W \in \mathbb{R}^{K\times p}$. We define the *confidence* of the prediction for class k as:

$$\hat{f}(\mathbf{x}, k) := \sigma^{(k)}(\mathbf{z}(\mathbf{x})) = \frac{e^{z_k(\mathbf{x})}}{\sum_{l\in[K]} e^{z_l(\mathbf{x})}} \in (0, 1) \tag{4.1}$$

where $\sigma : \mathbb{R}^K \to (0, 1)^K$ is the softmax activation function. In short, $\hat{f}(\mathbf{x}, k)$ defines a probability, as estimated by the network, that $\mathbf{x}$ belongs to class k. For a given $\mathbf{x} \in \mathbb{R}^d$, the final *prediction* of the model is then given by $\hat{y}(\mathbf{x}) = \arg\max_k \hat{f}(\mathbf{x}, k) \in [K]$, and the associated prediction confidence $\hat{f}(\mathbf{x}) = \max_k \hat{f}(\mathbf{x}, k) = \hat{f}(\mathbf{x}, \hat{y}(\mathbf{x})) \in (0, 1)$. As it is common practice, in what follows we will be mostly interested in the case in which the network is trained by minimizing the empirical risk

(ERM) with the cross-entropy loss:

$$\begin{aligned}\ell(\hat{f}(\mathbf{x}), y) &= -\log \hat{f}(\mathbf{x}, y) \\ &= \sum_{k=1}^{K} \delta(y = k) \log \sigma^{(k)}(W\phi(\mathbf{x})),\end{aligned}$$

although many of the concepts introduced here straightforwardly generalize to other training procedures. The quality of the training is typically assessed by the capacity of the model to generalize on unseen data. This can be quantified by the test misclassification error and the test loss:

$$\mathscr{E}_g = \mathbb{E}_{\mathbf{x},y}\left[\delta\left(\hat{y}(\mathbf{x}) \neq y\right)\right], \quad \mathscr{L}_g = -\mathbb{E}_{\mathbf{x},y}\left[\log \hat{f}(\mathbf{x}, y)\right]$$

These are point performance measures. However, often we are also interested in quantifying the quality of the network prediction confidence. Different uncertainty metrics exist in the literature, but some of the most current ones are the *calibration*, *expected calibration error* (ECE) and *Brier score* (BS), defined as:

$$\begin{cases} \Delta_p & = p - \mathbb{P}_{\mathbf{x},y}\left(\hat{y}(\mathbf{x}) = y | \hat{f}(\mathbf{x}) = p\right) \\ \text{ECE} & = \mathbb{E}_{\mathbf{x},y}\left(|\Delta_{\hat{f}(\mathbf{x})}|\right) \\ BS & = \mathbb{E}_{\mathbf{x},y}\left(\sum_{k=1}^{K} (\hat{f}(\mathbf{x}, k) - \delta(y = k))^2\right) \end{cases} \tag{4.2}$$

Note that the Brier score is a proper loss, meaning that it is minimized when $\hat{f}(\mathbf{x}, k)$ is the true marginal distribution $\mathbb{P}(y = k|\mathbf{x})$. This is not the case of the ECE: indeed, the estimator defined as $\hat{f}(\mathbf{x}, k) = \mathbb{P}(y = k)$ has 0 ECE but does not correspond to the marginal distribution of y conditioned on $\mathbf{x}$ and has suboptimal test error. Finally, we introduce the confidence function with temperature $T > 0$:

$$\hat{f}_T(\mathbf{x}, k) = \sigma^{(k)}(^{W\varphi(\mathbf{x})}/_T). \tag{4.3}$$

4.3 Expectation consistency Calibration

The method proposed in this work acts similarly as the temperature scaling method [101] discussed in the related work section, with a key difference in how the temperature parameter is chosen. The popular and widely adopted temperature scaling (TS) procedure will also serve as the main benchmark in what follows.

Temperature scaling — Although the score-based confidence measure introduced in (4.1) might appear natural, numerical evidence suggests that for modern neural network architectures, it tends to be overconfident [101]. In other words, it overestimates the probability of class

Dataset	Model	$\mathscr{E}_g$	T_{TS}	T_{EC}	ECE	ECE_{TS}	ECE_{EC}	BS	BS_{TS}	BS_{EC}
SVHN	Resnet20	6.8 %	1.59	1.55	2.6 %	1.5 %	1.3 %	10.5 %	10.4 %	10.4 %
CIFAR10	Resnet20	13.5 %	1.37	1.38	5.3 %	1.9 %	1.9 %	20.0 %	19.43%	19.2 %
CIFAR10	Resnet56	13.1 %	1.42	1.43	6.0%	2.5 %	2.4 %	20.2 %	19.3 %	19.3 %
CIFAR10	Densenet121	12.5 %	1.78	1.86	7.9 %	3.0 %	2.5 %	20.4 %	18.6 %	18.5 %
CIFAR100	Resnet20	31.0 %	1.44	1.44	10.2 %	1.7 %	1.7 %	44.3 %	42.5 %	42.5 %
CIFAR100	Resnet56	27.3 %	1.73	1.79	14 %	2.6 %	2.2 %	41 %	38.0 %	38.0 %
CIFAR100	VGG19	26.4 %	2.14	2.28	19.9 %	5.3 %	4.8 %	44.8 %	37.2 %	36.9 %
CIFAR100	RepVGG-A2	22.5 %	1.07	1.16	5.3 %	4.6 %	4.4 %	32.1 %	31.9 %	32.0 %

Figure 4.1: Comparison of expected calibration error (ECE) and Brier score (BS) of temperature scaling (TS) and expectation consistency (EC) on various models and data sets. We see very minor differences between the two calibration methods. Given how well TS works in practice we conjecture at least the same for EC.

belonging. To mitigate overconfidence, [101] has introduced a post-training calibration method known as *temperature scaling* (TS) [174, 236]. Temperature scaling consists of rescaling the trained network output $\mathbf{z} \mapsto {}^{\mathbf{z}}/_{T}$ by a positive constant $T > 0$ (the "temperature") which is then be tuned to adjust the prediction confidence. Equivalently, TS can be seen as a re-scaling of the norm of the last-layer weights W. [101] has found that choosing T that minimizes the cross-entropy loss on the validation set $\{(\mathbf{x}_i, y_i)_{i\in[n_{val}]}$:

$$T_{\mathrm{TS}} = \arg\min_{T>0} \left(-\sum_{i=1}^{n_{val}} \ell(\hat{f}_T(\mathbf{x}_i, y_i)) \right) \tag{4.4}$$

results in a better calibrated rescaled predictor $\hat{f}_{T_{\mathrm{TS}}}$. To get a feeling for its effect on the confidence, it is instructive to look at the two extreme limits of TS. On one hand, if $T \ll 1$, the softmax will be dominated by the class with the largest confidence, eventually converging to a hard-thresholding $T \to 0^+$. This will typically lead to an overconfident predictor. On the other hand, for $T \gg 1$, the softmax will be less and less sensitive to the trained weights, converging to a uniform vector at $T \to \infty$. This will typically correspond to an underconfident predictor. Therefore, by tuning T, we can either make a predictor less overconfident (by lowering the temperature $T < 1$) or less underconfident (by increasing the temperature $T > 1$).

Temperature scaling is a specific instance of matrix/vector scaling, where the logits z_i are multiplied by a matrix/vector before the softmax. Despite being more general, matrix and vector scaling have been observed in [101] to perform worse than TS. Different variants of TS have been developed. Similarly to vector scaling, class-based temperature scaling [84] computes one temperature per class and finds the best temperature by minimizing the validation ECE instead of the validation loss. While TS can be naturally applied to the last-layer output of neural networks, [142] has extended TS to more general multi-class classification models.

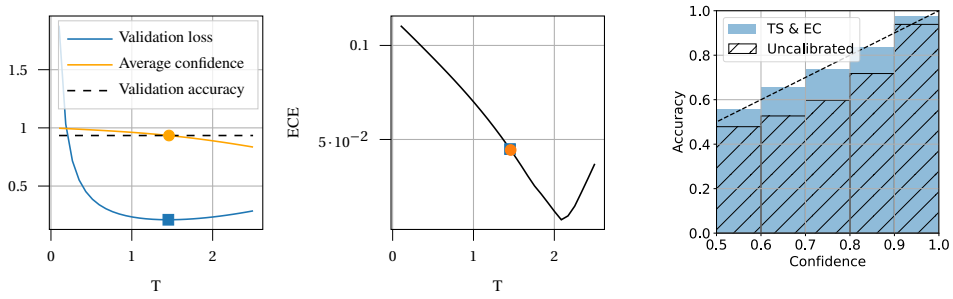


Figure 4.2: Left: The validation loss and average confidence of the model, as a function of the temperature T, model is DenseNet121 trained on CIFAR10. The dark dashed line is the accuracy for the validation set. Orange (respectively blue) cross corresponds to T_{EC}, T_{TS}. Middle: ECE of the model as a function of T, blue and orange dots respectively correspond to TS and EC. Right: Reliability diagram of Resnet20 trained on CIFAR10, before and after Temperature scaling. The reliability diagram after EC is indistinguishable from the one of TS.

Expectation consistency — In this work, we introduce a novel calibration method, which we will refer to as *Expectation Consistency* (EC). As for TS, the starting point is a pre-trained confidence function $\hat{f}$ which we rescale $\hat{f}_T$ by introducing a temperature $T > 0$. The key difference resides in the procedure we use to tune the temperature. Instead of minimizing the validation loss (4.4), we search for a temperature such that the average confidence is equal to the proportion of correct labels in the test set. In mathematical terms, we define T_{EC} such that the following is satisfied:

$$\frac{1}{n_{\mathrm{val}}} \sum_{i=1}^{n_{\mathrm{val}}} \hat{f}_{T_{\mathrm{EC}}}(\mathbf{x}_i) = \frac{1}{n_{\mathrm{val}}} \sum_{i=1}^{n_{\mathrm{val}}} \mathbf{1}(\hat{y}(\mathbf{x}_i) = y_i) \tag{4.5}$$

The intuition behind this choice is the following: a calibrated classifier is such that for all $p \in (0,1), \Delta_p = 0$. This condition is not achievable by tuning the temperature parameter T, so a less strict condition is to enforce it in expectation $\mathbb{E}_{\mathbf{x}}\left[\Delta_{\hat{f}(\mathbf{x})}\right] = 0$, ensuring that the classifier is calibrated on average. This is equivalent to enforcing the average confidence to be equal to the probability of predicting the correct class on a validation set. Note that the fact that we directly compare to the confidence on the validation set is analogous to what is done in the conformal prediction [192] methods to estimate prediction sets (as opposed to calibration that we are aiming at here).

It is instructive to consider a Bayesian perspective on EC. For the sake of this paragraph, assume that both the training and validation data were independently drawn from a parametric probability distribution $p(\mathbf{x}, y|\theta)$. If we had access to the distribution of the data (but not the specific realization of the parameters θ), the Bayes-optimal confidence function would be given by the expectation of $f_\star(\mathbf{x}|\theta) = p(y|\mathbf{x}, \theta)$ with respect to the posterior distribution of the weights given the training data $p(\theta|(\mathbf{x}_i, y_i)_{i\in[n]})$. In this case, one would not even need a validation set since the expected test accuracy would be predicted by the uncertainties under the posterior. In Section 4.5.1

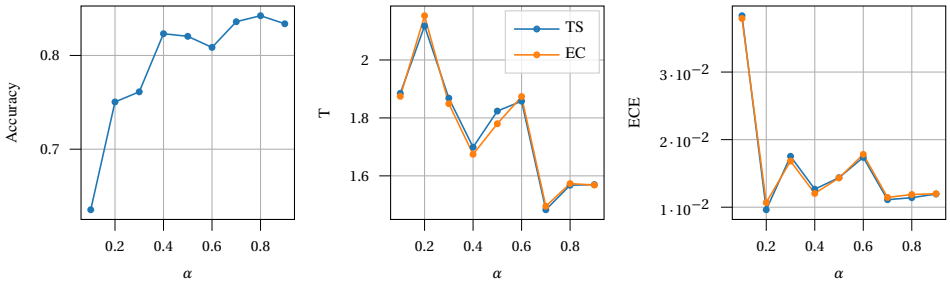


Figure 4.3: Left: Accuracy of Resnet20 model (Left), the temperature returned by TS and EC (Middle) and ECE of the model (Right) as a function of the size of the training set $\alpha = {}^{n_{\text{train}}}/_{50000}$. The model is trained with the same hyperparameters as in Figure 4.1. Again we see that the two methods are comparable even at largely different sample sizes.

.

we illustrate this discussion for concrete data distribution. This expectation consistency property of the Bayes-optimal predictor is known as the *Nishimori condition* in the information theory and statistical physics literature [112, 167, 247]. Therefore, from this perspective requesting condition (4.5) to hold can be seen as enforcing the Nishimori conditions for the rescaled confidence function. The Nishimori conditions are also used within the expectation-maximization algorithm for learning hyperparameters [63]. We describe in Section 4.5 how to interpret both temperature scaling and expectation consistency as learning procedures for the hyperparameter T. The main idea behind the EC method proposed here is that even in the absence of knowledge of the data-generating model, the expectation consistency (4.5) relation should hold for a calibrated uncertainty quantification method.

Note that T_{EC} exists and is unique. Indeed, the average confidence is a decreasing function of the temperature, converging to one when $T \to 0^+$ and to zero when $T \to \infty$. Therefore, there is a unique T_{EC} that satisfies the constraint (4.5), and in practice, it can be found by bisection. We refer to Figure 4.2 for an illustration of the uniqueness of T_{EC}. Moreover, note that expectation consistency is more flexible than temperature scaling: in multi-class classification problems, we can fix the temperature so that the average confidence is equal to the top N accuracy for any $N \in [K]$. In this work, we focus on the top 1 accuracy.

4.4 Experiments on real data

In this section, we present numerical experiments carried out on real data sets and compare the performance of EC and TS. As we will see, both methods yield similar calibration performances in practical scenarios.

Experimental setup — We consider the performance of the calibration methods from Section 4.3 in image classification tasks. Experiments were conducted on three popular image

classification data sets:

- SVHN [184] is made of colored 32×32 labelled digit images. Train/validation/test set sizes are 65931/7325/26032.
- CIFAR10 and CIFAR100 data sets [137], consisting of 32×32 colored images from 10/100 classes (dog, cat, plane, etc.), respectively. Train/validation/test sets sizes are 45000/5000/10000 images for CIFAR10, 50000/5000/5000 for CIFAR100.

We consider different neural network architectures adapted to image classification tasks: ResNets [106], DenseNets [110], VGG [213] and RepVGG [67]. For CIFAR100, pre-trained models available online were employed. More details on the training procedure are available in Appendix A.

Results — We refer to Table 4.1 for a comparison of TS and EC on the various data sets and models discussed above. Curiously, we observe that both EC and TS yield very similar temperatures across the different tasks and architectures, implying a similar ECE and Brier score. In particular, note that both methods give $T > 1$, consistent with the fact that the original networks were overconfident. Therefore, as expected, both methods improve the calibration of the classifiers.

The right panel of Figure 4.2 shows the reliability diagram of the ResNet 20 trained on CIFAR10: we observe that before applying TS and EC, the accuracy is lower than the confidence. In other words, the model is overconfident and both TS and EC improve the calibration of the model.

Note that both methods improve the Brier score and yield very similar results. From the computational cost perspective, EC is as efficient to run as TS, and requires only a few lines of code, see the GitHub repository where we provide the code to reproduce the experiments discussed here. However, we believe expectation consistency is a more principled calibration method, as it constrains the confidence of the model to correspond to the accuracy and has a natural Bayesian interpretation. Moreover, as we will discuss in Section 4.5, we can derive explicit theoretical results for EC.

Our experiments suggest that the similarity between TS and EC is independent of the accuracy of the model. Indeed, in Figure 4.3, we observe the accuracy and ECE of a ResNet model trained on different amounts of data. As expected, the accuracy of the model increases with the amount of training data. We observe in the middle and right panels that the temperatures and ECE obtained from both methods are extremely similar, independently of the accuracy of the model. Finally, we plot in the middle panel of Figure 4.2 the ECE as a function of the temperature and observe that neither T_{TS} nor T_{EC} is close to the minimum of ECE. However, as we have discussed in Section 4.3, ECE is only one uncertainty quantification metric and is not a proper loss, so we wish not to optimize the temperature for this metric in particular.

Experiments on corrupted data — In Appendix C we compare the performance of EC and TS on an image classification task where the test data is corrupted. We use the same datasets and architectures as in Section 4.4, but for some image classes, the target labels on the test data are randomly chosen. The goal of introducing a class-dependent noise is to evaluate both methods in a more realistic scenario where there is a distribution shift between the training and test data, as done in [108]. We report in Table 1 of the Appendix the performance of EC and TS in terms of ECE and Brier score. We observe that EC yields an reduction of the test ECE of 7 % on average, showing that EC is a competitive alternative to TS in more realistic scenarios. Note that in this setting, EC and TS yield different temperatures, contrary to the results described in Table 4.1. The full experimental details are described in Appendix C.

4.5 Theoretical analysis of the EC

As we have seen in Section 4.4, our experiments with real data and neural network models suggest that despite their different nature, EC and TS achieve a similar calibration performance across different architectures and data sets. In this section, we investigate EC and TS in specific settings where we can derive theoretical guarantees on their calibration properties.

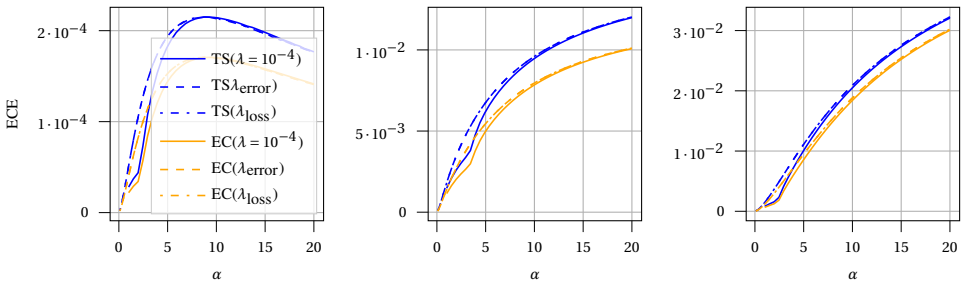


Figure 4.4: ECE of regularized logistic regression with three different values of λ ($10^{-4}, \lambda_{\text{error}}, \lambda_{\text{loss}}$): uncalibrated, after temperature scaling, and after expectation consistency. From left to right: $\sigma_\star = \sigma_{\text{logit}}, \sigma_{\text{affine}}, \sigma_{\text{constant}}$ respectively.

For concreteness, in the examples that follow we will focus on binary classification problems for which, without loss of generality, we can assume $y \in \{-1, +1\}$. In this encoding, the softmax function is equivalent to the logit $\sigma(t) := (1 + e^{-t})^{-1}$, and the hard-max is given by the sign function. Further, we assume that both the training $(\mathbf{x}_i, y_i)_{i\in[n]}$ and validation set $(\mathbf{x}_i, y_i)_{i\in[n_{val}]}$ were independently drawn from the following data generative model:

$$\begin{aligned} f_\star(\mathbf{x}) &:= \mathbb{P}(y^\mu = 1|\mathbf{x}^\mu) = \sigma_\star\left(\frac{\theta_\star{}^\top \mathbf{x}^\mu}{T_\star}\right) \\ \mathbf{x}^\mu &\sim \mathcal{N}(\mathbf{0}, {}^1\!/_d \mathbf{I}_d), \quad \theta_\star \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d) \end{aligned} \tag{4.6}$$

with $\sigma_\star : \mathbb{R} \to (0, 1)$ an activation function and $T_* > 0$ explicitly parametrizing the norm of the weights.

First, in Section 4.5.1 we provide a Bayesian interpretation of both the TS and EC methods, in an

example where $T_{\mathrm{TS}} = T_{\mathrm{EC}} = T_\star$. Next, in Section 4.5.2 we analyze a misspecified empirical risk minimization setting where they yield different results. Finally, we discuss in Section 4.5.3 one case in which EC consistently outperforms TS.

4.5.1 Relation with Bayesian estimation

Consider a Bayesian inference problem: given the training data $\mathscr{D} \coloneqq \{(\mathbf{x}_i, y_i)_{i\in[n]}\}$, what is the predictor that maximizes the accuracy? If the statistician had complete access to the data generating process (4.6), this would be given by integrating the likelihood of the data over the posterior distribution of the weights given the data:

$$f_{\mathrm{bo}}(\mathbf{x}) \coloneqq \mathbb{P}(y = 1|\mathscr{D}, \mathbf{x}) = \int \mathrm{d}\mathbf{w}\, \sigma\left(\frac{\mathbf{w}^\top \mathbf{x}}{T_*}\right) p(\mathbf{w}|\mathscr{D}, T_\star)$$

where the posterior distribution is explicitly given by:

$$p(\mathbf{w}|\mathscr{D}, T_\star) \propto \mathscr{N}(\mathbf{w}|0, \mathbf{I}_d) \prod_{i\in[n]} \sigma_*\left(y_i \frac{\mathbf{w}^\top \mathbf{x}_i}{T_\star}\right). \tag{4.7}$$

The Bayes-optimal predictor above is well calibrated [53], and consequently satisfies the expectation consistency condition: its average confidence equates to its accuracy. Consider now a scenario where the statistician only has *partial* information about the data-generating process: she knows the prior and likelihood but does not have access to the true temperature $T_\star$. In this case, she could still write a posterior distribution but would need to estimate the temperature T from the data. This can be done by finding the T that minimizes the classification error, or equivalently the generalisation loss, yet equivalently this would correspond to expectation maximization as discussed e.g. in [62, 141]. This estimation of the temperature would lead to $T = T_*$ and recovers the Bayes-optimal estimator f_{bo}. Hence, in the *well-specified* Bayesian setting, doing temperature scaling amounts to expectation consistency, providing a very natural interpretation of both the temperature scaling and expectation consistency methods in a Bayesian framework.

Note that in this paper we are concerned with frequentist estimators trained via empirical risk minimization. In that case, even in the well-specified setting, neither TS nor EC will recover the correct temperature T_* in the high-dimensional limit. This impossibility to recover T_* comes from the fact that we are not sampling a distribution anymore but instead consider a point estimate and do not have enough samples to be in the regime where point estimators are consistent.

4.5.2 Misspecified ERM

Consider now the case in which the statistician only has access to the training data $\mathscr{D}$, with no knowledge of the underlying generative model. A popular classifier for binary classification in

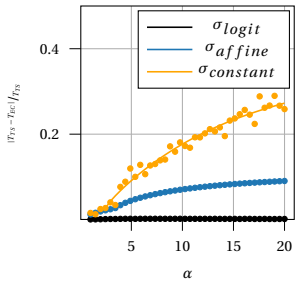


Figure 4.5: Relative difference $|T_{EC} - T_{TS}|/T_{TS}$ as a function of the sampling ratio α with three different $\sigma_\star$, and $\lambda = 10^{-4}$. We observe that when $\sigma_\star$ differs more from σ, EC and TS yield different results. Points are simulations done at $d = 200$.

this case is logistic regression, for which:

$$\hat{f}_{\text{erm}}(\mathbf{x}) = \sigma(\hat{\mathbf{w}}^\top \mathbf{x}) \tag{4.8}$$

and the weights are obtained by minimizing the empirical risk over the training data $\hat{\mathbf{w}} = \text{argmin}\hat{\mathscr{R}}_n(\mathbf{w})$ where:

$$\hat{\mathscr{R}}_n(\mathbf{w}) = -\sum_{i\in[n]} \log\sigma(\mathbf{w}^\top \mathbf{x}) + {}^{\lambda}/_{2}\|\mathbf{w}\|^2 \tag{4.9}$$

and we remind that σ is the sigmoid/logit function. In this setting, the calibration is given by $\Delta_\ell = \ell - \mathbb{E}_{\mathbf{x}}\left[f_*(\mathbf{x})|\hat{f}_{\text{erm}}(\mathbf{x}) = \ell\right]$, and the ECE by $\mathbb{E}_{\mathbf{x}}\left[|\Delta_{\hat{f}_{\text{erm}}(\mathbf{x})}|\right]$.

Note that logistic regression can also be seen as the maximum likelihood estimator for the logit model, which given the data model (4.6) for $\sigma_\star \neq \sigma$ is misspecified. [219] have shown that even in the well-specified case $\sigma_\star = \sigma$, non-regularized logistic regression yields a biased estimator of $\mathbf{w}_\star$ in the high-dimensional limit where $n, d \to \infty$ at a proportional rate $\alpha = {}^{n}/_{d}$, which [16] has shown to be overconfident. [53] characterized the calibration as a function of the regularization strength and the number of samples, and has shown that overconfidence can be mitigated by properly regularizing.

The goal in this section is to leverage these results on high-dimensional logistic regression in order to provide theoretical results on the calibration properties of TS and EC. In particular, we will be interested in comparing the following three choices of data likelihood function $\sigma_\star$:

$$\begin{cases} \sigma_{\text{logit}}(z) & = \frac{1}{1+e^{-z}} \\ \sigma_{\text{affine}}(z) & = 0 \text{ if } z < -1, 1 \text{ if } z > 1, \frac{t+1}{2} \text{ else} \\ \sigma_{\text{constant}} & = 0 \text{ if } z < -1, 1 \text{ if } z > 1, {}^{1}/_{2} \text{ else} \end{cases} \tag{4.10}$$

Asymptotic uncertainty metrics — The starting point of the analysis is to note that the uncertainty metrics of interest (4.2) only depend on the weights through the pre-activations $(\mathbf{w}_\star^\top \mathbf{x}, \hat{\mathbf{w}}^\top \mathbf{x})$ on a test point $\mathbf{x}$. Since the distribution of the inputs is Gaussian, the joint statistics of the pre-activations is Gaussian:

$$(\mathbf{w}_\star^\top \mathbf{x}, \hat{\mathbf{w}}^\top \mathbf{x}) \sim \mathcal{N}\left(\mathbf{0}_2, \begin{bmatrix} {}^1/_d ||\mathbf{w}_\star||_2^2 & {}^1/_d \mathbf{w}_\star^\top \hat{\mathbf{w}}_{\text{erm}} \\ {}^1/_d \mathbf{w}_\star^\top \hat{\mathbf{w}}_{\text{erm}} & {}^1/_d ||\hat{\mathbf{w}}_{\text{erm}}||_2^2 \end{bmatrix}\right)$$

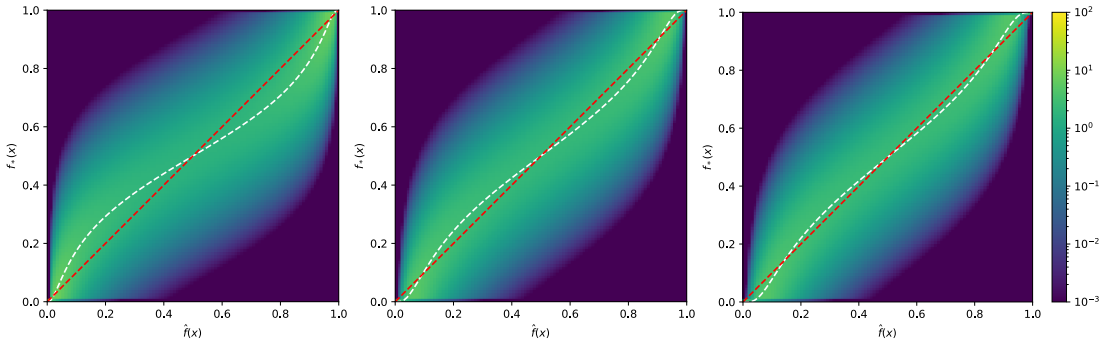


Figure 4.6: Plots of the density of $(\hat{f}_{\text{erm}}(\mathbf{x}), f_\star(\mathbf{x}))$ (Left), after Temperature scaling (Middle) and expectation consistency (Right), for the sampling ratio ${}^n/_d = 20$ and regularization $\lambda = 10^{-4}$. Dashed white lines represent the accuracy as a function of the confidence, the red line is the diagonal. The difference between red and white lines corresponds to the calibration. ECE of $\hat{f}_{\text{erm}}$ is, from left to right: 2.1 %, 1.2 %, 1.0 %. We have $T_{TS} = 1.24, T_{EC} = 1.35$.

As discussed above, different recent works [219, 16, 53] have derived exact asymptotic formulas for these statistics in different levels of generality for logistic regression. In particular, the following theorem from [53], which considers a general misspecified model will be used for the analysis:

Theorem 4.5.1 (Thm. 3.2 from [53])**.** *Consider the logit classifier* (4.8) *trained by minimizing the empirical risk* (4.9) *on a data set* $(\mathbf{x}_i, y_i)_{i\in[n]}$ *independently sampled from model* (4.6)*. Then, in the high-dimensional limit when* $n, d \to \infty$ *at fixed* $\alpha = {}^n/_d$*:*

$$({}^1/_d \mathbf{w}_\star^\top \hat{\mathbf{w}}_{\text{erm}}, {}^1/_d ||\hat{\mathbf{w}}_{\text{erm}}||_2^2) \xrightarrow[d\to\infty]{} (m, q) \tag{4.11}$$

where $(m, q) \in \mathbb{R}_+^2$ *are explicitly given by the solution of a set of low-dimensional self-consistent equations depending only on* $(\alpha, \lambda, \sigma, \sigma_*)$*, and which for the sake of space are discussed in Appendix B.*

Leveraging on Thm. 4.5.1, we can derive an asymptotic characterization for the asymptotic limit of the uncertainty metrics defined in (4.2).

Proposition 4.5.2. *Under the same assumptions of Theorem 4.5.1, the asymptotic limit of the*

uncertainty metrics defined in (4.2) *is given by:*

$$\begin{cases} \Delta_\ell(m,q) & = \ell - \mathcal{Z}_\star(1, {}^{m}/_{q}\sigma^{-1}(\ell), 1 - {}^{m^2}/_{q}) \\ \mathrm{ECE}(m,q) & = \int_0^\infty dz |\Delta_{\sigma(z)}(m,q)| \mathcal{N}(z|0,q) \end{cases}$$

where $(m,q) \in \mathbb{R}_+^2$ *are the asymptotic limits of the correlation functions in* (4.11) *and*

$$\mathcal{Z}_\star(y,\omega,V) = \mathbb{E}_{\xi\sim\mathcal{N}(\omega,V)}\left[\sigma_\star\left({}^{y\xi}/_{T_\star}\right)\right] \tag{4.12}$$

The proof of this result is given in Appendix B. Proposition 4.5.2 provides us with all we need to fully characterize the calibration properties of TS and EC in our setting. In the next paragraphs, we discuss its implications.

In practice, the ℓ_2 regularization parameter λ in the empirical risk (4.9) is optimized by cross-validation. [53, 53] has shown that appropriately regularizing the risk not only improves the prediction accuracy but also the calibration and ECE of the logistic classifier. In particular, it was shown that cross-validating on the loss function yields different results from cross-validation on the misclassification error, with a larger difference arising in the case of misspecified models. Curiously, [53] has shown that in this case, good performance and calibration can be achieved by combining a ℓ_2 penalty with TS. In the following, we discuss how this compares with EC. Note that the exact asymptotic characterization from Thm. 4.5.1 allows us to bypass cross-validation, allowing us to find the optimal λ by directly optimizing the low-dimensional formulas. We thus define λ_{error} (respectively λ_{loss}) as the value of λ such that $\hat{\theta}_\lambda$ yields the lowest test misclassification error (respectively test loss).

4.5.3 EC outperforms TS

In Section 4.4, we have numerically observed that EC and TS yield almost the same temperature and thus have similar performance in terms of different uncertainty quantification metrics for different architectures trained on real data sets. Figure 4.5 shows the relative difference $\delta T = {}^{|T_{TS}-T_{EC}|}/_{T_{TS}}$ between the two methods for logistic regression on the synthetic data model (4.6) for the different choice of target activation $\sigma_\star \in \{\sigma_{\text{logit}}, \sigma_{\text{affine}}, \sigma_{\text{constant}}\}$ defined in (4.10). Contrary to the real data scenario in Section 4.4, we observe a significant difference between the two methods for $\sigma_\star \in \{\sigma_{\text{affine}}, \sigma_{\text{constant}}\}$. For instance, for the piece-wise constant function $\sigma_\star = \sigma_{\text{constant}}$, δT is a non-decreasing function of the sampling ratio α, and is around 30% at $\alpha = 20$.

Figure 4.4 shows that expectation consistency yields a lower ECE than Temperature scaling in all the settings considered in Section 4.5. On one hand, the effect is small in the well-specified case where the target and model likelihoods are the same: the ECE of Temperature scaling is higher by around 0.01%. This is quite intuitive from the discussion in Section 4.5.1, since in this case, we are closer to the Bayesian setting where both methods were shown to coincide. On the other hand, this difference increases in the misspecified setting, suggesting that model misspecification plays

an important role in these calibration methods. In particular, note that in all cases considered here, EC has a lower ECE than TS for all three regularizations considered: $\lambda = 10^{-4}, \lambda_{\text{error}}, \lambda_{\text{loss}}$.

Figure 4.6 shows the joint probability density function of the variables $(f_\star(\mathbf{x}), \hat{f}_{\text{erm}}(\mathbf{x})) \in [0,1]^2$. In particular, we show in white-dashed lines the conditional mean $\mathbb{E}\left[f_*(\mathbf{x})|\hat{f}_{\text{erm}}(\mathbf{x})\right]$ which corresponds to the accuracy-confidence chart in Figure 4.2. As in the real data case, we observe that the ERM estimator is consistently overconfident, i.e $\forall \ell \geqslant 1/2, \Delta_\ell \geqslant 0$. Moreover, we see that after TS and EC, the conditional mean gets closer to the diagonal (red curve), implying that the model is more calibrated. The phenomenology of the simple data model seems to correspond to what we observe with real data and suggests that expectation consistency is a better approach to calibration.

Interpretation of the results — Temperature scaling corresponds to rescaling the outputs of the network by minimizing the validation loss. In the literature, the cross-entropy loss is one of the most widespread choices, both for training and for measuring uncertainty scores (with the softmax). From a Bayesian perspective, minimizing the cross-entropy loss corresponds to maximizing the likelihood under the assumption that the has been generated from a softmax (a.k.a. multinomial logit) model. Hence, the underlying assumption behind temperature scaling is that the labels are generated using a softmax likelihood. Therefore, we expect it to perform better when this assumption is met. Indeed, our experiments in Section 4.5 confirm this intuition. In the case where the ground truth model is indeed given by a logit, TS performs well and is close to EC. However, in the misspecified case, where this assumption does not hold, TS performs worse than EC.

4.6 Conclusion and future work

In this work, we introduced *Expectation Consistency*, a new post-training calibration method for neural networks. We have shown that EC is close to temperature scaling across different image classification tasks, giving almost the same expected calibration error and Brier score, while having comparable computational cost. Additionally, we provided an analysis of the asymptotic properties of both methods in a synthetic setting where data is generated by a ground truth model, showing that while EC and TS yield the same performance for well-specified methods, EC provides a better and more principled calibration method under model misspecification.

Our experiments on simple data models showed that when there is a discrepancy between our linear model and the true data model, EC performs better than TS. However, our experiments on real data show a very similar performance across different architectures, data models and overall model accuracy. In future work we aim to understand better why both methods are so similar in practical scenarios.

5 Bootstrap and subsampling in high dimensions

Authors : Lucas Clarté, Adrien Vandenbroucque, Guillaume Dalle, Bruno Loureiro, Florent Krzakala, Lenka Zdeborová

5.1 Introduction

Estimating and quantifying errors is a central aspect of statistical practice. Nevertheless, a solid understanding of how uncertainty can be reliably quantified in modern machine learning practice is largely missing, despite being a key endeavor towards a reliable use of these methods across sensitive applications. This chapter delves into a comprehensive mathematical analysis of conventional resampling methods to estimate uncertainty, such as subsampling, the bootstrap and the jackknife, specifically in the context of high-dimensional regression and classification tasks.

Let $Z_1, \cdots, Z_n \sim p_\theta$ denote n independent samples from a parametric probability distribution. Given an estimator $\hat{\theta}$ of θ (e.g. the maximum likelihood estimator), one is interested not only in the absolute performance of $\hat{\theta}$ but also in estimating how reliable it is, e.g. error bars. In particular, even if the estimator is consistent, i.e. $\hat{\theta} \to \theta$ when $n \to \infty$, having access only to a finite amount of data n introduces uncertainty in our estimation θ. A central question in statistics is *how to quantify this uncertainty* [237].

A classical family of non-parametric methods developed to address this question are *resampling methods* [232, 113], which consist in estimating the statistics of interest from the empirical distribution $p_n = {}^1/_n \sum_{i=1}^n \delta_{Z_i}$. Our goal is to investigate the statistical properties of three popular resampling methods in the context of the most widespread machine learning task: *supervised learning*. Here the samples are given by pairs $Z_i = (\mathbf{x}_i, y_i)$ from a joint distribution $p_\theta(\mathbf{x}, y)$, with $\mathbf{x}_i \in \mathbb{R}^d$ being the covariates and $y_i \in \mathscr{Y} \subset \mathbb{R}$ the labels. Given the parameter $\hat{\theta}$ learned by a fitting model, say ridge or logistic regression, the goal is to estimate the actual bias and variance of $\hat{\theta}$.

We focus on the *high-dimensional* regime, where both the number of samples n and their dimension d are comparatively large, with a fixed ratio $\alpha = n/d$. We provide a tight asymptotic

description of the biases and variances estimated by resampling methods for generalized linear models, such as ridge and logistic regression or any M-estimator. We show that resampling methods are fraught with problems in high-dimensions, either overestimating or underestimating the mean and variances. Reliable error estimation can only be reached in the regime when $\alpha \gg 1$, for which we provide asymptotic rates of convergences. However, in the overparametrized regime $\alpha < 1$, relevant to modern machine learning practice, the predictions of resampling methods are clearly off, even when optimally regularizing.

5.2 Setting & motivation

We consider the class of generalized linear estimation problems, where the goal is to estimate a parameter $\theta_\star \in \mathbb{R}^d$ from n independent samples $\mathscr{D} = \{(\mathbf{x}_i, y_i)_{i\in[n]}\}$ drawn from the following distribution:

$$y_i \sim p(\cdot|\theta_\star^\top \mathbf{x}_i), \qquad \mathbf{x}_i \sim \mathcal{N}(0, {}^1\!/_d \mathbf{I}_d) \tag{5.1}$$

for a general likelihood $p(y|z)$. Therefore, in this case, the joint distribution reads $p_{\theta_\star}(\mathbf{x}, y) = p(y|\theta_\star^\top \mathbf{x})p(\mathbf{x})$. For concreteness, we assume $\theta_\star \sim \mathcal{N}(0, \mathbf{I}_d)$. In the following, we focus on the (regularized) maximum likelihood estimator:

$$\hat{\theta}_\lambda(\mathscr{D}) = \underset{\theta\in\mathbb{R}^d}{\operatorname{argmin}} \sum_{i=1}^{n} -\log p\left(y_i|\theta^\top \mathbf{x}_i\right) + \frac{\lambda}{2}\|\theta\|_2^2 \tag{5.2}$$

also known as *empirical risk minimizer* in the context of supervised machine learning, where the loss function coincides with minus the empirical log-likelihood: $\ell(y, z) = -\log p(y|z)$. When it is clear from the context, we omit the training data dependence $\mathscr{D}$ in the MLE estimator and write $\hat{\theta}_\lambda$.

We will focus on two particular examples of generalized linear estimation: ridge and logistic regression. Ridge regression is a regression problem $\mathscr{Y} = \mathbb{R}$, which corresponds to the Gaussian likelihood $p(y|z) = \mathcal{N}(z|y, \Delta)$ (or equivalently the square loss function $\ell(y, z) = \frac{1}{2\Delta}(y - z)^2$) for $\Delta > 0$. Instead, logistic regression is a binary classification problem $\mathscr{Y} = \{-1, +1\}$ which corresponds to a logit likelihood $p(y|z) = \sigma(yz)$ for $\sigma(t) = (1 + e^{-t})^{-1}$ the logistic function (this corresponds to the logistic or cross-entropy loss function $\ell(y, z) = \log(1 + e^{-yz})$).

Note that the estimation problem introduced above is well-specified, and therefore enjoys strong mathematical guarantees in the classical statistical regime where $n \to \infty$ at fixed d. For instance, a well-known result is the asymptotic normality of the MLE for $\lambda = 0$ [237]:

$$\sqrt{n}\left(\hat{\theta}_0 - \theta_\star\right) \overset{(d)}{\to} \mathcal{N}(0, \mathscr{I}^{-1}), \qquad n \to \infty \tag{5.3}$$

where $\mathscr{I} \in \mathbb{R}^{d\times d}$ is the Fisher information matrix, in particular implying consistency and calibration of the maximum likelihood estimator. However, those guarantees break down when the

number of samples is comparable with the dimension of the covariates $n = \Theta(d)$. This is precisely the regime of interest in our work, and applying it to resampling methods will be our goal in the following.

5.2.1 What statisticians want

"Bias" and "variance" depend on the underlying data sampling process, and therefore, different notions co-exist, whether one takes, for instance, a frequentist or Bayesian viewpoint. Below, we define these different quantities, which resampling methods try to approximate.

Frequentist bias and variance — In the classical frequentist approach, the statistician seeks to estimate the bias and variance with respect to the data sampling process. This induces the classical *bias-variance decomposition* of the mean squared error for the estimator $\hat{\theta}_\lambda$:

$$\mathrm{MSE}(\hat{\theta}_\lambda) = \frac{1}{d}\mathbb{E}_{\mathscr{D},\theta_\star}\left[\|\hat{\theta}_\lambda - \theta_\star\|^2\right] = \mathrm{Bias}^2_{\mathscr{D}}(\hat{\theta}_\lambda) + \mathrm{Var}_{\mathscr{D}}(\hat{\theta}_\lambda)$$

with:

$$\mathrm{Bias}^2_{\mathscr{D}}(\hat{\theta}_\lambda) = \frac{1}{d}\left\|\mathbb{E}_{\mathscr{D},\theta_\star}\left[\hat{\theta}_\lambda\right] - \theta_\star\right\|^2 \tag{5.4}$$

$$\mathrm{Var}_{\mathscr{D}}(\hat{\theta}_\lambda) = \frac{1}{d}\mathbb{E}_{\mathscr{D},\theta_\star}\left[\left\|\hat{\theta}_\lambda - \mathbb{E}_{\mathscr{D},\theta_\star}\left[\hat{\theta}_\lambda\right]\right\|^2\right]. \tag{5.5}$$

We emphasize that in this case, the expectations are taken with respect to sampling of the full data set $\mathscr{D} = \{(\mathbf{x}_i, y_i)_{i\in[n]}\} \sim p_{\theta_\star}^{\otimes n}$.

Conditional bias and variance — Alternatively, in a supervised learning setting one can define the bias and variance only with respect to the sampling of the labels $y_i \sim p(\cdot|\mathbf{x}_i^\top\theta_\star)$, i.e. conditionally on the covariates $\mathbf{x}_i$. This is known as a *fixed design* analysis. We will refer to the corresponding notions as *conditional* bias and variance:

$$\mathrm{Bias}^2_{\mathscr{D}|\mathbf{X}}(\hat{\theta}_\lambda) = \frac{1}{d}\left\|\mathbb{E}_{\mathscr{D}}[\hat{\theta}_\lambda|\mathbf{X}] - \theta_\star\right\|^2 \tag{5.6}$$

$$\mathrm{Var}_{\mathscr{D}|\mathbf{X}}(\hat{\theta}_\lambda) = \frac{1}{d}\mathbb{E}_{\mathscr{D}}\left\|\hat{\theta}_\lambda - \mathbb{E}[\hat{\theta}_\lambda|\mathbf{X}]\right\|^2, \tag{5.7}$$

where for convenience we defined the covariate matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ with rows given by the covariates $\mathbf{x}_i \in \mathbb{R}^d$.

Bayesian estimator and variance — Finally, it is natural to compare the maximum likelihood estimator above with the best estimator (in mean squared error) conditioned on the full training data $\mathscr{D}$, also known as the *Bayes-optimal* estimator. It requires, however, the knowledge of the *a*

priori distribution of the "true" weights.

$$\hat{\theta}_{\text{bo}} = \underset{\hat{\theta}\in\mathbb{R}^d}{\text{argmin}}\, \mathbb{E}\left[\|\hat{\theta} - \theta_\star\|^2\right] = \mathbb{E}[\theta|\mathscr{D}] \tag{5.8}$$

where the conditional expectation is taken with respect to the posterior distribution:

$$p(\theta|\mathscr{D}) \propto \mathcal{N}(\theta|0, \mathbf{I}_d) \prod_{i=1}^{n} p(y_i|\theta^\top \mathbf{x}_i) \tag{5.9}$$

Note that, by definition, $\hat{\theta}_{\text{bo}}$ is an unbiased and calibrated estimator of $\theta_\star$ [53]. Nevertheless, it captures the irreducible variance due to the fact we have a finite sample $\mathscr{D}$ of the population distribution:

$$\text{Var}_{\text{bo}} = \frac{1}{d}\mathbb{E}\left[\|\theta - \theta_{\text{bo}}\|^2 |\mathscr{D}\right] \tag{5.10}$$

where, again, the expectation is taken over the posterior distribution $p(\theta|\mathscr{D})$.

5.2.2 Resampling estimates

A central problem in statistics is the estimation of the biases (5.4) & (5.6) and variances (5.5) & (5.7), which involve population expectations, from a finite number of samples $\mathscr{D} = \{(\mathbf{x}_i, y_i)_{i\in[n]}\}$. Resampling methods are a popular class of statistical procedures that fit a family of estimators $\hat{\theta}_k \equiv \hat{\theta}_\lambda(\mathscr{D}_k^\star)$ from resampled data $\mathscr{D}_k^\star$ generated from the original samples $\mathscr{D} = \{(\mathbf{x}_i, y_i)_{i\in[n]}\}$, and from which the bias and variance of $\hat{\theta}_\lambda$ can be estimated:

$$\widehat{\text{Bias}}^2 = \frac{1}{d}\left\|\frac{1}{B}\sum_{k=1}^{B}\hat{\theta}_k - \hat{\theta}_\lambda\right\|^2, \tag{5.11}$$

$$\widehat{\text{Var}} = \frac{1}{dB}\sum_{k=1}^{B}\left\|\hat{\theta}_k - \frac{1}{B}\sum_{k=1}^{B}\hat{\theta}_k\right\|^2 \tag{5.12}$$

In this work, we will focus on the following methods:

- **Pair bootstrap:** Consists in resampling $\mathscr{D}_k^\star$ from $\mathscr{D}$ with sample replacements, or in other words, sampling $\mathscr{D}_k^\star = \{(\mathbf{x}_{k,i}^\star, y_{k,i}^\star)_{i\in[n]}\} \sim p_n^{\otimes n}$ from the empirical distribution. For simplicity, we always assume $B = n$ for the bootstrap in the following.

- **Residual bootstrap:** Akin to the pair bootstrap method, but for the conditional distribution $p(y|z)$. In practice, one first fits an estimator $\hat{\theta}_\lambda(\mathscr{D})$ on the original samples (the MLE (5.2) in our setting), and given a statistical model for $\hat{p}(y|z)$, one resamples only the labels from $\hat{p}(y|\hat{\theta}_\lambda(\mathscr{D})^\top \mathbf{x}_i)$, generating new datasets $\mathscr{D}_k^\star = \{\mathbf{x}_i, y_{k,i}^\star\}_{i=1}^n$. This allows for the estimation of conditional statistical errors.

- Subsampling: Consists of generating new datasets $\mathscr{D}_k^\star$ of a smaller size $\lfloor r\,n \rfloor$ by subsampling $\mathscr{D}$ without replacement, where $r \in (0,1)$. While bootstrap creates datasets of the right size but from the wrong distribution (as elements of $\mathscr{D}$ are duplicated), subsampling relies on data of the wrong size but from the right distribution.[1]

- Jackknife: Consists of creating $B = n$ datasets $\mathscr{D}_k^\star = \{(\mathbf{x}_i, y_i)_{i \neq k}\}$, each of which leaves a single sample out. Note that when $n \to \infty$, as in our high-dimensional regime, this is equivalent to subsampling with $r \to 1$.

For notational convenience, we will refer to these statistics as $\widehat{\mathrm{Bias}_k^2}, \widehat{\mathrm{Var}_t}$ with $t \in \{\mathrm{pb}, \mathrm{rb}, \mathrm{ss}, \mathrm{jk}\}$ for pair (pb) and residual bootstrap (rb), subsampling (ss) and jackknife (jk).

The resampling methods above have been widely studied in the classical statistical literature, with whole books dedicated to proving their mathematical soundness [74, 60, 60]. However, as discussed in section 5.2 most of the classical guarantees hold in the regime where the quantity of data n available to the statistician is large in comparison with data dimension d — a regime that falls short in the context of modern machine learning practice. Of particular importance was the work of Karoui and Purdom [131] who have pointed out the lack of consistency of the bootstrap method for *unregularized* least squares, in the *underparametrized regime* $n > d$. One of our goals in this manuscript is to fill the gap, providing a complete evaluation of the aforementioned methods (beyond bootstrap), including the effect of regularization and over-parametrization.

More precisely, our **main contributions** are:

- We provide a closed-form expression for the biases and variances in the proportional high-dimensional limit where $n, d \to \infty$ at fixed rate $\alpha = n/d$ for all the cases discussed in section 5.2: the pair and residual biases and variances and their bootstrap, subsample, and jackknife estimates. Our result holds for generic log-concave likelihoods (corresponding to convex losses) and convex regularizers.
- Our formulas are derived from mapping to an Approximate Message Passing (AMP) scheme admitting a rigorous asymptotic characterization in terms of *state evolution* equations [26, 27, 114, 78, 154]. We believe this derivation has an interest on its own, as we show how simultaneously tracking *coupled* AMP trajectories provides the biases and variances for all the resampling methods. Our construction is quite generic and can be extended to other variants of interest.
- Our examination into the effectiveness and limitations of these methods yields three key insights. Firstly, we demonstrate that resampling techniques face significant challenges in high-dimensional contexts, resulting in a double-descent behavior typical of such scenarios. Secondly, we find that these methods yield consistent and reliable error estimates only when the ratio α is sufficiently large, for which we also present convergence rates. Thirdly,

[1] Since $\mathscr{D}_k^\star$ are independent conditionally on $\mathscr{D}$.

in the overparametrized regime where $\alpha < 1$, the predictions remain inconsistent despite optimal regularization.

Related work — Resampling methods are a classical topic in statistics. The jackknife method was introduced in Quenouille [200], refined by Tukey [234] and analysed by Efron and Stein [75]. Bootstrap was introduced by Efron [74], and studied in the context of least squares estimation in Freedman [83] and Wu [242].

The impact of high-dimensionality for the bootstrap method was first investigated by [131] in the context of unregularized M-estimation with $n > d$, where it was shown that methods that pair bootstrap under-estimates the true variance, while residual bootstrap overestimates it.

The asymptotic theory of high-dimensional statistical generalized linear problems has witnessed a burst of activity over the last decades. Pioneered by the statistical physics community in the late 80s [87, 190, 138, 211, 125], it is now an established field of research encompassing applications to machine learning, statistics, and signal processing among others [27, 77, 69, 229, 228, 68, 219, 220, 96, 225, 154, 153, 32, 30]. Bayes-optimal generalization guarantees for generalized linear models were established by Donoho, Javanmard, and Montanari [71], Krzakala et al. [140], Barbier et al. [22], and Maillard et al. [159]. [220] have shown that, besides not being well-defined when $n < d$, the unregularized maximum likelihood estimator is biased [77, 128, 28, 219, 31] for $n > d$. One consequence is that the variance of the MLE underestimates the true variance of $\theta_\star$, leading to an overconfident prediction [16, 17, 53]. Indeed, Clarté et al. [53, 52] highlighted the importance of properly regularizing the MLE in the high-dimensional regime, showing that cross-validation over λ can mitigate some of these issues. Clarté et al. [51] showed that post-training *temperature scaling* can mitigate overconfidence, regardless of the regularization used.

Bagging (the combination of subsampling with ensembling) has been studied in the high-dimensional regime by [216, 139, 145, 193, 73, 46, 8, 194]. Ensembling has also been investigated in the context of the random features model as a tool to decouple the different sources of randomness [58, 149, 3, 152]. The performance of *AdaBoost* [208] and its link to minimum ℓ_1-norm classifiers were studied in [146]. The performance of bootstrap averaging has been studied in the context of Gaussian Processes and Support Vector Machines using the replica method by [161, 162]. A replicated AMP algorithm for computing bootstrap averages of GLMs was proposed by [226] and studied in the context of LASSO [189] and Elastic Net [224].

Finally, we note that resampling methods in the context of generalized linear models are not just theoretical abstractions but are used in machine learning practice. For instance, Musil et al. [180] use subsampling to estimate the uncertainty in kernel regression for the energy of molecular compounds. Their observation that subsampling yields a better uncertainty estimation than Bootstrap or Gaussian processes is one motivation for the present work.

Consistency of bootstrap and subsampling in high-dimensions — Resampling methods such as subsampling and bootstrap have been studied in various tasks in the high-dimensional regime beyond supervised learning :

- The authors of [243] studied the estimation of the eigenvalues of a covariance matrix using the bootstrap, and showed that the estimation of the leading-eigenvalues is consistent in high dimensions. The main assumption that allows the bootstrap to work in their setting is the one that despite being high-dimensional, the covariance matrix to be estimated has an effective rank scaling as $\sqrt{n}$ at most. While the estimation problem studied in this chapter differs from theirs, note that in our setting the covariance of the data will be the identity, and the assumption of low effective rank does not hold.

- Additonally, [169] studied stability selection, a method for variable selection in high-dimensional linear models : in the case of Lasso regression where the parameter to infer is sparse, stability selection combines subsampling with randomizing the L_1 regularization to select the variables in the regression. The authors show that under some constraints on the sparsity and on the regularity of the input matrix, variable selectoin with stability selection is consistent, in the sense that the method selects the right variables with high probability. The setting of this work differs with the present chapter, as we do not assume the true parameter to be sparse.

Both works cited above show that resampling-based methods can be consistent in high-dimensions when the effective dimensionality of the problem, either through the input covariance matrix or the teacher's parameters, is much lower than the true dimension. This will not be our case, which explains why the bootstrap will fail in our specific setting. However, real data often presents a low effective dimensionality, which can explain why resampling methods such as bootstrap and subsampling work in practice.

5.3 Main technical results

The key observation in the results that follow is that in order to asymptotically characterize the biases and variances associated with any of the resampling methods in section 5.2, it is sufficient to characterize only a few correlations. For example, the resampling variance (5.12):

$$\widehat{\mathrm{Var}} = \frac{1}{d}\left(\frac{1}{B}\sum_{k=1}^{B}\|\hat{\theta}_k\|^2 - \frac{1}{B^2}\sum_{k,k'=1}^{B}\hat{\theta}_k^\top\hat{\theta}_{k'}\right). \tag{5.13}$$

Assuming the data sets $\mathscr{D}_k^\star$ are independently resampled from $\mathscr{D}$, it is then enough to characterize the norm of $\hat{\theta}_1$ and the correlation between two independent (conditionally on $\mathscr{D}$) resampled estimators $\hat{\theta}_1^\top\hat{\theta}_2$ - with all the rest being statistically similar. The results that follow precisely characterize these quantities asymptotically.

Finally, the methods defined in section 5.2 naturally divide into two categories: estimators for the statistics of the joint distribution $p_{\theta_\star}(\mathbf{x}, y)$ (we refer to them as *pair resampling*) and for the conditional distribution $p(y|{\theta_\star}^\top \mathbf{x})$ (we refer to them as *conditional* or *residual resampling*). Below, we start by discussing our results for the former.

5.3.1 Pair resampling

The key idea is to reframe the regularized MLE problem (5.2) as a *weighted empirical risk minimization* (wERM) problem:

$$\hat{\theta}_\lambda\left(\mathscr{D}, \mathbf{p}\right) = \operatorname*{arg\,min}_{\theta\in\mathbb{R}^d} \sum_{i=1}^{n} -p_i \log p\left(y_i|\theta^\top \mathbf{x}_i\right) + \frac{\lambda}{2}\|\theta\|^2 \tag{5.14}$$

where for each sample $(\mathbf{x}_i, y_i) \in \mathscr{D}$, we have introduced a sample weight p_i. When $p_i = 1$ for all $i \in [n]$, this reduces to standard MLE (5.2), which we sometimes refer to as full resampling. However, by taking the p_i's at random from a judiciously chosen distribution, we can asymptotically cover all pair resampling methods from section 5.2.

Indeed, it is immediate to see that by choosing $p_i \in \{0, 1\}$ at random from a Bernoulli distribution with probability $r \in (0, 1]$, the wERM (5.14) asymptotically corresponds to doing subsampling. Intuitively, this can be seen as throwing a coin for each sample $i \in [n]$ in order to decide whether to include it in the subsampled batch $\mathscr{D}^\star_{\text{ss}}$, which on average will contain precisely r samples. The jackknife estimator can then be obtained as the $r \to 1^-$ limit of subsampling.

Similarly, pair bootstrap is asymptotically equivalent to taking $p_i \sim \text{Pois}(1)$ independently. Indeed, for finite n, pair bootstrap exactly corresponds to taking $\mathbf{p} \in \mathbb{R}^n$ from the multinomial distribution $\text{Multinomial}(n, {}^1\!/_n)$. As $n \to \infty$, this is marginally equivalent to choosing $p_i \sim \text{Pois}(1)$ independently [131].

To summarize, each resampling method can be thought of as applying sampling weights which are i.i.d., with distributions defined as

$$\begin{cases} \mu_{\text{pb}}(p) & := \frac{1}{ep!} \\ \mu_{\text{ss(r)}}(p) & := r^p(1-r)^{1-p} \text{ for } r \in (0, 1). \end{cases} \tag{5.15}$$

We note that a key assumption which permits to retrieve our result is that for a particular resampling method, the sample weights p_i, $i \in [n]$ are *i.i.d.*. We are now ready to state our first two results for pair resampling. For the sake of clarity, we state our results for ridge regression and refer to appendix D.1 for the derivation of our results and a statement for general convex loss and penalties.

In the following, the asymptotic values of correlations needed to compute biases and variances

will be referred to as *overlaps*. For $\mathrm{t} \in \{\mathrm{pb}, \mathrm{ss}, \mathrm{jk}\}$, these overlaps read:

$$\begin{cases} Q_{11}^{\mathrm{t}} & := \lim_{n,d\to\infty} \mathbb{E}_{\theta_\star,\mathscr{D},\mathbf{p}} \left[\|\hat{\theta}_\lambda(\mathscr{D},\mathbf{p})\|^2 \right] \\ Q_{12}^{\mathrm{t}} & := \lim_{n,d\to\infty} \mathbb{E}_{\theta_\star,\mathscr{D}} \left[\|\mathbb{E}_{\mathbf{p}}[\hat{\theta}_\lambda(\mathscr{D},\mathbf{p})]\|^2 \right] \\ Q_{11}^{\mathrm{fr}} & := \lim_{n,d\to\infty} \mathbb{E}_{\theta_\star,\mathscr{D}} \left[\|\hat{\theta}_\lambda(\mathscr{D})\|^2 \right] \\ Q_{12}^{\mathrm{fr}} & := \lim_{n,d\to\infty} \mathbb{E}_{\theta_\star} \left[\|\mathbb{E}_{\mathscr{D}}[\hat{\theta}_\lambda(\mathscr{D})]\|^2 \right] \\ Q_{12}^{\mathrm{fr,t}} & := \lim_{n,d\to\infty} \mathbb{E}_{\theta_\star,\mathscr{D},\mathbf{p}} \left[\hat{\theta}_\lambda(\mathscr{D})^\top \hat{\theta}_\lambda(\mathscr{D},\mathbf{p}) \right] \\ m_1^{\mathrm{t}} & := \lim_{n,d\to\infty} \mathbb{E}_{\theta_\star,\mathscr{D},\mathbf{p}} \left[\hat{\theta}_\lambda(\mathscr{D},\mathbf{p})^\top \theta_\star \right] \\ m_1^{\mathrm{fr}} & := \lim_{n,d\to\infty} \mathbb{E}_{\theta_\star,\mathscr{D}} \left[\hat{\theta}_\lambda(\mathscr{D})^\top \theta_\star \right] \end{cases}, \tag{5.16}$$

where $\mathbf{p} = (p_1, \dots, p_n) \overset{\text{i.i.d.}}{\sim} \mu_{\mathrm{t}}$.

Theorem 5.3.1 (Biases and Variances for pair resampling in ridge regression)**.** *Let $\mathscr{D} = \{(\mathbf{x}_i, y_i)_{i\in[n]}\}$ denote n independent samples drawn from model* (5.1) *with log-concave likelihood $p(y|z)$. In the high-dimensional proportional regime $n, d \to \infty$ with $n/d = \alpha$, the overlaps of interest* (5.16) *are given by the unique solution $\mathbf{m} \in \mathbb{R}^2$, $\mathbf{Q} \in \mathbb{R}^{2\times 2}$, $\mathbf{V} \in \mathbb{R}^2$ to the following set of self-consistent equations:*

$$\begin{cases} \mathbf{m} & = \left(\lambda \mathbf{I}_2 + \hat{\mathbf{V}}\right)^{-1} \hat{\mathbf{m}} \\ \mathbf{Q} & = \left(\lambda \mathbf{I}_2 + \hat{\mathbf{V}}\right)^{-1} \left(\hat{\mathbf{m}}\hat{\mathbf{m}}^\top + \hat{\mathbf{Q}}\right) \left(\lambda \mathbf{I}_2 + \hat{\mathbf{V}}\right)^{-1\top} \\ \mathbf{V} & = \left(\lambda \mathbf{I}_2 + \hat{\mathbf{V}}\right)^{-1} \end{cases} \tag{5.17}$$

$$\begin{cases} \hat{\mathbf{m}} & = \alpha \mathbb{E}_{\mathbf{p}} \left[\mathbf{G}(\mathbf{p}) \right] \mathbf{1}_2 \\ \hat{\mathbf{Q}} & = \alpha \mathbb{E}_{\mathbf{p}} \left[\mathbf{G}(\mathbf{p}) \left((v_\star + \Delta) \mathbf{1}_{2\times 2} + \mathbf{B}\mathbf{Q}\mathbf{B}^\top \right) \mathbf{G}(\mathbf{p})^\top \right], \\ \hat{\mathbf{V}} & = \alpha \mathbb{E}_{\mathbf{p}} \left[\mathbf{G}(\mathbf{p}) \right] \end{cases} \tag{5.18}$$

for a careful choice of the joint distribution of $\mathbf{p} = (p_1, p_2)$. In the above, $\mathbf{G}(\mathbf{p}) = (\mathbf{I}_2 + \mathbf{P}\mathbf{V})^{-1}\mathbf{P}$ with $\mathbf{P} = \mathrm{Diag}(\mathbf{p})$, $\mathbf{B} = \begin{pmatrix} \mathbf{m}^\top \\ \mathbf{m}^\top \end{pmatrix} \mathbf{Q}^{-1} - \mathbf{I}_2$ and $v_\star = 1 - \mathbf{m}^\top Q^{-1} \mathbf{m}$.

Then, the following holds:

- *the variance of resampling method $\mathrm{t} \in \{\mathrm{pb}, \mathrm{ss}, \mathrm{jk}\}$ is given by*

$$\widehat{\mathrm{Var}}_t = Q_{11}^{\mathrm{t}} - Q_{12}^{\mathrm{t}}, \tag{5.19}$$

where overlaps with superscript t are obtained by solving (5.17)*,* (5.18) *using joint distribution $\mu(p_1, p_2) = \mu_{\mathrm{t}}(p_1) \cdot \mu_{\mathrm{t}}(p_2)$.*

- *the true variance is given by*

$$\mathrm{Var}_{\mathscr{D}}(\hat{\theta}_\lambda) = Q_{11}^{\mathrm{fr}} - Q_{12}^{\mathrm{fr}}, \tag{5.20}$$

where overlaps with superscript fr *(indicating full resampling) are obtained by solving* (5.17)*,* (5.18) *using joint distribution*

$$\mu(p_1, p_2) = (\mathbb{1}(p_1 = 0, p_2 = 1) + \mathbb{1}(p_1 = 1, p_2 = 0)).$$

- *the squared bias of resampling method* t *is given by*

$$\widehat{\mathrm{Bias}}_{\mathrm{t}}^2 = Q_{11}^{\mathrm{fr}} + Q_{12}^{\mathrm{t}} - 2Q_{12}^{\mathrm{t,fr}}, \tag{5.21}$$

where overlaps with superscript t, fr *are obtained by solving* (5.17)*,* (5.18) *using distribution* $\mu(p_1, p_2) = \mu_{\mathrm{t}}(p_1) \cdot \mathbb{1}\{p_2 = 1\}$ *for* p_1, p_2*.*

- *the true squared bias is given by*

$$\mathrm{Bias}_{\mathscr{D}}^2(\hat{\theta}_\lambda) = 1 - 2m_1^{\mathrm{fr}} + Q_{12}^{\mathrm{fr}}. \tag{5.22}$$

5.3.2 Conditional resampling

Similar to pair resampling, we leverage the fact that the conditional bias and variance, together with the estimates by residual bootstrap, can be written in terms of correlations between estimators. The key difference here is that the covariate $\mathbf{x}_i$ remain constant, and only the labels are resampled. Focusing on linear regression, in the case of residual resampling, the labels are sampled from the true distribution $y_i^\star \sim \mathcal{N}(\theta_\star{}^\top \mathbf{x}_i, \Delta)$, whereas for residual bootstrap, we use the ERM estimator to approximate this distribution and $y_i^\star \sim \mathcal{N}(\hat{\theta}_\lambda^\top \mathbf{x}_i, \tilde{\Delta})$ with $\tilde{\Delta}$ an estimator of Δ. Similarly to pair bootstrap, we now just need the correlation between B estimators $\hat{\theta}_{\lambda,k}$ trained on resampled datasets $\mathscr{D}_k^\star = \{(\mathbf{x}_i, y_{i,k}^\star)_{i=1}^n\}$. This can be done by considering the minimization problem (5.24). Despite minimizing each $\hat{\theta}_{\lambda,k}$ independently, they see the same covariates $\mathbf{x}_i$. In appendix D.2.1, we discuss how this correlation can be exactly captured by designing a particular approximate message passing, and also provide more details and an extension to more generic losses. As in the previous section, we first define the overlaps of interest

$$\begin{cases} Q_{11}^{\mathrm{rb}} & := \lim_{n,d\to\infty} \mathbb{E}_{\theta_\star,\mathscr{D}} \left[\mathbb{E}_{\mathbf{y}^\star|\mathscr{D}} \left[\|\hat{\theta}_\lambda(\mathbf{X}, \mathbf{y}^\star)\|^2 \right] \right] \\ Q_{12}^{\mathrm{rb}} & := \lim_{n,d\to\infty} \mathbb{E}_{\theta_\star,\mathscr{D}} \left[\|\mathbb{E}_{\mathbf{y}^\star|\mathscr{D}} [\hat{\theta}_\lambda(\mathbf{X}, \mathbf{y}^\star)]\|^2 \right] \\ Q_{11}^{\mathrm{rr}} & := \lim_{n,d\to\infty} \mathbb{E}_{\theta_\star,\mathscr{D}} \left[\|\hat{\theta}_\lambda\|^2 | \mathbf{X} \right] \\ Q_{12}^{\mathrm{rr}} & := \lim_{n,d\to\infty} \mathbb{E}_{\theta_\star} \left[\|\mathbb{E}_{\mathscr{D}} [\hat{\theta}_\lambda | \mathbf{X}]\|^2 \right] \\ m_1^{\mathrm{rb}} & := \lim_{n,d\to\infty} \mathbb{E}_{\theta_\star,\mathscr{D}} \left[\hat{\theta}_\lambda(\mathscr{D})^\top \mathbb{E}_{\mathbf{y}^\star|\mathscr{D}} \left[\hat{\theta}_\lambda(\mathbf{X}, \mathbf{y}^\star) \right] \right] \\ m_1^{\mathrm{rr}} & := \lim_{n,d\to\infty} \mathbb{E}_{\theta_\star} \left[\mathbb{E}_{\mathscr{D}} \left[\hat{\theta}_\lambda | \mathbf{X} \right]^\top \theta_\star \right]. \end{cases} \tag{5.23}$$

and the minimization problem for conditional resampling

$$\hat{\theta}_{\lambda,k} = \arg\min_{\theta} \sum_{i=1}^{n} -\log p(y^{\star}_{k,i}|\theta^{\top}\mathbf{x}_i) + {}^{\lambda}/_{2}\|\theta\|^2. \tag{5.24}$$

Theorem 5.3.2 (Biases and Variances for conditional resampling in ridge regression)**.** *Let* $\mathscr{D} = \{(\mathbf{x}_i, y_i)_{i\in[n]}\}$ *denote* n *independent samples drawn from model* (5.1) *with log-concave likelihood* $p(y|z)$*. In the high-dimensional proportional regime* $n, d \to \infty$ *with* ${}^{n}/_{d} = \alpha$*, the overlaps of interest* (5.23) *are given by the unique solution* $\mathbf{m} \in \mathbb{R}^2$*,* $\mathbf{Q} \in \mathbb{R}^{2\times 2}$*,*$\mathbf{V} \in \mathbb{R}^2$ *to the following set of self-consistent equations:*

$$\begin{cases} \mathbf{m} &= \tilde{\rho}\left(\lambda\mathbf{I}_2 + \hat{\mathbf{V}}\right)^{-1}\hat{\mathbf{m}} \\ \mathbf{Q} &= \left(\lambda\mathbf{I}_2 + \hat{\mathbf{V}}\right)^{-1}\left(\tilde{\rho}\hat{\mathbf{m}}\hat{\mathbf{m}}^{\top} + \hat{\mathbf{Q}}\right)\left(\lambda\mathbf{I}_2 + \hat{\mathbf{V}}\right)^{-1\top} \\ \mathbf{V} &= \left(\lambda\mathbf{I}_2 + \hat{\mathbf{V}}\right)^{-1} \end{cases} \tag{5.25}$$

and

$$\begin{cases} \hat{\mathbf{m}} &= \alpha\mathbf{G}\mathbf{1}_2 \\ \hat{\mathbf{Q}} &= \alpha\mathbf{G}\left(v_{\star}\mathbf{1}_{2\times 2} + \tilde{\Delta}\mathbf{I}_2 + \mathbf{B}\mathbf{Q}\mathbf{B}^{\top}\right)\mathbf{G}^{\top} \\ \hat{\mathbf{V}} &= \alpha\mathbf{G} \end{cases} \tag{5.26}$$

where $\mathbf{G} = (\mathbf{I}_B + \mathbf{V})^{-1}$*,* $\mathbf{B}$ *is defined as in Theorem 5.3.1 and* $v_{\star} = \tilde{\rho} - \mathbf{m}^{\top}\mathbf{Q}^{-1}\mathbf{m}$ *where* $\tilde{\rho}$ *depends on the method and is defined below. Then, the following holds:*

- *the variance of residual bootstrap is given by*

$$\widehat{\mathrm{Var}_{\mathrm{rb}}} = Q^{\mathrm{rb}}_{11} - Q^{\mathrm{rb}}_{12}, \tag{5.27}$$

  where $\mathbf{Q}^{\mathrm{rb}}$ *solves* (5.17)*,* (5.18) *using* $\tilde{\rho} = Q^{\mathrm{fr}}_{11}$ *and* $\tilde{\Delta} = {}^{(1+\Delta-2m^{\mathrm{fr}}_1+Q^{\mathrm{fr}}_{11})}/_{(1+V^{\mathrm{fr}}_{11})^2}$*. Note that the overlaps with superscript* fr *are specified in theorem 5.3.1.*

- *the true variance* $\mathrm{Var}_{\mathscr{D}|\mathbf{X}}(\hat{\theta}_{\lambda})$ *is given by*

$$\mathrm{Var}_{\mathscr{D}|\mathbf{X}}(\hat{\theta}_{\lambda}) = Q^{\mathrm{rr}}_{11} - Q^{\mathrm{rr}}_{12}, \tag{5.28}$$

  where $\mathbf{Q}^{\mathrm{rr}}$ *is obtained by solving* (5.25) *and* (5.26) *with* $\tilde{\rho} = 1, \tilde{\Delta} = \Delta$*.*

- *the squared bias of residual bootstrap*

$$\widehat{\mathrm{Bias}^2_{\mathrm{rb}}} = Q^{\mathrm{fr}}_{11} + Q^{\mathrm{rb}}_{12} - 2m^{\mathrm{rb}}_1 \tag{5.29}$$

Pair resampling rates

	Rate	Error
$\mathrm{Var}_{\mathscr{D}}(\hat{\theta}_\lambda)$	$1/\alpha$	–
$\widehat{\mathrm{Var}_{\mathrm{ss}}}$	$1/\alpha$	$1/\alpha$
$\widehat{\mathrm{Var}_{\mathrm{jk}}}$	$1/\alpha$	$1/\alpha^2$
$\widehat{\mathrm{Var}_{\mathrm{pb}}}$	$1/\alpha$	$1/\alpha^3$
$\mathrm{Bias}^2_{\mathscr{D}}(\hat{\theta}_\lambda)$	$1/\alpha^2$	–
$\widehat{\mathrm{Bias}^2_{\mathrm{ss}}}$	$1/\alpha^2$	$1/\alpha^2$
$\widehat{\mathrm{Bias}^2_{\mathrm{jk}}}$	$1/\alpha^2$	$1/\alpha^3$
$\widehat{\mathrm{Bias}^2_{\mathrm{pb}}}$	$1/\alpha^4$	$1/\alpha^2$

Residual resampling rates

	Rate	Error
$\mathrm{Var}_{\mathscr{D}\|\mathbf{X}}(\hat{\theta}_\lambda)$	$1/\alpha$	–
$\widehat{\mathrm{Var}_{\mathrm{rb}}}$	$1/\alpha$	$1/\alpha^2$
$\mathrm{Bias}^2_{\mathscr{D}\|\mathbf{X}}(\hat{\theta}_\lambda)$	$1/\alpha^2$	–
$\widehat{\mathrm{Bias}^2_{\mathrm{rb}}}$	$1/\alpha^2$	$1/\alpha^3$

Table 5.1: Summary of large α rates for ridge regression (see appendix D.3.2 for details).

- *the true conditional squared bias is given by*

$$\mathrm{Bias}^2_{\mathscr{D}|\mathbf{X}}(\hat{\theta}_\lambda) = 1 - 2m_1^{\mathrm{rr}} + Q_{12}^{\mathrm{rr}}. \tag{5.30}$$

The details for the derivations of theorem 5.3.2 are shown in appendix D.2.1. Compared to pair resampling, residual resampling does not involve introducing sample weights, only the labels are resampled from a conditional distribution. However, for residual bootstrap, the main idea is that the target weights $\theta_\star$ are replaced by $\hat{\theta}_\lambda$. Moreover, for ridge regression, we approximate the variance Δ by the averaged residual:

$$\tilde{\Delta} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{\theta}_\lambda^\top \mathbf{x}_i)^2 \tag{5.31}$$

In the high-dimensional regime, the analytical expression of this training error is given by the overlaps of state-evolution, and $\tilde{\Delta} = {}^{(1+\Delta-2m_1^{\mathrm{fr}}+Q_{11}^{\mathrm{fr}})}/_{(1+V_{11}^{\mathrm{fr}})^2}$. The derivation of this expression can be found in Loureiro et al. [154]. We end this section by observing that so far, we considered only the variance on the weights. However, one could be interested in other types of variances such as *predictive variance*, which we discuss in appendix D.4.

5.4 Discussions and main findings

In this section we discuss the consequences of the technical results from section 5.3 on the performance of resampling methods, and compare with empirical values. We refer to appendix D.5 for more details on the plots.

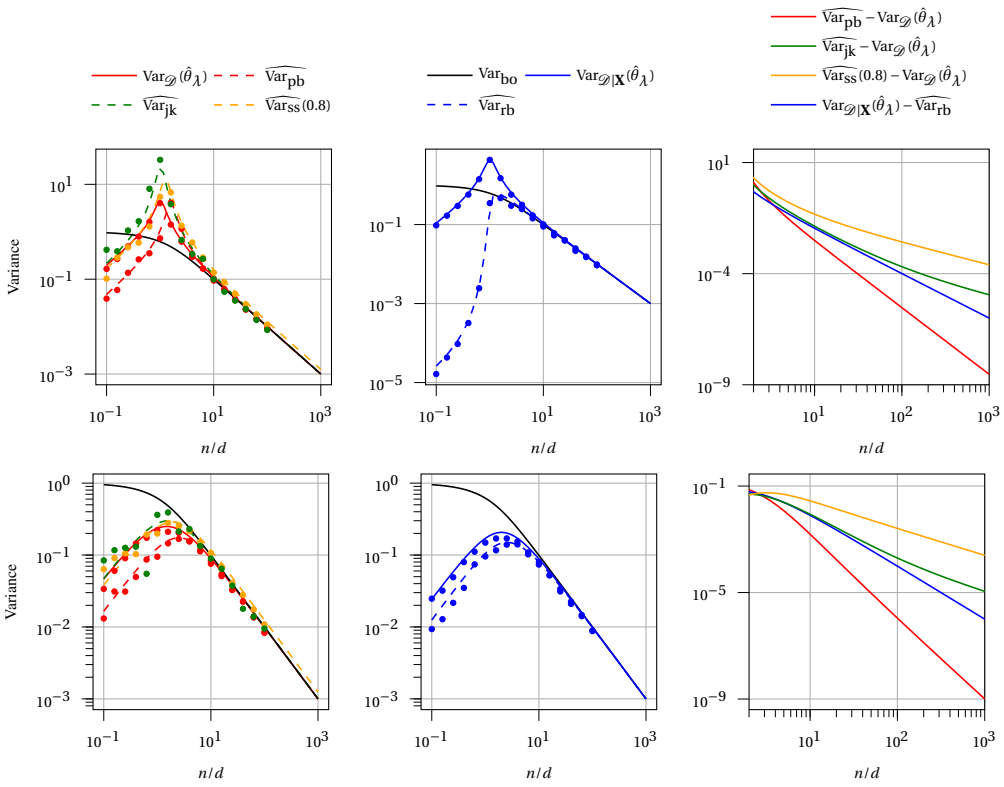


Figure 5.1: Variances for ridge regression at $\lambda = 10^{-2}$ (Top) and $\lambda = 1$ (Bottom). Left: variance of pair resampling methods and of Bayes-posterior. Middle: variance of conditional resampling and residual bootstrap. Right: difference between the true variances $\mathrm{Var}_{\mathscr{D}}(\hat{\theta}_\lambda)$, $\mathrm{Var}_{\mathscr{D}|\mathbf{X}}(\hat{\theta}_\lambda)$ and their estimation. Dots are simulations done at $d = 200$, with $B = 10$ resamples for bootstrap and subsampling.

5.4.1 Ridge regression

Variance – fig. 5.1 shows the different variances for ridge regression. We consider two important choices of regularization: $\lambda = 10^{-2}$ to approximate the behavior of unpenalized estimators, and $\lambda = \sigma^2 = 1$ which is the optimal value of λ: this regularization minimizes the generalization error of $\hat{\theta}_\lambda$ and its test error is the same as the Bayes-optimal estimator. As explained in section 5.2.2, the variance of Jackknife is approximated by doing subsampling with $r = 0.99$. Note that the subsampling variances with ratio r are rescaled by a factor $1 - r$. We compare our theoretical predictions with numerical experiments on Gaussian data and observe an excellent agreement. For $\lambda = 10^{-2}$ in the regime where $n > d$, our results are qualitatively consistent with [131], who showed that pair (respectively residual) bootstrap overestimates (resp. underestimates) the variance. On the other hand, our results allow us to study the variances at $d > n$. In this regime, we observe that both pair and residual bootstrap suffer from under-coverage: for residual bootstrap, it is easy to understand why, as without regularization $d > n$ the ERM interpolates the training data. Thus, the residual is exactly 0, and the residual bootstrap thus fatally underestimates the true level of noise in the data. On the other hand, subsampling and Jackknife are closer to

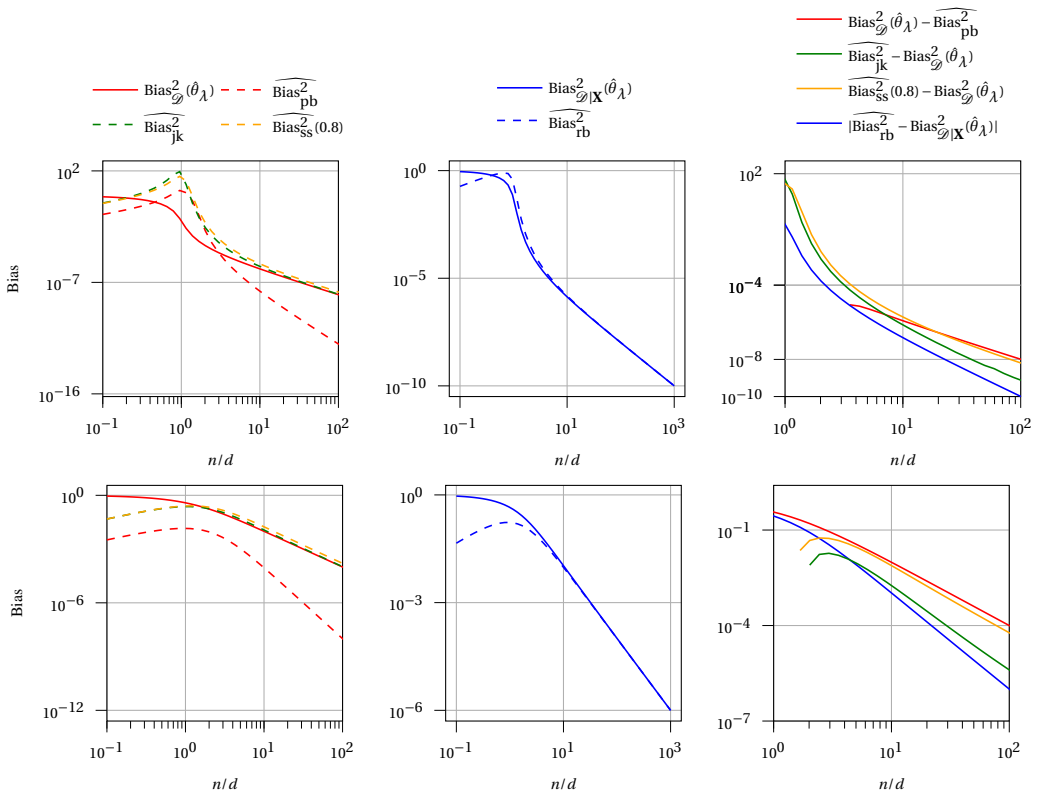


Figure 5.2: Bias of ridge regression and its estimation using pair bootstrap and subsampling at $\lambda = 10^{-2}$ (Top) and $\lambda = 1$ (Bottom). Left: bias of pair resampling methods. Middle: conditional bias and bias of residual bootstrap. Right: difference between the various biases.

$\mathrm{Var}_{\mathscr{D}}(\hat{\theta}_\lambda)$ than pair bootstrap, and as is classically known [75], the Jackknife estimate provides an upper bound of the true variance. On the right panel, we see that all variances conv erge to 0 with rate $1/n$, and pair bootstrap converges to $\mathrm{Var}_{\mathscr{D}}(\hat{\theta}_\lambda)$ the fastest. On the bottom row of Fig. fig. 5.1, we observe that optimal regularization greatly mitigates the under-coverage of bootstrapping, most notably for residual bootstrap. We thus conclude that for small values n/d, bootstrap fails to accurately capture the true variances, and appropriately regularizing partially mitigates this issue.

Note that conditioned on $\mathscr{D}$ and if the data generating process is known, the Bayes-optimal posterior variance $\mathrm{Var}_{\mathrm{bo}}$ is the best estimation of uncertainty on the weights. As in theorem 5.3.1 and 5.3.2, this variance can be obtained by solving a corresponding set of self-consistent equations [53]. We observe that at large α, all variances agree with $\mathrm{Var}_{\mathrm{bo}}$. However, at optimal λ and small n/d, resampling will underestimate the actual posterior variance.

Bias – In fig. 5.2, we plot the bias of the different resampling methods for ridge regression with regularization $\lambda \in \{10^{-2}, 1\}$. For the Jackknife and subsampling, the estimation of the squared bias is rescaled by a factor $(1-r)^2$. We observe that as $\alpha \to \infty$, $\mathrm{Bias}^2_{\mathscr{D}}(\hat{\theta}_\lambda)$ and $\widehat{\mathrm{Bias}^2_{\mathrm{pb}}}$ converge

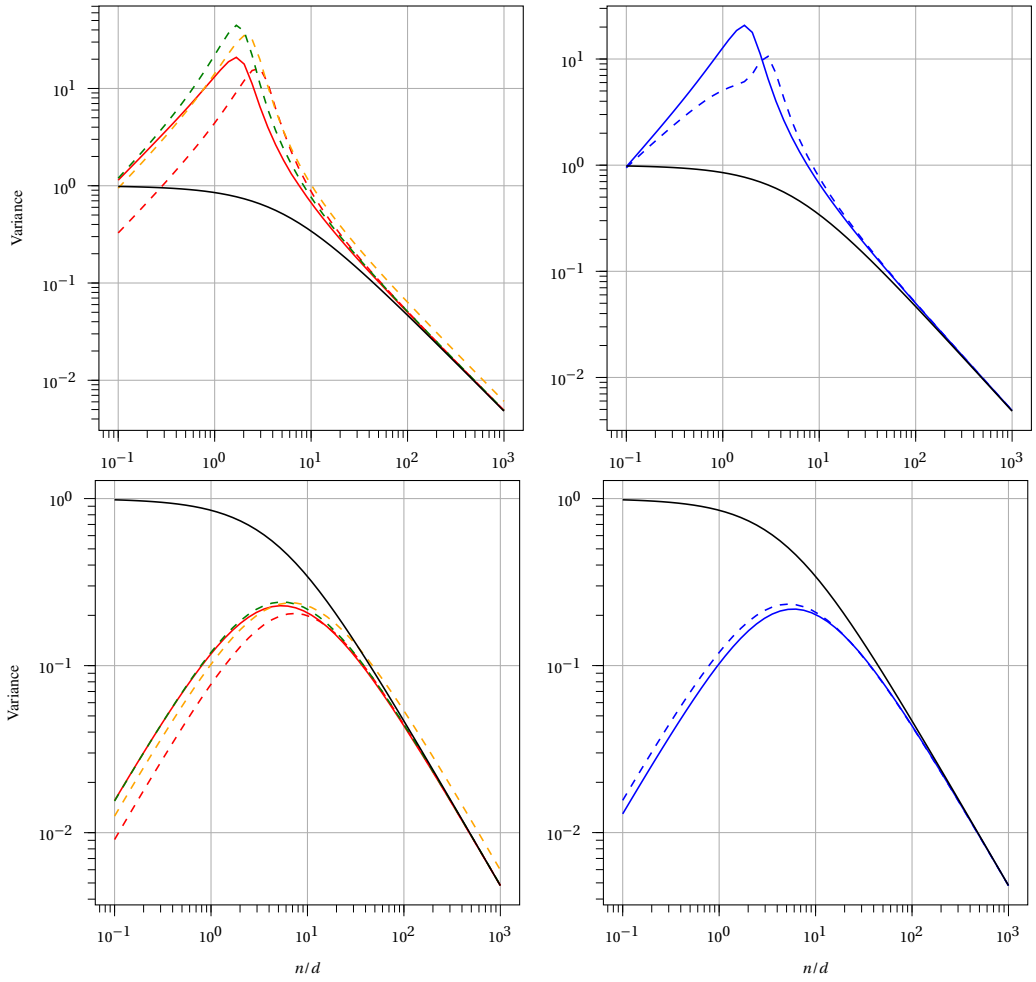


Figure 5.3: Variance for logistic regression at $\lambda = 10^{-2}$ (Top) and $\lambda = 1$ (Bottom). Left: variance of full resampling, pair bootstrap, subsampling. Right: variance of label resampling, residual bootstrap. See fig. 5.1 for the legend.

to zero, as expected by the consistency of the MLE estimator (5.3). However, $\widehat{\text{Bias}^2_{\text{pb}}}$ converges as $1/\alpha^4$, while $\text{Bias}^2_{\mathscr{D}}(\hat{\theta}_\lambda) \sim 1/\alpha^2$, and pair bootstrap underestimates the true bias. We deduce that in our model, subsampling or Jackknife should thus be preferred to estimate $\text{Bias}^2_{\mathscr{D}}(\hat{\theta}_\lambda)$.

5.4.2 Logistic regression

Our results extend beyond ridge regression, and the quantities of interest can be computed for any convex loss. fig. 5.3 displays the true variances and their estimation for regularized logistic regression with $\lambda \in \{10^{-2}, 1\}$, similarly to fig. 5.1. However, contrary to the ridge case, $\lambda = 1$ yields the maximum-a-posteriori estimator but does not minimize the misclassification error.

Qualitatively, we observe similar results as for ridge regression : at large α, all methods con-

sistently estimate the true variance and the Jackknife provides an upper bound of $\mathrm{Var}_{\mathscr{D}}(\hat{\theta}_\lambda)$. Moreover, at low α, regularization improves the estimation of the variance, even though λ is not optimal.

Finally, at $\lambda = 0.01$ for both ridge and logistic regression, we observe a local maximum in the true and resampled bias and variance around $d = n$. This behavior is reminiscent of the double-descent behavior observed e.g. in random features models or neural networks : the test error achieves a local maximum at the interpolation threshold where the model can perfectly fit the training data, then decreases with the number of parameters. Moreover, we see that regularization can mitigate this "double-descent" phenomenon.

5.5 Conclusion

In this work, we have provided an exact asymptotic comparison of the uncertainty estimations provided by different resampling methods, in the context of high-dimensional regularized maximum likelihood with generalized linear models.

Our results highlight the limitations of these methods in the high-dimensional regime relevant to modern machine learning practice and discuss how cross-validation can, to some extent, mitigate some of these limitations.

Avenues for future work are manifold. For instance, how would our results change in a misspecified scenario? Can structure in the data help or hinder resampling methods? These interesting questions are left for future investigation.

6 Ensembling overparametrized models

This chapter focuses on extending and combining the results of both chapter 5 and [152] :

- On one hand, the analysis done in Íchapter 5 was only done for the case of linear models, and a natural extension is to consider the case of overparametrized models. This extension is motivated by the empirical observation that the bootstrap and subsampling methods do not work well in modern machine learning models, unless they are re-calibrated after training [187, 180].
- On the other hand, the analysis done in [152] does not include the correlation between two Bayes-optimal estimators. This correlation is of interest to study the optimal performance in the context of ensembles of random features.

Our technical contribution is an extension of the computations done in [152], in which the authors computed the correlation between two empirical risk minimizers trained on two different random features. In this chapter, we compute the correlation of two Bayes-optimal estimators trained on two different random features. The computations are similar to the proof of theorem 5.3.1, which is contained in appendix D.

6.1 Setting

Consider a dataset $\mathscr{D} = \left(\mathbf{x}_i, y_i\right)_{i=1}^n$ of n samples, sampled from the Gaussian teacher-student model. The input comes from a Gaussian distribution $\mathbf{x} \sim \mathcal{N}(0, I_d/d)$.

6.1.1 Ensemble of random features

We will consider the case of estimators trained on different random features, each defined as

$$\varphi^k(\mathbf{x}) = \phi\left(F^k\mathbf{x}\right)/\sqrt{p}, \quad \forall 1 \leqslant k \leqslant K \tag{6.1}$$

where F^k is a random matrix with i.i.d Gaussian entries $\mathcal{N}(0, 1/d)$ of size $p \times d$ and ϕ is a non-linear activation function. The ratio $\gamma = p/d$ quantifies the overparametrization of the students. Without loss of generality, we will assume that $\mathbb{E}_{z\sim\mathcal{N}(0,1)}(\phi(z)) = 0$. This assumption simply means that the random features are centered in expectation, and is verified e.g. for the tanh function. We will write $\mathscr{D}^k = \left(\varphi^k(\mathbf{x}_i), y_i\right)_{i=1}^n$ the dataset in the random feature space. Our goal in this chapter is to characterize the variance in the estimators coming from the randomness of the random features, in contrast to chapter 5 where we analyzed the variance coming from the resampling procedure. The variance we are aiming to compute is

$$\mathrm{Var}_{\mathbf{F}}\left(\hat{f}_{\mathrm{bo}}(\mathbf{x})\right) = \mathrm{Var}_{\mathbf{F}}\left(\theta_{\mathrm{bo}}^\top \mathbf{x}\right) \tag{6.2}$$

here the randomness is on the random features $\varphi^k(\mathbf{x})$ or equivalently the matrix of random projections $\mathbf{F}$. As in the previous chapter, computing this variance reduces to computing the overlap $\theta_{\mathrm{bo}}^{1\top}\theta_{\mathrm{bo}}^2$ between two Bayes posterior trained on two different random features. Then, in the rest of this chapter, we will consider $K = 2$.

The joint posterior distribution that interests us here is

$$p_{\mathrm{bo}}(\theta^1, \theta^2) = \mathbb{P}(\theta^1|\mathscr{D}^1) \times \mathbb{P}(\theta^2|\mathscr{D}^2) \tag{6.3}$$

with $\hat{\theta}_{\mathrm{bo}}^1 = \mathbb{E}_{p_{\mathrm{bo}}}\left[\theta^1\right], \hat{\theta}_{\mathrm{bo}}^1 = \mathbb{E}_{p_{\mathrm{bo}}}\left[\theta^1\right]$ the means of the two marginal distributions.

We introduce below the local fields of the teacher and the two students. The joint distribution of these local fields will allow us to compute the variance (6.2).

$$\left(\nu, \lambda^1, \lambda^2\right) = \left(\theta_\star{}^\top \mathbf{x}, \theta_{\mathrm{bo}}^{1\top}\varphi^1(\mathbf{x}), \theta_{\mathrm{bo}}^{2\top}\varphi^2(\mathbf{x})\right) \tag{6.4}$$

Gaussian equivalence As in chapter 3, we use the Gaussian Equivalence Theorem [99] to analyze the performance of the estimators in high-dimensions. As such, we will assume that the estimators $\hat{f}_{\mathrm{bo}}^1, \hat{f}_{\mathrm{bo}}^2$ are trained on the datasets $\mathscr{D}^j = \left(\mathbf{v}_i^j, y_i\right)_{i=1}^n$ where the student's observations $\mathbf{v}^j$ follow the joint distribution :

$$(\mathbf{x}, \mathbf{v}^1, \mathbf{v}^2) \sim \mathcal{N}\left(\mathbf{0}, \begin{pmatrix} \Psi & \Phi^1 & \Phi^2 \\ \Phi^{1\top} & \Omega^1 & \Omega^{1,2} \\ \Phi^{2\top} & \Omega^{1,2\top} & \Omega^2 \end{pmatrix}\right) \tag{6.5}$$

With the covariances defined from the random projections

$$\begin{cases} \Phi^j & = \kappa_1 F^j \\ \Omega^j & = \kappa_1^2 F^j F^{j\top} + \kappa_\star^2 I_p\,, \quad \kappa_1 = \mathbb{E}_{z\sim\mathcal{N}(0,1)}(z\phi(z)), \quad \kappa_\star = \mathbb{E}_{z\sim\mathcal{N}(0,1)}(\phi(z)^2) - \kappa_1^2 \\ \Omega^{1,2} & = \kappa_1^2 F^1 F^{2\top} \end{cases} \tag{6.6}$$

The scalar factors κ_1 and $\kappa_\star$ are described more in details in chapter 3 and appendix B.

The local field of the teacher, expressed as a function of the student's observation $\mathbf{v}^j$ is

$$\theta_\star{}^\top \mathbf{x} = \theta_\star{}^\top (\Phi^j \mathbf{v}^j + \mathbf{z}) = \mathbf{w}_\star{}^{j\top} \mathbf{v}^j + \theta_\star{}^\top \mathbf{z} \tag{6.7}$$

$$\mathbf{z} \sim \mathcal{N}\left(\mathbf{0}, I_d - \Phi^j (\Omega^j)^{-1} \Phi^{j\top}\right) \tag{6.8}$$

where we have defined

$$\mathbf{w}_\star^j = \Phi^{j\top} \theta_\star \sim \mathcal{N}\left(\mathbf{0}, \Sigma_\star^j\right) \tag{6.9}$$

$$\Sigma_\star^j = \Phi^{j\top} \Phi^j \tag{6.10}$$

the scalar product $\theta_\star{}^\top \mathbf{z}$ follows a Gaussian distribution

$$\varepsilon \sim \mathcal{N}\left(0, \frac{1}{d} \theta_\star{}^\top \left(I_d - \Phi^j (\Omega^j)^{-1} \Phi^{j\top}\right) \theta_\star\right)$$

whose variance converges to $\tau^2_{\text{add}}{}^j = \lim_\infty \frac{1}{d} \text{Tr}\left(I_d - \Phi^j (\Omega^j)^{-1} \Phi^{j\top}\right)$ in the high-dimensional limit. Then, the likelihood of the teacher in the random feature space is

$$p'(y|\mathbf{w}_\star{}^j \mathbf{v}^j) = \int \mathrm{d}\varepsilon \, p(y|\mathbf{w}_\star \, j\top \mathbf{v}^j + \varepsilon) \mathcal{N}(\varepsilon|0, \tau^2_{\text{add}}{}^j) \tag{6.11}$$

And in this model, the local fields follow a 3-dimensional Gaussian distribution

$$\left(\nu, \lambda^1, \lambda^2\right) = \left(\theta_\star{}^\top \mathbf{x}, \theta_{\text{bo}}^{1\top} \mathbf{v}^1, \theta_{\text{bo}}^{2\top} \mathbf{v}^2\right) \sim \mathcal{N}\left(\mathbf{0}, \begin{pmatrix} \rho & \mathbf{m} \\ \mathbf{m}^\top & \mathbf{Q} \end{pmatrix}\right) \tag{6.12}$$

With similar overlaps as in chapter 5 :

$$\begin{cases} \rho &= \|\mathbf{w}_\star\|^2 / d \\ \mathbf{m} &= \theta_\star{}^\top \left(\Phi^1 \hat{\theta}^1_{\text{bo}}, \Phi^2 \hat{\theta}^2_{\text{bo}}\right) / d \\ \mathbf{Q} &= \frac{1}{d} \begin{pmatrix} \frac{1}{d} \hat{\theta}^1_{\text{bo}} \Omega^1 \hat{\theta}^1_{\text{bo}} & \frac{1}{d} \hat{\theta}^1_{\text{bo}} \Omega^{1,2} \hat{\theta}^2_{\text{bo}} \\ \frac{1}{d} \hat{\theta}^2_{\text{bo}} \Omega^{1,2\top} \hat{\theta}^1_{\text{bo}} & \frac{1}{d} \hat{\theta}^2_{\text{bo}} \Omega^2 \hat{\theta}^2_{\text{bo}} \end{pmatrix} \end{cases} \tag{6.13}$$

6.2 Technical result

In this section we present our technical result : state-evolution equations to compute the ensemble variance for frequentist and Bayesian estimators, trained with subsampling or bootstrap resampling on different random features.

Theorem 6.2.1. *Let $\mathscr{D} = \left(\mathbf{x}_i, y_i\right)_{i=1}^n$ be a dataset of n samples, sampled from the teacher-student*

model of chapter 3. Consider the joint distribution defined in eq. (6.3) *of the two Bayes posterior trained respectively on* $\mathscr{D}^1, \mathscr{D}^2$ *as defined in eq.* (6.1)*. In the high-dimensional proportional regime* $n, d \to \infty$ *with* $n/d = \alpha$*, the overlaps of interest* m, q, V *are given by the solutions of the following equations*

$$\begin{cases} Q_{j,j} & = \frac{1}{d}\mathrm{Tr}\left(\left(\hat{Q}_{j,j}\Omega^j + (\hat{m}_j)^2\Phi^{j\top}\Phi^j\right)\Omega^j\left((\Sigma^j_\star)^{-1} + \hat{V}^j\Omega^j\right)^{-2}\right) \\ Q_{1,2} = Q_{2,1} & = (1 + \hat{Q}_{1,2}/\hat{m}_1 \times \hat{m}_2) \times m_1 \times m_2 \\ m_j & = \frac{1}{d}\hat{m}_j\mathrm{Tr}\left(\Phi^{j\top}\Phi^j\left(\left(\Sigma^j_\star\right)^{-1} + \hat{V}_j\Omega^j\right)^{-1}\right) \\ V^j & = \frac{1}{d}\mathrm{Tr}\left(\left(\left(\Sigma^j_\star\right)^{-1} + \hat{V}^j\Omega^j\right)^{-1}\Omega^j\right) \end{cases} \tag{6.14}$$

$$\begin{cases} \hat{\mathbf{m}} & = \alpha\mathbb{E}_\omega\left[\int dy \partial_\mu \mathscr{Z}_0(y, \mu_\star(\omega), v_\star)\cdot \mathbf{g}_0(y,\omega,\mathbf{V})\right] \\ \hat{\mathbf{Q}} & = \alpha\mathbb{E}_\omega\left[\int dy \mathscr{Z}_0(y, \mu_\star(\omega), v_\star)\cdot \left[\mathbf{g}_0\mathbf{g}_0^\top\right](y,\omega,\mathbf{V})\right] \\ \hat{\mathbf{V}} & = -\alpha\mathbb{E}_\omega\left[\int dy \mathscr{Z}_0(y, \mu_\star(\omega), v_\star)\cdot \partial_\omega \mathbf{g}_0(y,\omega,\mathbf{V})\right] \end{cases} \tag{6.15}$$

where

$$\begin{cases} \mathscr{Z}_0(y,\omega,V) & = \int dz p'(y|z)\times e^{-\frac{1}{2V}(z-\omega)^2}/\sqrt{2\pi V} \\ \mathbf{g}_0(y,\omega,\mathbf{V}) & = \left(\partial_\omega \log \mathscr{Z}_0(y,\omega_1,V_1), \partial_\omega \log \mathscr{Z}_0(y,\omega_2,V_2)\right) \end{cases} \tag{6.16}$$

and

$$\begin{cases} \mu_\star(\omega) & = \mathbf{m}^\top\mathbf{Q}^{-1}\omega \\ v_\star & = \rho - \mathbf{m}^\top\mathbf{Q}^{-1}\mathbf{m} \end{cases} \tag{6.17}$$

Note that the equations for the overlaps $\mathbf{m}, \mathbf{V}$ and the diagonal term $Q_{1,1} = Q_{2,2}$ are the same as in chapter 3, where we study the performance of a single estimator. The only addition is the off-diagonal term $Q_{1,2}$. In section 6.3, we will prove the state-evolution equations (6.15), (6.14).

6.3 Proof of state-evolution equations

Proof idea The proof is similar to appendix B.2.2 : we introduce an AMP algorithm that computes the two Bayes-optimal estimator and derive its state-evolution equations. Note that the equations of the overlaps for the magnetization $\mathbf{m}$, the diagonal term of the variance V and the diagonal terms $Q_{1,1}, Q_{2,2}$ remain identical as in chapter 3, for the simple reason that the marginal distribution of the posteriors are unchanged.

Change of variables To simplify the algebra in the computations that follow, we will do the change of variable

$$\mathbf{v}^j \to (\Omega^j)^{-1/2}\mathbf{v}^j \tag{6.18}$$
$$\theta^j \to (\Omega^j)^{1/2}\theta^j \tag{6.19}$$
$$\tag{6.20}$$

such that the local fields $\theta^{j\top}\mathbf{v}^j$ are unchanged but the student's inputs have i.i.d covariance and follow the distribution $\mathcal{N}(\mathbf{0}, I_p/p)$. Consequently, the covariance matrices become :

$$\Omega^j \to I_p$$
$$\Omega^{1,2} \to (\Omega^1)^{-1/2}\Omega^{1,2}(\Omega^2)^{-1/2}, \quad \Phi^j \to \Phi^j(\Omega^j)^{-1/2}$$

6.3.1 AMP for Bayes-optimal on a pair of random features

In algorithm 4, we write down the AMP algorithm to infer the posterior distribution of B Bayes-optimal estimators. The algorithm is similar to the one introduced in algorithm 8. The difference lies in the denoiser function as the prior here is not an isotropic Gaussian : here, the denoisers for each estimator $i = 1, 2$ are related to their prior $\theta^j_\star \sim \mathcal{N}(0, \Sigma_{\star,j})$ where the prior covariance is given by

$$\Sigma_{\star,i} = \Omega_i^{-1/2}\Phi_i^T\Phi_i\Omega_i^{1/2} \tag{6.21}$$

and the denoiser for a prior $\mathcal{N}(0, \Sigma_\star)$ is

$$f_a(\mathbf{b}, \mathbf{A}) = \left(\mathbf{A} + \Sigma_\star^{-1}\right)^{-1}\mathbf{b} \in \mathbb{R}^p, \quad f_v(\mathbf{b}, \mathbf{A}) = \left(\mathbf{A} + \Sigma_\star^{-1}\right)^{-1} \in \mathbb{R}^p \tag{6.22}$$

6.3.2 State-evolution equations

As we do in the rest of the manuscript, we consider the vectors from belief propagation.

$$\begin{cases} \omega^{j(t)}_{\mu\to i} & = \sum_{k\neq i} v^j_{\mu,k}\hat{\theta}^{j(t)}_{k\to\mu} \\ V^{j(t)}_{\mu\to i} & = \sum_{k\neq i}(v^j_{\mu,k})^2\hat{C}^{j(t)}_{k\to\mu} \end{cases}, \quad \begin{cases} \mathbf{g}\text{out}^{(t)}_{\mu\to i} & = \mathbf{g}\text{out}(y_\mu, \omega^{(t)}_{\mu\to i}, \mathbf{V}^{(t)}_{\mu\to i}) \\ \partial\mathbf{g}\text{out}^{(t)}_{\mu\to i} & = \partial_\omega\mathbf{g}\text{out}(y_\mu, \omega^{(t)}_{\mu\to i}, \mathbf{V}^{(t)}_{\mu\to i}) \end{cases} \tag{6.23}$$

$$\begin{cases} b^{j(t)}_{\mu\to i} & = \sum_{\nu\neq\mu} v^j_{\nu,i}g^{j(t)}{}_{\nu\to i} \\ A^{j(t)}_{\mu\to i} & = -\sum_{\nu\neq\mu}(v^j_{\nu,i})^2\partial g^{j(t)}{}_{\nu\to i} \end{cases}, \quad \begin{cases} \hat{\theta}^{j(t)}_{\cdot\to\mu} & = \mathbf{f}_a(\mathbf{b}^{(t)}_{\cdot\to\mu}, \mathbf{A}^{j(t)}_{\cdot\to\mu}) \\ \hat{\mathbf{C}}^{j(t)}_{\cdot\to\mu} & = \partial_\mathbf{b}\mathbf{f}_a(\mathbf{b}^{(t)}_{\cdot\to\mu}, \mathbf{A}^{j(t)}_{\cdot\to\mu}). \end{cases} \tag{6.24}$$

Algorithm 4 GAMP for Bayes-optimal on random features $\mathbf{v}_\mu^j$ for $1 \leqslant j \leqslant B$ different random feature matrices F_j.

Input: $(\mathbf{v}_\mu^j)_{\mu=1}^n \in \mathbb{R}^{n\times p}$ $(1 \leqslant j \leqslant B)$, $\mathbf{y} \in \mathbb{R}^n$
Initialize: $\mathbf{g}_{\text{out}_\mu}^{(0)} = \mathbf{0}$ for $1 \le \mu \le n$, $\quad \mathbf{A}_i^{(0)} = \mathbf{1}_B$ for $1 \le i \le p$
Initialize: $\hat{\theta}_i^{(1)} \in \mathbb{R}^B$ and $\hat{\mathbf{C}}_i^{(1)} \in \mathbb{R}^B$ for $1 \le i \le d$
Repeat for $t = 1, 2, \ldots$**:**
// Update of the means $\omega_\mu^j \in \mathbb{R}$ and variances $\mathbf{V}_\mu \in \mathbb{R}^B$ for $1 \le \mu \le n$:
$\omega_\mu^{j(t)} = \sum_{i=1}^d v_{\mu,i}^j \mathbf{w}_i^{j,(t)} - \mathbf{v}_{\mu,i}^{j2} \hat{\mathbf{C}}_i^{j(t)} \mathbf{g}_{\text{out}_\mu}^{j(t-1)} \mid V_\mu^{j(t)} = \sum_{i=1}^d (\mathbf{v}_{\mu,i}^j)^2 \hat{\mathbf{C}}_i^{j(t)}$ for $1 \leqslant j, k \leqslant B$
// Update of $\mathbf{g}_{\text{out}_\mu} \in \mathbb{R}^B$ and $\partial_\omega \mathbf{g}_{\text{out}_\mu} \in \mathbb{R}^B$ for $1 \le \mu \le n$:
$\mathbf{g}_{\text{out}_\mu}^{(t)} = \mathbf{g}_{\text{out}}\left(\omega_\mu^{(t)}, y_\mu, \mathbf{V}_\mu^{(t)}\right) \mid \partial_\omega \mathbf{g}_{\text{out}_\mu}^{(t)} = \partial_\omega \mathbf{g}_{\text{out}}\left(\omega_\mu^{(t)}, y_\mu, \mathbf{V}_\mu^{(t)}\right)$
// Update of means $\mathbf{b}_i \in \mathbb{R}^B$ and variances $\mathbf{A}_i \in \mathbb{R}^B$ for $1 \le i \le p$:
$A_i^{j(t)} = -\sum_{\mu=1}^n (v^j)_{\mu,i}^2 \partial_\omega g_{\text{out}_\mu}^{j(t)} \mid b_i^{j(t)} = A_i^{j(t)} \hat{\theta}_i^{j(t)} + \sum_{\mu=1}^n v_{\mu,i}^j g_{\text{out}_\mu}^{j(t)}$
// Update of the estimated marginals $\hat{\theta}_i \in \mathbb{R}^B$ and $\hat{\mathbf{C}}_i \in \mathbb{R}^B$ for $1 \le i \le p$:
$\hat{\theta}^{(t+1)} = \mathbf{f}_a(\mathbf{b}^{(t)}, \mathbf{A}^{(t)}) \mid \hat{\mathbf{C}}^{(t+1)} = \partial_{\mathbf{b}} \mathbf{f}_a(\mathbf{b}^{(t)}, \mathbf{A}^{(t)})$
Until convergence
Output: $\hat{\theta}_1, \ldots, \hat{\theta}_d$ and $\hat{\mathbf{C}}_1, \ldots, \hat{\mathbf{C}}_d$

where $\mathbf{A}_{:\to\mu}^{j(t)}$ and $\mathbf{b}_{:\to\mu}^{j(t)}$ refer to the vectors $\left(\mathbf{A}_{i\to\mu}^{j(t)}\right)_{i=1}^p$ and $\left(\mathbf{b}_{i\to\mu}^{j(t)}\right)_{i=1}^p$.

The overlaps of these vectors are defined as

$$\mathbf{m}_j^{(t)} \equiv \lim_{d\to\infty} \frac{1}{d} \theta^{j\top} \mathbf{w}_\star{}^j, \qquad \mathbf{Q}_{ij}^{(t)} \equiv \lim_{d\to\infty} \frac{1}{d} \mathbf{w}_\star{}^{(t)i\top} \Omega^{i,j} \mathbf{w}_\star{}^{(t)j} \tag{6.25}$$

$$\mathbf{V}^{(t)} \equiv \lim_{d\to\infty} \frac{1}{d} \sum_{i=1}^d \hat{\mathbf{C}}_i^{(t)}, \qquad \rho = \lim_{d\to\infty} \frac{\|\mathbf{w}_\star\|^2}{d}. \tag{6.26}$$

To derive the asymptotic behavior of these overlap parameters, we compute the overlap distributions starting from the rBP equations above. In the $d \to \infty$ limit :

$$\left(z_\mu, \omega_{\mu\to i}^{1(t)}, \omega_{\mu\to i}^{2(t)}\right) \sim \mathcal{N}\left(0, \begin{bmatrix} \rho & \mathbf{m}^{(t)\top} \\ \mathbf{m}^{(t)} & \mathbf{Q}^{(t)} \end{bmatrix}\right) \tag{6.27}$$

Distribution of $\mathbf{b}_{\mu\to i}^{(t)}$ Recall from our setting that for a given input $\mathbf{x}_\mu$, the corresponding label is distributed as $y_\mu \sim p(\cdot|z_\mu)$. In fact, one can equivalently write $y^\mu = \varphi_0(z_\mu)$ for some (random) function φ_0. For example, the choice $\varphi_0(x) = x + \sqrt{\Delta} v$ corresponds to the linear regression, where

$\nu \sim \mathcal{N}(0, \Delta)$ is a Gaussian noise. With this representation for y_μ, we have

$$b_{\mu\to i}^{j(t)} = \sum_{\nu\neq\mu} v_{\nu,i}^{j}\, g_{\text{out}}{}_{\nu\to i}^{j}) \tag{6.28}$$

$$= \sum_{\nu\neq\mu} v_{\nu,i}^{j}\, g_{\text{out}}(\varphi_0\left(z_{\nu\to i}^{j} + \mathbf{w}_\star{}_i^{j} v_{\nu,i}^{j} + \varepsilon^j\right), \omega_{\nu\to i}^{j(t)}, \mathbf{V}_{\nu\to i}^{j(t)}) \tag{6.29}$$

$$= \sum_{\nu\neq\mu} v_{\nu,i}^{j}\, g_{\text{out}}(\varphi_0\left(z_{\nu\to i}^{j} + \varepsilon^j\right), \omega_{\nu\to i}^{j(t)}, V_{\nu\to i}^{j(t)}) \tag{6.30}$$

$$+ (v_{\nu,i}^{j})^2 \mathbf{w}_\star{}_i^{j} \partial_z \mathbf{g}_{\text{out}}(\varphi_0\left(z_{\nu\to i}^{j} + \varepsilon^j\right), \omega_{\nu\to i}^{j(t)}, V_{\nu\to i}^{j(t)}) + O(d^{-3/2}) \tag{6.31}$$

where in the last equality we have expanded the denoising function at leading order. The noise $\varepsilon^j \sim \mathcal{N}\left(0, \tau_{\text{add}}^2\right)$ stems from eq. (6.8). Taking expectation on both sides yields

$$\mathbb{E}[b_{\mu\to i}^{j(t)}] = \frac{\mathbf{w}_\star{}_i^{j}}{d} \sum_{\nu\neq\mu} \partial_z \mathbf{g}_{\text{out}}(\varphi_0\left(z_{\nu\to i}\right), \omega_{\nu\to i}^{j(t)}, V_{\nu\to i}^{j(t)}) + O(d^{-3/2}) \tag{6.32}$$

$$= \frac{\mathbf{w}_\star{}^{j}{}_i}{d} \sum_{\nu=1}^{n} \partial_z \mathbf{g}_{\text{out}}(\varphi_0\left(z_{\nu\to i} + \varepsilon^j\right), \omega_{\nu\to i}^{j(t)}, V_{\nu\to i}^{j(t)}) \tag{6.33}$$

$$- \frac{\mathbf{w}_\star{}^{j}{}_i}{d} \partial_z \mathbf{g}_{\text{out}}(\varphi_0\left(z_{\mu\to i} + \varepsilon^j\right), \omega_{\mu\to i}^{j(t)}, V_{\mu\to i}^{j(t)}) + O(d^{-3/2}), \tag{6.34}$$

Note that as $d \to \infty$, it follows from our computations above that for all ν, $(z_{\nu\to i}, \omega_{\nu\to i}^{(t)})$ are identically distributed according to eq. (6.27). Consequently, by the Law of Large Numbers,

$$\frac{n}{d} \cdot \frac{1}{n} \sum_{\nu=1}^{n} \partial_z \mathbf{g}_{\text{out}}(\varphi_0\left(z_{\nu\to i} + \varepsilon^j\right), \omega_{\nu\to i}^{(t)}, \mathbf{V}_{\nu\to i}^{(t)}) \overset{n,d\to\infty}{\longrightarrow} \alpha \mathbb{E}_{(z,\omega),\varepsilon}\left[\partial_z \mathbf{g}_{\text{out}}(\varphi_0\left(z+\varepsilon\right), \omega, \mathbf{V}^{(t)})\right] \equiv \hat{m}_j^{(t)}, \tag{6.35}$$

from which we find that

$$\mathbb{E}[b_{\mu\to i}^{j(t)}] \overset{n,d\to\infty}{\longrightarrow} \mathbf{w}_\star{}_i^{j} m^{j(t)}. \tag{6.36}$$

The second moment can be computed in a similar fashion.

$$\mathbb{E}[b_{\mu\to i}^{j(t)} b_{\mu\to i}^{k(t)\top}] = \sum_{\nu\neq\mu} \sum_{\kappa\neq\mu} \mathbb{E}[v_{\nu,i}^{j} v_{\kappa,i}^{k}]\, g_{\text{out}}{}_{\nu\to i}^{j}\, g_{\text{out}}{}_{\kappa\to i}^{k} \tag{6.37}$$

$$= \frac{1}{d}\Omega^{j,k} \sum_{\nu\neq\mu} g_{\text{out}}{}_{\nu\to i}^{j}\, g_{\text{out}}{}_{\kappa\to i}^{k} + O(d^{-2}) \tag{6.38}$$

$$\overset{n,d\to\infty}{\longrightarrow} \gamma \times \alpha\Omega^{j,k} \mathbb{E}_{(z,\omega^{(t)}),\varepsilon}\left[g_{\text{out}}{}^{j}\, g_{\text{out}}^{k}\right] \equiv \hat{\mathbf{Q}}_{j,k}^{(t)} \tag{6.39}$$

Hence, $\mathbf{b}_{\mu\to i}^{(t)} = \mathbf{w}_{\star\, i} \hat{\mathbf{m}}^{(t)} + \left(\hat{\mathbf{Q}}^{(t)}\right)^{1/2} \xi$ with $\xi \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_B)$.

Concentration of $\mathbf{A}^{(t)}_{\mu\to i}$ It remains to show that the covariances $\mathbf{A}^{(t)}_{\mu\to i}$ concentrate. We have

$$A^{j(t)}_{\mu\to i} = -\sum_{\nu\neq\mu}(v^j_{\nu,i})^2\partial_\omega g_{\text{out}}{}^j_{\nu\to i} \tag{6.40}$$

$$= -\sum_{\nu\neq\mu}(v^j_{\nu,i})^2\partial_\omega\partial_\omega g_{\text{out}}{}^j_{\nu\to i} \tag{6.41}$$

$$= -\sum_{\nu\neq\mu}(v^j_{\nu,i})^2\partial_\omega g_{\text{out}}(\varphi_0(z_{\nu\to i}),\omega^{(t)}_{\nu\to i},\mathbf{V}^{(t)}_{\nu\to i}) + O(d^{-3/2}). \tag{6.42}$$

Taking the expectation gives

$$\mathbb{E}[A^{j(t)}_{\mu\to i}] = -\frac{1}{d}\sum_{\nu\neq\mu}\partial_\omega g_{\text{out}}(\varphi_0(z_{\nu\to i}),\omega^{j(t)}_{\nu\to i},V^{j(t)}_{\nu\to i}) + O(d^{-3/2}) \tag{6.43}$$

$$\overset{n,d\to\infty}{\longrightarrow} -\alpha\mathbb{E}_{(z,\omega^{(t)})}\left[\partial_\omega g_{\text{out}}(\varphi_0(z),\omega^{(t)},V^{(t)})\right] \equiv \hat{V}^{j(t)} \tag{6.44}$$

Summary

Having shown the distribution of messages and concentration, we are ready to characterize the asymptotic distribution of the estimator:

$$\hat{\theta}_i \sim \mathbf{f}_a\left(\theta_{\star i}\hat{\mathbf{m}}^{(t)} + \left(\hat{\mathbf{Q}}^{(t)}\right)^{1/2}\xi, \hat{\mathbf{V}}^{(t)}\right) \quad \forall i\in\{1,\dots,d\}, \tag{6.45}$$

where $\xi\sim\mathcal{N}(\mathbf{0},\mathbf{I}_B)$.

From that, the definitions of overlaps at time $t+1$, and the message distributions, we obtain the state-evolution equations of the GAMP algorithm described in :

$$\begin{cases}\mathbf{m}^{j(t+1)} &= \mathbb{E}_{\mathbf{w}_\star,\xi}\left[\mathbf{f}_a\left(\hat{m}_j\mathbf{w}_\star + \sqrt{\hat{Q}^{(t)}_{j,j}}\xi, \hat{\mathbf{V}}^{(t)}\right)\theta_\star\right]\\ \mathbf{Q}^{(t+1)} &= \mathbb{E}_{\mathbf{w}_\star,\xi}\left[\mathbf{f}_a\left(\hat{\mathbf{m}}\theta_\star + \sqrt{\hat{\mathbf{Q}}^{(t)}}\xi, \hat{\mathbf{V}}^{(t)}\right)\mathbf{f}_a\left(\hat{\mathbf{m}}\theta_\star + \sqrt{\hat{\mathbf{Q}}^{(t)}}\xi, \hat{\mathbf{V}}^{(t)}\right)^\top\right]\\ \mathbf{V}^{(t+1)} &= \mathbb{E}_{\mathbf{w}_\star,\xi}\left[\partial_{\mathbf{b}}\mathbf{f}_a\left(\hat{\mathbf{m}}\theta_\star + \sqrt{\hat{\mathbf{Q}}^{(t)}}\xi, \hat{\mathbf{V}}^{(t)}\right)\right]\end{cases} \tag{6.46}$$

where $\xi\sim\mathcal{N}(0,\mathbf{I}_B)$, and by rewriting Equations (6.35), (6.39), (6.44)

$$\begin{cases}\hat{\mathbf{m}}^{(t)} &= \alpha\mathbb{E}_{(z,\omega^{(t)}),}\left[\partial_z\mathbf{g}_{\text{out}}(\varphi_0(z),\omega^{(t)},\mathbf{V}^{(t)})\right]\\ \hat{\mathbf{Q}}^{(t)} &= \alpha\mathbb{E}_{(z,\omega^{(t)})}\left[\mathbf{g}_{\text{out}}(\varphi_0(z),\omega^{(t)},\mathbf{V}^{(t)})\mathbf{g}_{\text{out}}(\varphi_0(z),\omega^{(t)},\mathbf{V}^{(t)})^\top\right],\\ \hat{\mathbf{V}}^{(t)} &= -\alpha\mathbb{E}_{(z,\omega^{(t)})}\left[\partial_\omega\mathbf{g}_{\text{out}}(\varphi_0(z),\omega^{(t)},\mathbf{V}^{(t)})\right]\end{cases} \tag{6.47}$$

where $\left(z,\omega^{(t)}\right)\sim\mathcal{N}\left(0,\begin{bmatrix}\rho & \mathbf{m}^{(t)\top}\\ \mathbf{m}^{(t)} & \mathbf{Q}^{(t)}\end{bmatrix}\right)$.

and the denoising function is, as we saw above

$$\mathbf{f}_a^j(\mathbf{b},\mathbf{A}) = (\mathrm{Diag}(\mathbf{A}) + \Sigma_{\star,j})^{-1}\mathbf{b}$$

6.3.3 Closed form for Gaussian prior

Consider the magnetization $\mathbf{m} \in \mathbb{R}^2$. Asymptotically, we have that

$$\mathbf{m} \in \mathbb{R}^2 = \frac{1}{d}\mathbb{E}_{\theta_\star,\xi}\left[\begin{pmatrix}\mathbf{f}_a\left(\hat{m}_1{\theta_\star}^1 + (\xi)_1, \mathbf{A}^1\right)^\top {\theta_\star}^1 \\ \mathbf{f}_a\left(\hat{m}_2{\theta_\star}^2 + (\xi)_2, \mathbf{A}^2\right)^\top {\theta_\star}^2\end{pmatrix}\right] \tag{6.48}$$

The expression of e.g. the first coordinate of the expectation is

$$\mathbb{E}\left[\mathbf{f}_a\left(\hat{m}_1{\theta_\star}^1 + (\xi)_1, \hat{V}_1\right)^\top {\theta_\star}^1\right] = \mathbb{E}\left[{\theta_\star}^{1\top}(\mathbf{A} + \Sigma_{\star,j}^{-1})^{-1}\left({\theta_\star}_i^1\hat{m}_1 + \xi\right)\right] \tag{6.49}$$

which equates

$$m_1 = \frac{1}{d}\hat{m}_1 \times \mathbb{E}\left[Tr({\theta_\star}^1{\theta_\star}^{1\top}(\mathbf{A} + \Sigma_{\star,1}^{-1})^{-1}\right] = \hat{m}_1 \times Tr\left(\Sigma_{\star,1}(\mathbf{A} + \Sigma_{\star,1}^{-1})^{-1}\right) \tag{6.50}$$

Consider now the overlap $\mathbf{Q} \in \mathbb{R}^{2\times 2}$: for two indices i, j the overlap is given by

$$\begin{aligned}
Q_{i,j} &= \mathbb{E}_{\mathbf{w}_\star,\xi}\left[\mathbf{f}_a\left({\mathbf{w}_\star}^i\hat{m}_i + \xi^i, \hat{V}_i\right)^\top \mathbf{f}_a\left({\mathbf{w}_\star}^j\hat{m}_j + \xi^j, \hat{V}_j\right)\right] && (6.51)\\
&= \hat{m}_i\hat{m}_j \times \mathbb{E}\left[{\mathbf{w}_\star}^{i\top}\left(\hat{\mathbf{V}}_i + \Sigma_{\star,i}^{-1}\right)^{-1}\left(\hat{\mathbf{V}}^j + \Sigma_{\star,j}^{-1}\right)^{-1}{\theta_\star}^j\right] && (6.52)\\
&+ \mathbb{E}\left[\xi^{i\top}\left(\hat{\mathbf{V}}_i + \Sigma_{\star,i}^{-1}\right)^{-1}\left(\hat{\mathbf{V}}_j + \Sigma_{\star,j}^{-1}\right)^{-1}\xi^j\right] && (6.53)\\
&= \hat{m}_i\hat{m}_j \times \mathrm{Tr}\left(Cov({\mathbf{w}_\star}^i, {\mathbf{w}_\star}^j)\left(\hat{V}_i + \Sigma_{\star,i}^{-1}\right)^{-1}\left(\hat{V}_j + \Sigma_{\star,j}^{-1}\right)^{-1}\right) + && (6.54)\\
&\hat{Q}_{i,j} \times \mathrm{Tr}\left(\Omega^{i,j}\left(\hat{V}^i + \Sigma_{\star,i}^{-1}\right)^{-1}\left(\hat{V}_j + \Sigma_{\star,j}^{-1}\right)^{-1}\right) && (6.55)
\end{aligned}$$

The last line comes from the fact that $\mathrm{Cov}\left(\xi^1, \xi^j\right) = \hat{Q}^{i,j}\Omega^{i,j}$. $\mathrm{Cov}({\mathbf{w}_\star}^1, {\mathbf{w}_\star}^2)$ is the covariance between the two projections of the teacher in both random feature spaces :

$${\mathbf{w}_\star}^j = (\Omega^j)^{-1}\Phi^{j\top}\theta_\star \Rightarrow Cov({\mathbf{w}_\star}^1, {\mathbf{w}_\star}^2) = (\Omega^1)^{-1}\Phi^{1\top}\Phi^2(\Omega^2)^{-1} \tag{6.56}$$

The overlap $\mathbf{V}$ is given by

$$V^j = \frac{1}{d}\mathbb{E}\left[\mathrm{Tr}\left((\mathbf{A} + \Sigma_{\star,j}^{-1})^{-1}\right)\right] \tag{6.57}$$

6.4 Numerics

Setting We consider here the case of Ridge regression with $n/d = 5$. The data is generated by the Gaussian likelihood

$$y = \theta_\star{}^\top \mathbf{x} + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0,1)$$

The random features are generated by $\varphi^j(\mathbf{x}) = \mathrm{erf}\left(F^j \mathbf{x}\right)$.

Concerning the empirical risk minimizer, the numerics are done with two choices of λ. First, we consider $\lambda = 10^{-4}$: this value of λ very close to zero yields the double-descent phenomenon that we observed in chapter 3. The second choice of λ is the one that minimizes the test error. Because the numerical experiments are done here for Ridge regression, this choice of λ corresponds to

$$\lambda_{opt} = \mathrm{argmin}\, \mathbb{E}_{\mathbf{x},y}\left(y - \theta(\lambda)^\top \mathbf{x}\right)^2 \tag{6.58}$$

$$\theta(\lambda) = \mathrm{argmin} \sum_{i=1}^{n} (y_i - \theta^\top \varphi(\mathbf{x}_i)) + \lambda/2 \|\theta\|^2 \tag{6.59}$$

Comparison of Bayes optimal and ERM estimators In Figure 6.1, we plot the ensemble variance of the Bayes-optimal and the ERM estimators, along with their respective test errors. Unsurprisingly, the ensemble variance converges to 0 in the large overparametrization limit $\gamma \to \infty$, with a rate of $1/\gamma$. Even at optimal λ, there is a mismatch between the ensemble and the Indeed in this limit, the feature map converges

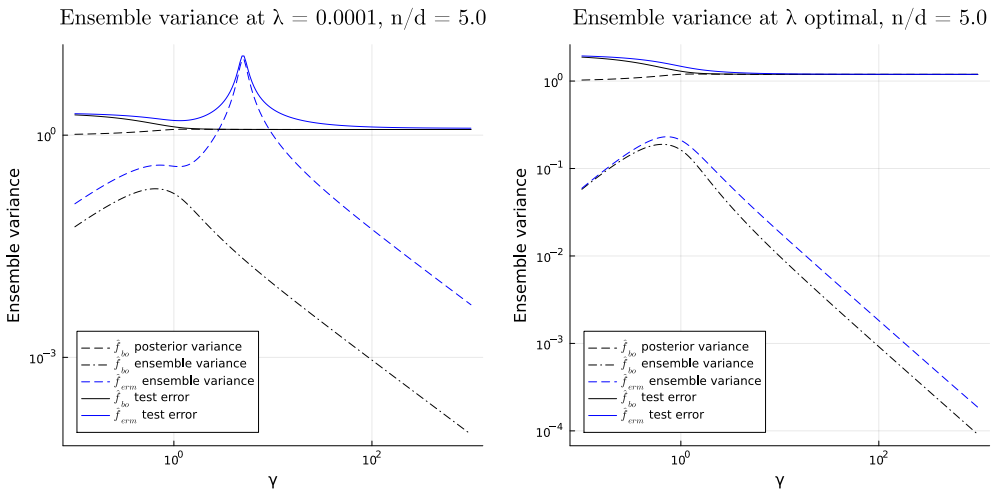


Figure 6.1: Comparison of ensemble variance for Bayes-optimal estimator $\hat{f}_{\mathrm{bo}}$ and ERM estimator $\hat{f}_{\mathrm{erm}}$ with no resampling, at $\lambda = 10^{-4}$ (Left) and λ_{opt} (Right), as a function of the overparametrization $\gamma = p/d$, with setting of section 6.4

Bootstrap and subsampling In Figure 6.2 and Figure 6.3, we plot the ensemble variance of $\hat{f}_{\text{erm}}$ and $\hat{f}_{\text{bo}}$ trained on data resampled using bootstrap (respectively subsampling with subsampling ratio 0.8). Note that here, the randomness comes from the random features and the resampling procedure. In the kernel limit corresponding, the ensemble variance converges to a non-zero limit, which is to be expected as the two estimators are trained with two resampled datasets. Hence, even in the kernel limit where the random features converge to the kernel feature map, the resampled datasets are different. Then, with these resampling schemes, the ensemble variance of the frequentist estimator does not correspond to any other notion of variance. Moreover, the reduction of the size of the training data induces a degradation of the performance of the accuracy, even of optimal λ as we can see in the figures. Hence, these resampling methods do not see, relevant in the context of overparametrized models as they do not provide any meaningful quantification of uncertainty and degrade the performance of the classifier.

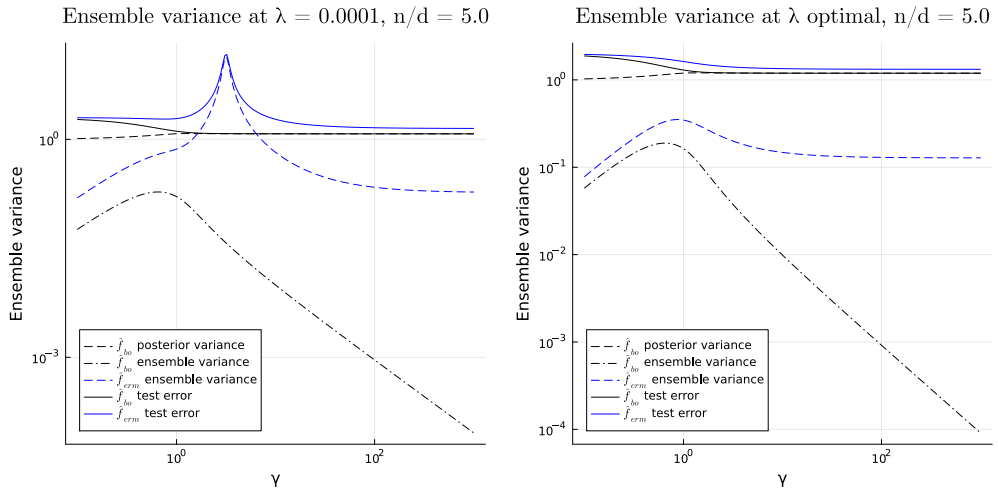


Figure 6.2: Comparison of ensemble variance for Bayes-optimal estimator $\hat{f}_{\text{bo}}$ and ERM estimator $\hat{f}_{\text{erm}}$ with bootstrap resampling, at $\lambda = 10^{-4}$ (Left) and λ_{opt} (Right), as a function of the overparametrization $\gamma = {}^{p}/_{d}$, with setting of section 6.4

6.5 Extension

The random features model studied in this section constitute a toy-model to analyze ensembles of overparametrized neural networks. A significan limitation of this model is its simplicity : indeed, for regression tasks, the original work on deep ensembles [143] considers models that output the variance of their prediction. Formally, the models considered in the paper are functions

$$\hat{f}(\mathbf{x}) = (\hat{y}, \hat{\sigma}^2)$$

The quesiton is to know whether the estimated variance $\hat{\sigma}^2$ captures anything meaningful about the randomness of the data. To analyze this type of estimators, we consider a teacher-student

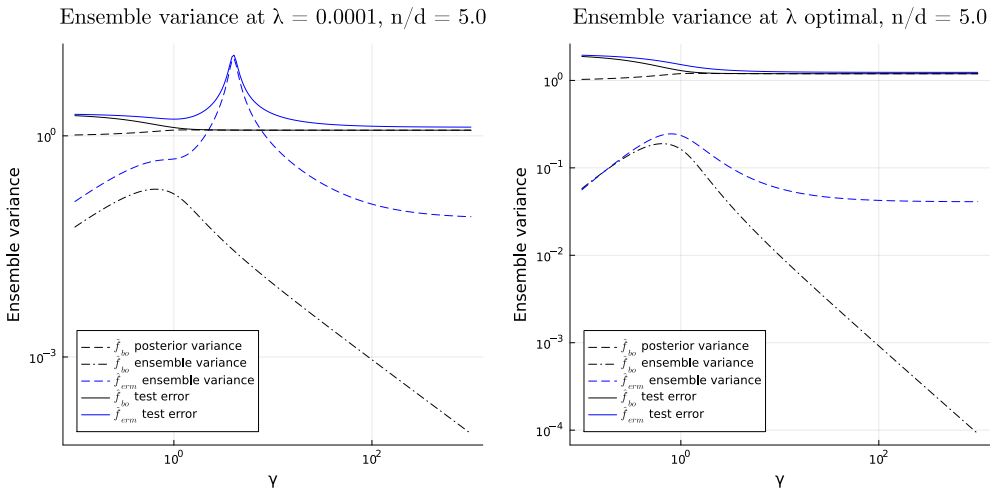


Figure 6.3: Comparison of ensemble variance for Bayes-optimal estimator $\hat{f}_{\text{bo}}$ and ERM estimator $\hat{f}_{\text{erm}}$ with subsampling (ratio 0.8), at $\lambda = 10^{-4}$ (Left) and λ_{opt} (Right), as a function of the overparametrization $\gamma = {}^{p}/_{d}$, with setting of section 6.4

model where the teacher is made of two vectors $\theta_\star, \mathbf{v}_\star$ and the data is generated by

$$y = \theta_\star^\top \mathbf{x} + \mathcal{N}(0, \sigma(\mathbf{v}_\star^\top \mathbf{x}))$$

where σ is a non-linearity. In words, the Gaussian noise is not constant but determined by another teacher vector $\mathbf{v}_\star$. This model is an instance of the *multi-index models*. As for the models studied in this thesis, one can use the state-evolution equations of AMP to analyze its performance asymptotically [233]. This study is left for future work.

7 Building conformal prediction intervals with approximate message passing

Authors : Lucas Clarté, Lenka Zdeborová

7.1 Introduction

Quantifying uncertainty is a central task in statistics, especially in sensitive applications. For regression tasks, the goal is to produce prediction sets instead of point estimates: consider here a dataset $\mathscr{D} = \left((\mathbf{x}_i, y_i)\right)_{i=1}^n$ with independent samples of the same distribution, with $\left(\mathbf{x}, y\right) \in \mathbb{R}^d \times \mathbb{R}$. Given a new input $\mathbf{x}$, we aim to produce a set of prediction $\mathscr{S}(\mathbf{x})$ that contains the observed label y with probability $1-\kappa$ for $\kappa \in (0,1)$. Conformal methods constitute a general framework used to produce such prediction sets with guarantees on their coverage. Among these methods, we can cite full and split conformal prediction (FCP and SCP) [235, 212] and Jackknife+ [19]. In full conformal prediction, the prediction set of $\mathbf{x}$ is the set of labels y whose *typicalness* is sufficiently high. The computation of this typicalness is based on leave-one-out residuals that are computed on an augmented dataset that includes the test data. Full conformal prediction has been shown to provide the correct coverage under the exchangeability of the data samples and symmetry of the scoring function under the permutation of the data. However, the computation cost of FCP is proportional to the number of training samples and the number of possible labels, making it computationally very heavy in practice. Split conformal prediction (SCP) [212, 116] is an efficient alternative to FCP, in which data is split between training and validation sets, the latter being used to calibrate the model after training. SCP is much more efficient than FCP, at the expense of statistical efficiency. Indeed, because the model is fitted on a lower amount of data than in FCP, the intervals of SCP are wider and thus less informative than FCP, as illustrated in [116]. Other works are concerned with accelerating full conformal prediction. For instance, [144, 183] approximate the computations of FCP by linearizing the solution of an empirical risk minimization problem. While [144] is limited to the Lasso, [183] is applicable to general convex empirical risks. Additionally, the work of [47] leverages incremental learning in the context of classification, kernel density estimation and k-NN regression.

Uncertainty quantification in high dimensions – In this work, we will focus our attention on the *high-dimensional* regime, where the number of samples n and the dimension d are both large with a fixed ratio $\alpha = n/d$. In this regime, many common uncertainty quantification methods are not applicable or quantify the true uncertainty wrongly. Full conformal prediction is computationally demanding as it needs to fit n estimators for each possible label. Alternatives, such as split conformal prediction or the Jackknife+ [19] are more tractable, at the expense of statistical efficiency. On the other hand, the bootstrap [60] has been shown to fail in high-dimensional linear regression [50, 131] and with deep neural networks [187]. Other methods based on ensembling, like the jackknife [200] or Adaboost [248], have been analyzed in high-dimension [223, 50, 152, 146] and have been shown to be problematic in that setting as well. Authors of [17] have shown that unpenalized quantile regression achieves under-coverage in high dimensions.

High-dimensional inference with AMP – Approximate message passing (AMP) algorithms are a class of iterative equations used to solve inference problems in high-dimension under certain distributional assumptions [72, 247]. These equations are usually derived by relaxing belief propagation equations in a graphical model [195]. A central property of AMP algorithms is their state-evolution equations that track their behaviour in high dimensions. Thanks to these state-evolution equations, AMP has been used as an analytical tool to tackle a wide range of problems in high-dimensional statistics [219, 72, 26]. In the context of uncertainty quantification, AMP has been used to study the calibration of frequentist and Bayesian classifiers [16, 53, 52] and for change point detection [11]. Additionally to these analyses, AMP algorithms have also been used in practical scenarios, such as compressed sensing [72], genomics [65], to accelerate cross-validation [188] or for change point detection [11]. Finally, in Bayesian learning, AMP can be used to compute marginals of the posterior distributions faster than with Monte-Carlo methods [53], or it can be used to establish fast sampling rigorously [76]. However, to our knowledge, no work has applied AMP to accelerate the computation of full conformal prediction.

Contributions – Our contributions are three-fold:

- First, we apply the AMP algorithm on generalized linear regression to compute the prediction intervals of full conformal prediction. AMP accelerates FCP by approximating the n leave-one-out estimators simultaneously. We show that it still provides coverage guarantees under the standard assumption that the data is exchangeable.
- Second, we introduce the Taylor-AMP algorithm, which further accelerates the computations by removing the need to fit an estimator for each possible label. We claim that Taylor-AMP is a good approximation of AMP if the empirical risk minimizer only weakly depends on each sample.
- Finally, we show that in a teacher-student model with Gaussian data and in the high-dimensional limit, AMP recovers the prediction intervals obtained by computing the leave-one-out scores exactly. As a consequence, our algorithm allows the study of conformal

prediction in high dimensions and provides a non-trivial benchmark for other methods in this regime. We also demonstrate the performance of Taylor-AMP on real data.

To our knowledge, our work is the first to apply ideas from the area of approximate message-passing algorithms to full conformal prediction and opens the door to a new research direction in which methods from high-dimensional statistics can be used practically for uncertainty quantification. The AMP-based method has the coverage guarantees celebrated in conformal prediction, with possible wide prediction intervals if the scores are estimated inacurately. The method can be used with practical advantages in scenarios where the AMP is usable for estimation, for instance, genomics [65] or MRI reconstruction [173]. Another practical interest of our work stems from the utility of having non-trivial high-dimensional settings where FCP can be evaluated rapidly, as this may be useful for theoretical research and benchmarking of other more general speed-up methods.

Notation – For a set of real values $\mathbf{z} = z_1, \cdots, z_n$ we will write $\hat{q}_\kappa(\mathbf{z})$ the κ quantile of $\mathbf{z}$ (i.e the $\kappa \times n$ largest value). The normal distribution of mean μ and variance σ^2 will be noted $\mathcal{N}(\mu, \sigma^2)$ while we will denote by $\mathcal{L}(\mu, b)$ the Laplace distribution with density $p(x) = \frac{1}{2b} e^{-\frac{|x-\mu|}{b}}$. The element-wise product between two vectors or matrices A, B will be written $A \otimes B$. Jac denotes the Jacobian of a vector-valued function.

7.2 Setting

We consider here the framework of generalized linear models for regression. Assume a training set $\mathcal{D} = \left(\mathbf{x}_i, y_i\right)_{i=1}^n$ where the n samples $\mathbf{x}_i, y_i \in \mathbb{R}^d \times \mathbb{R}$ are i.i.d. Given a test sample $\mathbf{x}$, we want to build a prediction set $\mathcal{S}(\mathbf{x})$ that contains the true label y with probability $1 - \kappa$

$$\mathbb{P}_{\mathcal{D},\mathbf{x}}\left(y \in \mathcal{S}(\mathbf{x})\right) \geqslant 1 - \kappa\,. \tag{7.1}$$

In (7.1), the randomness is on the training data and the test sample. We are interested in methods that provide the correct coverage with prediction sets of minimal size. In this work, we will focus on generalized linear models trained using empirical risk minimization

$$\hat{\theta} = \arg\min_\theta \mathcal{R}(\theta) = \arg\min_\theta \sum_{i=1}^n \ell\left(y_i, \theta^\top \mathbf{x}_i\right) + \sum_{\mu=1}^d r(\theta_\mu) \tag{7.2}$$

where ℓ is a convex loss and r is a convex regularizer. For concreteness, we will consider the cases of Ridge ($r(\theta) = \frac{\lambda}{2}\theta^2$) and Lasso ($r(\theta) = \lambda|\theta|$) regression, but our results are easily extendable to other problems such as quantile regression. Because the methods that we investigate rely on the computation of leave-one-out residuals, we introduce the leave-one out estimators $\hat{\theta}_{-i}$ that are learned on the whole dataset except sample i

7.2.1 Full conformal prediction

The basic procedure of full conformal prediction is to iterate over any possible label y, for which we define the augmented dataset $\mathscr{D}^+(y) = \mathscr{D} \cup (\mathbf{x}, y)$. We then compute the $n+1$ leave-one-out estimators $\hat{\theta}_{-i}$ trained on $\mathscr{D}^+(y)$ from which we compute the conformity scores $\sigma_i(y)$. These scores will be used to compute test statistics that will determine the inclusion y in the prediction set $\mathscr{S}(\mathbf{x})$. We first define

$$\hat{\theta}_{-i}(y) = \arg\min_{\theta} \sum_{j \neq i} \ell\left(y_j, \theta^\top \mathbf{x}_j\right) + \ell\left(y, \theta^\top \mathbf{x}\right) + \sum_{\mu} r\left(\theta_\mu\right) \tag{7.3}$$

that minimizes the empirical risk on $\mathscr{D}^+(y)$. We then define the conformity scores as the leave-one-out residuals:

$$\sigma_i(y) = |\hat{\theta}_{-i}(y)^\top \mathbf{x}_i - y_i| \tag{7.4}$$

From these scores, the prediction set $\mathscr{S}_{\text{fcp}}(\mathbf{x})$ is defined by

$$y \in \mathscr{S}_{\text{fcp}}(\mathbf{x}) \Leftrightarrow \sigma_{n+1}(y) \leqslant \hat{q}_{\lceil (1-\kappa)(n+1) \rceil / n}(\sigma(y)) \tag{7.5}$$

in other words, a label y is included in the prediction set if the conformity score of the test sample, when using the $y_{n+1} = y$, is lower than the ${}^{\lceil (1-\kappa)(n+1) \rceil}/_{n}$ quantile of the scores $\sigma_1(y), \cdots, \sigma_{n+1}(y)$. [235, 10].

In what follows, we will refer as *exact LOO* the computation of the conformity scores (7.4) by solving the minimization problems (7.3) exactly. The prediction set $\mathscr{S}_{\text{fcp}}$ achieves the desired coverage on average under the assumption that the data is exchangeable and the regression function used to produce the conformity scores is symmetric [235]. However, as noted before, fitting a model for all possible labels and computing the residuals by solving the minimization problem (7.3) is computationally heavy in practice. Methods have been developed to accelerate the computation of full conformal prediction, and in this paper, we introduce two algorithms that leverage tools from high-dimensional statistics, namely the AMP and Taylor-AMP algorithms. Contrary to exact LOO, our methods approximate the computation of the leave-one-out estimators (7.3) used to build prediction intervals.

7.2.2 Split conformal prediction

Split conformal prediction (SCP, also known as inductive conformal prediction) [192, 235] is an alternative to FCP that is computationally much cheaper. In the simplest form of SCP, $\mathscr{D}$ is split between the training and calibration sets $\mathscr{D}_{\text{train}}, \mathscr{D}_{\text{cal}}$. An estimator $\hat{\theta}$ will be fit using $\mathscr{D}_{\text{train}}$, and the conformity scores $(\sigma_i)_{i=1}^{|\mathscr{D}_{\text{cal}}|}$ are computed on the calibration set. We then extract the

$\lceil (1-\kappa) \times (n+1) \rceil$ quantile of the scores.

$$\sigma_i = |y_i - \hat{\theta}^\top \mathbf{x}_i|, \qquad Q = \hat{q}^{\lceil (1-\kappa)\times(n+1)\rceil / n}(\sigma_i) \tag{7.6}$$

$$\mathscr{S}_{\text{SCP}}(\mathbf{x}) = \left[\hat{\theta}^\top \mathbf{x} - Q, \hat{\theta}^\top \mathbf{x} + Q\right] \tag{7.7}$$

One drawback of (7.7) is that its prediction intervals are of the same size for all test samples. In this context, [206] introduced conformal quantile regression, which combines split conformal prediction and quantile regression to accommodate potential heteroskedasticity and produce intervals with data-dependent length.

7.2.3 Bayes-optimal estimator

Consider the Bayesian setting where the parameter to infer $\theta_\star$ is sampled from a prior $p_{\theta_\star}$ and the labels are generated by the likelihood distribution $p(y|{\theta_\star}^\top \mathbf{x})$. One can then compute the Bayes posterior

$$\theta \sim p(\theta|\mathscr{D}) \propto \prod_{i=1}^{n} p\left(y_i|{\theta_\star}^\top \mathbf{x}_i\right) p_{\theta_\star}(\theta_\star) \tag{7.8}$$

which yields the *Bayes-optimal* estimator, with the lowest generalisation error. This posterior distribution yields the predictive posterior distribution

$$p(y|\mathscr{D}, \mathbf{x}) = \int \mathrm{d}\theta\, p(y|\theta^\top \mathbf{x})\, p(\theta|\mathscr{D}) \tag{7.9}$$

One can then build a prediction interval $\mathscr{S}_{\text{bo}}(\mathbf{x})$ for the Bayes-optimal estimator using the *highest density interval*, which for a coverage $1-\kappa$ is the smallest set with measure $1-\kappa$.

Bayes posterior and maximum a posteriori In some settings, the empirical risk (7.2) corresponds to the logarithm of the Bayes-posterior. For instance, Ridge regression with $\lambda = 1$ corresponds to the log-posterior for the Gaussian prior $p_{\theta_\star} = \mathscr{N}(0,1)$ while Lasso with $\lambda = 1$ matches the log posterior for the Laplace prior $p_{\theta_\star} = \mathscr{L}(0,1)$.

7.3 Approximate message passing for uncertainty quantification

7.3.1 Computing residuals using AMP

We first introduce the AMP algorithm, stated in Algorithm 5. Given the regression problem (7.2), AMP approximates $\hat{\theta}_{\text{gamp}}$ of the empirical risk minimizer $\hat{\theta}$. As we will show later, using AMP to solve eq. (7.2) will allow us to simultaneously compute all the leave-one-out estimators instead of fitting the model n times, thus dramatically accelerating the computations. While AMP has

been discussed extensively in the literature, for example, in [72, 247, 170], we point the reader to Appendix E.1 for its derivation.

Algorithm 5 requires to define a *channel* and *denoising* functions, respectively noted as $\mathbf{g}_{\text{out}}$ and $\mathbf{f}_a$ and defined as follows depending on the choice of loss and regularization:

$$\mathbf{g}_{\text{out}}(y, \omega, V) = \arg\min_z \ell\left(z, y\right) + \frac{1}{2V}(z-\omega)^2 \tag{7.10}$$

$$\mathbf{f}_a(b, A) = \arg\min_z r(z) + \frac{1}{2A}(z - Ab)^2 \tag{7.11}$$

Above, $\mathbf{g}_{\text{out}}$ and $\mathbf{f}_a$ take scalar arguments but are applied on vectors in Algorithm 5 by applying the functions component-wise.

Channel and denoiser for Ridge and Lasso – In the general setting, computing $\mathbf{g}_{\text{out}}$ and $\mathbf{f}_a$ requires minimizing a scalar function. In this work, we will focus on Ridge regression and the Lasso, where these functions have a closed-form expression

$$\begin{cases} \mathbf{g}_{\text{out}}^{\text{Ridge}}(y, \omega, V) & = \frac{y-\omega}{1+V} \\ \mathbf{f}_a^{\text{Ridge}}(b, A) & = \frac{b}{\lambda + A} \end{cases}, \tag{7.12}$$

$$\begin{cases} \mathbf{g}_{\text{out}}^{\text{Lasso}}(y, \omega, V) & = \frac{y-\omega}{1+V} \\ \mathbf{f}_a^{\text{Lasso}}(b, A) & = \frac{b-\lambda}{A} \text{ if } b > \lambda, \frac{b+\lambda}{A} \text{ if } b < -\lambda \text{ else } 0 \end{cases}$$

Leave-one-out estimation – Using AMP, one can approximate the leave-one-out-estimators (7.3) and the associated residuals (7.4) with a single fit of the algorithm: for any sample i, an approximation of the $\hat{\theta}_{-i}$ is given by the following expression

$$\hat{\theta}_{-i,\text{gamp}}\left(y\right) = \hat{\theta}_{\text{gamp}}\left(y\right) - g_{i,\text{gamp}}\left(y\right) \times \mathbf{x}_i^\top \otimes \hat{\mathbf{v}}_{\text{gamp}}\left(y\right) \tag{7.13}$$

where all the vectors $\hat{\theta}_{\text{gamp}}, \hat{\mathbf{v}}_{\text{gamp}}, \mathbf{g}_{\text{gamp}}$ are computed in Algorithm 5, and the dependency on the last label y is made explicit. We refer the reader to Appendix E.1 for a justification of the above expression. The derivation is based on a close cousin of AMP, relaxed Belief Propagation (rBP), which is equivalent in the high-dimensional limit under Gaussianity assumptions on the data distribution, which we discuss in section 7.3.3. At finite dimensions d the leave-one-out estimators $\hat{\theta}_{-i,\text{gamp}}$ from (7.13) are only approximations of the solutions of (7.3) and may not be very good approximations. However, they still provide valid coverage guarantees, as essential in the conformal prediction.

Coverage guarantees for AMP – A central property of conformal prediction is that under very weak assumptions, one get prediction sets that have the correct coverage. Indeed, a standard property of FCP is that if the data is exchangeable and the score function f, which maps samples

Algorithm 5 AMP

Input: Dataset $\mathscr{D} = \left(\mathbf{x}_i, y_i\right)_{i=1}^n$

Define $\mathbf{X}^2 = \mathbf{X} \otimes \mathbf{X} \in \mathbb{R}^{n \times d}$ and initialize $\hat{\theta}^{t=0} = \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, $\hat{\mathbf{v}}^{t=0} = \mathbf{1}_d$, $\mathbf{g}^{t=0} = \mathbf{0}_n$.
for $t \leq t_{\max}$ or until convergence **do**
  $\mathbf{V}^t = \mathbf{X}^2 \hat{\mathbf{v}}^t$; $\omega^t = \mathbf{X}\hat{\theta}^t - \mathbf{V}^t \otimes \mathbf{g}^{t-1}$; /* *Update channel mean and variance*
  $\mathbf{g}^t = \mathbf{g}_{\text{out}}(\mathbf{y}, \omega^t, \mathbf{V}^t)$; $\partial \mathbf{g}^t = \partial_\omega \mathbf{g}_{\text{out}}(\mathbf{y}, \omega^t, \mathbf{V}^t)$; /* *Update channel*
  $\mathbf{A}^t = -\mathbf{X}^{2\top} \partial \mathbf{g}^t$; $\mathbf{b}^t = \mathbf{X}^\top \mathbf{g}^t + \mathbf{A}^t \otimes \hat{\theta}^t$; /* *Update prior mean and variance*
  /* *Update marginals* */
  $\hat{\theta}^{t+1} = \mathbf{f}_a(\mathbf{b}^t, \mathbf{A}^t)$; $\hat{\mathbf{v}}^{t+1} = \partial_b \mathbf{f}_a(\mathbf{b}^t, \mathbf{A}^t)$
end for
/* *Compute the leave-one-out estimators with eq.* (7.13)
for $1 \leqslant i \leqslant n$ **do**
  $\hat{\theta}_{-i,\text{gamp}} = \hat{\theta}_{\text{gamp}} - g_{\text{gamp},i} \mathbf{x}_i \otimes \hat{\mathbf{v}}_{\text{gamp}}$
end for
Return: $\hat{\theta}_{\text{gamp}}, (\hat{\theta}_{-i,\text{gamp}})_{i=1}^n$

to confirmity scores, is symmetric, then the prediction intervals given by f satisfy eq. (7.1), as shown in [235]. Recall that *symmetric* means here that for any permutation $s : [1, n] \to [1, n]$, then $\hat{f}((\mathbf{x}_{s(i)}, y_{s(i)}))_{i=1}^n = \left(\sigma_{s(i)}\right)_{i=1}^n$. We show in Appendix appendix E.3 that AMP is symmetric, which leads to the following property:

Property 7.3.1. *Consider training data $\mathscr{D} = \left(\mathbf{x}_i, y_i\right)_{i=1}^n$ and a test sample $\mathbf{x}$ where the data is exchangeable. Consider the conformity scores $\left(\sigma_{i,\text{gamp}}\right)_i = |y_i - \hat{\theta}_{-i,\text{gamp}}^\top \mathbf{x}_i|$ where the leave-one-out estimators are computed using AMP:*

$$\hat{\theta}_{-i,\text{gamp}} = \hat{\theta}_{\text{gamp}} - g_{i,\text{gamp}} \mathbf{x}_i^\top \otimes \hat{\mathbf{v}}_{\text{gamp}}$$

and the confidence set with target coverage $1 - \kappa$, defined as

$$\mathscr{S}_{\text{fcp}}(\mathbf{x}) = \left\{ y | \sigma_{n+1} \leqslant \hat{q}_{\lceil (1-\kappa)(n+1) \rceil / n}(\sigma_i) \right\}$$

then, $\mathscr{S}_{\text{fcp}}$ achieves coverage at $1 - \kappa$ on average

$$\mathbb{P}_{\mathscr{D},\mathbf{x}}\left(y \in \mathscr{S}_{\text{fcp}}(\mathbf{x})\right) \geqslant 1 - \kappa \tag{7.14}$$

Note that theorem 7.3.1 is valid at finite dimension and independently of the data distribution : AMP needs not to approximate precisely the leave-one-out residuals to achieve the correct coverage.

7.3.2 Taylor-AMP

In the previous paragraphs, we saw that AMP can be used to accelerate the computation of the conformity scores $\sigma_i(y)$ by computing the n leave-one-out estimators simultaneously for a fixed label y of the test data. In this section, we present a variant of AMP called Taylor-AMP and described in Algorithm 6, whose goal is to further accelerate AMP by approximating the iteration over the set of possible labels: Taylor-AMP will compute the leave-one out estimators $\hat{\theta}_{-i,\text{gamp}}(y)$ without fitting the model for each label y. The general idea is to approximate the quantities $\hat{\theta}_{-i}^\top \mathbf{x}_i$ by an affine function around a reference value $\hat{y}$. To do so, we will compute the derivative of the estimators $\hat{\theta}_{-i}(y)$ with respect to y, around $\hat{y}$. Then, for any possible label y, the corresponding scores will be approximated with

$$\begin{aligned}\sigma_i(y) &= |y_i - \hat{\theta}_{-i,\text{gamp}}(y)^\top \mathbf{x}_i| \\ &= |y_i - \left(\hat{\theta}_{-i,\text{gamp}}(\hat{y}) + (y-\hat{y})\frac{\partial \hat{\theta}_{-i,\text{gamp}}}{\partial y}(\hat{y})\right)^\top \mathbf{x}_i|\end{aligned}$$

The central part is the estimation of $\frac{\partial \hat{\theta}_{-i,\text{gamp}}}{\partial y}$ using AMP. Indeed, $\hat{\theta}_{\text{gamp}}$ solves a fixed point equation of the form

$$\mathbf{f}_{\text{gamp}}(\hat{\theta}_{\text{gamp}}(y_{n+1}), y_{n+1}) = \hat{\theta}_{\text{gamp}}(y_{n+1})$$

where we only make explicit its dependency y_{n+1} as the rest of the training data is fixed. Using the implicit function theorem, one can compute the derivative $\frac{\partial \hat{\theta}_{\text{gamp}}}{\partial y_{n+1}}$ from the implicit equation

$$\frac{\partial \hat{\theta}_{\text{gamp}}}{\partial y}(\hat{y}) = \left(\mathbf{I} - \text{Jac}(\mathbf{f}_{\text{gamp}})\right)^{-1} \frac{\partial \mathbf{f}_{\text{gamp}}}{\partial y}(\hat{y}) \tag{7.15}$$

which can be solved iteratively:

$$\Delta\hat{\theta}^{t+1} = \text{Jac}(\mathbf{f}_{\text{gamp}})(\Delta\hat{\theta}^t) + \frac{\partial \mathbf{f}_{\text{gamp}}}{\partial y}(\hat{y}). \tag{7.16}$$

In Algorithm 6, we iterate eq. (7.16) until convergence, at which point $(\Delta\hat{\theta}, \Delta\hat{\mathbf{v}}, \Delta\mathbf{g}) = \left(\frac{\partial \hat{\theta}}{\partial y}, \frac{\partial \hat{\mathbf{v}}}{\partial y}, \frac{\partial \mathbf{g}}{\partial y}\right)$. We provide more details, in particular the explicit form of the function $\mathbf{f}_{\text{gamp}}$ in Appendix E.2.

To summarize, Algorithm 6 computes the derivatives $\Delta\hat{\theta}_{\text{gamp}}, \Delta\hat{\mathbf{v}}_{\text{gamp}}, \Delta\mathbf{g}_{\text{gamp}}$ of $\hat{\theta}_{\text{gamp}}, \hat{\mathbf{v}}_{\text{gamp}}, \mathbf{g}_{\text{gamp}}$ around some value $\hat{y} = \hat{\theta}^\top \mathbf{x}_n$ where $\hat{\theta}$ minimizes (7.2) on $\mathscr{D}$. We can then approximate the leave-one-out estimators $\hat{\theta}_{-i,\text{gamp}}(y)$ by differentiating the expression of the leave-one-out estimators (7.13), which yields

$$\begin{aligned}\frac{\partial \hat{\theta}_{-i,\text{gamp}}}{\partial y}(y) = \Delta\hat{\theta} &- g_{i,\text{gamp}}(\hat{y}) \times \mathbf{x}_i \otimes \Delta\hat{\mathbf{v}}_{\text{gamp}} \\ &- \Delta g_{i,\text{gamp}} \mathbf{x}_i \otimes \hat{\mathbf{v}}_{\text{gamp}}(\hat{y})\end{aligned}$$

Algorithm 6 Taylor-AMP

Input: Data $\mathbf{X} \in \mathbb{R}^{n \times d}$, $\mathbf{y} \in \mathbb{R}^n$

Compute $\left(\hat{\theta}, \hat{\mathbf{v}}, \omega, \mathbf{V}, \mathbf{A}, \mathbf{b}, \mathbf{g}, \partial\mathbf{g}\right)$ using Algorithm 5
Initialize $\Delta\hat{\theta}^0 = \mathbf{0}, \Delta\hat{\mathbf{v}}^0 = \mathbf{0}, \Delta\mathbf{V}^0 = 0, \Delta\omega^0 = \mathbf{0}$
for $t \leq t_{\max}$ or until convergence **do**
$\Delta\mathbf{V}^t = \mathbf{X}^2 \Delta\hat{\mathbf{v}}^{t-1}$
$\Delta\omega^t = X\Delta\hat{\theta}^{t-1} - \Delta V \otimes \mathbf{g}^{t-1} - V \otimes \Delta\mathbf{g}^{t-1}$
$\Delta\mathbf{g}^t = \partial_\omega \mathbf{g}_{\text{out}} \Delta\omega^t + \partial_{\mathbf{V}} \mathbf{g}_{\text{out}} \Delta\mathbf{V}^t + \left(\partial_y \mathbf{g}_{\text{out}|n}\right) \mathbf{e}_n$
$\Delta\partial\mathbf{g}^t = \partial^2_{\omega^2} \mathbf{g}_{\text{out}} \Delta\omega^t + \partial_{\mathbf{V}}\partial_\omega \mathbf{g}_{\text{out}} \Delta\mathbf{V}^t + \left(\partial_y \partial_\omega \mathbf{g}_{\text{out}|n}\right) \mathbf{e}_n$
$\Delta\mathbf{A}^t = -X^{2\top} \Delta\partial\mathbf{g}^t$
$\Delta\mathbf{b}^t = X^\top \Delta\mathbf{g}^t$
$\Delta\hat{\theta}^t = \partial_b f_w \Delta\mathbf{b}^t + \partial_A f_w \Delta\mathbf{A}^t$
$\Delta\hat{\mathbf{v}}^t = \partial_b \left(\partial_b f_w\right) \Delta\mathbf{b}^t + \partial_A \left(\partial_b f_w\right) \Delta\mathbf{A}^t$
end for
Return: Derivatives $\left(\Delta\hat{\theta}_{\text{gamp}}, \Delta\hat{\mathbf{v}}_{\text{gamp}}, \Delta\mathbf{g},\right)$

which allows us to compute the conformity scores of FCP in eq. (7.4).

Justification of Taylor-AMP – Taylor-AMP is based on the idea that the value of the last sample only weakly affects the value of the estimator $\hat{\theta}_{\text{gamp}}$. More precisely, in high-dimensions as $n, d \to \infty$, $\frac{\hat{\theta}_{\text{gamp}}}{\partial y} \to 0$. This implies for instance that the data contains no outliers, whose value would induce a significant change in $\hat{\theta}_{\text{gamp}}$. We refer the reader to appendix E.2.1 for more details: we numerically observe for synthetic Gaussian data that Taylor-AMP accurately approximates the leave-one-out predictions $\hat{\theta}_{-i}^\top \mathbf{x}_i$ in high dimensions.

7.3.3 Exactness in high dimensions for Gaussian data

In this section, we provide guarantees on the size of the prediction intervals using conformity scores produced by AMP in high dimensions. Suppose that the samples $(\mathbf{x}, y)$ follow the distribution

$$y_i \sim p(\cdot|{\theta_\star}^\top \mathbf{x}_i), \qquad \mathbf{x}_i \sim \mathcal{N}(0, {}^{I_d}/_d) \tag{7.17}$$

for $\theta_\star$ *teacher* vector that is to be recovered from the training data and with a likelihood function $p(\cdot|z)$ that is not known to the statistician e.g. $y = {\theta_\star}^\top \mathbf{x} + \varepsilon$, with $\varepsilon \sim \mathcal{N}(0,1)$. Assume also that $\theta_\star$ is random and its components are independently sampled from the same distribution $p_{\theta_\star}$. In what follow we will assume that $p_{\theta_\star}$ is either the standard normal $p_{\theta_\star} = \mathcal{N}(0,1)$ or the Laplace distribution $p_{\theta_\star}(z) = \frac{1}{2}e^{-|z|}$. Then, under these assumptions on $\theta_\star$ and the data, in the high-dimensional limit where $n, d \to \infty$ with ${}^n/_d$ fixed, the estimator $\hat{\theta}_{\text{gamp}}$ converges to the true empirical risk minimizer, provided the samples $\mathbf{x}_i, y_i$ come from the distribution (7.17) as shown

Problem	exact LOO	Taylor-AMP	SCP	CQP	Coverage of Taylor-AMP
Lasso ($\lambda = 1$)	3.9 ± 0.45	4.2 ± 0.8	4.3 ± 0.9	4.7 ± 0.9	0.9
Ridge ($\lambda = 1$)	3.7 ± 0.34	3.9 ± 0.4	4.4 ± 0.8	4.7 ± 0.9	0.89
Ridge ($\lambda = 0.01$)	4.4 ± 0.5	4.7 ± 0.7	5.7 ± 1.2	4.8 ± 0.9	0.91

Table 7.1: Mean and standard deviation, of the size of prediction intervals at coverage $q = 0.9$, with random data at $n = 100, d = 50$ generated from a Gaussian teacher. For all methods except exact LOO, values are averaged over 1000 test samples.

in [247, 170, 72]. Thus, for any test sample $\mathbf{x}$ and any $\varepsilon > 0$

$$\mathbb{P}_{\mathscr{D},\mathbf{x}}\left(|\hat{\theta}^\top_{\text{gamp}}\mathbf{x} - \hat{\theta}^\top\mathbf{x}| < \varepsilon\right) \xrightarrow[n,d\to\infty,{}^{n}/_{d}=\alpha]{} 1 \tag{7.18}$$

Moreover, we show in appendix E.1 that in this high-dimensional limit, the estimators $\hat{\theta}_{i,\text{gamp}}$ of eq. (7.13) converge to the true leave-one-out estimators eq. (7.3).

7.4 Numerical experiments

In this section, we first show that on synthetic Gaussian data, our method correctly approximates the conformity scores while accelerating their computations by orders of magnitude. This allows us to compare FCP to other methods such as split conformal prediction and the Bayes-optimal estimator in a non-trivial high-dimensional setting. We then evaluate the methods on real datasets, showing the usefulness of AMP for uncertainty quantification beyond synthetic data with no distributional assumptions. In all of our numerical experiments, the prediction intervals will have a target coverage of 90%

7.4.1 Synthetic high-dimensional benchmark

Coverage and size of prediction intervals – In this section, we consider synthetic data generated by the model described in eq. (7.17). In table 7.1, we first compute the coverage of Taylor-AMP for the Ridge and Lasso regressions at different values of λ. We see in the right-most column that our method provides the desired coverage. Moreover, on this synthetic data we compare the size of prediction intervals produced by exact LOO and observe that the average length are almost equal. This numerically validates the statement of section 7.3.3 and shows that with Gaussian data, even at moderate dimension, Taylor-AMP is very close to exact LOO.

We also compute the similarity between the prediction intervals produced by Taylor-AMP with those returned by exact LOO, to show that both methods return the same intervals. To this end, we compute the *Jaccard index* between the exact and approximate intervals. Recall that the

Problem	JI (Taylor-AMP)	JI (SCP)
Ridge ($\lambda = 0.01$)	0.93 ± 0.04	0.80 ± 0.12
Ridge ($\lambda = 0.1$)	0.95 ± 0.04	0.83 ± 0.1
Ridge ($\lambda = 1$)	0.98 ± 0.02	0.84 ± 0.04
Lasso ($\lambda = 0.01$)	0.90 ± 0.06	0.86 ± 0.11
Lasso ($\lambda = 0.1$)	0.92 ± 0.05	0.87 ± 0.09
Lasso ($\lambda = 1$)	0.97 ± 0.03	0.88 ± 0.08

Table 7.2: Jaccard index (JI) between exact LOO and Taylor-AMP and SCP for different estimators, with data generated from a Gaussian teacher, and $d = 100, n = 200$. We report the averages and standard deviation over 20 test samples.

Jaccard index between two sets $\mathscr{S}_1, \mathscr{S}_2$ is defined as

$$\mathscr{J}(\mathscr{S}_1, \mathscr{S}_2) = \frac{|\mathscr{S}_1 \cap \mathscr{S}_2|}{|\mathscr{S}_1 \cup \mathscr{S}_2|} \in [0, 1]$$

values closer to 1 indicate more precise approximations. We report our findings in table 7.2, where we evaluate the Jaccard index $\mathscr{J}\left(\mathscr{S}_{\text{fcp}}(\mathbf{x}), \mathscr{S}_{\text{Taylor-AMP}}(\mathbf{x})\right)$ and $\mathscr{J}\left(\mathscr{S}_{\text{fcp}}(\mathbf{x}), \mathscr{S}_{SCP}(\mathbf{x})\right)$. Values are the averaged over 20 test samples. We observe that Taylor-AMP has a higher similarity to FCP than SCP. This confirms that even though our method is an approximation of FCP, even at moderate dimensions it provides intervals that are very close to the exact ones.

Computation speed – In Figure 7.1, we compare the time to compute $\mathscr{S}(\mathbf{x})$ for a single test sample $\mathbf{x}$, as a function of the dimension for a fixed sampling ratio $\alpha = {}^{n}/_{d}$. We observe that our method provides a speed-up over exact LOO by more than two orders of magnitude, and allows us to quantify the uncertainty for dimensions about 10 times higher for the same amount of time. With the Taylor-AMP algorithm, we can readily treat problems of dimension 10^4.

To summarize, so far our numerical results show that our algorithm approximates precisely exact LOO, while being order of magnitudes faster. This allows to benchmark FCP against other methods in large dimensions, as we do in the following paragraphs.

Comparison with Bayes posterior – We compare the prediction intervals of conformal prediction with those of the Bayes-optimal estimator as defined in Section 7.2.3. Recall that the Bayes-optimal estimator has the lowest generalisation error when the data-generating process is known. When the prior $p_{\theta_\star}$ is Gaussian, the log-posterior exactly corresponds to Ridge regression with $\lambda = 1$. Likewise, for a Laplace prior on $\theta_\star$, the log-posterior is exactly the empirical risk of Lasso, with $\lambda = 1$. In Table 7.3, we compare the average length of the prediction intervals provided by FCP with the highest density intervals of the Bayes posterior. Note that for a Gaussian prior, the posterior distribution is also Gaussian and can be easily sampled. However, this is not the case for a Laplace prior. In general, one would sample the posterior using Monte-Carlo

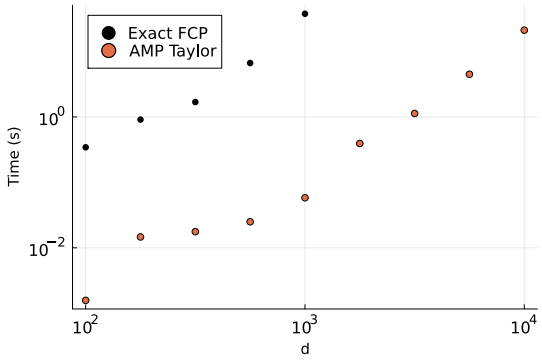


Figure 7.1: Computation time to produce a single prediction interval, for exact LOO and Taylor-AMP , for Lasso at $\lambda = 1$ and $n/d = 0.5$.

methods. However, within our synthetic data setting, we can leverage the AMP algorithm 5 to sample the posterior [53]. AMP is much faster than costly Monte-Carlo sampling, while being exact in the high-dimensional limit. Lines in bold represent the matched settings where the minimized empirical risk matches the true log posterior. We see that in these settings, FCP has almost optimal length, as it is very close to those of the Bayes-optimal estimator. On the other hand, when λ has a value that does not match the true prior, then the intervals obtained with Taylor-AMP are significantly larger than those of Bayes, for instance with $\lambda = 0.1$.

Teacher	Regularization	Bayes	Taylor-AMP
Gaussian	Ridge ($\lambda = 0.1$)	4.4	4.9
	Ridge ($\lambda = 1.0$)		**4.4**
	Lasso ($\lambda = 1.0$)		4.9
Laplace	Lasso ($\lambda = 0.1$)	5.1	13.2
	Lasso ($\lambda = 1.0$)		**5.6**
	Ridge ($\lambda = 1.0$)		4.9

Table 7.3: Average length of prediction intervals of Bayes posterior and FCP with Taylor-AMP , at $d = 250, n = 125$. Measures are averaged over 200 samples of both $\mathscr{D}$ and the single test sample. Bold lines correspond to the matched setting where the empirical risk corresponds to the log-posterior of the data-generating process.

Comparison with split conformal prediction – In table 7.1, we compare the length of the prediction intervals of Taylor-AMP with SCP described in 7.7, and to conformalized quantile regression (CQP) [206], where split conformal prediction is applied on two estimators of the quantile functions of the likelihood $p(y|\mathbf{x})$. We observe that as expected, our method provides tighter intervals while having the correct coverage.

Comparison with approximate homotopy– In this section, we compare the performance of Taylor-AMP with *Approximate homotopy* introduced in [183]. Like Taylor-AMP , the goal of

approximate homotopy is to approximate the computation of FCP. In table 7.4, we compare both methods in terms of computation time and coverage, on Lasso regression with $\lambda = 1, \alpha = 0.5$ for a target coverage of 0.9. For approximate homotopy we used the code provided by the authors of [183]with the default sets of parameters. From the table, we see that at a relatively low dimension, both methods have acceptable runtime but from $d = 150$ onwards, approximate homotopy becomes extremely slow, making it unusable in the high-dimensional regime.

Dataset	Regularization	Method	Size	Time	Coverage
Gaussian ($d = 100, n = 1000$)	Lasso ($\lambda = 1$)	Taylor-AMP	4.4	0.007	0.89
		Approximate homotopy	4.6	0.025	0.9
Gaussian ($d = 250, n = 100$)	Lasso ($\lambda = 1$)	Taylor-AMP	4.6	0.044	0.9
		Approximate homotopy	4.5	10.6	0.89
Boston	Lasso ($\lambda = 1$)	AMP	1.6	0.03	0.88
		Taylor-AMP	9.1	0.03	0.91
		Approximate homotopy	1.5	0.02	0.95
Riboflavin	Lasso ($\lambda = 0.25$)	AMP	2.34	0.42	0.89
		Taylor-AMP	9.5	0.4	0.95
		Approximate homotopy	0.23	17.7	1.0

Table 7.4: Comparison of Taylor-AMP and approximate homotopy on synthetic (Top) and real (Bottom) datasets. We observe that at high dimension, Taylor-AMP is faster by several orders of magnitude, while providing the correct coverage.

7.4.2 Real data

We also compare the performance of both AMP and Taylor-AMP with approximate homotopy on real data. We use for this two datasets, the Boston housing dataset and Riboflavin production data[42]. We validate that our methods provide the correct coverage.

In particular, we observe that our method significantly speeds up FCP compared to approximate homotopy on the riboflavin dataset, where $d = 4088$, in exchange of wide prediction intervals. On the other hand, we observe that Taylor-AMP behaves poorly as the prediction intervals are much wider than the two other methods.

Note that the convergence of AMP heavily depends on the properties of the input matrix X, making it somewhat fragile for real data. However, in cases in which it converges such as the ones shown here, the coverage is competitive.

7.5 Conclusion

In this paper, we introduce a method to accelerate the computations of full conformal prediction while guaranteeing confidence sets with the correct coverage. Our method leverages methods stemming from high-dimensional statistics literature, namely the approximate message passing (AMP) algorithm. Our numerical experiments on synthetic and real data show that the method has the potential to provide narrow confidence sets (with coverage guaranteed) while reducing the computation time by almost three orders of magnitude compared to the baseline. Our method has a particular theoretical interest, as Taylor-AMP can be used to investigate more easily the properties of full conformal prediction in high dimensions by drastically speeding up the simulations. The proposed algorithm can leverage the fact that it is asymptotically exact on the synthetic Gaussian data and these data can thus be used as a benchmark for other speed-up methods in high-dimensions.

Possible extensions – In appendix E.5, we describe how to extend our method to classification, and leave a more detailed study to future work. While we only investigated conformal prediction for frequentist estimators, AMP can be used to sample from Bayesian posteriors more efficiently than Monte-Carlo methods. Our results could thus be extended to Bayesian conformal prediction, where the conformity scores are given by the predictive posterior [82]. Moreover, the computation of leave-one-out estimators could be applied to other methods such as the Jackknife+[19]. Finally, we did not leverage the state-evolution equations of AMP, which allow us to characterize precisely the behaviour of AMP in high dimensions. This theoretical study is left to future work.

Limitations – One limitation of our work is the assumption weak dependence on every sample in Taylor-AMP . Further, while we show that our method is applicable to real data, the stability of AMP heavily depends on the input matrix. The extension of our method to more complex algorithms of a similar kind such as VAMP [203], which would make our method applicable to a broader set of data, and is left to future work.

A Appendix of chapter 2

A.1 Cavity derivation of the analytical results

In this appendix, we sketch how the self-consistent equations (2.16) and (2.18) characterizing the sufficient statistics $(q_{\text{bo}}, m, q_{\text{erm}})$ can actually be derived via the heuristic cavity method [171, 170] from statistical physics.

We shall use the notation of Rangan's GAMP algorithm [202] and present our results as a derivation of GAMP algorithm from cavity, or belief propagation, as in [247]. This allows to connect all our results as well as the state evolution equations of the GAMP 2 algorithm in a single framework. Note that in its most general form, GAMP can be used both as an algorithm for estimating the marginals of the posterior distribution $\mathbf{w}_{\text{amp}} = \mathbb{E}[w|\mathscr{D}]$ or to minimize the empirical risk in 2.4 - the only difference between the two being the choice of denoising functions (f_{out}, f_w).

The novelty of our approach consists of running two GAMP algorithms in parallel *on the same instance* of data $\mathscr{D} = \{(\mathbf{x}^\mu, y^\mu)\}_{\mu=1}^n$ drawn from the probit model introduced in eq. (2.2). Although we run the two versions of GAMP independently, they are correlated through the data $\mathscr{D}$ - and our goal is to characterize exactly their joint distribution.

A.1.1 Joint state evolution

Consider we are running two AMPs in parallel, one for BO estimation and one for ERM. To distinguish both messages, we will denote ERM messages with a tilde: $\tilde{\omega}^t, \tilde{V}^t$, etc. To derive the asymptotic distribution of the estimators $(\hat{\mathbf{w}}_{\text{amp}}, \hat{\mathbf{w}}_{\text{erm}})$, it is more convenient to start from a close cousin of AMP: the reduced Belief Propagation equations (rBP). Note that in the high-dimensional limit that we are interested in this manuscript, rBP is equivalent to AMP, see for instance [14] or [15] for a detailed derivation. Written in coordinates, the rBP equations are given

by:

$$\begin{cases} \omega^t_{\mu\to i} = \sum_{j\neq i} x^\mu_j \hat{w}^t_{j\to\mu} \\ V^t_{\mu\to i} = \sum_{j\neq i} (x^\mu_j)^2 \hat{c}^t_{j\to\mu} \end{cases}, \qquad \begin{cases} g^t_{\mu\to i} = f_{\text{out}}(y^\mu, \omega^t_{\mu\to i}, V^t_{\mu\to i}) \\ \partial g^t_{\mu\to i} = \partial_\omega f_{\text{out}}(y^\mu, \omega^t_{\mu\to i}, V^t_{\mu\to i}) \end{cases} \tag{A.1}$$

$$\begin{cases} b^t_{\mu\to i} = \sum_{\nu\neq\mu} x^\nu_i g^t_{\nu\to i} \\ A^t_{\mu\to i} = -\sum_{\nu\neq\mu} (x^\nu_i)^2 \partial g^t_{\nu\to i} \end{cases}, \qquad \begin{cases} \hat{w}^{t+1}_{i\to\mu} f_w(b^t_{i\to\mu}, A^t_{i\to\mu}) \\ \hat{c}^{t+1}_{i\to\mu} \partial_b f_w(b^t_{\mu\to i}, A^t_{\mu\to i}) \end{cases} \tag{A.2}$$

where (f_{out}, f_w) denote the denoising functions that could be associated either to BO or ERM estimation, and that can be generically written in terms of an estimation likelihood P_{out} and prior P_w as:

$$\begin{cases} f_{\text{out}}(y,\omega,V) &= \partial_\omega \log \mathcal{Z}_{\text{out}}(y,\omega,V) \\ \mathcal{Z}_{\text{out}}(y,\omega,V) &= \int_{\mathbb{R}} \frac{\mathrm{d}x}{\sqrt{2\pi V}} e^{-\frac{(x-\omega)^2}{2V}} P_{\text{out}}(y|x) \end{cases}, \quad \begin{cases} f_w(b,A) &= \partial_b \log \mathcal{Z}_w(b,A) \\ \mathcal{Z}_w(b,A) &= \int_{\mathbb{R}} \mathrm{d}w\ P_w(w) e^{-\frac{1}{2}Aw^2 + bw} \end{cases} \tag{A.3}$$

By assumption, the rBP messages are independent from each other, and since we are running both BO and ERM independently, they are only coupled to each other through the data, which has been generated by the same data model:

$$y^\mu \sim P_0(\cdot|\theta_\star{}^\top \mathbf{x}^\mu), \qquad \mathbf{x}^\mu \sim \mathcal{N}(0, {}^1\!/_d \mathbf{I}_d), \qquad \theta_\star \sim \prod_{i=1}^d P_0(w_{\star i}). \tag{A.4}$$

Note that here we work in a more general setting than the one in the main manuscript (2.2). Indeed, the derivation presented here work for *any* factorised distribution of teacher weights $\theta_\star$ and any likelihood P_0 (of which the probit is a particular case). For convenience, define the so-called *teacher local field*:

$$z_\mu = \sum_{j=1}^d x^\mu_j w_{\star j} \tag{A.5}$$

Step 1: Asymptotic joint distribution of $(z_\mu, \omega^t_{\mu\to i}, \tilde{\omega}^t_{\mu\to i})$

Note that $(z_\mu, \omega^t_{\mu\to i}, \tilde{\omega}^t_{\mu\to i})$ are given by a sum of independent random variables with variance $d^{-1/2}$, and therefore by the Central Limit Theorem in the limit $d\to\infty$ they are asymptotically Gaussian. Therefore we only need to compute their means, variances and cross correlation. The means are straightforward, since x^μ_i have mean zero and therefore they will also have mean zero.

The variances are given by:

$$\mathbb{E}\left[z_\mu^2\right] = \mathbb{E}\left[\sum_{i=1}^{d}\sum_{j=1}^{d} x_i^\mu x_j^\mu w_{\star i} w_{\star j}\right] = \sum_{i=1}^{d}\sum_{j=1}^{d}\mathbb{E}\left[x_i^\mu x_j^\mu\right] w_{\star i} w_{\star j} = \frac{1}{d}\sum_{i=1}^{d}\sum_{j=1}^{d}\delta_{ij} w_{\star i} w_{\star j}$$
$$= \frac{||\theta_\star||_2^2}{d} \underset{d\to\infty}{\to} \rho \tag{A.6}$$

$$\mathbb{E}\left[\left(\omega_{\mu\to i}^t\right)^2\right] = \mathbb{E}\left[\sum_{j\neq i}^{d}\sum_{k\neq i}^{d} x_j^\mu x_k^\mu \hat{w}_{j\to\mu}^t \hat{w}_{k\to\mu}^t\right] = \sum_{j\neq i}^{d}\sum_{k\neq i}^{d}\mathbb{E}\left[x_j^\mu x_k^\mu\right] \hat{w}_{j\to\mu}^t \hat{w}_{k\to\mu}^t$$
$$= \frac{1}{d}\sum_{j\neq i}^{d}\sum_{k\neq i}^{d}\delta_{jk}\hat{w}_{j\to\mu}^t\hat{w}_{k\to\mu}^t = \frac{1}{d}\sum_{j\neq i}^{d}\left(\hat{w}_{j\to\mu}^t\right)^2 = \frac{||\hat{\mathbf{w}}^t||_2^2}{d} - \frac{1}{d}(\hat{w}_{i\to\mu}^t)^2 \underset{d\to\infty}{\to} q^t \tag{A.7}$$

$$\mathbb{E}\left[z_\mu \omega_{\mu\to i}^t\right] = \mathbb{E}\left[\sum_{j\neq i}^{d}\sum_{k=1}^{d} x_j^\mu x_k^\mu \hat{w}_{j\to\mu}^t w_{\star k}\right] = \sum_{j\neq i}^{d}\sum_{k=1}^{d}\mathbb{E}\left[x_j^\mu x_k^\mu\right] \hat{w}_{j\to\mu}^t w_{\star k}$$
$$= \frac{1}{d}\sum_{j\neq i}^{d}\sum_{k=1}^{d}\delta_{jk}\hat{w}_{j\to\mu}^t w_{\star k} = \frac{1}{d}\sum_{j\neq i}^{d}\hat{w}_{j\to\mu}^t w_{\star j} = \frac{\hat{\mathbf{w}}^t\cdot\theta_\star}{d} - \frac{1}{d}\hat{w}_{i\to\mu}^t w_{\star i} \underset{d\to\infty}{\to} m^t \tag{A.8}$$

$$\mathbb{E}\left[\omega_{\mu\to i}^t \tilde{\omega}_{\mu\to i}^t\right] = \mathbb{E}\left[\sum_{j\neq i}^{d}\sum_{k\neq i}^{d} x_j^\mu x_k^\mu \hat{w}_{j\to\mu}^t \tilde{\hat{w}}_{k\to\mu}^t\right]$$
$$= \sum_{j\neq i}^{d}\sum_{k\neq i}^{d}\mathbb{E}\left[x_j^\mu x_k^\mu\right]\hat{w}_{j\to\mu}^t\tilde{\hat{w}}_{k\to\mu}^t = \frac{1}{d}\sum_{j\neq i}^{d}\sum_{k\neq i}^{d}\delta_{jk}\hat{w}_{j\to\mu}^t\tilde{\hat{w}}_{k\to\mu}^t$$
$$= \frac{1}{d}\sum_{j\neq i}^{d}\hat{w}_{j\to\mu}^t\tilde{\hat{w}}_{j\to\mu}^t = \frac{\hat{\mathbf{w}}^t\cdot\tilde{\hat{\mathbf{w}}}^t}{d} - \frac{1}{d}\hat{w}_{i\to\mu}^t\tilde{\hat{w}}_{i\to\mu}^t \underset{d\to\infty}{\to} Q^t \tag{A.9}$$

where we have used that $\hat{w}_{i\to\mu}^t = O(d^{-1/2})$ to simplify the sums at large d. Summarising our findings:

$$(z_\mu, \omega_{\mu\to i}^t, \tilde{\omega}_{\mu\to i}^t) \sim \mathcal{N}\left(\mathbf{0}_3, \begin{bmatrix} \rho & m^t & \tilde{m}^t \\ m^t & q^t & Q^t \\ \tilde{m}^t & Q^t & \tilde{q}^t \end{bmatrix}\right) \tag{A.10}$$

with:

$$\rho \equiv \frac{1}{d}||\theta_\star||^2, \qquad q^t \equiv \frac{1}{d}||\hat{\mathbf{w}}_{\mathrm{BO}}^t||^2, \qquad \tilde{q}^t \equiv \frac{1}{d}||\hat{\mathbf{w}}_{\mathrm{ERM}}^t||^2$$
$$m^t \equiv \frac{1}{d}\hat{\mathbf{w}}_{\mathrm{BO}}\cdot\theta_\star, \qquad \tilde{m}^t \equiv \frac{1}{d}\hat{\mathbf{w}}_{\mathrm{ERM}}\cdot\theta_\star, \qquad Q^t \equiv \frac{1}{d}\hat{\mathbf{w}}_{\mathrm{BO}}\cdot\hat{\mathbf{w}}_{\mathrm{ERM}} \tag{A.11}$$

Step 2: Concentration of variances $V^t_{\mu\to i}, \tilde{V}^t_{\mu\to i}$

Since the variances $V^t_{\mu\to i}, \tilde{V}^t_{\mu\to i}$ depend on $(x^\mu_i)^2$, in the asymptotic limit $d\to\infty$ they concentrate around their means:

$$\mathbb{E}\left[V^t_{\mu\to i}\right] = \sum_{j\neq i}\mathbb{E}\left[\left(x^\mu_i\right)^2\right]\hat{c}^t_{j\to\mu} = \frac{1}{d}\sum_{j\neq i}\hat{c}^t_{j\to\mu} = \frac{1}{d}\sum_{j=1}^{d}\hat{c}^t_{j\to\mu} - \frac{1}{d}\hat{c}^t_{i\to\mu} \underset{d\to\infty}{\to} V^t \equiv \frac{1}{d}\sum_{j=1}^{d}\hat{c}^t_j \tag{A.12}$$

where we have defined the variance overlap V^t. The same argument can be used for $\tilde{V}^t_{\mu\to in}$. Summarising, asymptotically we have:

$$V^t_{\mu\to i}\to V^t, \qquad\qquad \tilde{V}^t_{\mu\to i}\to \tilde{V}^t \tag{A.13}$$

Step 3: Distribution of $b^t_{\mu\to i}, \tilde{b}^t_{\mu\to i}$

By definition, we have

$$b^t_{\mu\to i} = \sum_{\nu\neq\mu} x^\nu_i g^t_{\nu\to i} = \sum_{\nu\neq\mu} x^\nu_i f_{\text{out}}(y^\mu, \omega^t_{\nu\to i}, V^t_{\nu\to i}) = \sum_{\nu\neq\mu} x^\nu_i f_{\text{out}}(f_0(z_\nu+\tau\xi_\nu), \omega^t_{\nu\to i}, V^t_{\nu\to i}) \tag{A.14}$$

Note that in the sum $z_\nu = \sum_{j=1}^{d} x^\nu_j w_{\star j}$ there is a term $i=j$, and therefore z_μ is correlated with x^ν_i. To make this explicit, we split the teacher local field:

$$z_\mu = \sum_{j=1}^{d} x^\mu_j w_{\star j} = \underbrace{\sum_{j\neq i} x^\mu_j w_{\star j}}_{z_{\mu\to i}} + x^\mu_i w_{\star i} \tag{A.15}$$

and note that $z_{\mu\to i} = O(1)$ is independent from x^ν_i. Since $x^\mu_i w_{\star i} = O(d^{-1/2})$, to take the average at leading order, we can expand the denoising function:

$$\begin{aligned} f_{\text{out}}(f_0(z_\mu+\tau\xi_\mu), \omega^t_{\nu\to i}, V^t_{\nu\to i}) = f_{\text{out}}(f_0(z_{\nu\to i}+\tau\xi_\nu), \omega^t_{\nu\to i}, V^t_{\nu\to i}) \\ + \partial_z f_{\text{out}}(f_0(z_{\nu\to i}+\tau\xi_\nu), \omega^t_{\nu\to i}, V^t_{\nu\to i}) x^\nu_i w_{\star i} + O(d^{-1}) \end{aligned} \tag{A.16}$$

Inserting in the expression for $b^t_{\mu\to i}$,

$$\begin{aligned} b^t_{\mu\to i} = &\sum_{\nu\neq\mu} x^\nu_i f_{\text{out}}(f_0(z_{\nu\to i}+\tau\xi_\nu), \omega^t_{\nu\to i}, V^t_{\nu\to i}) \\ &+ \sum_{\nu\neq\mu} (x^\nu_i)^2 \partial_z f_{\text{out}}(f_0(z_{\nu\to i}+\tau\xi_\nu), \omega^t_{\nu\to i}, V^t_{\nu\to i}) w_{\star i} + O(d^{-3/2}) \end{aligned} \tag{A.17}$$

Therefore:

$$\begin{aligned}\mathbb{E}\left[b^t_{\mu\to i}\right] &= \frac{w_{\star i}}{d}\sum_{\nu\neq\mu}\partial_z f_{\text{out}}(f_0(z_{\nu\to i}+\tau\xi_\nu),\omega^t_{\nu\to i},V^t_{\nu\to i})+O(d^{-3/2})\\ &= \frac{w_{\star i}}{d}\sum_{\nu=1}^{n}\partial_z f_{\text{out}}(f_0(z_{\nu\to i}+\tau\xi_\nu),\omega^t_{\nu\to i},V^t_{\nu\to i})+O(d^{-3/2})\end{aligned} \tag{A.18}$$

Note that as $d\to\infty$, for fixed t and for all ν, the fields $(z_{\nu\to i},\omega^t_{\nu\to i})$ are identically distributed according to average in eq. (A.10). Therefore,

$$\frac{1}{d}\sum_{\nu=1}^{n}\partial_z f_{\text{out}}(f_0(z_{\nu\to i}+\tau\xi_\nu),\omega^t_{\nu\to i},V^t_{\nu\to i})\underset{d\to\infty}{\to}\alpha\,\mathbb{E}_{(\omega,z),\xi}\left[\partial_z f_{\text{out}}(f_0(z+\tau\xi),\omega,V^t)\right]\equiv\hat{m}^t \tag{A.19}$$

so:

$$\mathbb{E}\left[b^t_{\mu\to i}\right]\underset{d\to\infty}{\to}w_{\star i}\hat{m}^t. \tag{A.20}$$

Similarly, the variance is given by:

$$\text{Var}\left[b^t_{\mu\to i}\right] \tag{A.21}$$

$$\begin{aligned}&=\sum_{\nu\neq\mu}\sum_{\kappa\neq\mu}\mathbb{E}\left[x^\nu_i x^\kappa_i\right]f_{\text{out}}(f_0(z_{\nu\to i}+\tau\xi_\nu),\omega^t_{\nu\to i},V^t_{\nu\to i})f_{\text{out}}(f_0(z_{\kappa\to i}+\tau\xi_\kappa),\omega^t_{\kappa\to i},V^t_{\kappa\to i})+O(d^{-2})\\ &=\frac{1}{d}\sum_{\nu\neq\mu}f_{\text{out}}(f_0(z_{\nu\to i}+\tau\xi_\nu),\omega^t_{\nu\to i},V^t_{\nu\to i})^2+O(d^{-2})\\ &=\frac{1}{d}\sum_{\nu=1}^{n}f_{\text{out}}(f_0(z_{\nu\to i}+\tau\xi_\nu),\omega^t_{\nu\to i},V^t_{\nu\to i})^2+O(d^{-2})\\ &\underset{d\to\infty}{\to}\alpha\,\mathbb{E}_{(z,\omega),\xi}\left[f_{\text{out}}(f_0(z+\tau\xi),\omega,V^t)^2\right]\equiv\hat{q}^t\end{aligned} \tag{A.22}$$

The same discussion holds for the ERM. We now just need to compute the correlation between both fields:

$$\text{Cov}\left[b^t_{\mu\to i},\tilde{b}^t_{\mu\to i}\right] \tag{A.23}$$

$$\begin{aligned}&=\sum_{\nu\neq\mu}\sum_{\kappa\neq\mu}\mathbb{E}\left[x^\nu_i x^\kappa_i\right]f_{\text{out}}(f_0(z_{\nu\to i}+\tau\xi_\nu),\omega^t_{\nu\to i},V^t_{\nu\to i})\tilde{f}_{\text{out}}(f_0(z_{\kappa\to i}+\tau\xi_\kappa),\tilde{\omega}^t_{\kappa\to i},\tilde{V}^t_{\kappa\to i})+O(d^{-2})\\ &=\frac{1}{d}\sum_{\nu=1}^{n}f_{\text{out}}(f_0(z_{\nu\to i}+\tau\xi_\nu),\omega^t_{\nu\to i},V^t_{\nu\to i})\tilde{f}_{\text{out}}(f_0(z_{\nu\to i}+\tau\xi_\nu),\tilde{\omega}^t_{\nu\to i},\tilde{V}^t_{\nu\to i})+O(d^{-2})\\ &\underset{d\to\infty}{\to}\alpha\,\mathbb{E}_{(z,\omega,\tilde{\omega}),\xi}\left[f_{\text{out}}(f_0(z+\tau\xi),\omega,V^t)\tilde{f}_{\text{out}}(f_0(z+\tau\xi),\tilde{\omega},\tilde{V}^t)\right]\equiv\hat{Q}^t\end{aligned} \tag{A.24}$$

To summarise, we have:

$$(b^t_{\mu\to i},\tilde{b}^t_{\mu\to i})\sim\mathcal{N}\left(w_{\star i}\begin{bmatrix}\hat{m}^t\\ \tilde{\hat{m}}^t\end{bmatrix},\begin{bmatrix}\hat{q}^t & \hat{Q}^t\\ \hat{Q}^t & \tilde{\hat{q}}^t\end{bmatrix}\right) \tag{A.25}$$

Step 4: Concentration of $A^t_{\mu\to i}, \tilde{A}^t_{\mu\to i}$

The only missing piece is to determine the distribution of the prior variances $A^t_{\mu\to i}, \tilde{A}^t_{\mu\to i}$. Similar to the previous variance, they concentrate:

$$\begin{aligned} A^t_{\mu\to i} &= -\sum_{\nu\neq\mu}(x^\nu_i)^2\partial_\omega f_{\text{out}}(y^\nu, \omega^t_{\nu\to i}, V^t_{\nu\to i}) \qquad \text{(A.26)}\\ &= -\sum_{\nu\neq\mu}(x^\nu_i)^2\partial_\omega f_{\text{out}}(f_0(z_{\nu\to i}+\tau\xi_\nu), \omega^t_{\nu\to i}, V^t_{\nu\to i}) + O(d^{-3/2})\\ &= -\frac{1}{d}\sum_{\nu=1}\partial_\omega f_{\text{out}}(f_0(z_{\nu\to i}+\tau\xi_\nu), \omega^t_{\nu\to i}, V^t_{\nu\to i}) + O(d^{-3/2})\\ &\underset{d\to\infty}{\to} -\alpha\,\mathbb{E}_{(z,\omega),\xi}\left[\partial_\omega f_{\text{out}}(f_0(z+\tau\xi),\omega,V^t)\right] \equiv \hat{V}^t \qquad \text{(A.27)}\end{aligned}$$

Summary

We now have all the ingredients we need to characterise the asymptotic distribution of the estimators:

$$\hat{\mathbf{w}}_{\text{BO}} \sim f_{\text{out}}(\theta_\star \hat{m}^t + \sqrt{\hat{q}^t}\xi, \hat{V}^t) \qquad \text{(A.28)}$$

$$\hat{\mathbf{w}}_{\text{ERM}} \sim \tilde{f}_{\text{out}}(\theta_\star \tilde{\hat{m}}^t + \sqrt{\tilde{\hat{q}}^t}\eta, \tilde{\hat{V}}^t) \qquad \text{(A.29)}$$

where $\eta, \xi \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ are independent Gaussian variables. From that, we can recover the usual GAMP state evolution equations for the overlaps:

$$\begin{cases} V^{t+1} = \mathbb{E}_{(w_\star,\xi)}\left[\partial_b f_w(\hat{m}^t w_\star + \sqrt{\hat{q}^t}\xi, \hat{V}^t)\right] \\ q^{t+1} = \mathbb{E}_{(w_\star,\xi)}\left[f_w(\hat{m}^t w_\star + \sqrt{\hat{q}^t}\xi, \hat{V}^t)^2\right] \\ m^{t+1} = \mathbb{E}_{(w_\star,\xi)}\left[f_w(\hat{m}^t w_\star + \sqrt{\hat{q}^t}\xi, \hat{V}^t) w_{\star i}\right] \end{cases}, \quad \begin{cases} \hat{V}^t = -\alpha\mathbb{E}_{(z,\omega),\xi}\left[\partial_\omega f_{\text{out}}(f_0(z+\tau\xi),\omega,V^t)\right] \\ \hat{q}^t = \alpha\mathbb{E}_{(z,\omega),\xi}\left[f_{\text{out}}(f_0(z+\tau\xi),\omega,V^t)^2\right] \\ \hat{m}^t = \alpha\mathbb{E}_{(z,\omega),\xi}\left[f_{\text{out}}(f_0(z+\tau\xi),\omega,V^t)\right] \end{cases} \qquad \text{(A.30)}$$

which is also valid for the tilde variables. But we can also get a set of equations for the correlations:

$$\begin{cases} Q^t = \mathbb{E}_{w_\star,(b,\tilde{b})}\left[f_w(b,\hat{V}^t)\tilde{f}_w\left(\tilde{b},\tilde{\hat{V}}^t\right)\right] \\ \hat{Q}^t = \alpha\mathbb{E}_{(z,\omega,\tilde{\omega}),\xi}\left[f_{\text{out}}(f_0(z+\tau\xi),\omega,V^t)\tilde{f}_{\text{out}}(f_0(z+\tau\xi),\tilde{\omega},\tilde{V}^t)\right] \end{cases} \qquad \text{(A.31)}$$

A.1.2 Simplifications

Simplifying BO state evolution

State evolution of BO can be reduced to two equations. First, note that asymptotically

$$m := \frac{1}{d}\hat{\mathbf{w}}_{\text{bo}}\cdot\theta_\star = \frac{1}{d}\mathbb{E}_{\theta_\star,\mathscr{D}}\left[\hat{\mathbf{w}}_{\text{bo}}\cdot\theta_\star\right]$$

with high probability. By Nishimori identity, the vector $\theta_\star$ in the expectation can be replaced by an independent copy of the Bayesian posterior. This yields:

$$\frac{1}{d}\mathbb{E}_{\theta_\star,\mathscr{D}}\left[\hat{\mathbf{w}}_{\text{bo}}\cdot\theta_\star\right]=\frac{1}{d}\mathbb{E}_{\mathscr{D}}\left[\hat{\mathbf{w}}_{\text{bo}}\right]=q$$

Hence $m=q$. Similarly, noting $\langle\cdot\rangle$ the average over the posterior distribution:

$$V=\frac{1}{d}\langle\|\mathbf{w}-\hat{\mathbf{w}}_{\text{bo}}\|^2\rangle=\frac{1}{d}\mathbb{E}_{\mathscr{D}}\left[\langle\|\mathbf{w}-\hat{\mathbf{w}}_{\text{bo}}\|^2\rangle\right]=\frac{1}{d}\mathbb{E}_{\mathscr{D}}\left[\langle\|\mathbf{w}\|^2\rangle\right]-\frac{1}{d}\mathbb{E}_{\mathscr{D}}\left[\hat{\mathbf{w}}_{\text{bo}}\cdot\hat{\mathbf{w}}_{\text{bo}}\right]$$

Like before, we used the fact that in asymptotically, $\langle\|\mathbf{w}-\hat{\mathbf{w}}_{\text{bo}}\|^2\rangle$ concentrates around its mean. Using Nishimori, the first term is equal to $\mathbb{E}_{\theta_\star}\left[\|\theta_\star\|^2\right]=1$. By definition, the second term is equal to q, thus $V=1-q$.

Using similar arguments, $\hat{m}=\hat{q}=\hat{V}$. Thus, the state evolution can be reduced to two equations on q and $\hat{q}$.

Simplifying the $Q,\hat{Q}$ equations

In fact, the Nishimori property also allow us to show that the cross-correlation $Q,\hat{Q}$ are the same as the overlaps $\tilde{m},\hat{\tilde{m}}$, in a similar way to A.1.2. Indeed,

$$Q=\frac{1}{d}\hat{\mathbf{w}}_{\text{bo}}\cdot\hat{\mathbf{w}}_{\text{erm}}=\frac{1}{d}\mathbb{E}_{\mathscr{D}}\left[\hat{\mathbf{w}}_{\text{bo}}\cdot\hat{\mathbf{w}}_{\text{erm}}\right]=\frac{1}{d}\mathbb{E}_{\theta_\star,\mathscr{D}}\left[\theta_\star\cdot\hat{\mathbf{w}}_{\text{erm}}\right]=\tilde{m} \tag{A.32}$$

Alternatively, we can also prove that directly showing that the iterations for Q^t are a stable orbit of $\tilde{m}^t$. Indeed, assume that at time step t we have $Q^t=\tilde{m}^t$ and $\hat{Q}^t=\hat{\tilde{m}}^t$. Then, focusing at our specific setting, at time $t+1$ we have:

$$\begin{aligned}Q^{t+1}&=\mathbb{E}_{w_\star,b,\tilde{b}}[f_w(b,\hat{V})f_w(b,\hat{V})]=\mathbb{E}_{w_\star}\left[\frac{b}{\hat{V}+1}\frac{\tilde{b}}{\hat{\tilde{V}}+\lambda}\right]=\mathbb{E}_{w_\star}\left[\frac{\hat{Q}+\hat{\tilde{m}}\hat{m}}{(\hat{V}+1)(\hat{\tilde{V}}+\lambda)}\right]\\&=\mathbb{E}_{w_\star}\left[\frac{\hat{\tilde{m}}}{\hat{\tilde{V}}+\lambda}\right].\end{aligned}$$

Because as we have shown above $\hat{m}=\hat{q}$ and $\hat{Q}^t=\hat{\tilde{m}}^t$. This is precisely the equation for $\tilde{m}$.

A.1.3 Evaluating the equations

Bayes-optimal

In Bayes-optimal estimation, the estimation likelihood P_{out} and prior P_w match exactly that of the generating model for data, which for the model (2.2) is:

$$P_{\text{out}}(y|x) = \frac{1}{2}\text{erfc}\left(-\frac{y\omega}{\sqrt{2\Delta}}\right), \qquad P_w(w) = \mathcal{N}(0,1). \tag{A.33}$$

Therefore, it is easy to show that:

$$\mathcal{Z}_{\text{out}}(y,\omega,V) = \frac{1}{2}\text{erfc}\left(-\frac{y\omega}{\sqrt{2(\tau^2+V)}}\right), \qquad Z_w(b,A) = \frac{e^{\frac{b^2}{1+A}}}{1+A} \tag{A.34}$$

and therefore:

$$f_{\text{out}}(y,\omega,V) = \frac{2y\,\mathcal{N}(\omega y|0,V+\tau^2)}{\text{erfc}\left(-\frac{y\omega}{\sqrt{2(\tau^2+V)}}\right)}, \qquad f_w(b,A) = \frac{b}{1+A} \tag{A.35}$$

This form of the prior allow us to simplify some of the equations considerably:

$$q_{\text{bo}}^{t+1} = \mathbb{E}_{(w_\star,\xi)}\left[f_w(\hat{q}^t w_\star + \sqrt{\hat{q}^t}\xi, \hat{q}^t)^2\right] = \frac{1}{1+\hat{q}_{\text{bo}}^t} \tag{A.36}$$

which is the equation found in Theorem 2.3.2. The other equation cannot be closed analytically, however it can be considerably simplified:

$$\hat{q}_{\text{bo}} = -\alpha\mathbb{E}_{(z,\omega),\xi}\left[\partial_\omega f_{\text{out}}(f_0(z+\tau\xi),\omega,V^t)\right] \tag{A.37}$$

$$= \frac{2}{\pi}\frac{\alpha}{1+\tau^2-q_{\text{bo}}^t}\int_{\mathbb{R}} \mathrm{d}z\,\mathcal{N}\left(z\middle|0,\frac{q_{\text{bo}}^t}{2(1+\tau^2-q_{\text{bo}}^t)}\right)\frac{e^{-2z^2}}{\text{erfc}(z)\text{erfc}(-z)} \tag{A.38}$$

A.1.4 ERM estimation

For ERM, the estimation likelihood P_{out} and prior P_w are related to the loss and penalty functions:

$$P_{\text{out}}(y|x) = e^{-\beta\ell(y,x)}, \qquad P_w(w) = e^{-\beta r(w)}. \tag{A.39}$$

where the parameter $\beta > 0$ is introduced for convenience, and should be taken to infinity. Focusing on the regularisation part and redefining $(b,A) \to (\beta b, \beta A)$

$$\mathcal{Z}_w(b,A) = \int_{\mathbb{R}} \mathrm{d}w\; e^{-\beta(\frac{A}{2}w^2 - bw + r(w))} \underset{\beta\to\infty}{\asymp} e^{\beta\left[\frac{b^2}{2A} - \mathcal{M}_{A^{-1}r}(A^{-1}b)\right]} \tag{A.40}$$

where we have used Laplace's method and defined the *Moreau envelope*:

$$\mathscr{M}_{\tau f}(x) = \min_{z\in\mathbb{R}} \left[\frac{1}{2\tau}(x-z)^2 + f(z) \right] \tag{A.41}$$

$$\tag{A.42}$$

Therefore,

$$f_w(b, A) = \lim_{\beta\to\infty} \frac{1}{\beta} \partial_b \log Z_w(b, A) = \mathrm{prox}_{A^{-1}r}(A^{-1}b) \tag{A.43}$$

where we have defined the *proximal operator*:

$$\mathrm{prox}_{\tau f}(x) = \operatorname*{argmin}_{z\in\mathbb{R}} \left[\frac{1}{2\tau}(x-z)^2 + f(z) \right] \tag{A.44}$$

and used the well-known property $\partial_x \mathscr{M}_{\tau f}(x) = -\frac{1}{\tau}\Big(\mathrm{prox}_{\tau f}(x) - x\Big)$. In particular, for the ℓ_2-penalty $r(w) = \lambda/2 w^2$, we have:

$$\mathrm{prox}_{\lambda/2(\cdot)^2}(x) = \frac{x}{1+\lambda} \qquad \Leftrightarrow \qquad f_w(b, A) = \frac{b}{\lambda + A} \tag{A.45}$$

The simple form of the regularization allow us to simplify the state evolution equations considerably:

$$\begin{cases} \tilde{V}^{t+1} &= \mathbb{E}_{(w_\star,\xi)} \left[\partial_b f_w(\hat{\hat{m}}^t w_\star + \sqrt{\hat{\hat{q}}^t}\xi, \hat{\hat{V}}^t) \right] = \frac{1}{\lambda+\hat{\hat{V}}} \\ \tilde{q}^{t+1} &= \mathbb{E}_{(w_\star,\xi)} \left[f_w(\hat{\hat{m}}^t w_\star + \sqrt{\hat{\hat{q}}^t}\xi, \hat{\hat{V}}^t)^2 \right] = \frac{\hat{m}^2+\hat{\hat{q}}}{(\lambda+\hat{\hat{V}})^2} \\ \tilde{m}^{t+1} &= \mathbb{E}_{(w_\star,\xi)} \left[f_w(\hat{\hat{m}}^t w_\star + \sqrt{\hat{\hat{q}}^t}\xi, \hat{\hat{V}}^t) w_{\star i} \right] = \frac{\hat{m}}{\lambda+\hat{\hat{V}}} \end{cases} \tag{A.46}$$

which are the equations found in Theorem 2.3.2. A similar discussion can be carried for the loss term, and yields in general:

$$f_{\mathrm{out}}(y, \omega, V) = V^{-1}\Big(\mathrm{prox}_{\tau\ell(y,\cdot)}(x) - x\Big) \tag{A.47}$$

Unfortunately, the logistic loss $\ell(y, x) = \log(1 + e^{-yx})$ does not admit a closed-form solution for the proximal, and therefore for a given (y, ω, V) we need to compute it numerically.

A.2 Proof of theorems

A possible route for proving our result is to give a rigorous proof of the cavity equations. Instead, we shall use a shortcut, and leverage on recent progress for both the ERM cavity results [228, 219, 220, 178, 13, 153]), the Bayes performances [20, 22], as well as on the performance of GAMP [202, 115, 97].

A.2.1 GAMP optimality

The optimally of GAMP is a direct consequence of the generic results concerning its performance (the state evolution in [202, 115]) and the characterization of the Bayes performance in [22]. G-a works, one considers a sequence of inference problems indexed by the dimension d, with data $\mathscr{D}_d$ (which are defined in section 2.2 for our purpose). As d increases, both GAMP performances and Bayes errors converge with high probability to the same deterministic limit given by the so-called "replica", or "state evolution" equations.

To simplify the notation, all our statements involving the asymptotic limit $d \to \infty$ are implicitly defined for such sequences, and the convergence is assumed to be in terms of probability.

Let us prove that, indeed, GAMP estimates for posterior probability are asymptotically exact with high probability. First, we note that the estimation of the Bayes posterior probability for the signs corresponds to finding the estimators that minimize the MMSE. Indeed consider, for fixed data (this remains true averaging over data), the mean squared error for an estimator $\hat{Y}(\mathbf{X})$:

$$\mathrm{MSE}(\hat{Y}(\mathbf{X})) = \mathbb{E}_{Y,\mathbf{X}}\left[(Y - \hat{Y}(\mathbf{X}))^2\right] = \mathbb{E}_{\mathbf{X}}\mathbb{E}_{Y|\mathbf{X}}\left[(Y - \hat{Y}(\mathbf{X}))^2\right] \tag{A.48}$$

The mean square error is given by using the posterior mean [55], as can be seen immediately differentiating with respect to $\hat{Y}$ (for a given $\mathbf{x}$), so that:

$$\hat{Y}_{\mathrm{Bayes}}(\mathbf{x}) = \mathbb{E}_{Y|X=\mathbf{x}}[Y] = 2\mathbb{P}_{Y|X=\mathbf{x}}(Y = 1) - 1 \tag{A.49}$$

The Bayes estimator for the posterior probability is thus the MMSE estimator. We see here that the estimation of the posterior mean of Y is equivalent to the estimation of the probability it takes value one; both quantities are thus trivially related.

We can now use Proposition 2, page 12 in [22], which shows that indeed GAMP efficiently achieves Bayes-optimality for the MMSE on Y:

Theorem A.2.1 (GAMP generalisation error, [22])**.** *Consider a sequence of problems indexed by d, with data $\mathscr{D}_d$ in dimension d, then we have that GAMP estimator asymptotically achieves the Minimal Mean Square Error in estimating the error on new label Y. That is, with high probability:*

$$\lim_{d\to\infty} \mathbb{E}_{Y,\mathbf{X}|\mathscr{D}_d}\left[(Y - \hat{Y}_{\mathrm{GAMP}}(\mathbf{X}, \mathscr{D}_d)^2\right] = \mathrm{MMSE}(Y) \tag{A.50}$$

where $\hat{Y}_{\mathrm{GAMP}}(\mathbf{x}, \mathscr{D}) = 2p - 1$, and $p = \hat{f}^{AMP}(\mathbf{x})$ (eq. 2.10), with $\hat{\mathbf{c}}_{\mathrm{amp}}^{\top}(\mathbf{x} \odot \mathbf{x}) = 1 - q$, with q a fixed point of (2.16)*.*

The fact that GAMP asymptotically achieves the MMSE, coupled with the uniqueness of the Bayes estimator, implies the GAMP estimator for p is arbitrarily close to the Bayes estimated for p, with high probability over new Gaussian samples, as $d \to \infty$. More precisely, we can use the

following lemma:

Lemma A.2.2 (Bounds on differences of estimators for Y)**.** *Consider a sequence of estimation problems indexed by d with data $\mathscr{D}_d$. If a (sequence of) estimators $\hat{f}_d(\mathbf{x})$ achieves the MMSE performance of $\hat{g}_d^{\text{Bayes}}(\mathbf{x})$ as $d \to \infty$ for Gaussian distributed $\mathbf{x}$, then*

$$\lim_{d\to\infty} \mathbb{E}_{\mathbf{X}} |f_d(\mathbf{X}) - g_d^{\text{Bayes}}(\mathbf{X})|^2 \to 0 \tag{A.51}$$

Proof. The Bayes estimator $g_d^{\text{Bayes}}()$ is the minimum of the MMSE, therefore for any other estimator $f_d(\mathbf{X})$ we have

$$\mathbb{E}\left[(Y - f_d(\mathbf{X})^2\right] \geq \mathbb{E}\left[(Y - g_d^{\text{Bayes}}(\mathbf{X})^2\right]. \tag{A.52}$$

We have, denoting $\delta_d(X) = f_d(\mathbf{X}) - g_d^{\text{Bayes}}(\mathbf{X})$

$$\begin{aligned}
\mathbb{E}\left[(Y - f_d(\mathbf{X})^2\right] &= \mathbb{E}\left[(Y - g_d^{\text{Bayes}}(\mathbf{X}) + \delta_d(X))^2\right] && \text{(A.53)}\\
&= \text{MMSE} + \mathbb{E}\left[\delta_d(X)^2 + 2\delta_n(X)(Y - g_d^{\text{Bayes}}(X))\right] && \text{(A.54)}\\
&= \text{MMSE} + \mathbb{E}\left[\delta_d(X)^2\right] + \mathbb{E}_{X,\mathscr{D}}\mathbb{E}_{Y|X,\mathscr{D}}\left[2\delta_d(X)(Y - g_d^{\text{Bayes}}(X))\right] && \text{(A.55)}\\
&= \text{MMSE} + \mathbb{E}\left[\delta_d(X)^2\right] + \mathbb{E}_{X,\mathscr{D}}\left[2\delta_n(X)\mathbb{E}_{Y|X,\mathscr{D}}[Y - g_d^{\text{Bayes}}(X])\right] && \text{(A.56)}\\
&= \text{MMSE} + \mathbb{E}\left[\delta_d(X)^2\right] && \text{(A.57)}
\end{aligned}$$

where we have used $g_d^{\text{Bayes}}(X) = \mathbb{E}_{Y|X,\mathscr{D}}[Y]$. Using the fact that the f_d asymptotically achieves MMSE optimality, we thus obtain:

$$\lim_{d\to\infty} \mathbb{E}_{Y,X,\mathscr{D}}\left[|f_d(X) - g_d^{\text{Bayes}}(X)|^2\right] \to 0 \tag{A.58}$$

□

Applying this lemma to the GAMP estimator leads to Lemma 2.3.1: with high probability over new sample $\mathbf{x}$ and learning data $\mathscr{D}$, the GAMP estimate is asymptotically equivalent to the Bayes one.

A.2.2 Joint density of estimators

While a possible strategy to prove the second theorem would be to use state evolution to follow our joint GAMP algorithm (thus monitoring the Bayes *and* the ERM performance), we shall instead again leverage on recent progress on generic proofs of replica equations, in particular the Bayes one (in [22] and the ERM ones (that were the subject of many works recently [228, 219, 220, 178, 13, 153]). Again, all our statements involving the asymptotic limit $d \to \infty$ are

implicitly defined for sequences of problems, and the convergence is assumed to be in terms of probability. We start with the following lemma:

Lemma A.2.3 (Joint distribution of pre-activation)**.** *For a fixed set of data $\mathscr{D}$, consider the joint random variables (over $\mathbf{X}$) $\nu = \mathbf{X}\cdot\theta_\star, \lambda_{\rm erm} = \mathbf{X}\cdot\hat{\mathbf{w}}_{\rm erm}, \lambda_{\rm amp} = \mathbf{X}\cdot\hat{\mathbf{w}}_{\rm amp}$. Then we have*

$$\mathbb{P}(\nu, \lambda_{\rm amp}, \lambda_{\rm erm}) = \mathcal{N}\left(0, \begin{pmatrix} \frac{\theta_\star\cdot\theta_\star}{d} & \frac{\theta_\star\cdot\hat{\mathbf{w}}_{\rm amp}}{d} & \frac{\theta_\star\cdot\hat{\mathbf{w}}_{\rm erm}}{d} \\ \frac{\hat{\mathbf{w}}_{\rm amp}\cdot\theta_\star}{d} & \frac{\hat{\mathbf{w}}_{\rm amp}\cdot\hat{\mathbf{w}}_{\rm amp}}{d} & \frac{\hat{\mathbf{w}}_{\rm amp}\cdot\hat{\mathbf{w}}_{\rm erm}}{d} \\ \frac{\hat{\mathbf{w}}_{\rm erm}\cdot\theta_\star}{d} & \frac{\hat{\mathbf{w}}_{\rm erm}\cdot\hat{\mathbf{w}}_{\rm amp}}{d} & \frac{\hat{\mathbf{w}}_{\rm erm}\cdot\hat{\mathbf{w}}_{\rm erm}}{d} \end{pmatrix}\right) \tag{A.59}$$

Proof. This is an immediate consequence of the Gaussianity of the new data $\mathbf{x}$, with covariance $\mathbb{I}/d$. □

We now would like to know the asymptotic limit of the parameters of this distribution, for large d. While we have $\frac{w_\star\cdot w_\star}{d} \to \rho$, the other overlap has a deterministic limit given by the replica equations. For empirical risk minimisation, this has been proven in the aforementioned series of works, but we shall here use the notation of [153] and utilize use the following results:

Theorem A.2.4 (ERM overlaps [228, 13, 153])**.** *Consider a sequence of inference problems indexed by the dimension d, then with high probability:*

$$\lim_{d\to\infty} \frac{\hat{\mathbf{w}}_{\rm erm}\cdot\theta_\star}{d} \to m, \qquad \lim_{d\to\infty} \frac{\hat{\mathbf{w}}_{\rm erm}\cdot\hat{\mathbf{w}}_{\rm erm}}{d} \to q_{\rm erm} \tag{A.60}$$

With $q_{\rm erm}$ and m solutions of the self-consistent equations (2.18) *in the main text.*

GAMP is tracked by its state evolution [115], and is known to achieve the Bayes overlap:

Theorem A.2.5 (Bayes overlaps [22])**.** *Consider a sequence of inference problems indexed by the dimension d, then with high probability:*

$$\lim_{d\to\infty} \frac{\hat{\mathbf{w}}_{\rm amp}\cdot\theta_\star}{d} \to q_{\rm bo}, \qquad \lim_{d\to\infty} \frac{\hat{\mathbf{w}}_{\rm amp}\cdot\hat{\mathbf{w}_{\rm amp}}}{d} \to q_{\rm bo} \tag{A.61}$$

With $q_{\rm bo}$ given by the self-consistent Equation (2.16)*.*

The only overlap left to control is thus $Q = \hat{\mathbf{w}}_{\rm amp}\cdot\hat{\mathbf{w}}_{\rm erm}/d$. We shall now prove that it is also concentrating, with high probability, to m. To do this, we first prove the following lemma for the overlap between the Bayes estimate $\mathbf{w}_{\rm bo} = \mathbb{E}_{W|\mathscr{D}}[\mathbf{W}]$ and any other vector $\mathbf{V}$, possibly dependent on the data:

Lemma A.2.6 (Nishimori relation for Bayes overlaps)**.**

$$\mathbb{E}_{\mathscr{D}}\left[\mathbf{w}_{\rm bo}\cdot\mathbf{V}(\mathscr{D})\right] = \mathbb{E}_{\mathscr{D},W^*}\left[\mathbf{w}^*\cdot\mathbf{V}(\mathscr{D})\right] \tag{A.62}$$

Proof. The proof is an application of Bayes' formula, and an example of what is often called a Nishimori equality in statistical physics:

$$\begin{aligned}\mathbb{E}_{\mathscr{D},W^*}\left[\mathbf{w}^*\cdot\mathbf{V}(\mathscr{D})\right] &= \mathbb{E}_{\mathscr{D}}E_{W^*|\mathscr{D}}\left[\mathbf{w}^*\cdot\mathbf{V}(\mathscr{D})\right] &\text{(A.63)}\\ &= \mathbb{E}_{\mathscr{D}}\left[(E_{W^*|\mathscr{D}}\mathbf{w}^*)\cdot\mathbf{V}(\mathscr{D})\right] = \mathbb{E}_{\mathscr{D}}\left[\mathbf{w}_{\text{bo}}\cdot\mathbf{V}(\mathscr{D})\right] &\text{(A.64)}\end{aligned}$$

□

From this lemma, we see immediately that, in expectation

$$\lim_{d\to\infty}\mathbb{E}\left[\frac{\mathbf{w}_{\text{erm}}\cdot\theta_\star}{d}\right] = \lim_{d\to\infty}\mathbb{E}\left[\frac{\mathbf{w}_{\text{erm}}\cdot\mathbf{w}_{\text{bo}}}{d}\right] = m \tag{A.65}$$

Additionally, we already know that the left-hand side concentrates. It is easy to see that the right-hand side does as well:

Lemma A.2.7 (Concentration of the overlap Q)**.**

$$\lim_{d\to\infty}\mathbb{E}\left[\left(\frac{\mathbf{w}_{\text{bo}}\cdot\mathbf{w}_{\text{erm}}}{d}\right)^2\right] = \lim_{d\to\infty}\mathbb{E}\left[\frac{\mathbf{w}_{\text{bo}}\cdot\mathbf{w}_{\text{erm}}}{d}\right]^2 \tag{A.66}$$

Proof. The proof again uses the Nishimori identity.

$$\begin{aligned}\mathbb{E}\left[\left(\frac{\mathbf{w}_{\text{bo}}\cdot\mathbf{w}_{\text{erm}}}{d}\right)^2\right] &= \mathbb{E}\left[\left(\frac{\mathbf{w}_{\text{bo}}\cdot\mathbf{w}_{\text{erm}}}{d}\right)\left(\frac{\mathbf{w}_{\text{bo}}\cdot\mathbf{w}_{\text{erm}}}{d}\right)\right] &\text{(A.67)}\\ &= \mathbb{E}_{\mathscr{D}}\left[\left(\frac{\mathbb{E}_{W|\mathscr{D}}W\cdot\mathbf{w}_{\text{erm}}}{d}\right)\left(\frac{\mathbb{E}_{W|\mathscr{D}}W\cdot\mathbf{w}_{\text{erm}}}{d}\right)\right] &\text{(A.68)}\\ &= \mathbb{E}_{\mathscr{D}}\mathbb{E}_{W_1,W_2|\mathscr{D}}\left[\left(\frac{W_1\cdot\mathbf{w}_{\text{erm}}}{d}\right)\left(\frac{W_2\cdot\mathbf{w}_{\text{erm}}}{d}\right)\right] &\text{(A.69)}\\ &= \mathbb{E}_{\mathscr{D},\mathbf{w}^*}\left[\left(\frac{\mathbf{w}^*\cdot\mathbf{w}_{\text{erm}}}{d}\right)\left(\frac{\mathbb{E}_{W|\mathscr{D}}W\cdot\mathbf{w}_{\text{erm}}}{d}\right)\right] &\text{(A.70)}\\ &= \mathbb{E}_{\mathscr{D},\mathbf{w}^*}\left[\left(\frac{\mathbf{w}^*\cdot\mathbf{w}_{\text{erm}}}{d}\right)\left(\frac{\mathbf{w}_{\text{bo}}\cdot\mathbf{w}_{\text{erm}}}{d}\right)\right] &\text{(A.71)}\end{aligned}$$

Then, from Cauchy-Schwartz we have

$$\begin{aligned}\mathbb{E}\left[\left(\frac{\mathbf{w}_{\text{bo}}\cdot\mathbf{w}_{\text{erm}}}{d}\right)^2\right]^2 &\leq \mathbb{E}\left[\left(\frac{\mathbf{w}_{\text{bo}}\cdot\mathbf{w}_{\text{erm}}}{d}\right)^2\right]\mathbb{E}\left[\left(\frac{\mathbf{w}^*\cdot\mathbf{w}_{\text{erm}}}{d}\right)^2\right] &\text{(A.72)}\\ \mathbb{E}\left[\left(\frac{\mathbf{w}_{\text{bo}}\cdot\mathbf{w}_{\text{erm}}}{d}\right)^2\right] &\leq \mathbb{E}\left[\left(\frac{\mathbf{w}^*\cdot\mathbf{w}_{\text{erm}}}{d}\right)^2\right] &\text{(A.73)}\end{aligned}$$

and as $d \to \infty$, we can use the concentration of the right-hand side to m to obtain

$$\lim_{d\to\infty} \mathbb{E}\left[\left(\frac{\mathbf{w}_{\text{bo}} \cdot \mathbf{w}_{\text{erm}}}{d}\right)^2\right] \quad \leq \quad m^2 \tag{A.74}$$

so that, given the second moment has to be larger or equal to its (squared) mean:

$$\lim_{d\to\infty} \mathbb{E}\left[\left(\frac{\mathbf{w}_{\text{bo}} \cdot \mathbf{w}_{\text{erm}}}{d}\right)^2\right] \quad = \quad m^2 \tag{A.75}$$

□

We have thus proven that the overlap Q concentrates in quadratic mean to m as $d \to \infty$: with high probability, it is thus asymptotically equal to m. We shall now prove that $\mathbf{w}_{\text{bo}}$ can be approximated by $\mathbf{w}_{\text{amp}}$. In fact, given the concentration of overlap, it will be enough to prove that:

$$\lim_{d\to\infty} \mathbb{E}_{\mathscr{D}_d} \frac{\hat{\mathbf{w}}_{\text{amp}}(\mathscr{D}_d) \cdot \mathbf{w}_{\text{erm}}(\mathscr{D})}{d} = \lim_{d\to\infty} \mathbb{E}_{\mathscr{D}_d} \frac{\hat{\mathbf{w}}_{\text{bo}}(\mathscr{D}_d) \cdot \mathbf{w}_{\text{erm}}(\mathscr{D})}{d} \tag{A.76}$$

This can be done in two steps. First, similarly as in section A.2.1, we use the fact that GAMP achieves Bayes optimality for the estimation of W^* [22]. This leads to the following lemma

Lemma A.2.8 (Bounds on differences of estimators for $\mathbf{w}$)**.**

$$\lim_{d\to\infty} \mathbb{E}_{\mathscr{D}} \frac{\|\mathbf{w}_{\text{amp}} - \mathbf{w}_{\text{bo}}\|^2}{d} \to 0 \tag{A.77}$$

Proof. The proof proceeds similarly as in lemma A.2.2. Denoting $\delta\mathbf{w}(\mathscr{D}) = \mathbf{w}_{\text{amp}}(\mathscr{D}) - \mathbf{w}_{\text{bo}}(\mathscr{D})$ we write

$$\mathbb{E}_{\mathscr{D},\mathbf{W}^*} \frac{\|\mathbf{w}_{\text{amp}}(\mathscr{D}) - \mathbf{w}^*\|_2^2}{d} = \mathbb{E}_{\mathscr{D},\mathbf{W}^*} \frac{\|\mathbf{w}_{\text{bo}}(\mathscr{D}) + \delta\mathbf{w}(\mathscr{D}) - \mathbf{w}^*\|_2^2}{d} \tag{A.78}$$

$$= \mathbb{E}_{\mathscr{D},\mathbf{W}^*} \frac{\|\mathbf{w}_{\text{bo}}(\mathscr{D}) - \mathbf{w}^*\|_2^2}{d} + \mathbb{E}_{\mathscr{D}} \frac{\|\delta\mathbf{w}(\mathscr{D})\|_2^2}{d} + \frac{1}{d} 2\mathbb{E}_{\mathscr{D}} \mathbb{E}_{\mathbf{w}^*|\mathscr{D}} \left[\delta w(\mathscr{D}) \left(\mathbf{w}^* - \mathbf{w}_{\text{bo}}\right)\right] \tag{A.79}$$

$$= \mathbb{E}_{\mathscr{D}} \frac{\|\delta\mathbf{w}(\mathscr{D})\|_2^2}{d} \tag{A.80}$$

Using the optimality of GAMP for the MMSE yields the lemma. □

We can now prove the equality of overlaps

Lemma A.2.9.

$$\lim_{d\to\infty} \mathbb{E}_{\mathscr{D}_d} \frac{\hat{\mathbf{w}}_{\text{amp}}(\mathscr{D}_d) \cdot \mathbf{V}(\mathscr{D})}{d} = \lim_{d\to\infty} \mathbb{E}_{\mathscr{D}_d} \frac{\hat{\mathbf{w}}_{\text{bo}}(\mathscr{D}_d) \cdot \mathbf{V}(\mathscr{D})}{d} \tag{A.81}$$

Proof. The proof is an application of Cauchy-Schwartz inequality:

$$\left|\mathbb{E}_{\mathscr{D}_d}\left[\frac{(\hat{\mathbf{w}}_{\text{amp}}-\hat{\mathbf{w}}_{\text{bo}})(\mathscr{D}_d)\cdot\mathbf{V}(\mathscr{D})}{d}\right]\right| \leq \sqrt{\mathbb{E}\frac{\|\mathbf{V}\|_2^2}{d}\mathbb{E}\frac{\|\mathbf{w}_{\text{bo}}-\mathbf{w}_{\text{amp}}\|_2^2}{d}} \tag{A.82}$$

taking the limit $d\to\infty$ yields the lemma. □

Applying the lemma to the ERM estimator, and using the concentration of overlaps, finally leads to

Lemma A.2.10 (Asymptotic Joint distribution of pre-activation)**.** *Asymptotically, and with high probability over data, the joint distribution of pre-activation is asymptotically given by*

$$\lim_{d\to\infty}\mathbb{P}(\nu,\lambda_{\text{amp}},\lambda_{\text{erm}})=\mathcal{N}\left(0,\begin{pmatrix}\rho & q_{\text{bo}} & m\\ q_{\text{bo}} & q_{\text{bo}} & m\\ m & m & q_{\text{erm}}\end{pmatrix}\right) \tag{A.83}$$

To obtain Theorem 2.3.2, one simply applies the change of variable

$$(\nu,\lambda_{\text{amp}},\lambda_{\text{erm}})\to(f_\star(\nu),\hat{f}_{\text{amp}}(\lambda_{\text{amp}}),\hat{f}_{\text{erm}}(\lambda_{\text{erm}}) \tag{A.84}$$

$$=(\sigma_\star(\nu/\tau),\sigma_\star(\lambda_{\text{amp}}/\tau'),\sigma(\lambda_{\text{erm}})) \tag{A.85}$$

A.2.3 Proof of Theorem 2.3.3

Proof of Equation (2.20) Consider the local fields $(\nu,\lambda_{\text{erm}},\lambda_{\text{amp}})$ defined in Equation (A.59). As shown above, these local fields follow a Gaussian distribution with covariance matrix Σ given in Equation (2.13). Then, $(\nu,\lambda_{\text{erm}})$ follows a bivariate Gaussian and the density of ν conditioned on $\hat{f}_{\text{erm}}(\mathbf{x})=\sigma(\lambda_{\text{erm}})=p$ follows the Gaussian distribution with mean $\mu=\frac{m}{q_{\text{erm}}}\sigma^{-1}(p)$ and variance $v^2=\rho-\frac{m^2}{q_{\text{erm}}}$. Then,

$$\mathbb{E}_x\left[f_\star(\mathbf{x})|\hat{f}_{\text{erm}}(\mathbf{x})=p\right]=\int d\nu\frac{1}{2}\text{erfc}\left(-\frac{\nu}{\sqrt{2\tau^2}}\right)\mathcal{N}(\nu|\mu,v^2) \tag{A.86}$$

$$=\frac{1}{2}\text{erfc}\left(-\frac{\mu}{\sqrt{2\left(\tau^2+v^2\right)}}\right)=\frac{1}{2}\text{erfc}\left(-\frac{\frac{m}{q_{\text{erm}}}\sigma^{-1}(p)}{\sqrt{2(1-\frac{m^2}{q_{\text{erm}}}+\tau^2)}}\right) \tag{A.87}$$

$$=\sigma_\star\left(\frac{\frac{m}{q_{\text{erm}}}\sigma^{-1}(p)}{\sqrt{1-\frac{m^2}{q_{\text{erm}}}+\tau^2}}\right) \tag{A.88}$$

which yields Equation (2.20). We used the property that, for any a, b,

$$\int \mathrm{erf}(ax+b)\mathcal{N}(x|\mu,\sigma^2)\mathrm{d}x = \mathrm{erf}\left(\frac{a\mu+b}{\sqrt{1+2a^2\sigma^2}}\right) \tag{A.89}$$

Proof of Equation (2.21) We use the same computation as in the previous paragraph: since the conditioned on the Bayes local field $\hat{f}_{\mathrm{bo}}(\mathbf{x}) = \sigma_\star(\frac{\lambda_{\mathrm{amp}}}{\sqrt{\tau^2+1-q_{\mathrm{bo}}}}) = p$, the teacher local field is Gaussian with mean $\mu = \sqrt{\tau^2+1-q_{\mathrm{bo}}}\sigma_\star^{-1}(p)$ and variance $v^2 = 1-q_{\mathrm{bo}}$. As before, we have

$$\mathbb{E}_{\mathbf{x}}\left[f_\star(\mathbf{x})|\hat{f}_{\mathrm{bo}}(\mathbf{x}) = p\right] = \sigma_\star\left(\frac{\mu}{\sqrt{\tau^2+v^2}}\right) \tag{A.90}$$

$$= \sigma_\star\left(\frac{\sqrt{\tau^2+1-q_{\mathrm{bo}}}\sigma_\star^{-1}(p)}{\sqrt{\tau^2+1-q_{\mathrm{bo}}}}\right) = p \tag{A.91}$$

Hence the result of Equation (2.21).

Proof of Equation (2.22) The proof follows the same structure as the previous paragraphs: conditioned on $\sigma(\lambda_{\mathrm{erm}}) = p$, the law of λ_{amp} is $\mathcal{N}(\frac{m}{q_{\mathrm{erm}}}\sigma^{-1}(p), q_{\mathrm{bo}} - \frac{m^2}{q_{\mathrm{erm}}})$ and

$$\mathbb{E}_{\mathbf{x}}\left[\hat{f}_{\mathrm{bo}}(\mathbf{x})|\hat{f}_{\mathrm{erm}}(\mathbf{x}) = p\right] = \int \sigma_\star\left(\frac{-x}{\sqrt{\tau^2+1-q}}\right)\mathcal{N}(x|\frac{m}{q_{\mathrm{erm}}}\sigma^{-1}(p), q_{\mathrm{bo}} - \frac{m^2}{q_{\mathrm{erm}}}) \tag{A.92}$$

$$= \sigma_\star\left(\frac{\frac{m}{q_{\mathrm{erm}}}\sigma^{-1}(p)}{\sqrt{\tau^2+1-q_{\mathrm{bo}}+(q_{\mathrm{bo}}-\frac{m}{q_{\mathrm{erm}}})}}\right) \tag{A.93}$$

$$= \sigma_\star\left(\frac{\frac{m}{q_{\mathrm{erm}}}\sigma^{-1}(p)}{\sqrt{1-\frac{m^2}{q_{\mathrm{erm}}}+\tau^2}}\right) = \mathbb{E}_x\left[f_\star(\mathbf{x})|\hat{f}_{\mathrm{erm}}(\mathbf{x}) = p\right] \tag{A.94}$$

A.3 Additional figures

A.3.1 Logistic regression uncertainty supplement

Figure A.1 complements Figure 2.3 from the main text by showing the same plot as the right panel in Figure 2.3 for other values of sample complexity α and noise τ. We observe that at zero regularization the logistic regression is overconfident in all the depicted cases, in particular so at small α and small noise.

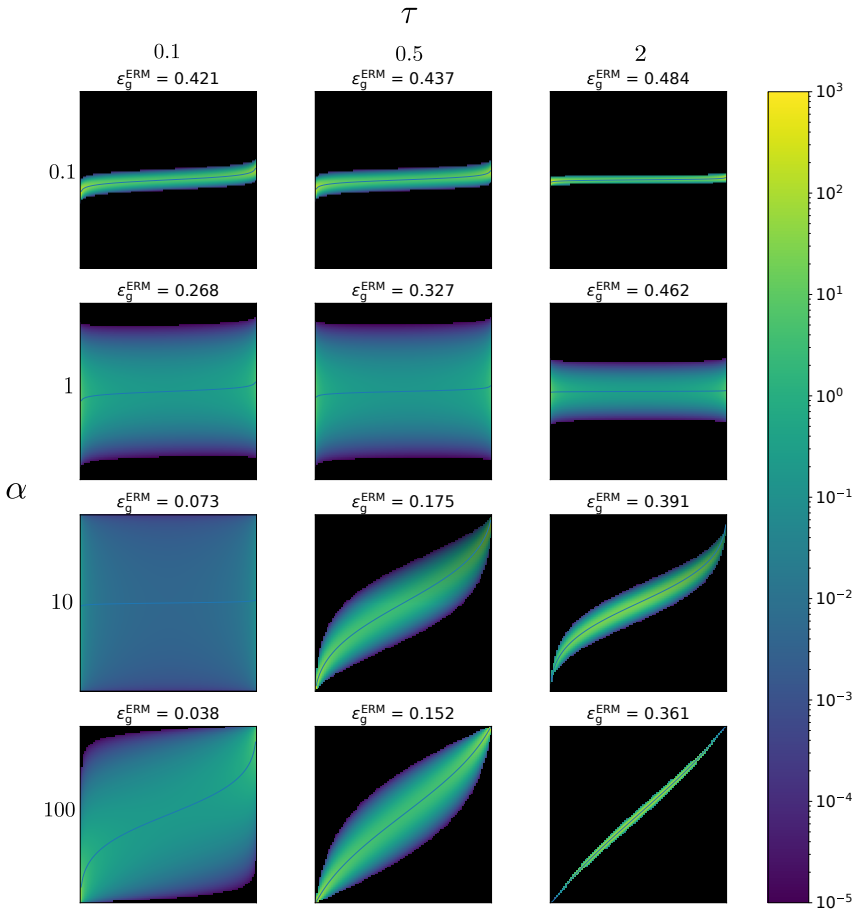


Figure A.1: Joint density of $\hat{f}_{\text{erm}}$ (x-axis) and $\hat{f}_{\text{bo}}$ (y-axis), at $\lambda = 0^+$. Blue curve is the mean of $\hat{f}_{\text{bo}}$ at fixed $\hat{f}_{\text{erm}}$. The test error of ERM is indicated above the corresponding plot. The test errors of Bayes for the same parameters are indicated in Figure 2.2.

A.3.2 Choosing optimal regularization supplement

Here we give additional illustration related to the section 2.4.3 in the Main text.

In figure A.2, the calibration Δ_p is shown as a function of λ at different levels p and different noise σ. First, observe that as λ grows the logistic regression is going from overconfident $\Delta_p > 0$

to underconfident $\Delta_p < 0$. For $\lambda \to \infty$, we have $\Delta_p \to p - 1$. Further, we observe that the value of λ at which the calibration is zero (the best calibration) has only mild dependence on the value of p. Finally, the vertical lines mark the values of regularization that minimize the validation error λ_{error}, and loss λ_{loss}. We see that λ_{loss} is closer to the well-calibrated region and that at small α this difference is more pronounced.

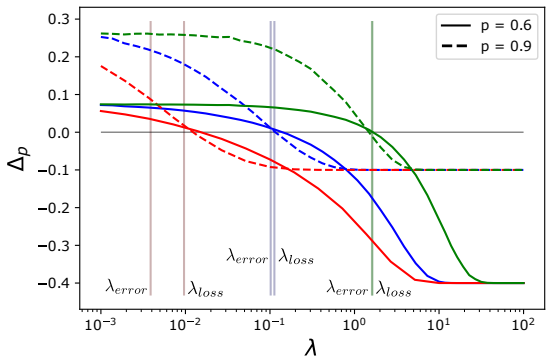


Figure A.2: Calibration Δ_p for $p = 0.9$ and $p = 0.6$ as a function of λ, for $\tau = 0$ (red curve) , $\tau = 0.5$ (blue curve), and $\tau = 2$ (green curve), at $\alpha = 5$. Vertical lines correspond to λ_{error} and λ_{loss} defined in 2.4.3. For $\tau = 2$, λ_{error} and λ_{loss} differ by only 10^{-2} and look indistinguishable on the plot.

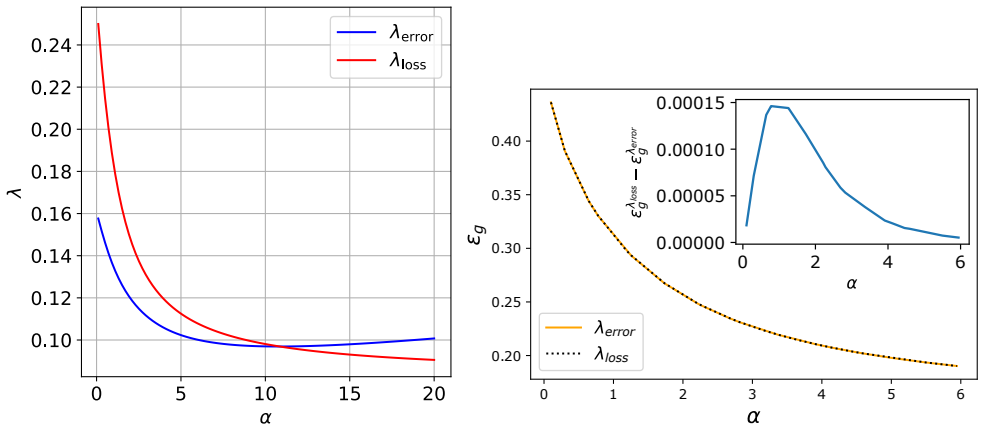
The left panel of Figure A.3 compares λ_{error} and λ_{loss} when $\tau = 0.5$.

The right panel of Figure A.3

then shows that the test error at λ_{loss} and λ_{error} are extremely close, with the difference being plot in the insert.

Figure A.4 depicts the joint density of $\hat{f}_{\text{erm}}$ (x-axis) and $\hat{f}_{\text{bo}}$ (y-axis) for several values of the regularization λ and the noise τ. As λ increases, we observe that the logistic regression changes from overconfident to underconfident, as we could also observe in figure A.2.

Next in Figure A.5 we depict the densities for λ_{error} and λ_{loss} for different values of α and τ. We observe an overall improvement in the calibration for these optimal regularizations.



(a) (b)

Figure A.3: Left: optimal penalization for logistic regression as a function of the sample complexity α, for $\tau = 0.5$. Right: Test error at optimal λ for $\sigma = 0.5$, as a function of α. Orange line (respectively black dotted line) corresponds to λ-error (respectively λ-loss). The two curves are indistinguishable on the plot. The blue curve in the inset shows $\varepsilon_g^{\lambda_{\text{loss}}} - \varepsilon_g^{\lambda_{\text{error}}}$ as a function of α: it appears that the difference is around $\sim 10^{-4}$.

A.4 Comparison to the data generated by logit model

As mentioned before, our state evolution equations can be adapted to data generated by the logit model, as studied in [16]. The likelihood is defined in Equation (A.99). Since this change only concerns the data distribution, Algorithm 2 is unchanged. However, state evolution is changed in the update of $\hat{m}, \hat{q}, \hat{V}$: the partition function $\mathcal{Z}_0$ is now

$$\mathcal{Z}_0(y, \omega, V) = \int \mathrm{d}z \; \sigma(y \times z) \mathcal{N}(z|\omega, V) \tag{A.95}$$

Note also that the expression of the calibration is now

$$\Delta_p = p - \int \mathrm{d}x \sigma(x) \mathcal{N}(x|{}^{m}/_{q} \times \sigma^{-1}(p), \rho - {}^{m^2}/_{q}) \tag{A.96}$$

State evolution equations The state evolution equations for the ERM estimator with this choice of likelihood is now

$$\begin{cases} \hat{m} &= \int \mathrm{d}z \partial_\omega \mathcal{Z}_0\left(y, \frac{m}{\sqrt{q}} z, V_\star\right) f_{erm}\left(y, \sqrt{q}, V\right) e^{-\frac{1}{2}z^2}/\sqrt{2\pi} \\ \hat{q} &= \int \mathrm{d}z \mathcal{Z}_0\left(y, \frac{m}{\sqrt{q}} z, V_\star\right) f_{erm}\left(y, \sqrt{q}, V\right)^2 e^{-\frac{1}{2}z^2}/\sqrt{2\pi} \\ \hat{V} &= \int \mathrm{d}z \mathcal{Z}_0\left(y, \frac{m}{\sqrt{q}} z, V_\star\right) \partial_\omega f_{erm}\left(y, \sqrt{q}, V\right) e^{-\frac{1}{2}z^2}/\sqrt{2\pi} \end{cases}, \begin{cases} m &= \frac{\hat{m}}{\lambda + \hat{V}} \\ q &= \frac{\hat{m}^2 + \hat{q}}{(\lambda + \hat{V})^2} \\ V &= \frac{1}{\lambda + \hat{V}} \end{cases} \tag{A.97}$$

where the partition function $\mathcal{Z}_0$ is written in (A.95) and f_{erm} is the same as Theorem 3.2. For the

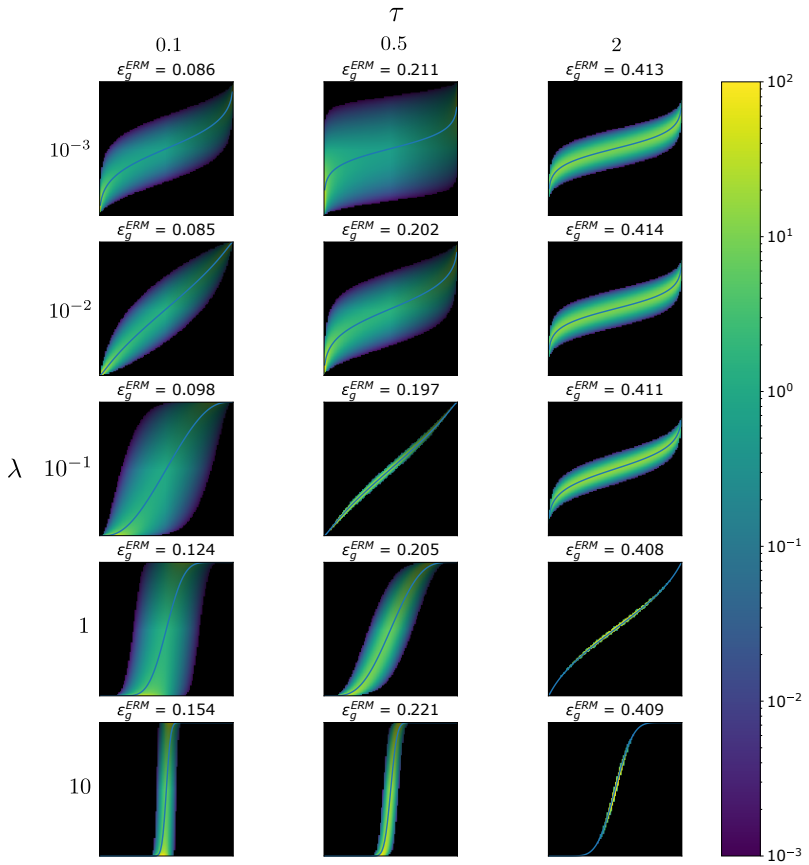


Figure A.4: Joint density of $\hat{f}_{\text{erm}}$ (x-axis) and $\hat{f}_{\text{bo}}$ (y-axis) at $\alpha = 5$. The best possible test errors are respectively $\varepsilon_g^\star = 0, 0.148, 0.352$ for $\tau = 0, 0.5, 2$. For the Bayes estimator with $\alpha = 5$, the test errors are $\varepsilon_g^{\text{bo}} = 0.083, 0.198, 0.402$

Bayes-optimal estimator, the state-evolution is now

$$\begin{cases} \hat{m} &= \int \mathrm{d}z \partial_\omega \mathcal{Z}_0\left(y, \frac{m}{\sqrt{q}}z, V_\star\right) f_0\left(y, \sqrt{q}, V\right) e^{-\frac{1}{2}z^2}/\sqrt{2\pi} \\ \hat{q} &= \int \mathrm{d}z \mathcal{Z}_0\left(y, \frac{m}{\sqrt{q}}z, V_\star\right) f_0\left(y, \sqrt{q}, V\right)^2 e^{-\frac{1}{2}z^2}/\sqrt{2\pi} \\ \hat{V} &= \int \mathrm{d}z \mathcal{Z}_0\left(y, \frac{m}{\sqrt{q}}z, V_\star\right) \partial_\omega f_0\left(y, \sqrt{q}, V\right) e^{-\frac{1}{2}z^2}/\sqrt{2\pi} \end{cases}, \begin{cases} m &= \frac{\hat{m}}{1+\hat{V}} \\ q &= \frac{\hat{m}^2+\hat{q}}{(1+\hat{V})^2} \\ V &= \frac{1}{1+\hat{V}} \end{cases} \tag{A.98}$$

where $f_0(y, \omega, V) = \frac{\partial \log \mathcal{Z}_0(y,\omega,V)}{\partial \omega}$.

Note that update of the overlaps m, q, V is unchanged compared to the probit case, as the update is related to the regularization function (through the *denoising* function f_w of AMP) which is still the L_2 norm.

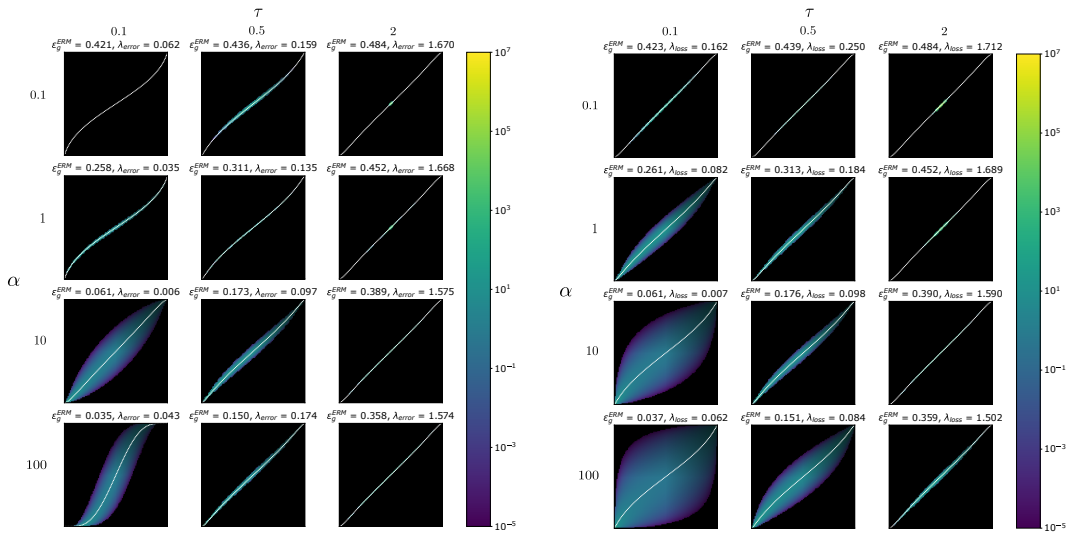


Figure A.5: Left: Joint density $\rho_{\text{erm,bo}}$, at $\lambda = \lambda_{\text{error}}$. λ_{error} and the test error of ERM are indicated above the corresponding plot. Right:Joint density $\rho_{\text{erm,bo}}$, at $\lambda = \lambda_{\text{loss}}$. λ_{loss} and the test error of ERM are indicated above the corresponding plot.

A.4.1 Behaviour at $\lambda = 0^+$

In [16], it has been shown that as the sampling ratio α goes to ∞, the unpenalized logistic classifier is calibrated when the data is generated by the *logit* model

$$\mathbb{P}(y_\star = 1) = \sigma(\mathbf{w}_\star \cdot \mathbf{x}) \tag{A.99}$$

In this section, we numerically recover the results from [16] i.e the unpenalized logistic estimator is calibrated asymptotically and the calibration decreases as $1/\alpha$. Figure A.6 plots the calibration at $p = 0.75, 0.9$ and 0.99 for $\alpha \in [10, 10^4]$. One can observe a decay of Δ_p with a power law, which confirms that with logistic data, the unpenalized logistic classifier is asymptotically calibrated at all levels. Fitting a linear model on these curves gives slopes equal to $-0.99, -1.00, -1.04$ for $p = 0.75, 0.9, 0.99$ respectively, which numerically validates the $1/\alpha$ rate derived in [16].

We compare here to the calibration with probit data, at $\tau = 0.5$. In particular, we exhibit that the logistic classifier cannot be calibrated at all levels p. Indeed, as $\alpha \to \infty$, it can be noted that $\cos(\hat{\mathbf{w}}_{\text{erm}}, \theta_\star) = m^2/q \to_\infty 1$. Moreover, we observe that $m/q = m^2/q \times 1/m \to_\infty m_\infty := \lim m$. Using the expression for calibration from Theorem 2.3.3, we get that for $p > 1/2$,

$$\Delta_p \to_\infty p - \sigma_\star\left(\frac{\sigma^{-1}(p)}{\tau \times m_\infty}\right) \tag{A.100}$$

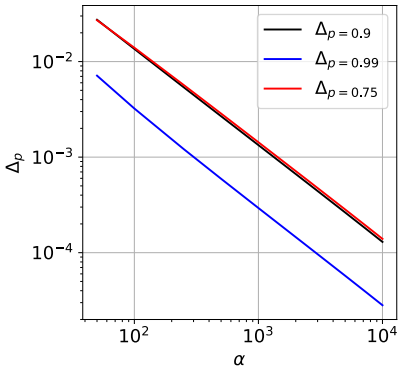


Figure A.6: Calibration Δ_p at $p = 0.75, 0.9$ and 0.99 of logistic regression with the logit model as a function of α. The plots are given in log-log scale. On this scale, the curves have respective slopes $-0.99, -1.01, -1.04$

And deduce that

$$\Delta_p = 0 \Leftrightarrow \frac{\sigma_\star^{-1}(p)}{\sigma^{-1}(p)} = \frac{1}{\tau \times m_\infty} \tag{A.101}$$

Noting $r(p) := \frac{\sigma_\star^{-1}(p)}{\sigma^{-1}(p)}$, we get the condition

$$p = r^{-1}(\frac{1}{\tau \times m_\infty}) \tag{A.102}$$

With $\tau = 0.5$, we numerically get $m_\infty \simeq 3.53 \Rightarrow \tau \times m_\infty \simeq 1.76$ The level p_0, defined as the only $p > 1/2$ such that $\Delta_p = 0$, is thus

$$p_0 = r^{-1}(\frac{1}{\tau \times m_\infty}) \simeq r^{-1}(0.57) \simeq 0.937 \tag{A.103}$$

For $1/2 < p < p_0$ (respectively $1 > p > p_0$), $\Delta_p > 0$ (respectively $\Delta_p < 0$). This can be observed in Figure A.7 where we have plotted Δ_p for several levels. For $p \neq p_0$, the calibration seems to converge to a finite value. On the other hand, at $p = p_0$, Δ_p converges to 0 as a power-law.

A.4.2 Behaviour a $\lambda = 1$, λ_{error} and λ_{loss}

In this section, we adapt the theoretical results of Figure 2.5 to the logit data model: we compute λ_{error} and λ_{loss} and plot their respective test errors and calibration. Note the definition of the test error and loss in this setting:

$$\begin{cases} \varepsilon_g & = \sum_y \mathbb{E}_{\xi\sim\mathcal{N}(0,1)} \left[\mathcal{Z}_0(y, m/\sqrt{q}\xi, 1 - m^2/q) \delta(\text{sign}(\xi) = y) \right] \\ \mathscr{L}_g & = -\sum_y \mathbb{E}_{\xi\sim\mathcal{N}(0,1)} \left[\mathcal{Z}_0(y, m/\sqrt{q}\xi, 1 - m^2/q) \log \sigma(y \times \sqrt{q}\xi) \right] \end{cases} \tag{A.104}$$

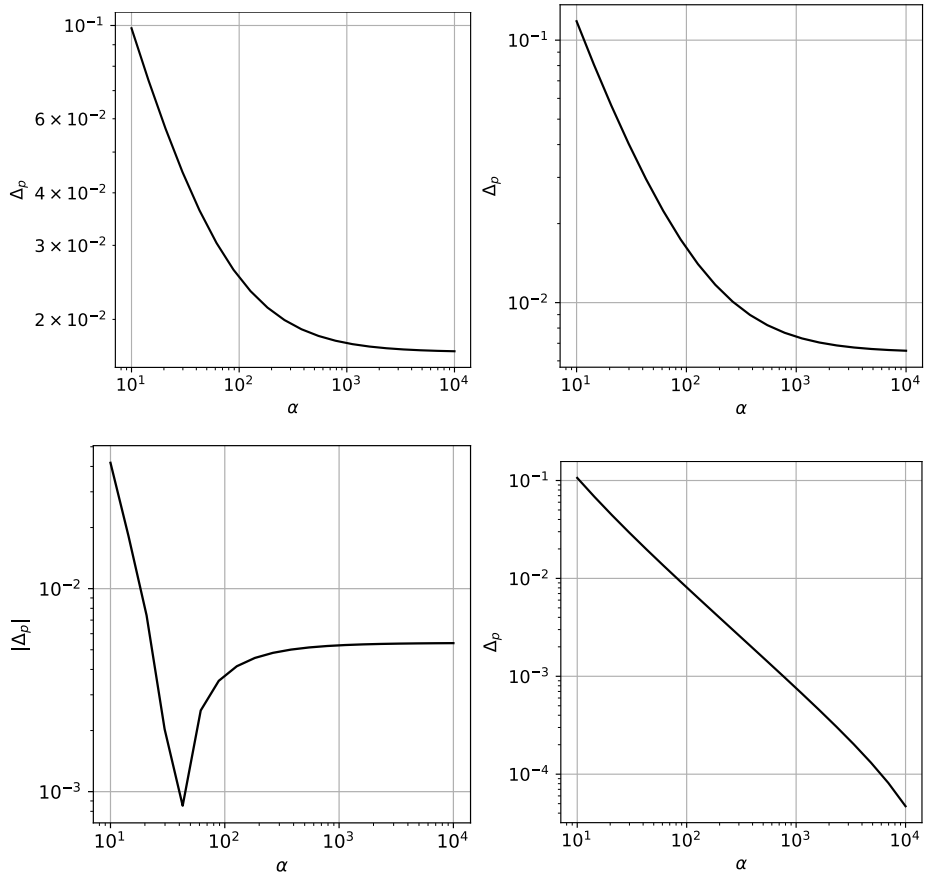


Figure A.7: Calibration for logistic regression with $\lambda = 0^+, \tau = 0.5$, at four different levels: $p = 0.75$ (Top left), $p = 0.9$ (Top right), $p = 0.99$ (Bottom left) and $p = p_0 \simeq 0.937$ (Bottom right). For $p = 0.99$, Δ_p becomes negative around $\alpha \simeq 50$ so the absolute value of Δ_p has been plotted instead. C seems to converge to zero for $p = p_0$ only.

Moreover, with the logit data model, the empirical risk at $\lambda = 1$ now has a Bayesian interpretation. The risk corresponds to the logarithm of the posterior distribution on $\mathbf{w}$, up to a normalization constant, because $\mathbf{w}_\star$ is sampled from a Gaussian with identity covariance. At $\lambda = 1$, the empirical risk minimizer $\hat{\mathbf{w}}_{\text{erm}}$ is the Maximum A Posteriori (MAP). In this section, we compare the performance of logistic regression with the two different optimal regularizations and with $\lambda = 1$.

The left panel of Figure A.8 shows the value of λ_{error} and λ_{loss}. As with the probit model, $\lambda_{\text{loss}} > \lambda_{\text{error}}$. Note also that both optimal values are bigger than 1 for this range of α. The right panel shows their respective test error ε_g. As with the probit model, λ_{error} has a lower error than λ_{loss}. Not surprisingly, $\lambda = 1$ has worse test error than both optimal λ. The left panel of Figure A.9 shows the calibration with the three different regularizations at $p = 0.75$. We observe that $\lambda = 1$ yields an overconfident estimator (consistent with the fact that λ_{error} and λ_{loss} are both bigger than 1), and as before, λ_{loss} is less confident than λ_{error}. Remark that an underconfident

estimator is not necessarily better than an overconfident one, and the calibration Δ_p is only a measure on one level p. To compare the different estimators more fairly, we can thus use a metric called *Expected Calibration Error* defined as

$$\mathrm{ECE} := \mathbb{E}_{\hat{f}(\mathbf{x})}\left(|\Delta_{\hat{f}(\mathbf{x})}|\right) = \int \mathrm{d}p |\Delta_p| \frac{\mathcal{N}(\sigma^{-1}(p)|0, q_{\mathrm{erm}})}{p(1-p)} \tag{A.105}$$

The ECE measures the average of $|\Delta_p|$ at all levels p weighted by the probability that $\hat{f}(\mathbf{x}) = p$. In other words, at a given level p, if $\mathbb{P}(\hat{f}(\mathbf{x}) = p) = 0$, the ECE of the estimator will not be affected by the calibration of the estimator at p. The right panel of Figure A.9 plots the ECE as a function of α for $\lambda = 1$, λ_{error} and λ_{loss}. We again observe that λ_{loss} has a lower ECE than λ_{error}, which confirms that optimizing λ for the test loss yields a more calibrated estimat or. Moreover, $\lambda = 1$ yields an estimator with the worst ECE, which is coherent with the left panel: at $p = 0.75$, the absolute value of its calibration is higher than λ_{error} and λ_{loss}. Our numerical results show that even if we know the prior distribution on the posterior and the likelihood, using only a point estimate for the parameter (here the maximum a posteriori) yields an overconfident estimator.

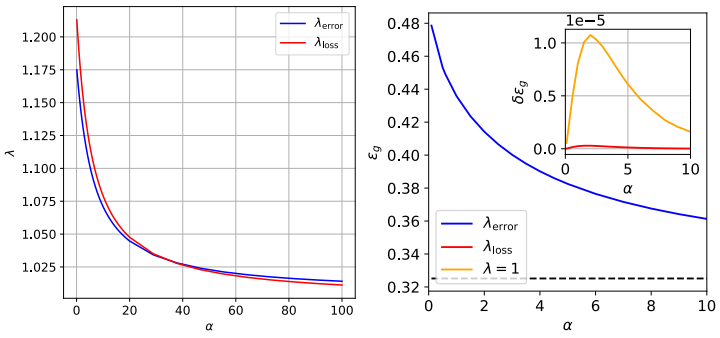


Figure A.8: Left: Values of λ_{error} and λ_{loss} as a function of α for the logistic data model. Center: Values of the test error ε_g for λ_{error} (blue curve) . The inset plots the difference of test error $\delta\varepsilon_{g,\mathrm{loss}} := \varepsilon_g(\lambda_{\mathrm{loss}}) - \varepsilon_g(\lambda_{\mathrm{error}})$ (red curve) and $\delta\varepsilon_{g,1} := \varepsilon_g(\lambda = 1) - \varepsilon_g(\lambda_{\mathrm{error}})$ (orange curve). Right: Calibration at $p = 0.75$ of logistic regression on logistic data, for $\lambda = 1$, λ_{error} and λ_{loss}. The curves are given by running state evolution.

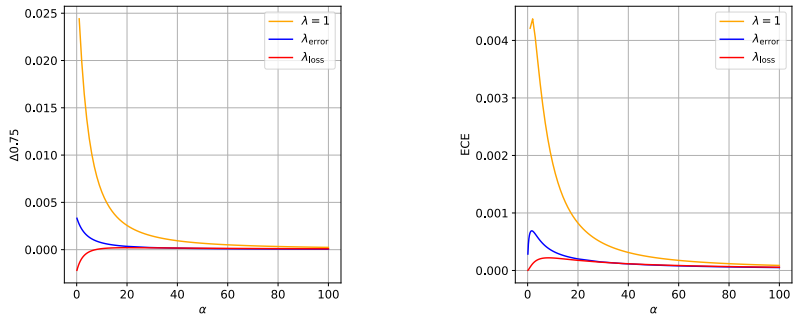


Figure A.9: Left: Calibration at $p = 0.75$ of logistic regression on logistic data, for $\lambda = 1$, λ_{error} and λ_{loss}. The curves are given by running state evolution. Right: Expected Calibration Error (ECE) for $\lambda = 1$, λ_{error}, λ_{loss}. The lower ECE, the better.

B Appendix of chapter 3

B.1 Gaussian equivalence

As discussed in the main, our analysis of the random features model introduced in Sec. 3.2.2 relies on a recent progress in high-dimensional statistics known as the *Gaussian equivalence theorem* (GET). In this Appendix, we recall the reader of the main results in this line of work.

B.1.1 Informal discussion and key idea

For convenience let's first recall the model of interest. Consider data $(\mathbf{x}^\mu, y^\mu)_{\mu\in[n]} \in \mathbb{R}^d \times \mathscr{Y}$ which we assume was independently drawn from the following model:

$$y^\mu = f_\star(\theta_\star^\top \mathbf{x}^\mu), \qquad \mathbf{x}^\mu \sim \mathcal{N}(\mathbf{0}, {}^1\!/_d \mathbf{I}_d), \qquad \theta_\star \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d) \tag{B.1}$$

where $f_\star : \mathbb{R} \to \mathscr{Y}$ is an activation function, which we assume can be potentially stochastic (as in the logit model studied in the main, Sec. 3.2.2). For convenience, define the matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ and the vector $\mathbf{y} \in \mathscr{Y}^n$ obtained by stacking together $\mathbf{x}^\mu$ and $\mathbf{y}^\mu$ row-wise. We are interested in studying the following generalized linear predictor:

$$\hat{y} = f(\hat{\theta}^\top \varphi(\mathbf{x})) \tag{B.2}$$

where $\varphi : \mathbb{R}^d \to \mathbb{R}^p$ is a feature map, and $\hat{\theta} \in \mathbb{R}^p$ are weights, which generally depend on the training data $\hat{\theta} = \hat{\theta}(\varphi(\mathbf{X}), \theta_\star)$, where for convenience we defined the feature matrix $\varphi(\mathbf{X}) \in \mathbb{R}^{n\times p}$. The *random features model* correspond to the specific feature map:

$$\varphi(\mathbf{x}) = \frac{1}{\sqrt{p}} \phi(\mathbf{F}\mathbf{x}) \tag{B.3}$$

where $\mathbf{F} \in \mathbb{R}^{p\times d}$ is a random matrix and $\phi : \mathbb{R} \to \mathbb{R}$ is a component-wise activation function. Our key goal is to characterize the statistics of the model, i.e. to compute expectations over functions

of the test and training predictions:

$$\mathbb{E}_{\mathbf{X},\mathbf{x},\theta_\star}\left[\psi\left(f_\star(\theta_\star^\top\mathbf{x}), f(\hat{\theta}(\varphi(\mathbf{X}),\theta_\star)^\top\varphi(\mathbf{x}))\right)\right], \qquad \mathbb{E}_{\mathbf{X},\theta_\star}\left[\tilde{\psi}\left(f_\star(\mathbf{X}\theta_\star), f(\varphi(\mathbf{X})\hat{\theta}(\varphi(\mathbf{X}),\theta_\star)\right)\right] \tag{B.4}$$

where $\psi : \mathscr{Y}^2 \to \mathbb{R}$ and $\tilde{\psi} : \mathscr{Y}^{2n} \to \mathbb{R}$ are test functions. Note in particular that the generalization errors eqs. (3.6), (3.7) and the density eq. (3.8) are examples of the above.

Different tools from high-dimensional statistics have been designed to compute such expectations in the limit where $n, p, d \to \infty$ at fixed rates $\alpha = {}^{n}/_{p}$ and $\gamma = {}^{d}/_{p}$, both rigorously and heuristically, e.g. the replica method [171, 247], CGMT [218, 230, 228], approximate message passing [26, 69, 141, 97], cavity / leave-one-out method [170, 132, 158], tools from random matrix theory [129, 68], among others. A shortcoming of all the aforementioned methods is that they typically rely on the Gaussianity of the input data, and therefore are not directly applicable to the random features model (note that even if $\mathbf{F}\mathbf{x} \in \mathbb{R}^p$ is a Gaussian vector, the features $\phi(\mathbf{F}\mathbf{x})$ are *not* Gaussian).

Gaussian equivalence provides a surprising answer to this hurdle. Assuming for simplicity that the features are centred $\mathbb{E}_{\mathbf{x}}[\varphi(\mathbf{x})] = 0$ and defining the covariances:

$$\Phi = \mathbb{E}_{\mathbf{x}}[\varphi(\mathbf{x})\mathbf{x}^\top] \in \mathbb{R}^{p\times d}, \qquad \Omega = \mathbb{E}_{\mathbf{x}}[\varphi(\mathbf{x})\varphi(\mathbf{x})^\top] \in \mathbb{R}^{p\times p} \tag{B.5}$$

Gaussian equivalence states that in the high-dimensional limit, the expectations in eq. (B.4) can be computed for an *equivalent Gaussian model* with matching second moments:

$$\begin{aligned}\mathbb{E}_{\mathbf{X},\mathbf{x},\theta_\star}\left[\psi\left(f_\star(\theta_\star^\top\mathbf{x}), f(\hat{\theta}(\varphi(\mathbf{X}),\theta_\star)^\top\varphi(\mathbf{x}))\right)\right] &\xrightarrow[p\to\infty]{} \mathbb{E}_{\mathbf{X},\mathbf{V},\mathbf{x},\mathbf{v},\theta_\star}\left[\psi\left(f_\star(\theta_\star^\top\mathbf{x}), f(\hat{\theta}(\mathbf{V},\theta_\star)^\top\mathbf{v})\right)\right]\\ \mathbb{E}_{\mathbf{X},\theta_\star}\left[\tilde{\psi}\left(f_\star(\mathbf{X}\theta_\star), f(\varphi(\mathbf{X})\hat{\theta}(\varphi(\mathbf{X}),\theta_\star)\right)\right] &\xrightarrow[p\to\infty]{} \mathbb{E}_{\mathbf{X},\mathbf{V},\theta_\star}\left[\tilde{\psi}\left(f_\star(\mathbf{X}\theta_\star), f(\mathbf{V}\hat{\theta}(\mathbf{V},\theta_\star)\right)\right]\end{aligned} \tag{B.6}$$

where $(\mathbf{x}^\mu, \mathbf{v}^\mu)_{\mu\in[n]}$ are n independent samples of jointly Gaussian random variables:

$$(\mathbf{x},\mathbf{v}) \sim \mathcal{N}\left(\mathbf{0}_{d+p}, \begin{bmatrix} {}^{\mathbf{I}_d}/_{d} & {}^{\Phi}/_{\sqrt{pd}} \\ {}^{\Phi^\top}/_{\sqrt{pd}} & {}^{\Omega}/_{p} \end{bmatrix}\right) \tag{B.7}$$

and as before we defined the matrices $\mathbf{X} \in \mathbb{R}^{n\times d}$ and $\mathbf{V} \in \mathbb{R}^{n\times p}$ by stacking the samples row-wise. For the random features model $\varphi(\mathbf{x}) = {}^{1}/_{\sqrt{p}}\,\phi(\mathbf{F}\mathbf{x})$, the asymptotic covariances Φ, Ω can be computed explicitly, and are given by:

$$\Phi \underset{p\to\infty}{\asymp} \frac{\kappa_1}{\sqrt{p}}\mathbf{F}, \qquad \Omega \underset{p\to\infty}{\asymp} \kappa_0^2\mathbf{1}_p\mathbf{1}_p^\top + \frac{\kappa_1^2}{p}\mathbf{F}\mathbf{F}^\top + \kappa_\star^2\mathbf{I}_p \tag{B.8}$$

where the constants $(\kappa_0, \kappa_1, \kappa_\star) \in \mathbb{R}^3$ are the Gaussian moments of the activation function ϕ:

$$\kappa_0 = \mathbb{E}_{z\sim\mathcal{N}(0,1)}[\phi(z)], \qquad \kappa_1 = \mathbb{E}_{z\sim\mathcal{N}(0,1)}[\phi'(z)], \qquad \kappa_\star = \sqrt{\mathbb{E}_{z\sim\mathcal{N}(0,1)}[\phi(z)^2] - \kappa_1^2 - \kappa_0^2}. \tag{B.9}$$

Therefore, for the random features problem the Gaussian equivalent model can be written

explicitly in terms of the input data $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, {}^1\!/_d \mathbf{I}_d)$ and the weights $\mathbf{F} \in \mathbb{R}^{p \times d}$ as:

$$\mathbf{v} = \kappa_0 \mathbf{1}_p + \frac{\kappa_1}{\sqrt{p}} \mathbf{F}\mathbf{x} + \kappa_\star \mathbf{z} \tag{B.10}$$

where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_p)$ is an effective noise vector which is independent from $\mathbf{F}$, $\mathbf{x}$ and $\theta_\star$. In summary, in the high-dimensional limit the statistics of the random features model is equivalent to the statistics of a Gaussian equivalent model with noisy features. The later can be directly characterized by the methods mentioned in the last paragraph.

B.1.2 Gaussian equivalence theorem

Related literature: Gaussian universality has a long history, and appeared in many contexts ranging from random matrix theory [79, 80, 227] to signal processing [70], statistical learning [1, 191, 135] and physics [43, 45]. In the context of random features, a precursor of the result discussed here is the observation that for Gaussian data the high-dimensional limit of kernel spectra is linearly related the spectrum of the inputs [130]. This result was generalized to random features kernels in [196, 147], and leveraged by [168] to derive exact asymptotic expressions for the generalization and training error of random features regression. For ridge regression, computing the performance boils down to computing traces of the feature matrix, and therefore Gaussian universality can be seen as an instance of spectral universality of random matrices [34]. In non-linear problems where a closed-form solution is not available (as in our classification setting), Gaussian universality for the random features model was shown to hold for the empirical risk minimizer in [94, 99], and was later proven in [109, 66]. More recently, [177] extended this proof to more general empirical risk minimization problems, and showed the universality of the free energy density associated to the empirical risk at finite temperature. Finally, [59] generalized this free energy universality result and proved universal weak convergence for both empirical risk minimizers and sampling from the empirical Bayes measure. In particular, this result covers the Bayesian classifier and the uncertainty quantification metrics (calibration, ECE, etc.) studied here. This version is better suited to our discussion, since it is closer to the Bayesian classifiers studied in the main.

Theorem B.1.1 (Lemma 1 from [177])**.** *Consider the random features model discussed in Sec. B.1.1. Assume that the activation ϕ is three times differentiable and has zero Gaussian mean $\kappa_0 = \mathbb{E}_{z\sim\mathcal{N}(0,1)}[\phi(z)] = 0$ and that the weight matrix $\mathbf{F} \in \mathbb{R}^{p\times d}$ has rows $\mathbf{f}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ for $i \in [p]$. Further, assume that the function $f_\star$ is Lipschitz with i.i.d. bounded sub-Gaussian noise. Define the free energy density at inverse temperature $\beta > 0$:*

$$f_\beta(\varphi(X)) = -\frac{1}{p\beta} \log \int_{\mathbb{R}^p} d\theta \, \exp\left\{ -\beta \left[-\sum_{\mu=1}^{n} \log\sigma\left(f_\star(\theta_\star^\top \mathbf{x}^\mu) \times \theta^\top \varphi(\mathbf{x}^\mu) \right) + \frac{\lambda}{2} ||\theta||_2^2 \right] \right\}. \tag{B.11}$$

Then for any bounded differentiable function ψ with Lipschitz derivative we have:

$$\lim_{p\to\infty} \left| \mathbb{E}\left[\psi\left(f_\beta(\varphi(X))\right)\right] - \mathbb{E}\left[\psi\left(f_\beta(\mathbf{V})\right)\right] \right| = 0 \tag{B.12}$$

We refer the reader to [177] for the technical details on the proof of this result.

B.1.3 Beyond random features

The Gaussian equivalence theorem for the random features model motivates the study of generalized linear models with general Gaussian covariates. For instance, consider n independently drawn Gaussian covariates $(\mathbf{u}^\mu, \mathbf{v}^\mu) \in \mathbb{R}^{d+p}$:

$$\begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}_{d+p}, \begin{bmatrix} \Psi & \Phi \\ \Phi^\top & \Omega \end{bmatrix} \right) \tag{B.13}$$

for positive definite matrices $\Psi \in \mathbb{R}^{d\times d}$, $\Omega \in \mathbb{R}^{p\times p}$ and $\Phi \in \mathbb{R}^{p\times d}$ such that $\Psi - \Phi\Omega^{-1}\Phi^\top$ is invertible. Labels $y^\mu \in \mathcal{Y}$ are generated from the covariate $\mathbf{u} \in \mathbb{R}^p$ from a generalized linear model:

$$y^\mu = f_\star(\theta_\star^\top \mathbf{u}^\mu/\sqrt{d}), \qquad \theta_\star \sim \mathcal{N}(0, \mathbf{I}_d). \tag{B.14}$$

However, the statistician only observes the pairs $(\mathbf{v}^\mu, y^\mu) \in \mathbb{R}^p \times \mathcal{Y}$, from which she tries to learn:

$$\hat{y} = f(\hat{\theta}^\top \mathbf{v}/\sqrt{p}). \tag{B.15}$$

The asymptotic statistics of this Gaussian covariate model has been derived and proven [153] for the particular case in which $\hat{\theta}$ is the empirical risk minimizer. In Appendix B.2, we recover and generalize this result to the other estimators defined in Sec. 3.2.1.

Note that thanks to Gaussian equivalence, in the proportional high-dimensional limit, the random features model discussed in Sec. B.1.1 is a particular case of this Gaussian covariate model where $\mathbf{u} = \mathbf{x}$ and $\mathbf{v} = \varphi(\mathbf{x})$. However, the Gaussian covariate model can accommodate a richer class of models. For instance, one could consider the case in which the target covariates themselves come from a feature map: $\mathbf{u} = \varphi_\star(\mathbf{x})$. Although Gaussian equivalence has only been established for a limited number of feature maps, [153] has empirically observed that the asymptotic formulas derived for Gaussian covariates are in good agreement with a rich class of feature maps, including case in which the fixed features are learned from neural networks. While the goal of this work *is not* to investigate Gaussian equivalence, this line of work motivate us to derive our result for general Gaussian covariates, hence making them readily applicable to equivalences proven in the future.

B.2 Derivation of theorem 3.3.1

In this Appendix we provide a derivation of the self-consistent equations (3.14) characterizing the sufficient statistics $(m_t^\star, q_t^\star, v_t^\star, \hat{m}_t^\star, \hat{q}_t^\star, \hat{v}_t^\star)$ for $t \in \{\text{bo}, \text{erm}, \text{eb}, \text{Lap}\}$. As motivated in Appendix B.1, our discussion will focus on the more general Gaussian covariate model, which contains the random features setting as a particular case. The key idea is to design an approximate message passing and show that the associated state evolution equations coincide exactly the with the self-consistent equations for the sufficient statistics in Theorem 3.3.1.

This Appendix is organized as follows. We start by a recap of the Gaussian covariate model for our specific setting in Sec. B.2.1, and introduce a convenient change of variables. Next, in Sec. B.2.2 we introduced a tailored message passing algorithm, and provide an informal derivation of the associated state evolution equations. In Sec. B.2.3 we provide a heuristic derivation of the self-consistent equations from the replica method, and show that it agrees with the state evolution equations for our algorithm. In Sec. B.2.4 we discuss how these equations are made rigorous from recent progress in the literature. Finally, in the three last subsections we discuss variations of this result to the context of the Laplace approx imation and temperature scaling, and a simplification for the the random features case.

B.2.1 Recap of the setting

As motivated in Sec. B.1.3, our goal is to derive the self-consistent equations in the more general setting of the Gaussian covariate model (GCM), which thanks to Gaussian universality contains the random features model as a special case. For the ease of reading, we first recall the reader of the model of interest, which specializes Sec. B.1.3 to binary classification.

Data model: Let $(\mathbf{u}, \mathbf{v})$ denote a pair of Gaussian covariates:

$$\begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}_{d+p}, \begin{bmatrix} \Psi & \Phi \\ \Phi^\top & \Omega \end{bmatrix}\right). \tag{B.16}$$

and define the oracle classifier as:

$$f_\star(\mathbf{u}) = \mathbb{P}(y = 1 | \theta_\star^\top \mathbf{u}) = \sigma_{\tau^2}\left(\frac{\theta_\star^\top \mathbf{u}}{\sqrt{d}}\right), \qquad \theta_\star \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d), \tag{B.17}$$

where we remind the reader of the convenient notation:

$$\sigma_\tau(x) := \int \sigma(z) \mathcal{N}(z | x, \tau) \mathrm{d}z \tag{B.18}$$

with $\sigma(z) = (1 + e^{-z})^{-1}$ the sigmoid function.

Classifiers: Given n independent pairs $(\mathbf{v}^\mu, y^\nu)_{\mu\in[n]} \in \mathbb{R}^p \times \{-1,1\}$ from the model above and defining the training data $\mathscr{D} = \{(\mathbf{v}^\mu, y^\mu)_{\mu\in[n]}\}$ we are interested in studying the family of probabilistic classifiers of the type:

$$\hat{f}_t(\mathbf{v}) = \mathbb{P}(y=1|\tau_t,\mathscr{D}) \int \mathrm{d}\theta\ \sigma_{\tau_t}\left(\frac{\theta^\top \mathbf{v}}{\sqrt{p}}\right) p_t(\theta|\mathscr{D}) \tag{B.19}$$

where the "posterior" $p_t(\theta|\mathscr{D})$ and the noise level τ_t depend on the specific classifier $t \in \{\mathrm{bo}, \mathrm{erm}, \mathrm{eb}, \mathrm{Lap}\}$ of interest introduced in Sec. 3.2.1.

A convenient rewriting: Since the covariate $\mathbf{u} \in \mathbb{R}^d$ is not observed by the statistician, it is useful to rewrite it explicitly as a function of $\mathbf{v}$ and an effective uncorrelated noise. Additionally, it is also convenient to write $\mathbf{v}$ in terms of an uncorrelated variable. Mathematically, this is given by a standard Gaussian partition:

$$\mathbf{u} = \Phi\Omega^{-1}\mathbf{v} + \left(\Psi - \Phi\Omega^{-1}\Phi^\top\right)^{1/2}\mathbf{z}, \tag{B.20}$$

for $\mathbf{z} \sim \mathscr{N}(0, I_d)$ uncorrelated with $\mathbf{v}$. This motivate us to define the *projected oracle weights*:

$$\mathbf{w}_\star = \Omega^{-1}\Phi^\top\theta_\star \tag{B.21}$$

Then, the oracle classifier can be equivalently written as:

$$\mathbb{P}(y=1|\mathbf{w}_\star^\top\mathbf{v}) = \int \sigma_{\tau^2}\left(\frac{\mathbf{w}_\star^\top\mathbf{v}}{\sqrt{d}} + \frac{1}{\sqrt{d}}\theta_\star^\top\left(\Psi - \Phi\Omega^{-1}\Phi^\top\right)^{1/2}\mathbf{z}\right)\mathscr{N}(\mathbf{z}|0,\mathbf{I}_d)\,\mathrm{d}\mathbf{z} \tag{B.22}$$

$$= \int \sigma_{\tau^2}\left(\frac{\mathbf{w}_\star^\top\mathbf{v}}{\sqrt{d}} + \xi\right)\mathscr{N}\left(\xi\middle|0, \frac{\theta_\star^\top\left(\Psi - \Phi\Omega^{-1}\Phi^\top\right)\theta_\star}{d}\right)\mathrm{d}\xi \tag{B.23}$$

which is a logit model on the observed features with an effective mismatch noise $\xi \sim \mathscr{N}(0, \frac{1}{d}\theta_\star^\top\left(\Psi - \Phi\Omega^{-1}\Phi^\top\right)\theta_\star)$. Recalling that $\theta_\star \sim \mathscr{N}(\mathbf{0}_d, \mathbf{I}_d)$, in the asymptotic limit the noise variance concentrates:

$$\frac{1}{d}\theta_\star^\top\left(\Psi - \Phi\Omega^{-1}\Phi^\top\right)\theta_\star \to \frac{1}{d}\mathrm{Tr}\left(\left(\Psi - \Phi\Omega^{-1}\Phi^\top\right)\right) =: \tau^2_{\mathrm{add}}. \tag{B.24}$$

Therefore, the oracle classifier is equivalent to:

$$f_\star(\mathbf{v}) = \mathbb{P}(y=1|\mathbf{w}_\star^\top\mathbf{v}) = \sigma_{\tau^2+\tau^2_{\mathrm{add}}}\left(\frac{\mathbf{w}_\star^\top\mathbf{v}}{\sqrt{d}}\right) \tag{B.25}$$

with

$$\mathbf{w}_\star \sim \mathscr{N}(0, \Sigma_\star), \quad \Sigma_\star = \Omega^{-1}\Phi^\top\Phi\Omega^{-1} \tag{B.26}$$

Finally, to further simplify the algebra it is convenient to consider the following change of variables:

$$\mathbf{v} \to \Omega^{-1/2}\mathbf{v}, \quad \mathbf{w}_\star \to \Omega^{1/2}\mathbf{w}_\star \tag{B.27}$$

Such that $\mathbf{v} \sim \mathcal{N}(\mathbf{0}_p, \mathbf{I}_p)$ and $\mathbf{w}_\star \sim \mathcal{N}(\mathbf{0}_d, \tilde{\Phi}^\top\tilde{\Phi})$ with $\tilde{\Phi} \equiv \Phi\Omega^{-1/2}$. Note that the labels are invariant under this change, and therefore we can assume input data with identity covariance.

B.2.2 State evolution for GAMP

Algorithm 7 GAMP for an estimator $t \in \{\text{erm}, \text{bo}, \text{eb}\}$

Input: Data $\mathbf{V} \in \mathbb{R}^{n\times p}$, $\mathbf{y} \in \{-1, 1\}^n$
Define $\mathbf{V}^2 = \mathbf{V} \odot \mathbf{V} \in \mathbb{R}^{n\times p}$ and Initialize $\hat{\theta}^{T=0} = \mathcal{N}(\mathbf{0}, \sigma_w^2\mathbf{I}_d)$, $\hat{\mathbf{c}}^{T=0} = \mathbf{1}_d$, $\mathbf{g}^{T=0} = \mathbf{0}_n$.
for $T \le T_{\max}$ **do**
$\mathbf{V}^T = \mathbf{V}^2\hat{\mathbf{c}}^T$; $\omega^T = \mathbf{V}\hat{\theta}^T - \mathbf{V}^T \odot \mathbf{g}^{t-1}$; /* *Update channel mean and variance* */
$\mathbf{g}^T = f_{\text{out},t}(\mathbf{y}, \omega^T, \mathbf{V}^T)$; $\partial\mathbf{g}^T = \partial_\omega f_{\text{out},t}(\mathbf{y}, \omega^T, \mathbf{V}^T)$; /* *Update channel* */
$\mathbf{A}^T = -\mathbf{V}^{2\top}\partial\mathbf{g}^T$; $\mathbf{b}^T = \mathbf{V}^\top\mathbf{g}^T + \mathbf{A}^T \odot \hat{\theta}^T$; /* *Update prior mean and variance* */
/* *Update marginals* */
$\hat{\theta}^{T+1} = f_{w,t}(\mathbf{b}^T, \mathbf{A}^T)$; $\hat{\mathbf{c}}^{T+1} = \text{diag}(\partial_{\mathbf{b}} f_{w,t}(\mathbf{b}^T, \mathbf{A}^T))$
end for
Return: Estimators $(\hat{\theta}_t^{\text{amp}}, \hat{\mathbf{c}}_t^{\text{amp}}) := (\hat{\theta}_t^{T_{\max}}, \hat{\mathbf{c}}_t^{T_{\max}})$.

With the model in hands, we now show discuss how the sufficient statistics $(v_t^\star, q_t^\star, m_t^\star)$ needed to characterize the asymptotic density defined in eq. (3.12) satisfy a set of self-consistent equations, which in the particular case of the random features model are explicitly written given in eq. (3.14). Our derivation follows from the analysis of an approximate message passing scheme, which provides a powerful tool to derive exact asymptotic results in an unified way, and has been employed in many works in the high-dimensional statistics literature, e.g. [27, 26, 202, 141, 69, 219, 154, 44, 97, 152, 54].

Given the training data $\mathscr{D} = (\mathbf{V}, \mathbf{y})$, the initial step is to consider the following set of iterates known as Generalized Approximate Message Passing (GAMP) algorithm 7, where the *denoising functions* $(f_{\text{out},t}, f_{w,t})$ depend on the specific classifier of interest $t \in \{\text{bo}, \text{erm}, \text{eb}\}$, and are summarized in table B.1. The convenience of the GAMP is precisely to allow us to deal with classifiers of very different nature ($t \in \{\text{bo}, \text{eb}\}$ are defined by sampling, while $t = \text{erm}$ is a point-estimator) in an unified framework. Note that the GAMP algorithm 7 is close to the one in [202], with the important difference that the denoising functions $f_{w,t}$ is vector valued - a consequence of the fact that implicitly the classifiers of interest have non-separable priors. A second convenient property of GAMP is that in the high-dimensional limit of interest here, the statistics of the sequence of estimators $\hat{\theta}_t^{T,\text{amp}}, \hat{\mathbf{c}}_t^{T,\text{amp}}$ can be exactly tracked by a set of equations known as *state evolution*. Therefore, the key idea in the proof strategy is to show that the statistics of the iterates $\hat{\theta}_t^{T,\text{amp}}, \hat{\mathbf{c}}_t^{T,\text{amp}}$ (given by the state evolution equations) coincide with the statistics of the classifiers defined in Sec. 3.2.1. The state evolution for GAMP with non-separable priors

Classifier	$f_{out,t}(y,\omega,v)$	$f_{w,t}(\mathbf{b},\mathbf{A})$
$\hat{f}_{\text{erm}}$	$\text{prox}_{\log\sigma(y\times\cdot)}(\omega)$	$(\lambda\mathbf{I}_p+\mathbf{A})\mathbf{b}$
$\hat{f}_{\text{bo}}$	$\partial_\omega \log\int \mathbb{P}(y=1\|z)\mathcal{N}(z\|\omega,v)\mathrm{d}z$	$(\Sigma_\star^{-1}+\mathbf{A})\mathbf{b}$
$\hat{f}_{\text{eb}}$	$\partial_\omega \log\int \sigma(\beta y\times z)\mathcal{N}(z\|\omega,v)\mathrm{d}z$	$(\lambda\mathbf{I}_p+\mathbf{A})\mathbf{b}$

Table B.1: GAMP denoising functions for the ERM, Bayes-optimal and empirical Bayes estimators. We recall that the covariance matrix is given by $\Sigma_\star = \Omega^{-1}\Phi^\top\Phi\Omega^{-1}$.

was rigorously derived in [35, 97]. Therefore, in the following we limit ourselves to an informal but intuitive derivation. In Sec. B.2.3 we show that the state evolution for the GAMP estimators indeed coincides with the fixed-point equations describing the statistics of the classifiers of interest according to the replica method. The fact that GAMP (rigorous) state evolution equations corresponds to the replica saddle-point equations is a very general fact [247], which is at the roots of many rigorous proofs to the replica predictions.

In the limit where $n,p\to\infty$ with fixed $\alpha = {}^{n}/_{p}$, it can be shown that the GAMP algorithm 7 is asymptotically equivalent to the following rBP equations (this is discussed in for instance [14, 15]):

$$\begin{cases} \omega^T_{\mu\to i} = \sum_{j\neq i} v^\mu_j \hat{\theta}^T_{j\to\mu} \\ V^T_{\mu\to i} = \sum_{j\neq i} (v^\mu_j)^2 \hat{c}^T_{j\to\mu} \end{cases}, \qquad \begin{cases} g^T_{\mu\to i} = f_{\text{out},t}(y^\mu, \omega^T_{\mu\to i}, V^T_{\mu\to i}) \\ \partial g^T_{\mu\to i} = \partial_\omega f_{\text{out},t}(y^\mu, \omega^T_{\mu\to i}, V^T_{\mu\to i}) \end{cases} \tag{B.28}$$

$$\begin{cases} b^T_{\mu\to i} = \sum_{\nu\neq\mu} v^\nu_i g^T_{\nu\to i} \\ A^T_{\mu\to i} = -\sum_{\nu\neq\mu} (v^\nu_i)^2 \partial g^T_{\nu\to i} \end{cases}, \qquad \begin{cases} \hat{\theta}^{T+1}_{i\to\mu} = f_{w,t}(b^T_{i\to\mu}, A^T_{i\to\mu}) \\ \hat{c}^{T+1}_{i\to\mu} = \partial_b f_{w,t}(b^T_{\mu\to i}, A^T_{\mu\to i}) \end{cases} \tag{B.29}$$

where we recall the reader $i\in[p]$, $\mu\in[n]$, and to lighten notation we have dropped the indexes $t\in\{\text{bo},\text{erm},\text{eb}\}$ for the classifier and $^{\text{amp}}$ which stresses that the messages concern GAMP estimators. By construction, the rBP messages are independent, and are only coupled to each other through the data, which we recall is given by:

$$y^\mu \sim P_0(\cdot|\mathbf{w}_\star^\top\mathbf{v}^\mu), \qquad \mathbf{v}^\mu\sim\mathcal{N}(0,\mathbf{I}_p), \qquad \mathbf{w}_\star\sim\mathcal{N}(\mathbf{0},\Sigma_\star) \tag{B.30}$$

For convenience, we define the so-called *teacher local field*:

$$z_\mu = \mathbf{w}_\star^\top\mathbf{v}^\mu/\sqrt{d} \tag{B.31}$$

Without loss of generality, we can write $y^\mu = f_0(z_\mu,\eta^\mu)$ for $\eta^\mu\sim\mathcal{N}(0,1)$. We now characterize the joint statistics of the rBP messages.

Step 1: Asymptotic joint distribution of $(z_\mu, \omega^T_{\mu\to i})$

Note that $(z_\mu, \omega^T_{\mu\to i})$ are given by a sum of independent random variables with variance $p^{-1/2}$, and therefore by the Central Limit Theorem in the limit $p\to\infty$ they are asymptotically Gaussian. Therefore we only need to compute their means, variances and cross correlation. The means are straightforward, since v^μ_i have mean zero and therefore they will also have mean zero. The variances are given by:

$$\mathbb{E}\left[z^2_\mu\right] = \frac{1}{d}\mathbb{E}\left[\sum_{i=1}^{p}\sum_{j=1}^{p} v^\mu_i v^\mu_j w_{\star i} w_{\star j}\right] = \frac{1}{d}\sum_{i=1}^{p}\sum_{j=1}^{p}\mathbb{E}\left[v^\mu_i v^\mu_j\right] w_{\star i} w_{\star j} = \frac{1}{d}\sum_{i=1}^{p}\sum_{j=1}^{p}\delta_{ij} w_{\star i} w_{\star j}$$
$$\underset{p\to\infty}{\to} \rho \tag{B.32}$$

$$\mathbb{E}\left[\left(\omega^T_{\mu\to i}\right)^2\right] = \frac{1}{p}\mathbb{E}\left[\sum_{j\neq i}^{p}\sum_{k\neq i}^{p} v^\mu_j v^\mu_k \hat\theta^T_{j\to\mu}\hat\theta^T_{k\to\mu}\right] = \frac{1}{p}\sum_{j\neq i}^{p}\sum_{k\neq i}^{p}\mathbb{E}\left[v^\mu_j v^\mu_k\right]\hat\theta^T_{j\to\mu}\hat\theta^T_{k\to\mu}$$
$$= \frac{1}{p}\sum_{j\neq i}^{p}\sum_{k\neq i}^{p}\delta_{jk}\hat\theta^T_{j\to\mu}\hat\theta^T_{k\to\mu} \underset{p\to\infty}{\to} q^T \tag{B.33}$$

$$\mathbb{E}\left[z_\mu\omega^T_{\mu\to i}\right] = \frac{1}{\sqrt{dp}}\mathbb{E}\left[\sum_{j\neq i}^{p}\sum_{k=1}^{p} v^\mu_j v^\mu_k \hat\theta^T_{j\to\mu} w_{\star k}\right] = \frac{1}{\sqrt{dp}}\sum_{j\neq i}^{p}\sum_{k=1}^{p}\mathbb{E}\left[v^\mu_j v^\mu_k\right]\hat\theta^T_{j\to\mu} w_{\star k}$$
$$= \frac{1}{\sqrt{dp}}\sum_{j\neq i}^{p}\sum_{k=1}^{p}\delta_{jk}\hat\theta^T_{j\to\mu} w_{\star k} \underset{p\to\infty}{\to} m^T \tag{B.34}$$

$$\tag{B.35}$$

where we have used that $\hat\theta^T_{i\to\mu} = O(p^{-1/2})$ to simplify the sums at large p. Summarising our findings:

$$(z_\mu, \omega^T_{\mu\to i}) \sim \mathcal{N}\left(\mathbf{0}_3, \begin{bmatrix} \rho & m^T \\ m^T & q^T \end{bmatrix}\right) \tag{B.36}$$

with:

$$\rho \equiv \frac{1}{d}\mathbf{w}^\top_\star\mathbf{w}_\star, \qquad q^T \equiv \frac{1}{p}(\hat\theta^T_t)^\top\hat\theta^T_t, \qquad m^T \equiv \frac{1}{\sqrt{dp}}(\hat\theta^T_t)^\top\mathbf{w}_\star \tag{B.37}$$

Step 2: Concentration of variances $V^T_{\mu\to i}$

Since the variance $V^T_{\mu\to i}$ depends on $(\mathbf{v}^\mu_i)^2$, in the asymptotic limit $p\to\infty$ it concentrates around its mean :

$$\mathbb{E}\left[V^T_{\mu\to i}\right] = \frac{1}{p}\sum_{j\neq i}\mathbb{E}\left[\left(v^\mu_i\right)^2\right]\hat c^T_{j\to\mu} = \frac{1}{p}\sum_{j\neq i}\hat c^T_{j\to\mu} = \frac{1}{p}\sum_{j=1}^{p}\hat c^T_{j\to\mu} - \frac{1}{p}\hat c^T_{i\to\mu} \underset{p\to\infty}{\to} V^T \equiv \frac{1}{p}\sum_{j=1}^{p}\hat c^T_j \tag{B.38}$$

where we have defined the variance overlap V^T. We thus have $V^T_{\mu\to i} \to V^T$. Note that V^T corresponds to the divergence with respect to $\mathbf{b}$ of $\log \mathcal{Z}_w(\mathbf{b},\mathbf{A})$.

Step 3: Distribution of $b^T_{\mu\to i}, \tilde{b}^T_{\mu\to i}$

By definition, we have

$$b^T_{\mu\to i} = \frac{1}{\sqrt{p}} \sum_{\nu\neq\mu} v^\nu_i g^T_{\nu\to i} = \frac{1}{\sqrt{p}} \sum_{\nu\neq\mu} v^\nu_i f_{\text{out}}(y^\mu, \omega^T_{\nu\to i}, V^T_{\nu\to i}) = \frac{1}{\sqrt{p}} \sum_{\nu\neq\mu} v^\nu_i f_{\text{out}}(f_0(z_\nu,\eta^\nu), \omega^T_{\nu\to i}, V^T_{\nu\to i}) \tag{B.39}$$

Note that in the sum $z_\nu = \frac{1}{\sqrt{d}} \sum_{j=1}^{p} v^\nu_j w_{\star j}$ there is a term $i = j$, and therefore z_μ is correlated with v^ν_i. To make this explicit, we split the teacher local field:

$$z_\mu = \frac{1}{\sqrt{d}} \sum_{j=1}^{p} v^\mu_j w_{\star j} = \underbrace{\frac{1}{\sqrt{d}} \sum_{j\neq i} v^\mu_j w_{\star j}}_{z_{\mu\to i}} + \frac{1}{\sqrt{d}} v^\mu_i w_{\star i} \tag{B.40}$$

and note that $z_{\mu\to i} = O(1)$ is independent from v^ν_i. Since $v^\mu_i w_{\star i} = O(p^{-1/2})$, to take the average at leading order, we can expand the denoising function:

$$\begin{aligned} f_{\text{out}}(f_0(z_\mu,\eta^\nu), \omega^T_{\nu\to i}, V^T_{\nu\to i}) = f_{\text{out}}(f_0(z_{\nu\to i},\eta^\nu), \omega^T_{\nu\to i}, V^T_{\nu\to i}) \\ + \frac{1}{\sqrt{d}} \partial_z f_{\text{out}}(f_0(z_{\nu\to i},\eta^\nu), \omega^T_{\nu\to i}, V^T_{\nu\to i}) v^\nu_i w_{\star i} + O(p^{-1}) \end{aligned} \tag{B.41}$$

Inserting in the expression for $b^T_{\mu\to i}$,

$$\begin{aligned} b^T_{\mu\to i} = \frac{1}{\sqrt{p}} \sum_{\nu\neq\mu} v^\nu_i f_{\text{out}}(f_0(z_{\nu\to i},\eta^\nu), \omega^T_{\nu\to i}, V^T_{\nu\to i}) \\ + \frac{1}{\sqrt{dp}} \sum_{\nu\neq\mu} (v^\nu_i)^2 \partial_z f_{\text{out}}(f_0(z_{\nu\to i},\eta^\nu), \omega^T_{\nu\to i}, V^T_{\nu\to i}) w_{\star i} + O(p^{-3/2}) \end{aligned} \tag{B.42}$$

Therefore:

$$\begin{aligned} \mathbb{E}\left[b^T_{\mu\to i}\right] &= \frac{w_{\star i}}{\sqrt{dp}} \sum_{\nu\neq\mu} \partial_z f_{\text{out}}(f_0(z_{\nu\to i},\eta^\nu), \omega^T_{\nu\to i}, V^T_{\nu\to i}) + O(p^{-3/2}) \\ &= \frac{w_{\star i}}{\sqrt{dp}} \sum_{\nu=1}^{n} \partial_z f_{\text{out}}(f_0(z_{\nu\to i},\eta^\nu), \omega^T_{\nu\to i}, V^T_{\nu\to i}) + O(p^{-3/2}) \end{aligned} \tag{B.43}$$

Note that as $p \to \infty$, for fixed t and for all ν, the fields $(z_{\nu\to i}, \omega^T_{\nu\to i})$ are identically distributed according to average in eq. (B.36). Therefore,

$$\frac{1}{\sqrt{dp}} \sum_{\nu=1}^{n} \partial_z f_{\text{out}}(f_0(z_{\nu\to i}, \eta^\nu), \omega^T_{\nu\to i}, V^T_{\nu\to i}) \underset{p\to\infty}{\to} \alpha\sqrt{\gamma}\, \mathbb{E}_{(\omega,z),\eta}\left[\partial_z f_{\text{out}}(f_0(z,\eta),\omega,V^T)\right] \equiv \hat{m}^T \tag{B.44}$$

so:

$$\mathbb{E}\left[b^T_{\mu\to i}\right] \underset{p\to\infty}{\to} w_{\star i}\hat{m}^T. \tag{B.45}$$

Similarly, the variance is given by:

$$\begin{aligned}
&\text{Var}\left[b^T_{\mu\to i}\right] && \text{(B.46)}\\
&= \frac{1}{p}\sum_{\nu\neq\mu}\sum_{\kappa\neq\mu} \mathbb{E}\left[v^\nu_i v^\kappa_i\right] f_{\text{out}}(f_0(z_{\nu\to i},\eta^\nu),\omega^T_{\nu\to i},V^T_{\nu\to i}) f_{\text{out}}(f_0(z_{\kappa\to i},\eta^\kappa),\omega^T_{\kappa\to i},V^T_{\kappa\to i}) + O(d^{-2})\\
&= \frac{1}{p}\sum_{\nu\neq\mu} f_{\text{out}}(f_0(z_{\nu\to i},\eta^\nu),\omega^T_{\nu\to i},V^T_{\nu\to i})^2 + O(p^{-2})\\
&= \frac{1}{p}\sum_{\nu=1}^{n} f_{\text{out}}(f_0(z_{\nu\to i},\eta^\nu),\omega^T_{\nu\to i},V^T_{\nu\to i})^2 + O(p^{-2})\\
&\underset{d\to\infty}{\to} \alpha\, \mathbb{E}_{(z,\omega),\xi}\left[f_{\text{out}}(f_0(z,\eta),\omega,V^T)^2\right] \equiv \hat{q}^T && \text{(B.47)}
\end{aligned}$$

To summarise, we have:

$$b^T_{\mu\to i} \sim \mathcal{N}\left(w_{\star i}\hat{m}^T, \hat{q}^T\right) \tag{B.48}$$

Step 4: Concentration of $A^T_{\mu\to i}, \tilde{A}^T_{\mu\to i}$

The only missing piece is to determine the distribution of the prior variances $A^T_{\mu\to i}, \tilde{A}^T_{\mu\to i}$. Similar to the previous variance, they concentrate:

$$\begin{aligned}
A^T_{\mu\to i} &= -\frac{1}{p}\sum_{\nu\neq\mu} (v^\nu_i)^2 \partial_\omega f_{\text{out},t}(y^\nu, \omega^T_{\nu\to i}, V^T_{\nu\to i}) && \text{(B.49)}\\
&= -\frac{1}{p}\sum_{\nu\neq\mu} (x^\nu_i)^2 \partial_\omega f_{\text{out},t}(f_0(z_{\nu\to i},\eta^\nu),\omega^T_{\nu\to i},V^T_{\nu\to i}) + O(p^{-3/2})\\
&= -\frac{1}{p}\sum_{\nu=1} \partial_\omega f_{\text{out},t}(f_0(z_{\nu\to i},\eta^\nu),\omega^T_{\nu\to i},V^T_{\nu\to i}) + O(d^{-3/2})\\
&\underset{p\to\infty}{\to} -\alpha\, \mathbb{E}_{(z,\omega),\xi}\left[\partial_\omega f_{\text{out},t}(f_0(z,\eta),\omega,V^T)\right] \equiv \hat{v}^T && \text{(B.50)}
\end{aligned}$$

Summary

We now have all the ingredients we need to characterize the asymptotic distribution of the GAMP iterates for any of the classifiers $t \in \{\text{bo}, \text{erm}, \text{eb}\}$:

$$\hat{\theta}_t^{T,\text{amp}} \sim f_{\text{out},t}(\mathbf{w}_\star \hat{m}_t^{T,\text{amp}} + \sqrt{\hat{q}_t^{T,\text{amp}}}\xi, \hat{v}_t^{T,\text{amp}}) \tag{B.51}$$

where $\xi \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_p)$ is an independent Gaussian vector. Therefore, we recover the GAMP state evolution equations [202, 35] for the overlaps:

$$\begin{cases} V^{T+1,\text{amp}} &= \mathbb{E}_{(\mathbf{w}_\star,\xi)}\left[\partial_{\mathbf{b}} \cdot f_{w,t}(\hat{m}^{T,\text{amp}}\mathbf{w}_\star + \sqrt{\hat{q}^{T,\text{amp}}}\xi, \hat{v}^{T,\text{amp}}\mathbf{I}_p)\right] \\ q^{T+1,\text{amp}} &= \mathbb{E}_{(\mathbf{w}_\star,\xi)}\left[f_{w,t}(\hat{m}^{T,\text{amp}}\mathbf{w}_\star + \sqrt{\hat{q}^{T,\text{amp}}}\xi, \hat{v}^T\mathbf{I}_p)^2\right] \\ m^{T+1,\text{amp}} &= \sqrt{\gamma}\mathbb{E}_{(\mathbf{w}_\star,\xi)}\left[f_{w,t}(\hat{m}^{T,\text{amp}}\mathbf{w}_\star + \sqrt{\hat{q}^{T,\text{amp}}}\xi, \hat{v}^{T,\text{amp}}\mathbf{I}_p)^\top \mathbf{w}_\star\right] \end{cases}, \tag{B.52}$$

$$\begin{cases} \hat{v}^{T,\text{amp}} &= -\alpha\mathbb{E}_{(z,\omega),\eta}\left[\partial_\omega f_{\text{out},t}(f_0(z,\eta),\omega,V^{T,\text{amp}})\right] \\ \hat{q}^{T,\text{amp}} &= \alpha\mathbb{E}_{(z,\omega),\eta}\left[f_{\text{out},t}(f_0(z,\eta),\omega,V^{T,\text{amp}})^2\right] \\ \hat{m}^{T,\text{amp}} &= \alpha\sqrt{\gamma}\mathbb{E}_{(z,\omega),\eta}\left[\partial_z f_{\text{out},t}(f_0(z,\eta),\omega,V^{T,\text{amp}})\right] \end{cases} \tag{B.53}$$

where $\mathbf{w}_\star \sim \mathcal{N}(\mathbf{0}, \Sigma_\star), \xi \sim \mathcal{N}(\mathbf{0}, I_p)$ and $(z,\omega) \sim \mathcal{N}(\mathbf{0}, \begin{bmatrix} \rho & m^T \\ m^T & q^T \end{bmatrix}), \eta \sim \mathcal{N}(0,1)$

Interestingly, we can show that the equations (B.53) are strictly equivalent to the self-consistent equations (3.14) of Theorem 3.3.1. Consider first the update equations for $V_t^{T,\text{amp}}, q_t^{T,\text{amp}}, m_t^{T,\text{amp}}$. First, note that for all the estimators considered here, the function $f_{w,t}$ has the form $(\Sigma_t^{-1} + \mathbf{A})\mathbf{b}$ for some matrix Σ_t. Now, let us introduce

$$\tilde{\Psi}(\mathbf{b}, \mathbf{A}, \Sigma) = \frac{1}{2p}\text{Tr}\left(\log\right)(\Sigma^{-1} + \mathbf{A}) + \frac{1}{2p}\mathbf{b}^\top(\Sigma^{-1} + \mathbf{A})^{-1}\mathbf{b} \tag{B.54}$$

$$\Psi_w(\hat{m}, \hat{q}, \hat{v}) = \mathbb{E}_{\mathbf{w}_\star,\xi}\tilde{\Psi}(\hat{m}\mathbf{w}_\star + \hat{q}\xi, \hat{v}\mathbf{I}_p) \tag{B.55}$$

With some algebra, we can see that for any estimator described in Table B.1 we have

$$\partial_{\hat{m}}\Psi_w(\hat{m}, \hat{q}, \hat{v}) = \mathbb{E}_{(\mathbf{w}_\star,\xi)}\left[f_{w,t}(\hat{m}^T\mathbf{w}_\star + \sqrt{\hat{q}^T}\xi, \hat{v}^T\mathbf{I}_p)^\top \mathbf{w}_\star\right] \tag{B.56}$$

$$\partial_{\hat{q}}\Psi_w - \partial_{\hat{v}}\Psi_w = \frac{1}{2}\mathbb{E}_{(\mathbf{w}_\star,\xi)}\left[f_{w,t}(\hat{m}^T\mathbf{w}_\star + \sqrt{\hat{q}^T}\xi, \hat{v}^T\mathbf{I}_p)^2\right] \tag{B.57}$$

$$\partial_{\hat{q}}\Psi_w = \frac{1}{2}\mathbb{E}_{(\mathbf{w}_\star,\xi)}\left[\partial_{\mathbf{b}} \cdot f_{w,t}(\hat{m}^T\mathbf{w}_\star + \sqrt{\hat{q}^T}\xi, \hat{v}^T\mathbf{I}_p)\right] \tag{B.58}$$

Thus the update equations (B.53) for m, q, v are equivalent to Equations (3.14). It is the same for

$\hat{m}, \hat{q}, \hat{v}$: consider the update equation for $\hat{q}^{T,\text{amp}}$. We can rewrite it with a Dirac delta

$$\hat{q}^{T,\text{amp}} = \alpha \sum_y \mathbb{E}_{(z,\omega),\eta} \left[f_{\text{out},t}(y, \omega, V^{T,\text{amp}})^2 \delta(y - f_0(z,\eta)) \right] \tag{B.59}$$

$$= \alpha \sum_y \mathbb{E}_{\omega} \left[f_{\text{out},t}(y, \omega, V^{T,\text{amp}})^2 \mathbb{E}_{z|\omega,\eta}(\delta(y - f_0(z,\eta)) \right] \tag{B.60}$$

$$\tag{B.61}$$

The distribution of z conditioned on ω is a Gaussian with mean $m^{T,\text{amp}}/q^{T,\text{amp}} \times \omega$ and variance $\rho - m^{T,\text{amp}} \times m^{T,\text{amp}}/q^{T,\text{amp}}$. Then, $\mathbb{E}_{z|\omega,\eta}(\delta(y - f_0(z,\eta))$ can be written

$$\begin{aligned}
\mathbb{E}_{z|\omega,\eta}(\delta(y - f_0(z,\eta)) &= \int \mathrm{d}z \mathbb{E}_{\eta}\left(\delta(y - f_0(z,\eta))\right) \mathcal{N}(z | m^{T,\text{amp}}/q^{T,\text{amp}} \times \omega, \rho - m^{T,\text{amp}} \times m^{T,\text{amp}}/q^{T,\text{amp}}) \\
&= \int \mathrm{d}z \mathbb{P}(y = 1|z) \mathcal{N}(z | m^{T,\text{amp}}/q^{T,\text{amp}} \times \omega, \rho - m^{T,\text{amp}} \times m^{T,\text{amp}}/q^{T,\text{amp}}) \\
&= \mathcal{Z}_0(y, m^{T,\text{amp}}/q^{T,\text{amp}} \times \omega, \rho - m^{T,\text{amp}} \times m^{T,\text{amp}}/q^{T,\text{amp}})
\end{aligned}$$

we thus recover the equation for $\hat{q}$ in (3.14) :

$$\hat{q}^{T+1} = \alpha \sum_y \mathbb{E}_{\omega} \left[f_{\text{out},t}(y, \omega, V^{T,\text{amp}})^2 \mathcal{Z}_0(y, m^T/q^T \omega, \rho - {}^{m^2}/_q) \right] \tag{B.62}$$

Similar computations can be done for $\hat{m}$ and $\hat{v}$.

We have thus eqs. (B.53) with the self-consistent equations (3.14) of Theorem 3.3.1. It remains to show two points. First, that in the particular case of the random feature model the expression of Ψ_w simplifies to Equation (3.15) - this is discussed in Appendix B.2.6. Second, to show that the fixed points of the state evolution equations $(m_t^{\text{amp}}, q_t^{\text{amp}}, v_t^{\text{amp}})$ indeed corresponds to the sufficient statistics $(m_t^\star, q_t^\star, v_t^\star)$ for the classifiers of interest. First, we provide a heuristic derivation of this fact, based on the replica method from statistical physics [171]. We defer the discussion of the formal aspects to Sec. B.2.4.

B.2.3 Self-consistent equation from the replica method

As discussed above, the goal of this section is to provide a derivation of the self-consistent equations in eq. (3.14) from the replica method. For the particular case of $\hat{f}_{\text{erm}}$, this derivation appeared [153], where it was also rigorously proven using CGMT. Here, we extend this analysis to $t \in \{\text{bo}, \text{eb}\}$.

We can treat the different classifiers of interest in the replica analysis by defining the following Gibbs distribution:

$$\mu_t(\theta|\mathscr{D}) = \frac{1}{\mathcal{Z}_t} \prod_{\mu \in [n]} P_\sigma^t(y^\mu | \theta^\top \mathbf{v}^\mu) \times P_\theta^t(\theta) \tag{B.63}$$

where (P_σ^t, P_θ^t) are a likelihood and priors (not necessarily normalized) depending on the particular classifier, and are explicitly given in Table B.2, and the normalization constant $\mathcal{Z}_t$ is the partition function.

Classifier	$P_\sigma^t(y\|z)$	$P_\theta^t(\theta)$
$\hat{f}_{\text{erm}}$	$\sigma(y\times z)^\beta$	$e^{-\beta\lambda/2\|\theta\|^2}$
$\hat{f}_{\text{eb}}$	$\sigma(\beta y\times z)$	$\mathcal{N}(\theta\|\mathbf{0}, {}^1/_\lambda I_p)$
$\hat{f}_{\text{bo}}$	$\sigma_{\tau^2+\tau^2_{\text{add}}}(y\times z)$	$\mathcal{N}(\theta\|\mathbf{0},\Sigma_\star)$

Table B.2: Prior and likelihood for the different estimators. For $\hat{f}_{\text{erm}}$, the temperature β must be taken in the limit $\beta\to\infty$, and the Gibbs measure $\mu_{\text{erm}}(\theta\,|\,\mathscr{D})$ is peaked around the minimizer of the empirical risk $\hat{\theta}_{\text{erm}}$.

The aim of the replica method is to compute the free energy density defined as:

$$\beta f_\beta = -\lim_{p\to\infty}\frac{1}{p}\mathbb{E}_{\mathscr{D}}\log\mathcal{Z}_t \tag{B.64}$$

The free energy is the cumulant generating function of the Gibbs measure, and therefore computing it give us access to the statistics of the measure, which in particular allow us to compute the test error and calibration (among others quantities) of the classifiers defined in Sec. 3.2.1. Since taking the expectation over the log is intractable, we resort to the *replica method* [171], which consists of the following trick:

$$\log\mathcal{Z}_t = \lim_{r\to 0^+}\frac{1}{r}\mathcal{Z}_t^r \tag{B.65}$$

Swapping the limit and the expectation, what we need to compute is:

$$\begin{aligned}\mathbb{E}_{\mathscr{D}}\mathcal{Z}_t^r &= \prod_{\mu=1}^n \mathbb{E}_{\mathbf{v}^\mu,y^\mu}\prod_{a=1}^r\int_{\mathbb{R}^d} P_\theta^t(\theta^a)\,P_\sigma^t\left(y^\mu|\frac{\mathbf{v}^{\mu\top}\theta^a}{\sqrt{d}}\right)\\ &= \prod_{\mu=1}^n\sum_y\int P(\mathbf{w}_\star)\int\left(\prod_a P_\theta^t(\theta^a)\right)\mathbb{E}_{\mathbf{v}^\mu}\left[P_0\left(y^\mu|\frac{\mathbf{v}^{\mu\top}\mathbf{w}_\star}{\sqrt{d}}\right)\prod_a P_\sigma^t\left(y^\mu|\frac{\mathbf{v}^{\mu\top}\theta^a}{\sqrt{p}}\right)\right]\end{aligned}$$

Next, we introduce the *local fields* $\nu_\star^\mu = \frac{1}{\sqrt{d}}\mathbf{v}^{\mu\top}\mathbf{w}_\star$ and $\nu_a^\mu = \frac{1}{\sqrt{d}}\mathbf{v}^{\mu\top}\theta^a$. Then, the term between brackets in the above equation is equal to

$$\int \mathrm{d}\nu_\star^\mu P_0(y^\mu|\nu_\star^\mu)\int\prod_a \mathrm{d}\nu_a^\mu P_\sigma^t(y^\mu|\nu_a^\mu)\mathbb{E}_{\mathbf{v}^\mu}\left[\delta(\nu_\star^\mu-\frac{\mathbf{v}^{\mu\top}\mathbf{w}_\star}{\sqrt{d}})\prod_a\delta(\nu_a^\mu-\frac{\mathbf{v}^{\mu\top}\theta^a}{\sqrt{p}}))\right] \tag{B.66}$$

Note that $\mathbb{E}_{\mathbf{v}^\mu}\left[\delta(\nu_\star^\mu-\mathbf{v}^{\mu\top}\mathbf{w}_\star)\prod_a\delta(\nu_a^\mu-\mathbf{v}^{\mu\top}\theta^a))\right]$ defines the joint distribution of the local fields.

It is straightforward to show that this is a Gaussian distribution on zero mean and covariance Σ_ν:

$$\mathbb{E}(\nu_\star,\nu_\star)=\frac{1}{d}\mathbf{w}_\star^\top\Omega\mathbf{w}_\star=\rho,\quad \mathbb{E}(\nu_\star,\nu_a)=\frac{1}{\sqrt{pd}}\mathbf{w}_\star^\top\Omega\theta^a=m^a,\quad \mathbb{E}(\nu_a,\nu_b)=\frac{1}{p}\theta^{a\top}\Omega\theta^b=Q^{ab} \tag{B.67}$$

Then, we have

$$\mathbb{E}\mathcal{Z}_t^r=\prod_\mu\sum_{y^\mu}\int P_{\theta,0}(\mathbf{w}_\star)\int\prod_a P_\theta^t(\theta^a)\int\mathrm{d}\nu_\star^\mu\prod_a\mathrm{d}\nu_a^\mu P_0(y^\mu|\nu_*^\mu)\prod_a P_\sigma^t(y^\mu|\nu_a^\mu)\times\mathcal{N}(\nu_*^\mu,\nu_a^\mu|\mathbf{0},\Sigma_\nu) \tag{B.68}$$

The elements of the covariance matrix Σ_ν are fixed by eq. (B.67). We can free these overlaps by doing the Fourier transform of the Dirac delta. We get in the end

$$\mathbb{E}_{\mathscr{D}}\mathcal{Z}_t^r\propto\int\mathrm{d}\rho\mathrm{d}\hat{\rho}\prod_a\mathrm{d}m^a\mathrm{d}\hat{m}^a\prod_{a,b}\mathrm{d}Q^{ab}\mathrm{d}\hat{Q}^{ab}e^{p\Phi(r)} \tag{B.69}$$

Where

$$\Phi(r)=-\frac{1}{\gamma}\rho\hat{\rho}-\frac{1}{\sqrt{\gamma}}\sum_a m^a\hat{m}^a-\sum_{a\leqslant b}Q^{ab}\hat{Q}^{ab}+\alpha\times\Psi_y^{(r)}+\Psi_w^{(r)} \tag{B.70}$$

$$\Psi_y^{(r)}=\frac{1}{p}\log\int P_{\theta,0}(\mathbf{w}_\star)\int\prod_a P_\theta^t(\theta^a)e^{\hat{\rho}\|\mathbf{w}_\star\|^2+\sum_a\hat{m}^a\mathbf{w}_\star\Omega\theta^a+\sum_{a\leqslant b}\hat{Q}^{a,b}\theta^a\Omega\theta^b} \tag{B.71}$$

$$\Psi_w^{(r)}=\frac{1}{p}\log\sum_y\int\mathrm{d}\nu_\star P_0(y|\nu_\star)\int\prod_a\mathrm{d}\nu_a P_\sigma^t(y|\nu_a)\mathcal{N}(\nu,\nu_a;\Sigma_\nu) \tag{B.72}$$

Replica symmetric ansatz

In the replica symmetric ansatz, we assume $m^a=m$, $Q^{ab}=q$ for $a\neq b$, $Q^{aa}=v+q$, $\hat{m}^a=\hat{m}$, $\hat{Q}^{ab}=\hat{q}$ for $a\neq b$, $\hat{Q}^{aa}=-\frac{1}{2}(\hat{v}-\hat{q})$ where the quantities $m,q,v,\hat{m},\hat{q},\hat{v}$ are to be determined.

We refer to [94, 13] for the detailed computation of $\lim_{r\to0^+}\Psi_y^{(r)}$ and $\lim_{r\to0^+}\Psi_w^{(r)}$. In the end,

we obtain :

$$\boxed{\begin{aligned} f_\beta &= \mathrm{extr}_{m,q,v,\hat{m},\hat{q},\hat{v}} \left\{ -\frac{1}{\sqrt{\gamma}} m\hat{m} + \frac{1}{2}(q\hat{v} - \hat{q}v + \hat{v}v) + \Psi_w + \alpha \times \Psi_y \right\} && \text{(B.73)} \\ \Psi_w &= \lim_{d\to\infty} \frac{1}{p} \mathbb{E}_{\xi,\mathbf{w}_\star} \log \int \mathrm{d}\theta P_\theta^t(\theta) e^{-\hat{v}^2\theta^\top \Omega\theta + \theta^\top(\hat{m}\Omega\mathbf{w}_\star + \hat{q}\Omega^{-1/2}\xi)} && \text{(B.74)} \\ \Psi_y &= \mathbb{E}_{\xi\sim\mathcal{N}(0,q)} \left[\sum_y \mathcal{Z}_0(y, {}^m/_q\xi, \rho - {}^{m^2}/_q) \log \mathcal{Z}_g(y,\xi,v) \right] && \text{(B.75)} \end{aligned}}$$

where

$$\mathcal{Z}_{0/g}(y,\omega,v) = \int \mathrm{d}z P_{0/g}(y|z)\mathcal{N}(z|\omega,v) \tag{B.76}$$

The self-consistent equations (3.14) are obtained by cancelling the derivative of the free energy with respect to each of $(m, q, v, \hat{m}, \hat{q}, \hat{v})$.

Ψ_w for Gaussian priors: For all the estimators considered here, the prior distribution $P_\theta^t(\theta)$ is Gaussian $\mathcal{N}(0,\Sigma)$, where Σ depends on the considered estimator. Then,

$$\begin{aligned} \Psi_w &= \int \mathrm{d}\theta e^{-\frac{1}{2}\theta^\top\Sigma^{-1}\theta} e^{-\frac{v}{2}\theta^\top\Omega\theta + \theta^\top(\hat{m}\Omega\mathbf{w}_\star + \sqrt{\hat{q}}\Omega^{-1/2}\xi)} && \text{(B.77)} \\ &= \frac{\exp\left(\frac{1}{2}(\hat{m}\mathbf{w}_\star + \sqrt{\hat{q}}\Omega^{-1/2}\xi)^\top(\Sigma+\hat{v}\Omega)^{-1}(\hat{m}\mathbf{w}_\star + \sqrt{\hat{q}}\Omega^{-1/2}\xi)\right)}{\sqrt{\det(\Sigma+\hat{v}\Omega)}} && \text{(B.78)} \\ &= \lim -\frac{1}{2p}\mathrm{Tr}\left(\log\right)(\Sigma+\hat{v}\Omega) + \frac{1}{2p}\mathrm{Tr}\left(()\,\hat{m}^2\Omega\mathbf{w}_\star\mathbf{w}_\star^\top\Omega + \hat{q}\Omega\right)(\Sigma+\hat{v}\Omega) && \text{(B.79)} \end{aligned}$$

We get in the end the following expression for Ψ_w

$$\Psi_w = -\frac{1}{2p}\mathrm{Tr}\left(\log\right)(\hat{v}\Omega+\Sigma) + \frac{1}{2p}\mathrm{Tr}\left(\left((\hat{m}^2\Omega\mathbf{w}_\star\mathbf{w}_\star^\top\Omega + \hat{q}\Omega)(\hat{v}\Omega+\Sigma)^{-1}\right)\right) \tag{B.80}$$

$$\tag{B.81}$$

Saddle-point equations: To compute the free energy, we cancel its derivative with respect to $m, q, v, \hat{m}, \hat{q}, \hat{v}$. We have :

$$\begin{cases} \partial_{\hat{m}} f_\beta &= -\frac{1}{\sqrt{\gamma}} m + \partial_{\hat{m}}\Psi_w \\ \partial_{\hat{q}} f_\beta &= -\frac{1}{2}v + \partial_{\hat{q}}\Psi_w \\ \partial_{\hat{v}} f_\beta &= \frac{1}{2}(v+q) + \partial_{\hat{v}}\Psi_w \end{cases} \tag{B.82}$$

Cancelling the derivatives gives the condition:

$$\boxed{\begin{cases} m = \sqrt{\gamma}\partial_{\hat{m}}\Psi_w \\ v = 2 \times \partial_{\hat{q}}\Psi_w \\ q = -v - 2 \times \partial_{\hat{v}}\Psi_w = 2 \times (\partial_{\hat{q}}\Psi_w - \partial_{\hat{v}}\Psi_w) \end{cases}} \tag{B.83}$$

Which are the first three equations of Theorem 3.3.1. The derivative of the free energy with respect to (m, q, v) is given by

$$\begin{cases} \partial_m f_\beta &= -\frac{1}{\sqrt{\gamma}}\hat{m} + \alpha\partial_m\Psi_y \\ \partial_q f_\beta &= \frac{1}{2}\hat{v} + \alpha\partial_q\Psi_y \\ \partial_v f_\beta &= \frac{1}{2}(\hat{v} - \hat{q}) + \alpha\partial_v\Psi_y \end{cases} \tag{B.84}$$

Cancelling the derivatives, and computing the derivatives of Ψ_y gives then

$$\boxed{\begin{cases} \hat{v} &= -\alpha\mathbb{E}_{\xi\sim\mathcal{N}(0,q)}\left[\sum_y \mathcal{Z}_0\left(y, {}^{m}/_{q}\xi, v_\star\right)\partial_\omega g_t\left(y,\xi,v\right)\right] \\ \hat{q} &= \alpha\mathbb{E}_{\xi\sim\mathcal{N}(0,q)}\left[\sum_y \mathcal{Z}_0\left(y, {}^{m}/_{q}\xi, v_\star\right) g_t\left(y,\xi,v\right)^2\right] \\ \hat{m} &= \alpha\sqrt{\gamma}\mathbb{E}_{\xi\sim\mathcal{N}(0,q)}\left[\sum_y \partial_\omega\mathcal{Z}_0\left(y, {}^{m}/_{q}\xi, v_\star\right) g_t\left(y,\xi,v\right)\right] \end{cases}} \tag{B.85}$$

which are the last three equations for $\hat{m}, \hat{q}, \hat{v}$ in Theorem 3.3.1.

Therefore, we have shown that the self-consistent equations characterizing the sufficient statistics in Theorem 3.3.1 can be obtained from the replica method by computing the asymptotic free energy density. Moreover, in Sec. B.2.2 we have shown that these equations exactly agree with the state evolution equations for a tailored GAMP algorithm 7.

B.2.4 Rigorous version of replica and self-consistent equations

As discussed in the introduction of this Appendix, the derivation of Theorem 3.3.1 consists in two steps. First, one constructs a tailored GAMP algorithm 7 for which the estimates can be exactly tracked by a set of state evolution equations. Second, one shows that these equations actually agree with the self-consistent equations describing the sufficient statistics for the joint density of interest in Theorem 3.3.1. The first part was discussed in Sec. B.2.2, and although we provided an informal derivation of the state evolution equations, they rigorously follow from the recent progress on state evolution proofs for structured message passing schemes with non-separable priors [35, 97]. For the second part, in Sec.B.2.3 we discussed a heuristic derivation of the self-consistent equations from the replica method, and showed it agrees with the state evolution equations from GAMP. Therefore, it remains to rigorously justify this last step. Thankfully, one can resort to a large number of recent progress on generic proofs of the replica predictions [27, 22, 219, 220, 178, 66, 153], which we now discuss in detail.

First, let us recall the statement of the theorem in the more general context of the Gaussian covariate model. Let $(\mathbf{u}, \mathbf{v})$ denote a pair of Gaussian covariates:

$$\begin{pmatrix}\mathbf{u}\\ \mathbf{v}\end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}_{d+p}, \begin{bmatrix}\Psi & \Phi\\ \Phi^\top & \Omega\end{bmatrix}\right). \tag{B.86}$$

For any of the classifiers $t \in \{\text{bo}, \text{erm}, \text{Lap}, \text{eb}\}$ from Sec. 3.2.1, the 2-dimensional vector $(f_\star(\mathbf{u}), \hat{f}_t(\mathbf{v}))$ is asymptotically distributed as $(\sigma(z), \sigma_{\tilde{v}}(z_t'))$ for some $\tilde{v}$ that depends on the estimator, where $(z, z_t') \sim \mathcal{N}(\mathbf{0}_2, \Sigma_t)$, and

$$\Sigma_t = \begin{pmatrix}\theta_\star^\top \Psi \theta_\star^\top & \hat{\theta}_t^\top \Phi \theta_\star\\ \theta_\star^\top \Phi^\top \hat{\theta}_t & \hat{\theta}_t^\top \Omega \hat{\theta}_t\end{pmatrix}$$

where $\hat{\theta}_t$ is either the unique minimizer the empirical risk in eq. (3.2) for $t \in \{\text{erm}, \text{Lap}\}$ or the mean over the respective posterior distribution for $t \in \{\text{bo}, \text{eb}\}$. The computation of $\rho_{\star,t}$ thus boils down to computing the sufficient statistics $(\hat{\theta}_t^\top \Phi \theta_\star, \hat{\theta}_t^\top \Omega \hat{\theta}_t)$. A first important point is that, asymptotically in p, these quantities converge in probability to single, deterministic quantities. This was shown in general for sampling problems with log concave measure (such as the one we use in the Bayes-optimal and empirical Bayes method) in [21], and for empirical risk minimization with convex risks in [153]. We shall thus use the following lemma:

Lemma B.2.1 (Overlap Concentration, from [21, 153])**.** *In the asymptotic limit $p \to \infty$, the random variables $(\hat{\theta}_t^\top \Phi \theta_\star, \hat{\theta}_t^\top \Omega \hat{\theta}_t)$ converge in probability to some value $(m_t^\star, q_t^\star)$ for $t \in \{\text{bo}, \text{erm}, \text{Lap}, \text{eb}\}$.*

The problem is thus reduced to the computation of these statistics, as a function of the parameters of the problems (α, γ, τ_0, etc.) for each of the estimators of interest. In Theorem 3.3.1, we claim that these are given by the replica equations derived in Appendix B.2.3. Thankfully, for different estimators these equations were proven in the literature in slightly different contexts, written as formal proofs of replica predictions.

- For $\hat{f}_{\text{erm}}$ on the random features model the self-consistent equations for $(m_{\text{erm}}^\star, q_{\text{erm}}^\star)$ were heuristically derived in [94] and rigorously proven in [66]. In the more general context of the Gaussian covariate model, analogous equations were proven in [153]. In both cases, they agree with our equations in eq. (3.14). While these works use the Gordon minimax approach to prove these equations, we note an independent GAMP-based proof for both the random features and Gaussian covariate models appeared in [152], leveraging recent progress on structured message passing schemes from [97].

- As noted in Sec. 3.2.1, the average over the Laplace posterior agrees exactly with the empirical risk minimizer:

$$p_{\text{Lap}}(\theta|\mathcal{D}) = \mathcal{N}(\theta|\hat{\theta}_{\text{erm}}, \mathcal{H}^{-1}) \tag{B.87}$$

Therefore, the self-consistent equations for $(m^\star_{\text{Lap}}, q^\star_{\text{Lap}})$ agree exactly with the ones for $(m^\star_{\text{erm}}, q^\star_{\text{erm}})$. Therefore, they are also rigorous.

In both cases, our result follows from:

Theorem B.2.2 (ERM statistics, Thms. 4 & 5 from [153], Informal)**.** *In the setting of Theorem 3.3.1, the ERM predictions from the replica are correct:* (m, q, v) *converges in probability to their replica fixed points* $(m^\star_{\text{erm}}, q^\star_{\text{erm}}, v^\star_{\text{erm}})$*, while the minimum training error converges in probability to the replica free energy density.*

- The "finite temperature" sampling problems related to Bayesian estimation pose different challenges. We start by discussing the Bayes-optimal $\hat{f}_{\text{bo}}$ classifier. For i.i.d. Gaussian data, the rigour of the replica prediction has been proven for Generalized linear models in [22], together with the GAMP optimality. Thanks to Gaussian equivalence, our problem can be framed as a Bayesian generalized linear reconstruction problem, but with data matrix that are instead correlated. In the random features case, the data matrix is a product of two random matrix (see eq.(B.10)). Thus, in this case replica predictions were for the Bayes-optimal problem was rigorously proven in [85, 24]. Note that while these works only prove the correctness of the replica free energy density, the techniques in [22] can be readily applied to generalize the proof to overlaps:

  Theorem B.2.3 (BO statistics, Th. 1 from [24] and Th. 1 from [24], Informal)**.** *In the setting of Theorem 3.3.1, the BO prediction from the replica is correct:* (m, q, v) *converges in probability to their replica fixed points* $(m^\star_{\text{bo}}, q^\star_{\text{bo}}, v^\star_{\text{bo}})$*, while the minimum training error converges in probability to the replica free energy density.*

  Additionally, given that the performance of the GAMP algorithms follows the same self-consistent equations as the replica's [35, 97], it follows that GAMP performs Bayes-optimal estimation for this problem, a classical property in Bayesian estimation [247]. [1]

- The remaining case is the empirical Bayes (EM) classifier $\hat{f}_{\text{eb}}$. In this case, where Bayesian estimation is performed *with mismatched noise*, the complete proof of the replica equation is not available in the literature. In principle, this can be done following the steps of [23] for the square loss (recall we consider the logistic loss in this work). Indeed, [23] shows how the concentration of $(m^\star, q^\star)$ (referred to as strong replica symmetry in [21]) can be used together with rigorous control of the cavity method [6] to prove the cavity equations. While this is, we believe, a worthwhile direction of research, we instead shall redefine the empirical Bayes method performance as the one of the *best empirical Bayesian estimator* in linear time, that is, the estimator achieved by the GAMP algorithm 7 with the corresponding empirical Bayes denoiser. It can indeed be shown that GAMP is the best *first-order algorithm* for this class of Bayesian estimation problems [44], and it is widely expected to perform an exact sampling for these problems [247] (as it was proven in for the Bayes-optimal case). With this definition, the

[1] Note that this crucially relies on the strong replica symmetry [21] condition, which impose the existence of an unique fixed point in our problem. Without this property, one could generically have more than a fixed point, associated to a so-called "hard phase" where GAMP is not optimal, see [247].

performance of GAMP is by construction given by its rigorous state evolution [35, 97], which we recall the reader matches the replica prediction.

B.2.5 Laplace approximation : computing the inverse Hessian

In this section, we show how to compute the prediction for the Laplace approximation

$$\hat{f}_{\text{Lap}}(\mathbf{v}) = \int \mathrm{d}z\sigma(z)\mathcal{N}(z|\hat{\theta}_{\text{erm}}^{\top}\mathbf{v}, \mathbf{v}^{\top}\mathscr{H}^{-1}\mathbf{v}) \tag{B.88}$$

with $\mathscr{H}$ the Hessian of the empirical risk at $\hat{\theta}_{\text{erm}}$. Note that in the high-dimensional limit, $\mathbf{v}^{\top}\mathscr{H}^{-1}\mathbf{v} \to_{p\to\infty} \mathrm{Tr}\,(()\,\mathscr{H}^{-1}\Omega)$. As shown in Appendix B.3, to compute this quantity we can add the term $\mathbf{h}^{\top}\theta$ to the loss and compute the second derivative of the free energy density with respect to $\mathbf{h}$. The computations are the same as those done in Section B.2.3, except that the Gibbs distribution $\mu_t(\theta)$ is replaced by

$$\mu_t(\theta) = \frac{1}{\mathcal{Z}_t(\mathbf{h})}\prod_i P_{\sigma}^t(y_i|\theta^{\top}\varphi(\mathbf{x}_i)) \times P_{\theta}^t(\theta) \times e^{\beta\mathbf{h}^{\top}\theta} \tag{B.89}$$

Adapting the derivation from Sec. B.2.3 for $\hat{f}_{\text{erm}}$ and taking the temperature $\beta \to \infty$ and get as before

$$f_0 := \lim_{\beta\to\infty} f_{\beta} = \underset{m,q,v,\hat{m},\hat{q},\hat{v}}{\mathbf{extr}} \left\{ -\frac{1}{\sqrt{\gamma}} m\hat{m} + \frac{1}{2}(q\hat{v} - \hat{q}v) + \Psi_w(\hat{m},\hat{q},\hat{v},\mathbf{h}) + \alpha\Psi_y(m,q,v) \right\} \tag{B.90}$$

$$\Psi_y = \mathbb{E}_{\xi\sim\mathcal{N}(0,q)}\left[\sum_y \mathcal{Z}_0(y, {}^{m}/_{q}\xi, \rho - {}^{m^2}/_{q}) \log \mathcal{Z}_g(y,\xi,v) \right] \tag{B.91}$$

However, now Ψ_w is

$$\Psi_w = -\frac{1}{2p}\mathrm{Tr}\big(\log\big)\,(\hat{v}\Omega + \Sigma) + \frac{1}{2p}\mathrm{Tr}\big(\big[((\hat{m}\Omega\mathbf{w}_{\star} + \mathbf{h})(\hat{v}\Omega + \lambda I)^{-1}(\hat{m}\Omega\mathbf{w}_{\star} + \mathbf{h}) + \hat{q}\Omega(\hat{v}\Omega + \Sigma)^{-1})\big]\big) \tag{B.92}$$

The second derivative of Ψ_w with respect to $\mathbf{h}$ is $(\lambda I_d + \hat{v}\Omega)^{-1}$. As a consequence, the second derivative of the free energy

$$(\nabla_{\mathbf{h}}^2 \log \mathcal{Z}_{\text{erm}})_{|\mathbf{h}=\mathbf{0}} = \nabla_{\mathbf{h}}^2 \Psi_w(m_{\text{erm}}^{\star}, q_{\text{erm}}^{\star}, v_{\text{erm}}^{\star}, \mathbf{h}) = (\lambda I_d + \hat{v}_{\text{erm}}^{\star}\Omega)^{-1}$$

and $\nabla_{\mathbf{h}}^2 f_0 = -(\lambda I_d + \hat{v}_{\text{erm}}^{\star}\Omega)^{-1}$. We then deduce that the inverse Hessian is equal to

$$\mathscr{H}^{-1} = \left(\lambda\mathbf{I}_d + \hat{v}_{\text{erm}}^{\star}\Omega\right)^{-1} \tag{B.93}$$

B.2.6 Simplification for random features

As discussed in Appendix B.2, the random features model $\varphi(\mathbf{x}) = \phi(\mathbf{F}\mathbf{x})$ is asymptotically equivalent to the Gaussian covariate model up to an identification of the covariances:

$$\Omega = \kappa_1^2 FF^\top + \kappa_\star^2 \mathbf{I}_p, \qquad \Phi = \kappa_1 F, \qquad \Psi = \mathbf{I}_d \tag{B.94}$$

where:

$$\kappa_1 = \mathbb{E}_{z\sim\mathcal{N}(0,1)}\left[\phi'(z)\right], \qquad \kappa_\star = \sqrt{\mathbb{E}_{z\sim\mathcal{N}(0,1)}\left[\phi(z)^2\right] - \kappa_1^2} \tag{B.95}$$

where for simplicity we assume $\kappa_0 = \mathbb{E}_{z\sim\mathcal{N}(0,1)}[\phi(z)] = 0$. Thus, in this case we can explicitly write:

$$\Sigma_\star = \frac{\kappa_1^2 \mathbf{F}\mathbf{F}^\top}{\kappa_\star^2 \mathbf{I}_p + \kappa_1^2 \mathbf{F}\mathbf{F}^\top}. \tag{B.96}$$

. Note that the matrices $\Omega, \Sigma, \Sigma_\star$ are diagonalizable in the same basis, since Σ is either a multiple of the identity, or a function of $\Omega, \Phi\Phi^\top$. Assuming that $\mathbf{F}\mathbf{F}^\top$ has an asymptotic spectral distribution μ, we can write Ψ_w directly in terms of an average over μ:

$$\Psi_w = \frac{1}{2}\mathbb{E}_{x\sim\mu)}\left[\log\left(\hat{v}(\kappa_1^2 x + \kappa_\star^2) + \pi(x)\right) + \left(\frac{\hat{m}^2 \frac{\kappa_1^2 x}{\kappa_1^2 x + \kappa_\star^2} + \hat{q}(\kappa_1^2 x + \kappa_\star^2)}{\hat{v}(\kappa_1^2 x + \kappa_\star^2) + \pi(x)}\right)\right] \tag{B.97}$$

where the function π represents the eigenvalues of Σ : since we can write $\Sigma = f(\Phi\Phi^\top)$ here, we have, $\pi(x) = f(x)$. For $\hat{f}_{\text{erm}}$ and $\hat{f}_{\text{eb}}$, $\pi(x) = \lambda$. For $\hat{f}_{\text{bo}}$, $\pi(x) = \frac{\kappa_1^2 x}{\kappa_1^2 x + \kappa_\star}$. This gives us the values of $\hat{\pi}_t$ in Table 3.1.

In particular, when $\mathbf{F}$ has Gaussian i.i.d. entries (as in all plots presented here), μ is simply the Marcenko-Pastur distribution with shape parameter γ.

B.2.7 Temperature scaling

In this section, we show how to compute the optimal temperature T that minimizes the test loss for $\hat{f}_{\text{erm}}$. Once the overlaps $m^\star, q^\star, v^\star, \hat{m}^\star, \hat{q}^\star, \hat{v}^\star$ are computed, we get the test loss with the expression

$$\mathscr{L}_{\text{gen.}}(m, q) = \sum_y \mathbb{E}_{\xi\sim\mathcal{N}(0,1)}\left[\mathscr{Z}_0(y, {}^{m^\star}/_{q^\star}\xi, \rho - {m^2}/{q}) \times (-\log\sigma(y \times \sqrt{q}\xi))\right] \tag{B.98}$$

Given a temperature T, temperature scaling will divide the weights such that the prediction is now $\sigma(\theta^\top \varphi(\mathbf{x}/T)$. It is easy to see that in this case, the overlaps $m^\star, q^\star$ now become $m^\star/T, q^\star/T^2$.

Then, temperature scaling amounts to finding

$$T^\star = \text{argmin}_T \mathscr{L}_{\text{gen.}}(m^\star / T, q^\star / T^2) \tag{B.99}$$

B.3 Confidence function and Hessian of Laplace method

B.3.1 Computing the Hessian of the training loss

In this section, we show how we can compute the (inverse of) the Hessian thanks to classical properties of Legendre transforms. We consider the ERM estimator $\hat{f}_{\text{erm}}$ trained by minimizing the following loss :

$$\mathscr{L}(\mathbf{w}) = -\sum_i \log\sigma(\theta^\top \varphi(\mathbf{x})_i \times y_i) + {}^{\lambda}/_{2}\|\theta\|^2 \tag{B.100}$$

whose Hessian at the minimum is given by

$$\mathscr{H} := \nabla^2 \mathscr{L} = -\sum_i (1-\sigma'(\theta^\top \varphi(\mathbf{x})_i \times y_i))\varphi(\mathbf{x})_i \varphi(\mathbf{x})_i^\top + \lambda I_d\big|_{\theta=\hat{\theta}_{\text{erm}}} \tag{B.101}$$

Our starting point to compute this Hessian is a very classical lemma in statistical mechanics, that uses the Legendre transform of the loss.

Lemma B.3.1 (Inverse Hessian from Legendre Transforms)**.** *We define the Legendre transform of the loss by adding a source term to the loss (an external field in the parlance of statistical mechanics)*

$$\mathscr{L}^L(\mathbf{h}) = \min_\theta \left[-\sum_i \log\sigma(y_i \theta^\top \varphi(\mathbf{x})_i) + {}^{\lambda}/_{2}\|\theta\|^2 + \mathbf{h}^\top\theta \right] = \min_\theta \left[\mathscr{L}(\theta) + \mathbf{h}^\top\theta\right] \tag{B.102}$$

then the Inverse of the Hessian (B.101) *is the Hessian of the Legendre transform* $\mathscr{L}(\mathbf{h})$

$$\mathscr{H}^{-1}(\hat{\theta}_{\text{erm}}) = -\frac{\partial^2 \mathscr{L}^L(\mathbf{h})}{\partial^2 \mathbf{h}}\Big|_{\mathbf{h}=0} \tag{B.103}$$

Proof This is a classical result from Legendre transform of strongly convex functions, which we informally recall. First notice that at the minimum of $\mathscr{L}(\theta) + \mathbf{h}^\top\theta$ over θ is characterized by

$$\frac{\partial \mathscr{L}(\theta)}{\partial \theta_j} + h_j = 0 \;\forall\, j \tag{B.104}$$

so that

$$\begin{aligned}
\frac{\partial \mathscr{L}^L(\mathbf{h})}{\partial h_i} &= \frac{\partial\left[\mathscr{L}(\theta) + \mathbf{h}^\top\theta\right]}{\partial h_i}\Big|_{\hat{\theta}_{\text{erm}}} = \sum_{j=1}^{p}\left[\frac{\partial \mathscr{L}(\theta)}{\partial \theta_j}\frac{\partial \theta_j}{\partial h_i} + h_j \frac{\partial \theta_j}{\partial h_i}\right]\Big|_{\hat{\theta}_{\text{erm}}} + \theta_i\big|_{\hat{\theta}_{\text{erm}}} \\
&= \sum_{j=1}^{p}\frac{\partial \theta_j}{\partial h_i}\left[\frac{\partial \mathscr{L}(\theta)}{\partial \theta_j} + h_j\right]\Big|_{\hat{\theta}_{\text{erm}}} + \theta_i\big|_{\hat{\theta}_{\text{erm}}} = \theta_{\text{erm},i}
\end{aligned} \tag{B.105}$$

It thus follows that

$$\frac{\partial^2 \mathscr{L}^L(\mathbf{h})}{\partial h_i \partial h_j} = \frac{\partial \theta_i}{\partial h_j}. \tag{B.106}$$

However, we have from eq.(B.104)

$$\frac{\partial^2 \mathscr{L}(\theta)}{\partial \theta_i \partial \theta_j} = -\frac{\partial h_j}{\partial \theta_i}. \tag{B.107}$$

Using both eqs (B.106) and (B.107) at $h = 0$ concludes the proof. □

Note that this relation is not asymptotic and is valid for a given instance of the problem. This lemma is however particularly practical in the large n limit, since an asymptotic expression for the loss $\mathscr{L}^L$ is known so that we can use it to obtain the asymptotic expression. Using the value of the minimal loss from [153, 152], we deduce, taking its second derivative, that for large n we must have (See Section B.2.5 for the derivation)

$$\mathscr{H}_{\text{rep.}}^{-1} = (\lambda \mathbf{I}_p + \hat{v}_{\text{erm}}^{\star} \Omega)^{-1} \tag{B.108}$$

Where $\hat{v}_{\text{erm}}^{\star}$ is the unique solution the following self-consistent equations:

$$\begin{cases} m &= \frac{\gamma \hat{m}}{p} \text{Tr}\left(\left(\Omega \mathbf{w}_\star \mathbf{w}_\star^\top \Omega (\lambda \mathbf{I}_p + \hat{v}\Omega)^{-1} \right) \right) \\ q &= \frac{1}{p} \text{Tr}\left(\left((\hat{q}\Omega + \hat{m}^2 \Omega \mathbf{w}_\star \mathbf{w}_\star^\top \Omega) \Omega (\lambda \mathbf{I}_p + \hat{v}\Omega)^{-2} \right) \right), \\ v &= \frac{1}{p} \text{Tr}\left(() \, \lambda \mathbf{I}_p + \hat{v}\Omega)^{-1} \Omega \right) \end{cases} \quad \begin{cases} \hat{v} &= -\alpha \mathbb{E}_{\xi \sim \mathcal{N}(0,q)} \left[\sum_y \mathcal{Z}_0\left(y, {}^m/_q \xi, v_\star\right) \partial_\omega f_{\text{out,erm}}\left(y, \xi, v \right. \right. \\ \hat{q} &= \alpha \mathbb{E}_{\xi \sim \mathcal{N}(0,q)} \left[\sum_y \mathcal{Z}_0\left(y, {}^m/_q \xi, v_\star\right) f_{\text{out,erm}}\left(y, \xi, v\right)^2 \right] \\ \hat{m} &= \alpha \mathbb{E}_{\xi \sim \mathcal{N}(0,q)} \left[\sum_y \partial_\omega \mathcal{Z}_0\left(y, {}^m/_q \xi, v_\star\right) f_{\text{out,erm}}\left(y, \xi, v\right) \right] \end{cases} \tag{B.109}$$

Note that the subscript emphasizes that this expression was obtained by differentiating the asymptotic free energy density.

It would be tempting to assume that the convergence of the free energy to its asymptotic value in [153] would also be valid for the second derivative, so that the replica Hessian would be close, pointwise, to the actual Hessian when $p \to \infty$. This, however, turns out to be wrong, as one can easily check in the ridge regression case. However, we conjecture that the limit holds in the sense of deterministic equivalents [103]. This leads us to the following conjecture:

Conjecture B.3.2 (Deterministic equivalent of the inverse Hessian)**.** *For any deterministic matrix $A \in \mathbb{R}^{p \times p}$, in the asymptotic limit where $n, d, p \to \infty$ at fixed ratios $\alpha = {}^n/_p$ and $\gamma = {}^d/_p$, we have:*

$$\lim_{p \to \infty} \frac{1}{p} \text{Tr}\left(A \mathscr{H}^{-1} \right) = \lim_{p \to \infty} \frac{1}{p} \text{Tr}\left[A (\lambda \mathbf{I}_p + \hat{v}_{\text{erm}}^{\star} \Omega)^{-1} \right] \tag{B.110}$$

For the purpose of characterizing the Laplace approximation, we apply this formula using $A = \mathbf{v}\mathbf{v}^\top$. Proving rigorously the convergence in the sense of deterministic equivalent remains an open problem. It is, however, easy to prove that it is valid for the square loss, as we show in the next section. Fortunately, it can be checked numerically to great precision that the conjecture is empirically satisfied for the study of Laplace method, as is also shown in the next section.

Additionally, we note that the statement is made in term of an expression of the inverse of the Hessian (which, conveniently, is actually what we want to know).

B.3.2 An instructive example: the square loss

Although in this work we only focus in the classification case, Conj. B.3.2 actually applies in the more general context of a convex loss function ℓ, for which the exact asymptotics was characterized in [153]. An instructive example is therefore given by looking at the square loss $\ell(y, x) = {}^1\!/_2(y - x)^2$, for which the Hessian is simply given by:

$$\mathscr{H} := \nabla^2 \mathscr{L} = V^\top V + \lambda I_p \tag{B.111}$$

where $V \in \mathbb{R}^{n \times p}$ is the feature matrix with rows given by $\varphi(\mathbf{x}^\mu) \in \mathbb{R}^p$ for $\mu \in [n]$. Therefore, it is independent of the minimizer $\hat{\theta}_{\text{erm}}$. In this case, Conj. B.3.2 boils down to the classical deterministic equivalents for the sample covariance matrix $\hat{\Omega}_n := V^\top V$. Deterministic equivalents for sample covariance matrices have been characterized under different levels of generality for V [18, 134, 151, 49, 209], including in particular the random features case $V = \sigma(FX^\top)$ with X i.i.d. Gaussian considered here. They state precisely that, for any deterministic matrix $A \in \mathbb{R}^{p \times p}$ and in the asymptotic limit considered here:

$$\lim_{p \to \infty} \frac{1}{p} \text{Tr}\left[A\left(V^\top V + \lambda I_p\right)^{-1}\right] = \lim_{p \to \infty} \frac{1}{p} \text{Tr}\left[A\left(\hat{v}^\star \Omega + \lambda I_p\right)^{-1}\right] \tag{B.112}$$

where $\Omega = \mathbb{E}\left[\sigma(F\mathbf{x})\sigma(F\mathbf{x})^\top\right]$ is the population covariance of the features and $\hat{v}^\star$ is the solution of the following self-consistent equation:

$$\frac{\alpha}{\hat{v}} - 1 = 1 - \lambda \int \frac{\mu_\Omega(\mathrm{d}t)}{\lambda + \hat{v}t} \tag{B.113}$$

with μ_Ω the asymptotic spectral density of Ω. It is not hard to check that this self-consistent equation (B.113) coincides exactly with the self-consistent equations (B.109) from [153] when ℓ is the square loss.

B.3.3 Comparison with numerics

In this section, we apply the computations of the previous section and show that they gives extremly good prediction even at very moderate sizes. In Figure B.1, we compare the theoretical

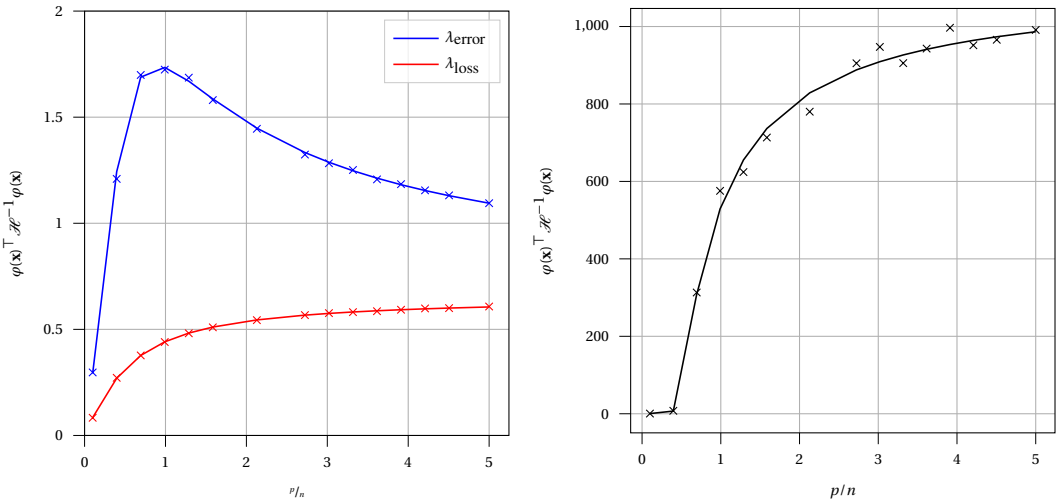


Figure B.1: (**Left**) Theoretical predictions (lines) and experimental values (crosses) of $\varphi(\mathbf{x})^\top \mathscr{H}^{-1} \varphi(\mathbf{x})$ with $n/d = 2, \tau^2 = 0.5$, $\varphi(\mathbf{x}) = \text{erf}(\mathbf{F}\mathbf{x})$ and $\mathbf{F}$ Gaussian, as in Figure 3.1, for λ_{error} and λ_{loss}. Experimental values are obtained by fixing $d = 256$. (**Right**) Theoretical and experimental values for $\lambda = 10^{-4}$.

value of $\varphi(\mathbf{x})^\top \mathscr{H}^{-1} \varphi(\mathbf{x})$ for $\varphi(\mathbf{x}) = \text{erf}(\mathbf{F}\mathbf{x})$ from eq. (B.108) and the one observed experimentally. Experiments are done by training the logistic classifier $\hat{f}_{\text{erm}}$ on training data $(\mathbf{x}^\mu, y^\mu)_{\mu \in [n]}$ and computing the Hessian (B.3.1) at the minimizer $\hat{\theta}_{\text{erm}}$. We observe a good fit between theory and experiment, validation our analysis.

B.4 Conditional variance of the Bayes-optimal estimator

In this section, we prove the expression of the variance of $\hat{f}_{\text{bo}}$ conditioned on the confidence of other estimators :

$$\text{Var}(\hat{f}_{\text{bo}}(\mathbf{x})|\hat{f}_t(\mathbf{x}) = \ell) = \int \mathrm{d}a \; \sigma_{\hat{y}^\star_{\text{bo}}+\tau^2+\tau^2_{\text{add}}}(a)^2 \times \mathcal{N}\left(a|^{m^\star_t}/_{q^\star_t}\sigma^{-1}_{\hat{\tau}_t}(\ell), q^\star_{\text{bo}} - {}^{m^{\star\,2}_t}/_{q^\star_t}\right) - (\ell - \Delta_\ell)^2 \tag{B.114}$$

The first step is to show that for any estimator $t \in \{\text{erm}, \text{eb}, \text{Lap}\}$, the joint density of the confidence of $\hat{f}_{\text{bo}}, \hat{f}$, defined as

$$\rho_{\text{bo},t}(a,b) = \mathbb{P}_{\mathbf{x}}(\hat{f}_{\text{bo}}(\mathbf{x}) = a, \hat{f}_t(\mathbf{x}) = b) \tag{B.115}$$

can be computed in the similar way as $\rho_{\star,t}$ in Theorem 3.3.1. This was shown previously for a simpler model in [53], where the teacher and input data have identity covariance.

Lemma B.4.1. *In the same setting as Theorem 3.3.1, in the asymptotic limit, the density* $\rho_{\text{bo},t}(a,b)$ *converges to* $\rho^{\lim}_{\text{bo},t}(a,b)$

$$\rho^{\lim}_{\text{bo},t}(a,b) = \frac{\mathcal{N}\left(\begin{bmatrix}\sigma^{-1}_{\tau^2+\tau^2_{\text{add}}}(a)\\ \sigma^{-1}_{\hat{\tau}^2_t}(b)\end{bmatrix}\middle|\mathbf{0}_2, \Sigma_{bo,t}\right)}{|\sigma'_{\tau^2+\tau^2_{\text{add}}}(\sigma^{-1}_{\tau^2+\tau^2_{\text{add}}}(a))||\sigma'_{\hat{\tau}^2_t}(\sigma^{-1}_{\hat{\tau}^2_t}(b))|} \tag{B.116}$$

where this time

$$\Sigma_{\text{bo},t} = \begin{bmatrix} q^\star_{\text{bo}} & m^\star_t \\ m^\star_t & q^\star_t \end{bmatrix} \tag{B.117}$$

To prove Lemma B.4.1, the main idea is to observe that, as with $f_\star$, to compute the density we need the covariance matrix

$$\frac{1}{d}\begin{pmatrix} \hat{\theta}^\top_{\text{bo}}\Omega\hat{\theta}_{\text{bo}} & \hat{\theta}^\top_{\text{bo}}\Omega\theta_t \\ \hat{\theta}^\top_{\text{bo}}\Omega\hat{\theta}_t & \hat{\theta}^\top_t\Omega\theta_t \end{pmatrix} \tag{B.118}$$

The diagonal terms are $q^\star_{\text{bo}}, q^\star_t$ respectively by definition. We then just need to compute the overlap $m_{\text{bo},t} = \frac{1}{d}\hat{\theta}^\top_{\text{bo}}\Omega\hat{\theta}_t$. Our goal is to prove that $m_{\text{bo},t} = m^\star_t$,

However, using the Nishimori identity from statistical physics, for any vector $\mathbf{z}(\mathscr{D})$ that can depend on the training data, we have

$$\mathbb{E}_{\mathscr{D}}\left(\hat{\theta}^\top_{\text{bo}}\mathbf{z}(\mathscr{D})\right) = \mathbb{E}_{\mathbf{w}_\star,\mathscr{D}}\left(\mathbf{w}^\top_\star\mathbf{z}(\mathscr{D})\right) \tag{B.119}$$

Equation B.119 is just an application of Bayes formula. In particular, if we take $\mathbf{z}(\mathscr{D}) = \hat{\theta}_t$, we obtain that

$$\mathbb{E}_{\mathscr{D}}(\hat{\theta}_{\text{bo}}^\top \hat{\theta}_t) = \mathbb{E}_{\mathbf{w}_\star,\mathscr{D}}(\mathbf{w}_\star^\top \hat{\theta}_t) \tag{B.120}$$

and we see that in expectation, $\mathbb{E}_{\mathscr{D}}(m_{\text{bo},t}) = \mathbb{E}_{\mathbf{w}_\star,\mathscr{D}}(m_t^\star)$. We already know that the right-hand side of the equality self-averages, i.e $\lim_{d\to\infty} \mathbb{E}_{\mathbf{w}_\star,\mathscr{D}}(m_t^\star) = m_t^\star$. It remains to show that the left-hand side also self-averages.

Lemma B.4.2 (Concentration of the overlap $m_{\text{bo},t}$)**.**

$$\lim_{d\to\infty} \mathbb{E}\left[\left(\frac{\hat{\theta}_{\text{bo}}^\top \theta_t}{d}\right)^2\right] = \lim_{d\to\infty} \mathbb{E}\left[\frac{\hat{\theta}_{\text{bo}}^\top \theta_t}{d}\right]^2 \tag{B.121}$$

Proof. The proof again uses Nishimori identity.

$$\begin{aligned}
\mathbb{E}\left[\left(\frac{\hat{\theta}_{\text{bo}}^\top \theta_t}{d}\right)^2\right] &= \mathbb{E}\left[\left(\frac{\hat{\theta}_{\text{bo}}^\top \theta_t}{d}\right)\left(\frac{\hat{\theta}_{\text{bo}}^\top \theta_t}{d}\right)\right] && \text{(B.122)}\\
&= \mathbb{E}_{\mathscr{D}}\left[\left(\frac{\mathbb{E}_{\hat{\theta}|\mathscr{D}}\hat{\theta}^\top \theta_t}{d}\right)\left(\frac{\mathbb{E}_{\hat{\theta}|\mathscr{D}}\hat{\theta}\cdot \theta_t}{d}\right)\right] && \text{(B.123)}\\
&= \mathbb{E}_{\mathscr{D}}\mathbb{E}_{\hat{\theta}_1,\hat{\theta}_2|\mathscr{D}}\left[\left(\frac{\hat{\theta}_1^\top \theta_t}{d}\right)\left(\frac{\hat{\theta}_2\cdot \theta_t}{d}\right)\right] && \text{(B.124)}\\
&= \mathbb{E}_{\mathscr{D},\mathbf{w}_\star}\left[\left(\frac{\mathbf{w}_\star^\top \theta_t}{d}\right)\left(\frac{\mathbb{E}_{\hat{\theta}|\mathscr{D}}\hat{\theta}\cdot \theta_t}{d}\right)\right] && \text{(B.125)}\\
&= \mathbb{E}_{\mathscr{D},\mathbf{w}_\star}\left[\left(\frac{\mathbf{w}_\star^\top \theta_t}{d}\right)\left(\frac{\mathbf{w}_{\text{bo}}^\top \theta_t}{d}\right)\right] && \text{(B.126)}
\end{aligned}$$

Then, from Cauchy-Schwartz we have

$$\begin{aligned}
\mathbb{E}\left[\left(\frac{\hat{\theta}_{\text{bo}}^\top \theta_t}{d}\right)^2\right]^2 &\leq \mathbb{E}\left[\left(\frac{\hat{\theta}_{\text{bo}}^\top \theta_t}{d}\right)^2\right]\mathbb{E}\left[\left(\frac{\mathbf{w}_\star^\top \theta_t}{d}\right)^2\right] && \text{(B.127)}\\
\mathbb{E}\left[\left(\frac{\hat{\theta}_{\text{bo}}^\top \theta_t}{d}\right)^2\right] &\leq \mathbb{E}\left[\left(\frac{\mathbf{w}_\star^\top \theta_t}{d}\right)^2\right] && \text{(B.128)}
\end{aligned}$$

and as $d \to \infty$, we can use the concentration of the right hand side to $m_t^\star$ to obtain

$$\lim_{d\to\infty} \mathbb{E}\left[\left(\frac{\hat{\theta}_{\text{bo}}^\top \theta_t}{d}\right)^2\right] \leq (m_t^\star)^2 \tag{B.129}$$

so that, given the second moment has to be larger or equal to its (squared) mean:

$$\lim_{d\to\infty} \mathbb{E}\left[\left(\frac{\hat{\theta}_{\rm bo}^\top \theta_t}{d}\right)^2\right] \quad = \quad (m_t^\star)^2 \tag{B.130}$$

□

We have thus shown that $m_t^\star = m_{{\rm bo},t}$, proving Lemma B.4.1.

Computing the conditional variance Fix now the confidence $\hat{f}_t = \ell$, the local field of the estimator t is $\nu_t := \sigma_{\hat{\tau}_t}^{-1}(\ell)$. The conditional distribution of the Bayes-optimal local field $\lambda_{\rm bo}$ is a Gaussian $\mathcal{N}(m_t^\star/q_t^\star \nu_t, q_{\rm bo}^\star - m_t^{\star 2}/q_t^\star)$. Thus,

$$\mathbb{E}\left(\hat{f}_{\rm bo}^2 | \hat{f} = \ell\right) = \int {\rm d}z \sigma_{v_{\rm bo}^\star + \tau^2 + \tau_{\rm add}^2}(z)^2 \mathcal{N}(z|m_t^\star/q_t^\star \nu_t, q_{\rm bo}^\star - m_t^{\star 2}/q_t^\star) \tag{B.131}$$

The last step to prove eq. (B.114) is to show that $\mathbb{E}\left(\hat{f}_{\rm bo}|\hat{f}_t = \ell\right) = \ell - \Delta_\ell$:

$$\begin{aligned}
\mathbb{E}\left(\hat{f}_{\rm bo}|\hat{f}_t = \ell\right) &= \int \sigma_{\tau^2+\tau_{\rm add}^2+\hat{v}_{\rm bo}^*}(z)\mathcal{N}(z|m_t^\star/q_t^\star \nu_t, q_{\rm bo}^\star - m_t^{\star 2}/q_t^\star){\rm d}z \\
&= \sigma_{\tau^2+\tau_{\rm add}^2+\hat{v}_{\rm bo}^*+q_{\rm bo}^\star - m_t^{\star 2}/q_t^\star}(m_t^\star/q_t^\star \nu_t) \\
&= \sigma_{\tau^2+\tau_{\rm add}^2+\rho - m_t^{\star 2}/q_t^\star}(m_t^\star/q_t^\star \nu_t) = \mathbb{E}\left(f_\star|\hat{f}_t = \ell\right) = \ell - \Delta_\ell
\end{aligned}$$

since, due to Bayes optimality, $\hat{v}_{\rm bo}^* = \rho - q_{\rm bo}^\star$.

B.5 Additional numerical evaluations

B.5.1 Calibration at different levels

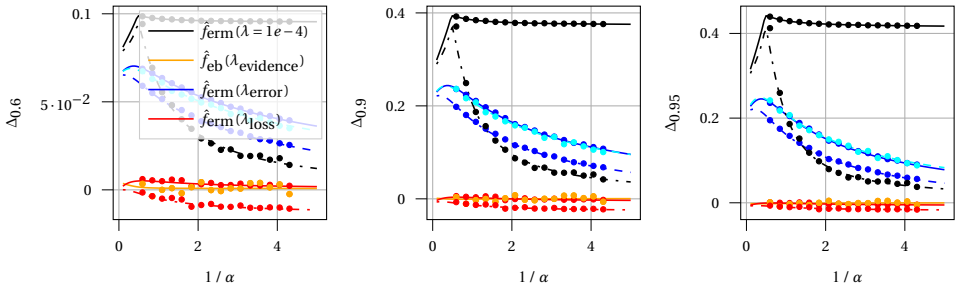


Figure B.2: Calibration of several estimators in the same setting as Figure 3.1 at level $\ell = 0.6$ (Left), $\ell = 0.9$ (Middle), $\ell = 0.95$ (Right). Dashed lines correspond to $\hat{f}_{\text{Lap}}$. We observe the same phenomenology as in Figure 3.1, as $\hat{f}_{\text{Lap}}$ tends to be underconfident, and $\hat{f}_{\text{eb}}(\lambda_{\text{evidence}})$ is the best calibrated estimator across all levels. Dots correspond to experiments at $d = 200$.

B.5.2 Additional setting : $\tau^2 = 0, {}^n/_d = 10.0$

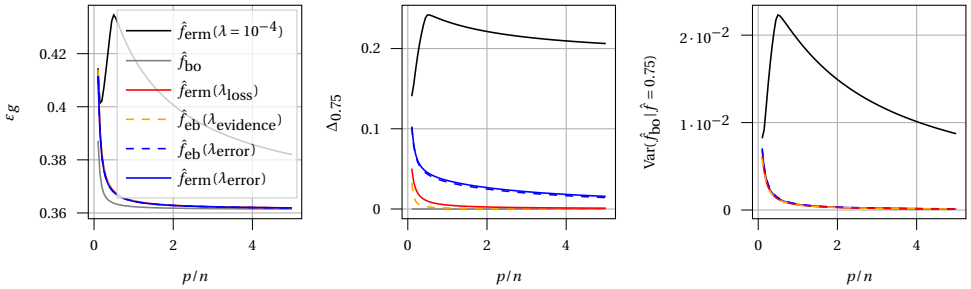


Figure B.3: (**Left**) Test error of the estimators as a function of ${}^1/_\alpha$ in the setting of Section B.5.2 : $\|\theta_\star\|^2 = 1, \tau^2 = 0, {}^n/_d = 10$. (**Middle**) Calibration of the estimators. (**Right**) Variance of $\hat{f}_{\text{bo}}$ conditioned on $\hat{f} = 0.75$ for the different estimators.

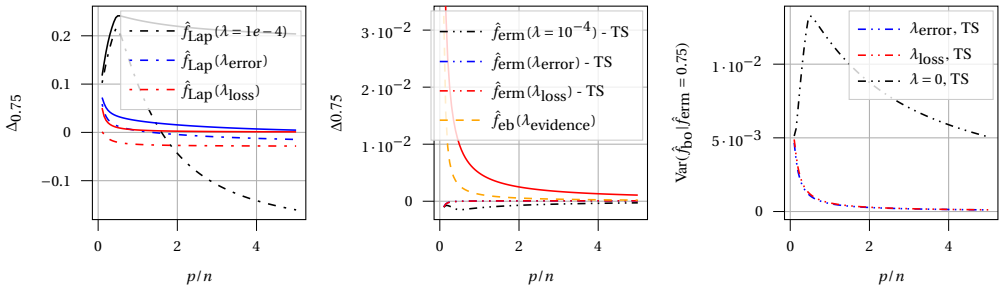


Figure B.4: (**Left**) Calibration of $\hat{f}_{\text{Lap}}$ and $\hat{f}_{\text{erm}}$ in the setting of Section B.5.2. (**Middle**) Calibration of $\hat{f}_{\text{erm}}$ after temperature scaling. Curves for λ_{error} and λ_{loss} are indistinguishable on the plot. (**Right**) Variance of $\hat{f}_{\text{bo}}$ conditioned on the confidence of temperature scaling.

In this section, we consider a setting where $\tau^2 = 0.0$. This allows us to consider a setting where the test error of our estimators will be lower, as we reduce the noise in the teacher and increase the amount of training data. This is confirmed by the first panel of Figure B.3, where the test error of the estimators is smaller than in Figure 3.1. Moreover, compared to the setting of Figure 3.1, the curves for $\hat{f}_{\text{erm}}(\lambda_{\text{error}})$, $\hat{f}_{\text{erm}}(\lambda_{\text{loss}})$ and $\hat{f}_{\text{eb}}(\lambda_{\text{evidence}})$ are much closer. Looking at the second panel, we note that as before, doing ERM with λ_{loss} or empirical-Bayes with $\lambda_{\text{evidence}}$ yields the best calibration. However, the calibration curves $\Delta_{0.75}$ do not exhibit the *double descent*-like behaviour shown in Figure 3.1. On Figure B.4, we see the calibration of $\hat{f}_{\text{Lap}}$ (left plot) and temperature scaling (center). We see that in this setting, $\hat{f}_{\text{Lap}}$ yields underconfident estimators for p/n large enough. On the other hand, temperature scaling yields a well-calibrated estimator, whether we apply it on $\hat{f}_{\text{erm}}(\lambda = 0)$ or $\hat{f}_{\text{erm}}(\lambda_{\text{error}})$

B.5.3 Additional setting 2 : $\tau^2 = 0, n/d = 20, \|\theta_*\|^2 = 50$

In the previous plots, we defined $\theta_* = 1$. This is of course not a limitation of our model and we can assume any norm for the teacher. In this section, we will assume $\|\theta_*\|^2 = 50$. This allows us to significantly reduce the noise in the data. Indeed, as $\|\theta_*\|^2 \to \infty$, the label becomes deterministic in the input. As before, figures B.5 and B.4 show the test error, calibration and variance for the different estimators. In the left panel of Figure B.6, we observe that $\hat{f}_{\text{Lap}}$ with $\lambda \in \{\lambda_{\text{error}}, \lambda_{\text{loss}}, 10^{-4}\}$ systematically under-confident for p/n large enough. As with the previous settings, we also note that $\hat{f}_{\text{erm}}(\lambda_{\text{error}})$ used in combination with temperature scaling is the most competitive estimator as it yields very good test error and calibration.

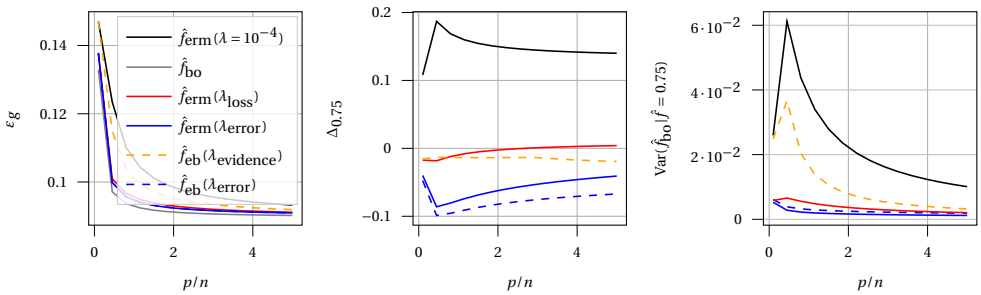


Figure B.5: (**Left**) Test error of the estimators as a function of $1/\alpha$ in the setting described in section B.5.3. (**Middle**) Calibration of the estimators. (**Right**) Variance of $\hat{f}_{\text{bo}}$ conditioned on $\hat{f} = 0.75$ for the different estimators.

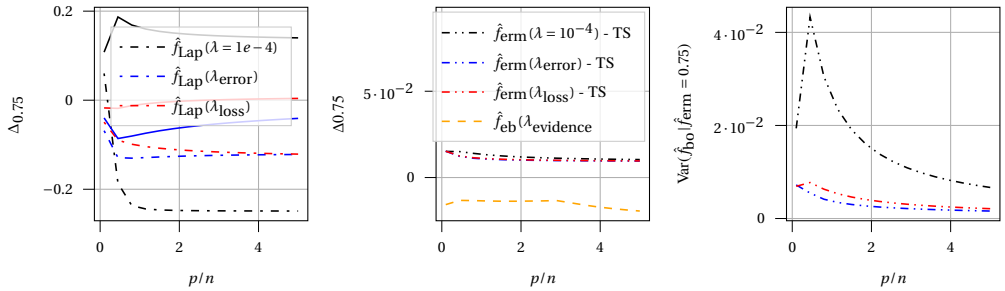


Figure B.6: (**Left**) Calibration of $\hat{f}_{\text{Lap}}$ and $\hat{f}_{\text{erm}}$ with the setting described in section B.5.3. (**Middle**) Calibration of $\hat{f}_{\text{erm}}$ after temperature scaling. Solid red line is $\hat{f}_{\text{erm}}(\lambda_{\text{loss}})$ before temperature scaling. (**Right**) Variance of $\hat{f}_{\text{bo}}$ conditioned on the confidence of temperature scaling.

C Appendix of chapter 4

C.1 Details on training procedure

SVHN For the SVHN dataset [184], the Resnet20 model of depth 20 and containing 0.27M parameters was trained for 50 epochs, using SGD with a learning rate $\eta = 0.1$, weight decay $1e-4$ and momentum 0.9. 90% of data points were used for training and the rest was used for validation.

CIFAR10 ResNet models (of depth 20, 56 with Resnet56 having 0.85M parameters) were trained for 50 epochs, using SGD with a learning rate $\eta = 0.1$, weight decay $1e-4$ and momentum 0.9. The DenseNet 121 (containing 7.9 parameters) was trained with the same parameters as the ResNets, except for the learning rate $\eta = 0.01$. As in [106], images in the training set were randomly cropped and flipped horizontally.

CIFAR100 On CIFAR100, we used pre-trained models from the Github repository https://github.com/chenyaofo/pytorch-cifar-models. These models were trained on the entirety of the training set, so the test set containing 10000 images was split in half into a validation and test set, containing 5000 images each.

C.1.1 Additional plots

In Figure C.1, we plot the reliability diagram of Resnet20 and Resnet56 on SVHN and CIFAR10 respectively. We observe that the uncalibrated models are overconfident (as the confidence is higher than the corresponding accuracy), and both TS and EC mitigate this overconfidence.

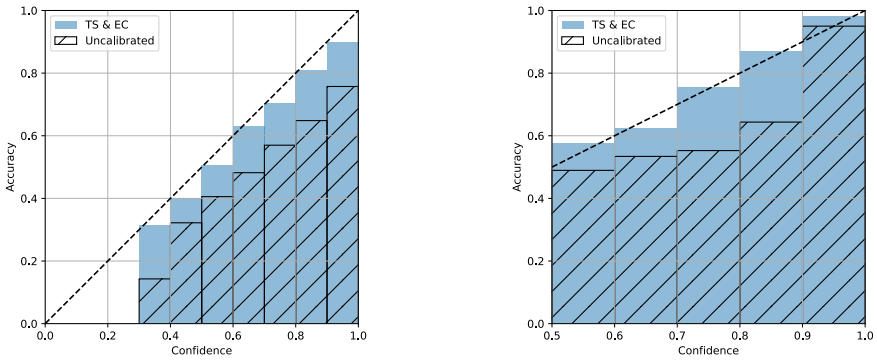


Figure C.1: Reliability diagram of Resnet20 on the SVHN dataset (Left) and Resnet56 on the CIFAR10 dataset (Right). Before calibration, both methods are overconfident. TS and EC improve calibration and mitigate overconfidence.

C.2 State evolution equation

In this section, we focus on the data model introduced in Section 5. Recall that we consider a dataset of n samples $\mathscr{D} = (x^\mu, y^\mu)_{\mu=1}^n$ generated by

$$\mathbf{x} \sim \mathscr{N}(\mathbf{0}, \mathscr{I}_d / d), \mathbf{w}_* \sim \mathscr{N}(\mathbf{0}, \mathscr{I}_d), \mathbb{P}(y = 1 | {\theta_\star}^\top \mathbf{x}) = \sigma_\star({\theta_\star}^\top \mathbf{x}) \tag{C.1}$$

and we fit the following logistic regression model, with σ the sigmoid function:

$$\hat{f}_{\text{erm}}(\mathbf{x}) = \sigma(\hat{\theta}_\lambda^\top \mathbf{x}) \tag{C.2}$$

by minimizing the following empirical risk

$$\mathscr{R}(\mathbf{w}, \mathscr{D}, \lambda) = \sum_{\mu=1}^{n} \log \sigma(y^\mu \mathbf{w}^\top \mathbf{x}) + {}^{\lambda}/_{2} \|\mathbf{w}\|^2 \tag{C.3}$$

we thus have $\hat{\theta}_\lambda = \arg\min_{\mathbf{w}} \mathscr{R}(\mathbf{w}, \mathscr{D}, \lambda)$. For a new sample $\mathbf{x}$, we are interested in the joint distribution of $f_*(\mathbf{x})$ and $\hat{f}_{\text{erm}}(\mathbf{x})$. As these two functions only depend on the scalar products ${\theta_\star}^\top \mathbf{x}, \hat{\theta}_\lambda^\top \mathbf{x}$ it suffices to compute the joint distribution of these scalar products. By the Gaussianity of $\mathbf{x}$, we just need to compute the *overlaps* $m = {\theta_\star}^\top \hat{\theta}_\lambda$ and $q = \|\hat{\theta}_\lambda\|^2$. In the high-dimensional limit where $n, d \to \infty$ but where we keep the *sampling ratio* constant ${}^{n}/_{d} = \alpha$, it is possible to compute the value of m and q. The idea is to introduce the distribution

$$\mu_{\beta,\mathscr{D},\lambda}(\mathbf{w}) = \frac{1}{\mathscr{Z}_\beta} \exp\left(-\beta \mathscr{R}(\mathbf{w}, \mathscr{D}, \lambda)\right) \tag{C.4}$$

where $\mathscr{Z}_\beta$ is a normalization constant. In the limit $\beta \to \infty$, $\mu_{\beta,\mathscr{D},\lambda}$ converges to a Dirac distribution peaked at $\hat{\theta}_\lambda = \arg\min \mathscr{R}(\mathbf{w}, \mathscr{D}, \lambda)$. To compute m, q, one needs to compute the expression of $\log \mathscr{Z}_\beta$ and its limit when $\beta \to \infty$. In the high-dimensional regime where both the dimension and

number of samples diverge with a fixed ratio, this can be done using the *replica method* from statistical physics [247]. As these computations are not the focus of this appendix, we refer to [153, 53] for the detailed computations. In the end, if we define

$$\mathcal{Z}_*(y,\omega,v_*) = \int \mathrm{d}z \sigma_*(y\times z)\mathcal{N}(z|\omega,v_*) \tag{C.5}$$

$$f(y,\omega,v) = \arg\min_z \left[\frac{(z-\omega)^2}{2v} - \log\sigma(z)\right] \tag{C.6}$$

then m, q are the solution of the following self-consistent equations:

$$\begin{cases} m &= \frac{\hat{m}}{\lambda+\hat{v}} \\ q &= \frac{\hat{q}+\hat{m}^2}{(\lambda+\hat{v})^2} \\ v &= \frac{1}{\lambda+\hat{v}} \end{cases}, \quad \begin{cases} \hat{m} &= \alpha\mathbb{E}_{\xi\sim\mathcal{N}(0,q)}\left[\int \mathrm{d}y \partial_\omega \mathcal{Z}_*(y, {}^m/_q\xi, v_*) f(y,\xi,v)\right] \\ \hat{q} &= \alpha\mathbb{E}_{\xi\sim\mathcal{N}(0,q)}\left[\int \mathrm{d}y \mathcal{Z}_*(y, {}^m/_q\xi, v_*) f^2(y,\xi,v)\right] \\ \hat{v} &= -\alpha\mathbb{E}_{\xi\sim\mathcal{N}(0,q)}\left[\int \mathrm{d}y \mathcal{Z}_*(y, {}^m/_q\xi, v_*) \partial_\omega f(y,\xi,v)\right] \end{cases} \tag{C.7}$$

with $v_* = \rho - {}^{m^2}/_q$.

Calibration in the high-dimensional regime Once we obtained the overlaps m, q, we can derive the expression the calibration Δ_ℓ:

$$\Delta_\ell = \mathbb{E}\left[f_*(\mathbf{x})|\hat{f}_{\mathrm{erm}}(\mathbf{x})\right] = \mathbb{P}\left[y=1|\hat{f}_{\mathrm{erm}}(\mathbf{x})\right] = \int \mathrm{d}z \sigma_\star(z)\mathcal{N}(z|\frac{m}{q}\hat{f}^{-1}_{\mathrm{erm}}(\mathbf{x}), \rho - {}^{m^2}/_q) \tag{C.8}$$

The second line comes from the fact that the scalar product $\theta_\star{}^\top\mathbf{x}$ conditioned on $\hat{\theta}_\lambda^\top\mathbf{x} = \sigma^{-1}(\ell)$ follows a Gaussian distribution with mean ${}^m/_q\xi$ and variance $\rho - {}^{m^2}/_q$. As a consequence, the expression of ECE is

$$ECE = \mathbb{E}_{\mathbf{x}}\left[|\Delta_{\hat{f}_{\mathrm{erm}}(\mathbf{x})}|\right] = \mathbb{E}_{\xi=\hat{\theta}_\lambda^\top\mathbf{x}}\left[|\Delta_{\sigma(\xi)}|\right] = \int \mathrm{d}\xi |\Delta_{\sigma(\xi)}| \mathcal{N}(\xi|0,q) \tag{C.9}$$

C.3 Experiments on corrupted dataset

We describe below an experiment where EC can significantly improve over TS for real data: we train different architectures on several image classification tasks, as in Figure 1. However, here for the validation and test set some classes are replaced with random labels. For SVHN and CIFAR10, the labels $y = 0$ are replaced by random labels. For CIFAR100, the labels $y = 0, ..., 9$ are replaced by random labels. By doing so, around 10% of validation/test data is corrupted, with a noise that depends on the class. Note that the training data is left unchanged: the goal of this experiment is to model a distribution shift between training and test data, similarly as what is done [108].

In the table below, we compare the performance (in ECE and Brier score) of EC and TS with these corrupted datasets. We observe that in this setting, EC outperforms TS by a significant

Dataset	Model	$\mathscr{E}_g$	T_{TS}	T_{EC}	ECE	ECE_{TS}	ECE_{EC}	BS	BS_{TS}	BS_{EC}
SVHN	Resnet20	12.5 %	2.69	2.23	8.3 %	10.7 %	7.5 %	21.9 %	23.4 %	22.1 %
CIFAR10	Resnet20	20.9 %	2.4	2.0	12.8 %	4.6 %	4.2 %	34.2 %	32.2 %	32.1 %
CIFAR10	Resnet56	21 %	2.58	2.15	13.8 %	5.4 %	4.9 %	35.2 %	32.9 %	32.8 %
CIFAR10	Densenet121	20.4 %	2.76	2.54	15.8 %	3.6 %	5.0 %	35.9 %	31.8 %	31.9 %
CIFAR100	Resnet20	38.1 %	2.04	1.70	16.5 %	9.6 %	5.9 %	57.0 %	54.9 %	53.9 %
CIFAR100	Resnet56	34.8 %	2.27	2.10	21.7 %	7.6 %	7.3 %	56.0 %	50.6 %	50.4 %
CIFAR100	VGG19	35.5 %	2.6	2.1	28.34 %	5.2	5.1 %	61.8 %	50.1 %	50.1 %
CIFAR100	RepVGG-A2	30.5 %	1.44	1.40	13.7 %	11.6 %	11.7 %	47.2 %	47.1 %	47.0 %

Table C.1: Comparison of expected calibration error (ECE) and Brier score (BS) of temperature scaling (TS) and expectation consistency (EC) when part of the validation and test data has been corrupted

margin on several datasets and architectures.

D Appendix of chapter 5

D.1 Derivation of the results for pair resampling

In this appendix we show how the self-consistent equations (5.17) and (5.18) can be derived from the state-evolution equation of GAMP (Generalized Approximate Message Passing), and how to extend them to generic log-concave losses.

As stated in section 5.3, the key observation is that in order to asymptotically characterize the biases and variances associated with any of the resampling methods in section 5.2, it is sufficient to characterize only the correlation $\hat{\theta}_\lambda(\mathscr{D}_b^\star)^\top \hat{\theta}_\lambda(\mathscr{D}_{b'}^\star)$ between two resampled datasets $\mathscr{D}_b^\star, \mathscr{D}_{b'}^\star$. Indeed, the resampling variances can be written

$$\widehat{\mathrm{Var}} = \frac{1}{d}\left(\frac{1}{B}\sum_{b=1}^{B}\|\hat{\theta}_b\|^2 - \frac{1}{B^2}\sum_{b,b'=1}^{B}\hat{\theta}_b^\top\hat{\theta}_{b'}\right). \tag{D.1}$$

It is natural to study these variances in the limit $B \to \infty$. In that limit, $\widehat{\mathrm{Var}}$ converges to

$$\widehat{\mathrm{Var}} = \frac{1}{d}\mathbb{E}_{\mathscr{D}^\star}\left[\|\hat{\theta}(\mathscr{D}^\star)\|^2\right] - \frac{1}{d}\mathbb{E}_{\mathscr{D}^\star,\mathscr{D}^{\star\prime}}\left[\hat{\theta}(\mathscr{D}^\star)\hat{\theta}(\mathscr{D}^{\star\prime})\right]$$

where the expectations are over resampled dataset conditioned on $\mathscr{D}$ and where the resampling depends on the method considered. In a similar way for the bias

$$\begin{aligned}\widehat{\mathrm{Bias}^2} &= \frac{1}{d}\left\|\frac{1}{B}\sum_{b=1}^{B}\hat{\theta}_b - \hat{\theta}\right\|^2 \\ &\overset{B\to\infty}{\to} \frac{1}{d}\left(\|\hat{\theta}\|^2 + \left\|\mathbb{E}_{\mathscr{D}^\star}\left[\hat{\theta}(\mathscr{D}^\star)\right]\right\|^2\right)\end{aligned}$$

To do so, we observe that computing the ERM estimator on a resampled dataset $\mathscr{D}^\star$ is equivalent to solving an wERM problem eq. (5.14), where for each sample $(\mathbf{x}_i, y_i) \in \mathscr{D}$, we introduce a sample weight p_i. The distribution on the sample weights depends on the way $\mathscr{D}$ is resampled:

for example, with $p_i = 1$ for all $i \in [n]$, this reduces to standard MLE (5.2). On the other hand, by choosing $p_i \in \{0, 1\}$ at random from a Bernoulli distribution with probability $r \in (0, 1]$, the wERM (5.14) asymptotically corresponds to doing subsampling. Also, pair bootstrap is asymptotically equivalent to taking $p_i \sim \text{Pois}(1)$ independently. The problem is thus to compute the correlation between estimators $\hat{\theta}_\lambda(\mathscr{D}, \mathbf{p})$ trained with different, possibly correlated vectors $\mathbf{p}$.

The use of GAMP for deriving high-dimensional asymptotics characterization is now a classic rigorous tool, that has been used in many situations [27, 114, 221, 78, 154, 152, 95]. The idea is to proceed in two steps: i) to propose a GAMP algorithm that solves the optimisation problem asymptotically, and ii) to use the fact that GAMP performance can be tracked with a rigorous state evolution [26, 97] . This was, to the best of our knwoledge, introduced first in [27] for studying the LASSO risk. We shall not repeat the proof technique, and refer the reader to [154, 152] for details with our current notation. Our results directly uses Thm. 1 in [154] or Thm 2.1 in [152].

The novelty of our approach consists in adapting these results to the bootstrap situation by introducing sample weights $\mathbf{p}$ and studying the performance of GAMP for several estimators. The properties of the estimators are given by the distribution on the weights $\mathbf{p}$. All previous proof still trivially apply: indeed the state evolution theorems generalize to vector estimations [115], and, since GAMP is applied to two problems in parallel, the convergence guarantees still independently apply to each of them. A similar strategy was used in [152].

Algorithm 8 GAMP with sample weights

Input: $\mathbf{X} \in \mathbb{R}^{n \times d}$, $\mathbf{y} \in \mathbb{R}^n$, and $\mathbf{p}_\mu \in \mathbb{R}^B$ for $1 \leq \mu \leq n$
Initialize: $\mathbf{g}_{\text{out}\mu}^{(0)} = \mathbf{0}$ for $1 \leq \mu \leq n$, $\quad \mathbf{A}_i^{(0)} = \mathbf{I}_B$ for $1 \leq i \leq d$
Initialize: $\hat{\theta}_i^{(1)} \in \mathbb{R}^B$ and $\hat{\mathbf{C}}_i^{(1)} \in \mathbb{R}^{B \times B}$ for $1 \leq i \leq d$
Repeat for $t = 1, 2, \ldots$**:**
// Update of the means $\omega_\mu \in \mathbb{R}^B$ and covariances $\mathbf{V}_\mu \in \mathscr{S}_B^+$ for $1 \leq \mu \leq n$:
$\omega_\mu^{(t)} = \sum_{i=1}^d X_{\mu,i} \hat{\theta}_i^{(t)} - X_{\mu,i}^2 \left(\mathbf{A}_i^{(t-1)}\right)^{-1} \hat{\mathbf{C}}_i^{(t)} \mathbf{A}_i^{(t-1)} \mathbf{g}_{\text{out}\mu}^{(t-1)} \mid \mathbf{V}_\mu^{(t)} = \sum_{i=1}^d X_{\mu,i}^2 \hat{\mathbf{C}}_i^{(t)}$
// Update of $\mathbf{g}_{\text{out}\mu}$ and $\partial_\omega \mathbf{g}_{\text{out}\mu}$ for $1 \leq \mu \leq n$:
$\mathbf{g}_{\text{out}\mu}^{(t)} = \mathbf{g}_{\text{out}}\left(\omega_\mu^{(t)}, y_\mu, \mathbf{V}_\mu^{(t)}, \mathbf{p}_\mu\right) \mid \partial_\omega \mathbf{g}_{\text{out}\mu}^{(t)} = \partial_\omega \mathbf{g}_{\text{out}}\left(\omega_\mu^{(t)}, y_\mu, \mathbf{V}_\mu^{(t)}, \mathbf{p}_\mu\right)$
// Update of means $\mathbf{b}_i \in \mathbb{R}^B$ and covariances $\mathbf{A}_i \in \mathbb{R}^{B \times B}$ for $1 \leq i \leq d$:
$\mathbf{A}_i^{(t)} = -\sum_{\mu=1}^n X_{\mu,i}^2 \partial_\omega \mathbf{g}_{\text{out}\mu}^{(t)} \mid \mathbf{b}_i^{(t)} = \mathbf{A}_i^{(t)} \hat{\theta}_i^{(t)} + \sum_{\mu=1}^n X_{\mu,i} \mathbf{g}_{\text{out}\mu}^{(t)}$
// Update of the estimated marginals $\hat{\theta}_i \in \mathbb{R}^B$ and $\hat{\mathbf{C}}_i \in \mathbb{R}^{B \times B}$ for $1 \leq i \leq d$:
$\hat{\theta}_i^{(t+1)} = \mathbf{f}_a(\mathbf{b}_i^{(t)}, \mathbf{A}_i^{(t)}) \mid \hat{\mathbf{C}}_i^{(t+1)} = \partial_{\mathbf{b}} \mathbf{f}_a(\mathbf{b}_i^{(t)}, \mathbf{A}_i^{(t)})$
Until convergence
Output: $\hat{\theta}_1, \ldots, \hat{\theta}_d$ and $\hat{\mathbf{C}}_1, \ldots, \hat{\mathbf{C}}_d$

Consider a convex loss function ℓ and regularizer r, and the following empirical risk minimization problem

$$(\hat{\theta}_1, \ldots, \hat{\theta}_B) = \arg \min_{\theta_1, \ldots, \theta_B \in \mathbb{R}^d} \mathscr{L}(\theta_1, \ldots, \theta_B) \tag{D.2}$$

where

$$\mathscr{L}(\theta_1,\dots,\theta_B) := \sum_{\mu=1}^{n} \ell_{\mathbf{p}}(y_\mu, \theta_1^\top \mathbf{x}_\mu, \dots, \theta_B^\top \mathbf{x}_\mu) + \sum_{b=1}^{B} r(\theta_b) \tag{D.3}$$

and

$$\ell_{\mathbf{p}}(y, z_1, \dots, z_B) := \sum_{b=1}^{B} p_b \ell(y, z_b) \tag{D.4}$$

We define a *channel function* associated to the function ℓ :

$$\mathbf{g}_{\text{out}}(y, \omega, \mathbf{V}, \mathbf{p}) = \mathbf{V}^{-1}\left(\text{prox}_{\mathbf{V}, \ell_{\mathbf{p}}(y,\cdot)}(\omega) - \omega\right), \tag{D.5}$$

where the proximal operator is

$$\text{prox}_{\mathbf{V}, \ell_{\mathbf{p}}(y,\cdot)}(\omega) = \underset{\mathbf{z}\in\mathbb{R}^B}{\text{arg min}}\left(\frac{1}{2}(\mathbf{z}-\omega)^\top \mathbf{V}^{-1}(\mathbf{z}-\omega) + \ell_{\mathbf{p}}(y, \mathbf{z})\right). \tag{D.6}$$

Let us also define the *denoising function* associated to the regularizer r:

$$\mathbf{f}_a(\mathbf{b}, \mathbf{A}) = \text{prox}_{\mathbf{A}^{-1}, r}(\mathbf{A}^{-1}\mathbf{b}) = \underset{\mathbf{z}\in\mathbb{R}^B}{\text{arg min}}\left(\frac{1}{2}(\mathbf{z}-\mathbf{A}^{-1}\mathbf{b})^\top \mathbf{A}(\mathbf{z}-\mathbf{A}^{-1}\mathbf{b}) + r(\mathbf{z})\right). \tag{D.7}$$

Using algorithm 8 with this choice of channel and denoising functions returns a set of vectors $\hat{\theta}_1, \cdots, \hat{\theta}_d \in \mathbb{R}^B$, where $\hat{\theta}_i$ contains the B estimates for $\theta_{\star i}$. Hence, these vectors allow to solve the minimization problem (D.2).

Intuition of GAMP algorithm We are interested in solving the minimization problem (D.2), which is equivalent to sampling from the distribution

$$p(\theta_1,\dots,\theta_B) \propto \exp\left(-\beta \mathscr{L}(\theta_1,\dots,\theta_B)\right) = \exp\left(-\beta\left(\sum_{\mu=1}^{n} \ell_{\mathbf{p}}(y_\mu, \theta_1^\top \mathbf{x}_\mu, \dots, \theta_B^\top \mathbf{x}_\mu) + \sum_{b=1}^{B} r(\theta_b)\right)\right) \tag{D.8}$$

in the limit $\beta \to \infty$. Sampling the distribution on a graphical model can be used with Belief Propagation, which iterates messages between different nodes (here the coordinates θ_{ij} for $i \le B, j \le d$). However in high dimensions, Belief Propagation is intractable as it involves computing d-dimensional integrals. To alleviate this issue, GAMP only computes the first two moments of the different messages. In the high-dimensional limit, the output of GAMP coincides with the true minimizer of (D.2).

Similarly to our work, in [14], the authors introduce a GAMP algorithm for for a generic coupled system of estimates. They provide a detailed analysis of GAMP and its state evolution to track its

behaviour in the asymptotic limit.

D.1.1 State evolution equations

In this section, we inspect the behavior of algorithm 8 in the $n, d \to \infty$ limit and derive the asymptotic distribution of $\hat{\theta}_1, \dots, \hat{\theta}_d$. To do so, we start from the more convenient relaxed Belief Propagation (rBP) equations, which are very close to GAMP. In the high-dimensional limit, rBP and GAMP are equivalent. The rBP equations are written,

$$\begin{cases} \omega^{(t)}_{\mu\to i} &= \sum_{j\neq i} X_{\mu,j}\hat{\theta}^{(t)}_{j\to\mu} \\ \mathbf{V}^{(t)}_{\mu\to i} &= \sum_{j\neq i} X^2_{\mu,j}\hat{\mathbf{C}}^{(t)}_{j\to\mu} \end{cases}, \quad \begin{cases} \mathbf{g}_{\text{out}}{}^{(t)}_{\mu\to i} &= \mathbf{g}_{\text{out}}(y_\mu, \omega^{(t)}_{\mu\to i}, \mathbf{V}^{(t)}_{\mu\to i}, \mathbf{p}_\mu) \\ \partial\mathbf{g}_{\text{out}}{}^{(t)}_{\mu\to i} &= \partial_\omega \mathbf{g}_{\text{out}}(y_\mu, \omega^{(t)}_{\mu\to i}, \mathbf{V}^{(t)}_{\mu\to i}, \mathbf{p}_\mu) \end{cases} \tag{D.9}$$

$$\begin{cases} \mathbf{b}^{(t)}_{\mu\to i} &= \sum_{\nu\neq\mu} X_{\nu,i}\mathbf{g}^{(t)}_{\text{out}\,\nu\to i} \\ \mathbf{A}^{(t)}_{\mu\to i} &= -\sum_{\nu\neq\mu} X^2_{\nu,i}\partial\mathbf{g}^{(t)}_{\text{out}\,\nu\to i} \end{cases}, \quad \begin{cases} \hat{\theta}^{(t)}_{i\to\mu} &= \mathbf{f}_a(\mathbf{b}^{(t)}_{i\to\mu}, \mathbf{A}^{(t)}_{i\to\mu}) \\ \hat{\mathbf{C}}^{(t)}_{i\to\mu} &= \partial_{\mathbf{b}}\mathbf{f}_a(\mathbf{b}^{(t)}_{i\to\mu}, \mathbf{A}^{(t)}_{i\to\mu}). \end{cases} \tag{D.10}$$

It turns out that the average asymptotic behavior of these equations can be tracked with some overlap parameters defined as follows:

$$\mathbf{m}^{(t)} \equiv \lim_{d\to\infty} \frac{1}{d}\sum_{i=1}^{d} \hat{\theta}^{(t)}_i \theta_\star{}^\top, \qquad \mathbf{Q}^{(t)} \equiv \lim_{d\to\infty} \frac{1}{d}\sum_{i=1}^{d} \hat{\theta}^{(t)}_i \hat{\theta}^{(t)\top}_i \tag{D.11}$$

$$\mathbf{V}^{(t)} \equiv \lim_{d\to\infty} \frac{1}{d}\sum_{i=1}^{d} \hat{\mathbf{C}}^{(t)}_i, \qquad \rho = \lim_{d\to\infty} \frac{\|\theta_\star\|^2}{d}. \tag{D.12}$$

To derive the asymptotic behavior of these overlap parameters, we compute the overlap distributions starting from the rBP equations above.

Messages Distribution

For convenience, let us define $z_\mu \equiv \sum_{i=1}^{d} X_{\mu,i}\theta_{\star i} = \mathbf{X}^\top_\mu \theta_\star$ and $z_{\mu\to i} \equiv \frac{1}{d}\sum_{j\neq i} X_{\mu,i}\theta_{\star j}$.

Distribution of $(z_\mu, \omega^{(t)}_{\mu\to i})$ By the Central Limit Theorem, since $(z_\mu, \omega^{(t)}_{\mu\to i})$ are the sum of independent variables, they follow Gaussian distributions in the $d \to \infty$ limit. Therefore, we only need to compute their means, variances, and cross-correlation. Recall that from our assumptions, the random variables $X_{\mu,j}$ are i.i.d. zero-mean Gaussian with variance $1/d$. Hence, the first and second-order statistics read

$$\mathbb{E}\left[z_\mu\right] = {\theta_\star}^\top \mathbb{E}[\mathbf{X}_\mu] = 0 \tag{D.13}$$

$$\mathbb{E}\left[z_\mu^2\right] = \sum_{i,j=1}^{d} \mathbb{E}[X_{\mu,i}X_{\mu,j}]\theta_{\star i}\theta_{\star j} = \sum_{i,j=1}^{d} \frac{1}{d}\delta_{ij}\theta_{\star i}\theta_{\star j} = \frac{\|\theta_\star\|^2}{d} \overset{d\to\infty}{\longrightarrow} \rho \tag{D.14}$$

$$\mathbb{E}\left[\omega_{\mu\to i}^{(t)}\right] = \sum_{j\neq i} \mathbb{E}[X_{\mu,j}]\hat{\theta}_{j\to\mu}^{(t)} = \mathbf{0} \tag{D.15}$$

$$\mathbb{E}\left[\omega_{\mu\to i}^{(t)}(\omega_{\mu\to i}^{(t)})^\top\right] = \sum_{j\neq i}^{d}\sum_{k\neq i}^{d} \mathbb{E}[X_{\mu,j}X_{\mu,k}]\hat{\theta}_{j\to\mu}^{(t)}\hat{\theta}_{k\to\mu}^{(t)\top} = \frac{1}{d}\sum_{j\neq i}^{d} \hat{\theta}_{j\to\mu}^{(t)}\hat{\theta}_{k\to\mu}^{(t)\top} \tag{D.16}$$

$$= \frac{1}{d}\sum_{j=1}^{d} \hat{\theta}_{j\to\mu}^{(t)}\hat{\theta}_{j\to\mu}^{(t)\top} - \frac{1}{d}\hat{\theta}_{i\to\mu}^{(t)}\hat{\theta}_{i\to\mu}^{(t)\top} \overset{d\to\infty}{\longrightarrow} \mathbf{Q}^{(t)} \tag{D.17}$$

$$\mathbb{E}\left[z_\mu\omega_{\mu\to i}^{(t)}\right] = \sum_{j=1}^{d}\sum_{k\neq i}^{d} \mathbb{E}[X_{\mu,j}X_{\mu,k}]\hat{\theta}_{k\to\mu}^{(t)}\theta_{\star j} = \frac{1}{d}\sum_{j\neq i} \hat{\theta}_{j\to\mu}^{(t)}\theta_\star \tag{D.18}$$

$$= \frac{1}{d}\sum_{j=1}^{d} \hat{\theta}_{j\to\mu}^{(t)}\theta_\star - \frac{1}{d}\hat{\theta}_{i\to\mu}^{(t)}\theta_\star \overset{d\to\infty}{\longrightarrow} \mathbf{m}^{(t)} \tag{D.19}$$

In summary, in the $d\to\infty$ limit :

$$\left(z_\mu, \omega_{\mu\to i}^{(t)}\right) \sim \mathcal{N}\left(0, \begin{bmatrix} \rho & \mathbf{m}^{(t)\top} \\ \mathbf{m}^{(t)} & \mathbf{Q}^{(t)} \end{bmatrix}\right) \tag{D.20}$$

Concentration of $\mathbf{V}_{\mu\to i}^{(t)}$ In the asymptotic limit, the variances $\mathbf{V}_{\mu\to i}^{(t)}$ concentrate around their means, which equates

$$\mathbb{E}\left[\mathbf{V}_{\mu\to i}^{(t)}\right] = \sum_{j\neq i}^{d} \mathbb{E}\left[X_{\mu,j}^2\right]\hat{\mathbf{C}}^{(t)} = \frac{1}{d}\sum_{j\neq i} \hat{\mathbf{C}}_j^{(t)} = \frac{1}{d}\sum_{j=1}^{d} \hat{\mathbf{C}}_j^{(t)} - \frac{1}{d}\hat{\mathbf{C}}_i^{(t)} \overset{d\to\infty}{\longrightarrow} \mathbf{V}^{(t)} \tag{D.21}$$

Distribution of $\mathbf{b}_{\mu\to i}^{(t)}$ Recall from our setting that for a given input $\mathbf{x}_\mu$, the corresponding label is distributed as $y_\mu \sim p(\cdot|z_\mu)$. In fact, one can equivalently write $y^\mu = \varphi_0(z_\mu)$ for some (random) function φ_0. For example, the choice $\varphi_0(x) = x + \sqrt{\Delta}\xi$ corresponds to the linear regression, where $\xi \sim \mathcal{N}(0,1)$ is Gaussian noise scaled by a variance $\Delta \geq 0$. With this representation for y_μ, we have

$$\mathbf{b}_{\mu\to i}^{(t)} = \sum_{\nu\neq\mu} X_{\nu,i}\mathbf{g}_{\text{out}}(\varphi_0(z_\nu), \omega_{\nu\to i}^{(t)}, \mathbf{V}_{\nu\to i}^{(t)}, \mathbf{p}_\nu) \tag{D.22}$$

$$= \sum_{\nu\neq\mu} X_{\nu,i}\mathbf{g}_{\text{out}}(\varphi_0\left(z_{\nu\to i} + \theta_{\star i}X_{\nu,i}\right), \omega_{\nu\to i}^{(t)}, \mathbf{V}_{\nu\to i}^{(t)}, \mathbf{p}_\nu) \tag{D.23}$$

$$= \sum_{\nu\neq\mu} X_{\nu,i}\mathbf{g}_{\text{out}}(\varphi_0(z_{\nu\to i}), \omega_{\nu\to i}^{(t)}, \mathbf{V}_{\nu\to i}^{(t)}, \mathbf{p}_\nu) + X_{\nu,i}^2\theta_{\star i}\partial_z\mathbf{g}_{\text{out}}(\varphi_0(z_{\nu\to i}), \omega_{\nu\to i}^{(t)}, \mathbf{V}_{\nu\to i}^{(t)}, \mathbf{p}_\nu) + O(d^{-3/2}), \tag{D.24}$$

where in the last equality we have expanded the denoising function at leading order. Taking expectation on both sides yields

$$\mathbb{E}[\mathbf{b}_{\mu\to i}^{(t)}] = \frac{\theta_{\star i}}{d}\sum_{\nu\neq\mu}\partial_z \mathbf{g}_{\text{out}}(\varphi_0(z_{\nu\to i}), \omega_{\nu\to i}^{(t)}, \mathbf{V}_{\nu\to i}^{(t)}, \mathbf{p}_\nu) + O(d^{-3/2}) \tag{D.25}$$

$$= \frac{\theta_{\star i}}{d}\sum_{\nu=1}^{n}\partial_z \mathbf{g}_{\text{out}}(\varphi_0(z_{\nu\to i}), \omega_{\nu\to i}^{(t)}, \mathbf{V}_{\nu\to i}^{(t)}, \mathbf{p}_\nu) - \frac{\theta_{\star i}}{d}\partial_z \mathbf{g}_{\text{out}}(\varphi_0(z_{\mu\to i}), \omega_{\mu\to i}^{(t)}, \mathbf{V}_{\mu\to i}^{(t)}, \mathbf{p}_\mu) + O(d^{-3/2}), \tag{D.26}$$

Note that as $d\to\infty$, it follows from our computations above that for all ν, $(z_{\nu\to i}, \omega_{\nu\to i}^{(t)})$ are identically distributed according to eq. (D.20). Consequently, by the Law of Large Numbers,

$$\frac{n}{d}\cdot\frac{1}{n}\sum_{\nu=1}^{n}\partial_z \mathbf{g}_{\text{out}}(\varphi_0(z_{\nu\to i}), \omega_{\nu\to i}^{(t)}, \mathbf{V}_{\nu\to i}^{(t)}, \mathbf{p}_\nu) \overset{n,d\to\infty}{\longrightarrow} \alpha\mathbb{E}_{(z,\omega),\mathbf{p}}\left[\partial_z \mathbf{g}_{\text{out}}(\varphi_0(z), \omega, \mathbf{V}^{(t)}, \mathbf{p})\right] \equiv \hat{\mathbf{m}}^{(t)}, \tag{D.27}$$

from which we find that

$$\mathbb{E}[\mathbf{b}_{\mu\to i}^{(t)}] \overset{n,d\to\infty}{\longrightarrow} \theta_{\star i}\hat{\mathbf{m}}^{(t)}. \tag{D.28}$$

The second moment can be computed in a similar fashion:

$$\mathbb{E}[\mathbf{b}_{\mu\to i}^{(t)}\mathbf{b}_{\mu\to i}^{(t)\top}] = \sum_{\nu\neq\mu}\sum_{\kappa\neq\mu}\mathbb{E}[X_{\nu,i}X_{\kappa,i}]\mathbf{g}_{\text{out}}(\varphi_0(z_\nu), \omega_{\nu\to i}^{(t)}, \mathbf{V}_{\nu\to i}^{(t)}, \mathbf{p}_\nu)\mathbf{g}_{\text{out}}(\varphi_0(z_\kappa), \omega_{\kappa\to i}^{(t)}, \mathbf{V}_{\kappa\to i}^{(t)}, \mathbf{p}_\kappa)^\top \tag{D.29}$$

$$= \frac{1}{d}\sum_{\nu\neq\mu}\mathbf{g}_{\text{out}}(\varphi_0(z_{\nu\to i}), \omega_{\nu\to i}^{(t)}, \mathbf{V}_{\nu\to i}^{(t)}, \mathbf{p}_\nu)\mathbf{g}_{\text{out}}(\varphi_0(z_{\nu\to i}), \omega_{\nu\to i}^{(t)}, \mathbf{V}_{\nu\to i}^{(t)}, \mathbf{p}_\nu)^\top + O(d^{-1/2}) \tag{D.30}$$

$$= \frac{1}{d}\sum_{\nu=1}^{n}\mathbf{g}_{\text{out}}(\varphi_0(z_{\nu\to i}), \omega_{\nu\to i}^{(t)}, \mathbf{V}_{\nu\to i}^{(t)}, \mathbf{p}_\nu)\mathbf{g}_{\text{out}}(\varphi_0(z_{\nu\to i}), \omega_{\nu\to i}^{(t)}, \mathbf{V}_{\nu\to i}^{(t)}, \mathbf{p}_\nu)^\top + O(d^{-1/2}) \tag{D.31}$$

$$\overset{n,d\to\infty}{\longrightarrow} \alpha\mathbb{E}_{(z,\omega^{(t)}),\mathbf{p}}\left[\mathbf{g}_{\text{out}}(\varphi_0(z), \omega^{(t)}, \mathbf{V}^{(t)}, \mathbf{p})\mathbf{g}_{\text{out}}(\varphi_0(z), \omega^{(t)}, \mathbf{V}^{(t)}, \mathbf{p})^\top\right] \equiv \hat{\mathbf{Q}}^{(t)}. \tag{D.32}$$

Hence, $\mathbf{b}_{\mu\to i}^{(t)} = \theta_{\star i}\hat{\mathbf{m}}^{(t)} + \left(\hat{\mathbf{Q}}^{(t)}\right)^{1/2}\xi$ with $\xi\sim\mathcal{N}(\mathbf{0}, \mathbf{I}_B)$.

Concentration of $\mathbf{A}_{\mu\to i}^{(t)}$ It remains to show that the covariances $\mathbf{A}_{\mu\to i}^{(t)}$ concentrate. We have

$$\mathbf{A}_{\mu\to i}^{(t)} = -\sum_{\nu\neq\mu}X_{\nu,i}^2\partial_\omega \mathbf{g}_{\text{out}}(y_\nu, \omega_{\nu\to i}^{(t)}, \mathbf{V}_{\nu\to i}^{(t)}, \mathbf{p}_\nu) \tag{D.33}$$

$$= -\sum_{\nu\neq\mu}X_{\nu,i}^2\partial_\omega \mathbf{g}_{\text{out}}(\varphi_0(z_\nu), \omega_{\nu\to i}^{(t)}, \mathbf{V}_{\nu\to i}^{(t)}, \mathbf{p}_\nu) \tag{D.34}$$

$$= -\sum_{\nu\neq\mu}X_{\nu,i}^2\partial_\omega \mathbf{g}_{\text{out}}(\varphi_0(z_{\nu\to i}), \omega_{\nu\to i}^{(t)}, \mathbf{V}_{\nu\to i}^{(t)}, \mathbf{p}_\nu) + O(d^{-3/2}). \tag{D.35}$$

Taking the expectation gives

$$\mathbb{E}[\mathbf{A}^{(t)}_{\mu\to i}] = -\frac{1}{d}\sum_{\nu\neq\mu}\partial_\omega \mathbf{g}_{\text{out}}(\varphi_0(z_{\nu\to i}),\omega^{(t)}_{\nu\to i},\mathbf{V}^{(t)}_{\nu\to i},\mathbf{p}_\nu) + O(d^{-3/2}) \tag{D.36}$$

$$= -\frac{1}{d}\sum_{\nu=1}^{n}\partial_\omega \mathbf{g}_{\text{out}}(\varphi_0(z_{\nu\to i}),\omega^{(t)}_{\nu\to i},\mathbf{V}^{(t)}_{\nu\to i},\mathbf{p}_\nu) - \frac{1}{d}\partial_\omega \mathbf{g}_{\text{out}}(\varphi_0(z_{\mu\to i}),\omega^{(t)}_{\mu\to i},\mathbf{V}^{(t)}_{\mu\to i},\mathbf{p}_\mu) + O(d^{-3/2}) \tag{D.37}$$

$$\overset{n,d\to\infty}{\longrightarrow} -\alpha\mathbb{E}_{(z,\omega^{(t)}),\mathbf{p}}\left[\partial_\omega \mathbf{g}_{\text{out}}(\varphi_0(z),\omega^{(t)},\mathbf{V}^{(t)},\mathbf{p})\right] \equiv \hat{\mathbf{V}}^{(t)} \tag{D.38}$$

Summary

Having shown the distribution of messages and concentration, we are ready to characterize the asymptotic distribution of the estimator:

$$\hat{\theta}_i \sim \mathbf{f}_a\left(\theta_{\star i}\hat{\mathbf{m}}^{(t)} + \left(\hat{\mathbf{Q}}^{(t)}\right)^{1/2}\xi, \hat{\mathbf{V}}^{(t)}\right) \quad \forall i\in\{1,\dots,d\}, \tag{D.39}$$

where $\xi\sim\mathcal{N}(\mathbf{0},\mathbf{I}_B)$.

From that, the definitions of overlaps in eq. (D.11) at time $t+1$, and the message distributions, we obtain the state-evolution equations of the GAMP algorithm described in algorithm 8:

$$\begin{cases}\mathbf{m}^{(t+1)} &= \mathbb{E}_{\theta_\star,\xi}\left[\mathbf{f}_a\left(\hat{\mathbf{m}}\theta_\star + \sqrt{\hat{\mathbf{Q}}^{(t)}}\xi, \hat{\mathbf{V}}^{(t)}\right)\theta_\star\right]\\ \mathbf{Q}^{(t+1)} &= \mathbb{E}_{\theta_\star,\xi}\left[\mathbf{f}_a\left(\hat{\mathbf{m}}\theta_\star + \sqrt{\hat{\mathbf{Q}}^{(t)}}\xi, \hat{\mathbf{V}}^{(t)}\right)\mathbf{f}_a\left(\hat{\mathbf{m}}\theta_\star + \sqrt{\hat{\mathbf{Q}}^{(t)}}\xi, \hat{\mathbf{V}}^{(t)}\right)^\top\right]\\ \mathbf{V}^{(t+1)} &= \mathbb{E}_{\theta_\star,\xi}\left[\partial_{\mathbf{b}}\mathbf{f}_a\left(\hat{\mathbf{m}}\theta_\star + \sqrt{\hat{\mathbf{Q}}^{(t)}}\xi, \hat{\mathbf{V}}^{(t)}\right)\right]\end{cases} \tag{D.40}$$

where $\xi\sim\mathcal{N}(0,\mathbf{I}_B)$, and

$$\begin{cases}\hat{\mathbf{m}}^{(t)} &= \alpha\mathbb{E}_{(z,\omega^{(t)}),\mathbf{p}}\left[\partial_z \mathbf{g}_{\text{out}}(\varphi_0(z),\omega^{(t)},\mathbf{V}^{(t)},\mathbf{p})\right]\\ \hat{\mathbf{Q}}^{(t)} &= \alpha\mathbb{E}_{(z,\omega^{(t)}),\mathbf{p}}\left[\mathbf{g}_{\text{out}}(\varphi_0(z),\omega^{(t)},\mathbf{V}^{(t)},\mathbf{p})\mathbf{g}_{\text{out}}(\varphi_0(z),\omega^{(t)},\mathbf{V}^{(t)},\mathbf{p})^\top\right],\\ \hat{\mathbf{V}}^{(t)} &= -\alpha\mathbb{E}_{(z,\omega^{(t)}),\mathbf{p}}\left[\partial_\omega \mathbf{g}_{\text{out}}(\varphi_0(z),\omega^{(t)},\mathbf{V}^{(t)},\mathbf{p})\right]\end{cases} \tag{D.41}$$

where $\left(z,\omega^{(t)}\right)\sim\mathcal{N}\left(0,\begin{bmatrix}\rho & \mathbf{m}^{(t)\top}\\ \mathbf{m}^{(t)} & \mathbf{Q}^{(t)}\end{bmatrix}\right)$.

Let us note that the overlaps $\hat{\mathbf{m}}^{(t)},\hat{\mathbf{Q}}^{(t)},\hat{\mathbf{V}}^{(t)}$ can be written slightly differently. For that, first notice that since $\left(z,\omega^{(t)}\right)$ is Gaussian, so is z conditioned on $\omega^{(t)}$, and in particular $z|\omega^{(t)}\sim\mathcal{N}(\mu_\star(\omega^{(t)}),\nu_\star)$ with $\mu_\star(\omega^{(t)}) = (\mathbf{m}^{(t)})^\top(\mathbf{Q}^{(t)})^{-1}\omega^{(t)}$, $\nu_\star = \rho - (\mathbf{m}^{(t)})^\top(\mathbf{Q}^{(t)})^{-1}\mathbf{m}^{(t)}$. Moreover,

using that $p(y|z) = \delta(y - \varphi_0(z))$, we have for an arbitrary function $\mathbf{f}: \mathbb{R} \times \mathbb{R}^B \to \mathbb{R}^B$ that

$$\begin{aligned}
\mathbb{E}_{(z,\omega^{(t)})}\left[f(\varphi_0(z), \omega^{(t)})\right] &= \mathbb{E}_{\omega^{(t)}}\left[\mathbb{E}_{z|\omega^{(t)}}\left[\mathbf{f}(\varphi_0(z), \omega^{(t)})\right]\right] && \text{(D.42)}\\
&= \mathbb{E}_{\omega^{(t)}}\left[\int \mathrm{d}z \mathcal{N}(z|\mu_\star(\omega^{(t)}), v_\star)\mathbf{f}(\varphi_0(z), \omega^{(t)})\right] && \text{(D.43)}\\
&= \mathbb{E}_{\omega^{(t)}}\left[\int \mathrm{d}z \mathcal{N}(z|\mu_\star(\omega^{(t)}), v_\star)\int \mathrm{d}y\, p(y|z)\mathbf{f}(y, \omega^{(t)})\right] && \text{(D.44)}\\
&= \mathbb{E}_{\omega^{(t)}}\left[\int \mathrm{d}y \mathcal{Z}_0(y, \mu_\star(\omega^{(t)}), v_\star)\mathbf{f}(y, \omega^{(t)})\right], && \text{(D.45)}
\end{aligned}$$

where we have defined $\mathcal{Z}_0(y, \mu, v) \equiv \int \mathrm{d}z \mathcal{N}(z|\mu, v) p(y|z)$. Consequently, we can rewrite

$$\begin{cases}
\hat{\mathbf{m}}^{(t)} &= \alpha \mathbb{E}_{\omega^{(t)},\mathbf{p}}\left[\int \mathrm{d}y \partial_\mu \mathcal{Z}_0(y, \mu_\star(\omega^{(t)}), v_\star) \cdot \mathbf{g}_{\text{out}}(y, \omega^{(t)}, \mathbf{V}^{(t)}, \mathbf{p})\right]\\
\hat{\mathbf{Q}}^{(t)} &= \alpha \mathbb{E}_{\omega^{(t)},\mathbf{p}}\left[\int \mathrm{d}y \mathcal{Z}_0(y, \mu_\star(\omega^{(t)}), v_\star) \cdot \mathbf{g}_{\text{out}}(y, \omega^{(t)}, \mathbf{V}^{(t)}, \mathbf{p})\mathbf{g}_{\text{out}}(y, \omega^{(t)}, \mathbf{V}^{(t)}, \mathbf{p})^\top\right],\\
\hat{\mathbf{V}}^{(t)} &= -\alpha \mathbb{E}_{\omega^{(t)},\mathbf{p}}\left[\int \mathrm{d}y \mathcal{Z}_0(y, \mu_\star(\omega^{(t)}), v_\star) \cdot \partial_\omega \mathbf{g}_{\text{out}}(\varphi_0(z), \omega^{(t)}, \mathbf{V}^{(t)}, \mathbf{p})\right]
\end{cases} \text{(D.46)}$$

where $\omega^{(t)} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}^{(t)})$.

Self-Consistent Equations

In the limit $t \to \infty$, the state-evolution equations derived above yield a set of self-consistent equations:

$$\begin{cases}
\mathbf{m} &= \mathbb{E}_{\theta_\star,\xi}\left[\mathbf{f}_a(\hat{\mathbf{m}}\theta_\star + \sqrt{\hat{\mathbf{Q}}}\xi, \hat{\mathbf{V}})\theta_\star\right]\\
\mathbf{Q} &= \mathbb{E}_{\theta_\star,\xi}\left[\left[\mathbf{f}_a\mathbf{f}_a^\top\right](\hat{\mathbf{m}}\theta_\star + \sqrt{\hat{\mathbf{Q}}}\xi, \hat{\mathbf{V}})\right],\\
\mathbf{V} &= \mathbb{E}_{\theta_\star,\xi}\left[\partial_{\mathbf{b}}\mathbf{f}_a(\hat{\mathbf{m}}\theta_\star + \sqrt{\hat{\mathbf{Q}}}\xi, \hat{\mathbf{V}})\right]
\end{cases}, \begin{cases}
\hat{\mathbf{m}} &= \alpha \mathbb{E}_{\omega,\mathbf{p}}\left[\int \mathrm{d}y \partial_\mu \mathcal{Z}_0(y, \mu_\star(\omega), v_\star) \cdot \mathbf{g}_{\text{out}}(y, \omega, \mathbf{V}, \mathbf{p})\right]\\
\hat{\mathbf{Q}} &= \alpha \mathbb{E}_{\omega,\mathbf{p}}\left[\int \mathrm{d}y \mathcal{Z}_0(y, \mu_\star(\omega), v_\star) \cdot \left[\mathbf{g}_{\text{out}}\mathbf{g}_{\text{out}}^\top\right](y, \omega, \mathbf{V}, \mathbf{p})\right]\\
\hat{\mathbf{V}} &= -\alpha \mathbb{E}_{\omega,\mathbf{p}}\left[\int \mathrm{d}y \mathcal{Z}_0(y, \mu_\star(\omega), v_\star) \cdot \partial_\omega \mathbf{g}_{\text{out}}(y, \omega, \mathbf{V}, \mathbf{p})\right]
\end{cases} \text{(D.47)}$$

where $\xi \sim \mathcal{N}(0, \mathbf{I}_B)$, $\omega \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$, and $\mu_\star(\omega) = \mathbf{m}^\top\mathbf{Q}^{-1}\omega$ and $v_\star = \rho - \mathbf{m}^\top\mathbf{Q}^{-1}\mathbf{m}$ with $\rho = {}^1/_d\|\theta_\star\|_2^2$.

Channels

Channel for square loss When the loss is the square loss $\ell(y, \omega) = \frac{1}{2\Delta}(y - \omega)^2$, we can conveniently write the proximal in a matrix form

$$\text{prox}(y, \omega, \mathbf{V}, \mathbf{p}) = \arg\min_{\mathbf{z}\in\mathbb{R}^B} \frac{1}{2}(\mathbf{z} - \omega)^\top\mathbf{V}^{-1}(\mathbf{z} - \omega) + \frac{1}{2\Delta}(\mathbf{z} - \mathbf{1}_B y)^\top\mathbf{P}(\mathbf{z} - \mathbf{1}_B y), \quad \text{(D.48)}$$

where we have defined $\mathbf{P} = \mathrm{Diag}(\mathbf{p})$. In that case, the vector $\mathbf{z}$ that cancels the derivative of the function to minimize is

$$\mathbf{z}_* = \left(\mathbf{V}^{-1} + \frac{\mathbf{P}}{\Delta}\right)^{-1} \left(\mathbf{V}^{-1}\omega + \frac{\mathbf{P}}{\Delta}\mathbf{1}_B y\right) \tag{D.49}$$

such that

$$\mathbf{g}_{\mathrm{out}}(y, \omega, \mathbf{V}, \mathbf{p}) = \left(\mathbf{I}_B + \frac{\mathbf{PV}}{\Delta}\right)^{-1} \frac{\mathbf{P}}{\Delta}(\mathbf{1}_B y - \omega) \tag{D.50}$$

$$\partial_\omega \mathbf{g}_{\mathrm{out}}(y, \omega, \mathbf{V}, \mathbf{p}) = -\left(\mathbf{I}_B + \frac{\mathbf{PV}}{\Delta}\right)^{-1} \frac{\mathbf{P}}{\Delta} \tag{D.51}$$

Channel for logistic loss In classification tasks one usually uses the logistic loss $\ell(y, z) = \log(1 + e^{-z})$. We thus aim to compute the proximal

$$\mathrm{prox}_{\ell(y,\cdot),\mathbf{V}}(\omega) = \underset{\mathbf{z}\in\mathbb{R}^B}{\mathrm{arg\,min}} \sum_{b=1}^{B} p_b \ell(y, z_b) + \frac{1}{2}(\mathbf{z} - \omega)\mathbf{V}^{-1}(\mathbf{z} - \omega) \tag{D.52}$$

We deduce the channel from it. On the other hand, to compute $\partial_\omega \mathbf{g}_{\mathrm{out}}$, one needs to compute the Hessian of the loss function:

$$\nabla^2 \ell(y, \mathbf{z}, \mathbf{p}) = \mathrm{Diag}\big(p_1 \sigma'(yz_1), \dots, p_B \sigma'(yz_B)\big) \tag{D.53}$$

Denoiser for ℓ_2 regularization

In a similar way, the denoiser is written

$$\mathbf{f}_a(\mathbf{b}, \mathbf{A}) = (\lambda \mathbf{I}_B + \mathbf{A})^{-1}\mathbf{b} \tag{D.54}$$

$$\partial_{\mathbf{b}} f_a(\mathbf{b}, \mathbf{A}) = (\lambda \mathbf{I}_B + \mathbf{A})^{-1} \tag{D.55}$$

D.1.2 Ridge regression

Using the channel for square loss and the denoiser for ℓ_2 regularization, we can compute the various overlaps for the ridge regression. First, defining $\mathbf{R}(\lambda) \equiv (\lambda \mathbf{I}_B + \hat{\mathbf{V}})^{-1}$, we find that

$$\mathbf{m} = \mathbb{E}_{\theta_\star, \xi}\left[\mathbf{R}(\lambda)\left(\hat{\mathbf{m}}\theta_\star + \sqrt{\hat{\mathbf{Q}}}\xi\right)\theta_\star\right] = \mathbf{R}(\lambda)\hat{\mathbf{m}}\mathbb{E}_{\theta_\star}\left[\theta_\star^2\right] = \mathbf{R}(\lambda)\hat{\mathbf{m}}\rho \tag{D.56}$$

$$\mathbf{Q} = \mathbb{E}_{\theta_\star, \xi}\left[\mathbf{R}(\lambda)\left(\hat{\mathbf{m}}\theta_\star + \sqrt{\hat{\mathbf{Q}}}\xi\right)\left(\hat{\mathbf{m}}\theta_\star + \sqrt{\hat{\mathbf{Q}}}\xi\right)^\top \mathbf{R}(\lambda)^\top\right] = \mathbf{R}(\lambda)\left(\rho\hat{\mathbf{m}}\hat{\mathbf{m}}^\top + \hat{\mathbf{Q}}\right)\mathbf{R}(\lambda)^\top \tag{D.57}$$

$$\mathbf{V} = \mathbb{E}_{\theta_\star, \xi}\left[\mathbf{R}(\lambda)\right] = \mathbf{R}(\lambda). \tag{D.58}$$

In order to compute the other overlaps, we must first evaluate $\mathcal{Z}_0(y,\mu,v) \equiv \int \mathrm{d}z \mathcal{N}(z|\mu,v)p(y|z)$. Since $p(y|z) = \mathcal{N}(y|z,\Delta)$ for ridge regression, $\mathcal{Z}_0(y,\mu,v)$ is simply the convolution of $\mathcal{N}(y|0,\Delta)$ and $\mathcal{N}(y|\mu,v)$, from which we can conclude $\mathcal{Z}_0(y,\mu,v)$ is equal to the density of $\mathcal{N}(0,\Delta) + \mathcal{N}(\mu,v) = \mathcal{N}(\mu_\star(\omega), v_\star + \Delta)$. Hence, $\mathcal{Z}_0(y,\mu,v) = \mathcal{N}(y|\mu, v+\Delta)$, and we also find that $\partial_\mu \mathcal{Z}_0(y,\mu,v) = \frac{y-\mu}{v+\Delta}\mathcal{N}(y|\mu, v+\Delta)$. Defining $\mathbf{G}(\mathbf{p}) \equiv (\mathbf{I}_2 + \mathbf{PV})^{-1}\mathbf{P}$ with $\mathbf{P} = \mathrm{Diag}(\mathbf{p})$, the overlaps are given by

$$\hat{\mathbf{m}} = \alpha \mathbb{E}_{\omega,\mathbf{p}}\left[\int \mathrm{d}y \mathcal{N}(y|\mu_\star(\omega), v_\star + \Delta)\frac{y-\mu_\star(\omega)}{v_\star+\Delta} G(\mathbf{p})(\mathbf{1}_B y - \omega)\right] \tag{D.59}$$

$$= \alpha \mathbb{E}_{\mathbf{p}}\left[G(\mathbf{p})\right]\mathbb{E}_\omega\left[\int \mathrm{d}y \mathcal{N}(y|\mu_\star(\omega), v_\star + \Delta)\left(\mathbf{1}_B \frac{y^2}{v_\star+\Delta} - \mathbf{1}_B \frac{y\mu_\star(\omega)}{v_\star+\Delta} - \frac{y-\mu_\star(\omega)}{v_\star+\Delta}\omega\right)\right] \tag{D.60}$$

$$= \alpha \mathbb{E}_{\mathbf{p}}\left[G(\mathbf{p})\right]\mathbb{E}_\omega\left[\left(\mathbf{1}_B \frac{v_\star + \Delta + \mu_\star(\omega)^2}{v_\star+\Delta} - \mathbf{1}_B \frac{\mu_\star(\omega)^2}{v_\star+\Delta}\right)\right] \tag{D.61}$$

$$= \alpha \mathbb{E}_{\mathbf{p}}\left[G(\mathbf{p})\right]\mathbf{1}_B \tag{D.62}$$

$$\hat{\mathbf{Q}} = \alpha \mathbb{E}_{\omega,\mathbf{p}}\left[\int \mathrm{d}y \mathcal{N}(y|\mu_\star(\omega), v_\star + \Delta) G(\mathbf{p})(\mathbf{1}_B y - \omega)(\mathbf{1}_B y - \omega)^\top G(\mathbf{p})^\top\right] \tag{D.63}$$

$$= \alpha \mathbb{E}_{\mathbf{p}}\left[G(\mathbf{p})\mathbb{E}_\omega\left[\mathbf{1}_{B\times B}(v_\star + \Delta + \mu_\star(\omega)^2) - \mathbf{1}_B\mu_\star(\omega)\omega^\top - \omega\mathbf{1}_B^\top\mu_\star(\omega) + \omega\omega^\top\right]G(\mathbf{p})^\top\right] \tag{D.64}$$

$$= \alpha \mathbb{E}_{\mathbf{p}}\left[G(\mathbf{p})\left(\mathbf{1}_{B\times B}(v_\star + \Delta + \mathbf{m}^\top Q^{-1}\mathbf{m}) - \mathbf{m}\mathbf{1}_B^\top - \mathbf{1}_B\mathbf{m}^\top + \mathbf{Q}\right)G(\mathbf{p})^\top\right] \tag{D.65}$$

$$= \alpha \mathbb{E}_{\mathbf{p}}\left[G(\mathbf{p})\left(\mathbf{1}_{B\times B}(v_\star + \Delta) + \mathbf{BQB}^\top\right)G(\mathbf{p})^\top\right] \tag{D.66}$$

$$\hat{\mathbf{V}} = -\alpha \mathbb{E}_{\omega,\mathbf{p}}\left[\int \mathrm{d}y \mathcal{N}(y|\mu_\star(\omega), v_\star + \Delta)(-G(\mathbf{p}))\right] = \alpha \mathbb{E}_{\mathbf{p}}\left[G(\mathbf{p})\right], \tag{D.67}$$

where $\mathbf{B} = \mathbf{1}_B\mathbf{m}^\top\mathbf{Q}^{-1} - \mathbf{I}_B$ in eq. (D.66).

Summary

Overall, the closed-form expressions for the state-evolution for ridge regression are

$$\begin{cases} \hat{\mathbf{m}} &= \alpha \mathbb{E}_{\mathbf{p}}\left[\mathbf{G}(\mathbf{p})\right]\mathbf{1}_B \\ \hat{\mathbf{Q}} &= \alpha \mathbb{E}_{\mathbf{p}}\left[\mathbf{G}(\mathbf{p})\left((v_\star+\Delta)\mathbf{1}_{B\times B} + \mathbf{BQB}^\top\right)\mathbf{G}(\mathbf{p})^\top\right] \\ \hat{\mathbf{V}} &= \alpha \mathbb{E}_{\mathbf{p}}\left[\mathbf{G}(\mathbf{p})\right] \end{cases}, \begin{cases} \mathbf{m} &= \rho\mathbf{R}(\lambda)\hat{\mathbf{m}} \\ \mathbf{Q} &= \mathbf{R}(\lambda)\left(\rho\hat{\mathbf{m}}\hat{\mathbf{m}}^\top + \hat{\mathbf{Q}}\right)\mathbf{R}(\lambda)^\top \\ \mathbf{V} &= \mathbf{R}(\lambda) \end{cases} \tag{D.68}$$

with $\mathbf{G}(\mathbf{p}) = (\mathbf{I}_B + \mathbf{PV})^{-1}\mathbf{P}, \mathbf{P} = \mathrm{Diag}(\mathbf{p})$, $\mathbf{B} = \mathbf{1}_B\mathbf{m}^\top\mathbf{Q}^{-1} - \mathbf{I}_B$, and $\mathbf{R}(\lambda) = \left(\lambda\mathbf{I}_B + \hat{\mathbf{V}}\right)^{-1}$, and $v_\star = \rho - \mathbf{m}^\top\mathbf{Q}^{-1}\mathbf{m}$.

D.2 Derivation of the results for residual resampling

As for pair resampling, one can consider the state-evolution equations of a well-chosen AMP algorithm to compute the conditional bias / variance and the bias and variance of residual bootstrap. Indeed, as for pair resampling, we leverage the fact that the conditional bias and variance, together with the estimates by residual bootstrap, can be written in terms of correlations between estimators trained on different resampled datasets $\mathscr{D}_b^\star$ with same covariates $\mathbf{X}$ but resampled labels $y^\star$. Introducing an augmented dataset $\tilde{\mathscr{D}} = (\mathbf{x}_i, \mathbf{y}_i^\star = (y_{b,i}^\star)_{b=1}^B)_{i=1}^n$ where the labels are now B-dimensional vectors comprised of the resampled labels, we see that eq. (5.24) is mathematically equivalent to the following minimization problem

$$(\hat{\theta})_{b=1}^B = \arg \min_{\theta_1,\dots,\theta_B \in \mathbb{R}^d} \sum_{b=1}^B \sum_{i=1}^n -\log p(y_{b,i}^\star | \theta_b^\top \mathbf{x}_i) + \frac{\lambda}{2} \|\theta_b\|^2 \tag{D.69}$$

While eq. (D.69) is equivalent eq. (5.24), formulating it as a joint minimization over B estimators allow us to solve it using a specific AMP algorithm. As for pair resampling, the state-evolution equations of AMP will yield the correlation between two estimators $\mathbb{E}_{\mathscr{D}_b^\star, \mathscr{D}_{b'}^\star}\left[\hat{\theta}(\mathscr{D}_b^\star)^\top \hat{\theta}(\mathscr{D}_{b'}^\star)\right]$ in the high-dimensional limit. These correlations are sufficient to compute the true variance and its estimation with the residual bootstrap, depending on the resampling process $\mathscr{D}^\star$.

For residual bootstrap, the AMP algorithm is similar to algorithm 8 to compute the estimators $\hat{\theta}_i$. The main difference with algorithm 8 is the absence of sample weights p_i, as all the covariates $\mathbf{x}_i$ are resampled only once. Equivalently, we can consider constant sample weights $p_i = 1 \; \forall i$. Moreover, the labels are now B-dimensional.

The overlaps can be computed using the state evolution equations (D.47) of algorithm 8, where the 2-dimensional channel function is

$$\mathbf{g}_{\text{out}}(\mathbf{y}, \omega, \mathbf{V}) = \arg\min_{\mathbf{z} \in \mathbb{R}^B} \frac{1}{2} (\mathbf{z} - \omega)^\top \mathbf{V}^{-1} (\mathbf{z} - \omega) + \sum_{b=1}^B \ell(y_b, z_b) \tag{D.70}$$

Note that here the channel function takes a vector label as input instead of scalar label. Moreover, the channel function does not depend on any sample weight $\mathbf{p}$. This yields the following equations:

$$\begin{cases} \mathbf{m} &= \mathbb{E}_{\theta_\star, \xi}\left[\mathbf{f}_a(\hat{\mathbf{m}}\theta_\star + \sqrt{\hat{\mathbf{Q}}}\xi, \hat{\mathbf{V}})\theta_\star\right] \\ \mathbf{Q} &= \mathbb{E}_{\theta_\star, \xi}\left[\mathbf{f}_a(\hat{\mathbf{m}}\theta_\star + \sqrt{\hat{\mathbf{Q}}}\xi, \hat{\mathbf{V}})\mathbf{f}_a(\hat{\mathbf{m}}\theta_\star + \sqrt{\hat{\mathbf{Q}}}\xi, \hat{\mathbf{V}})^\top\right] \\ \mathbf{V} &= \mathbb{E}_{\theta_\star, \xi}\left[\partial_{\mathbf{b}} \mathbf{f}_a(\hat{\mathbf{m}}\theta_\star + \sqrt{\hat{\mathbf{Q}}}\xi, \hat{\mathbf{V}})\right] \end{cases} \tag{D.71}$$

with $\xi \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_B)$ and

$$\begin{cases} \hat{\mathbf{m}} &= \alpha \mathbb{E}_\omega \left[\int \mathrm{d}\mathbf{y} \partial_\mu \mathcal{Z}_0(\mathbf{y}, \mu_\star(\omega), v_\star) \cdot \mathbf{g}_{\text{out}}(\mathbf{y}, \omega, \mathbf{V}) \right] \\ \hat{\mathbf{Q}} &= \alpha \mathbb{E}_\omega \left[\int \mathrm{d}\mathbf{y} \mathcal{Z}_0(\mathbf{y}, \mu_\star(\omega), v_\star) \cdot \mathbf{g}_{\text{out}}(\mathbf{y}, \omega, \mathbf{V}) \mathbf{g}_{\text{out}}(\mathbf{y}, \omega, \mathbf{V})^\top \right], \\ \hat{\mathbf{V}} &= -\alpha \mathbb{E}_\omega \left[\int \mathrm{d}\mathbf{y} \mathcal{Z}_0(\mathbf{y}, \mu_\star(\omega), v_\star) \cdot \partial_\omega \mathbf{g}_{\text{out}}(\mathbf{y}, \omega, \mathbf{V}) \right] \end{cases} \tag{D.72}$$

where $\omega \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$. Now the integrals in eq. (D.72) carry over vector labels $\mathbf{y}$ and the teacher partition $\mathcal{Z}_0$ is

$$\mathcal{Z}_0(\mathbf{y}, \mu, v) = \int \mathrm{d}z \mathcal{N}(z|\mu, v) \prod_{i=1}^{B} p(y_i|z) \tag{D.73}$$

In Equations (D.71) and (D.72), ρ is the squared norm $^1/_d \|\theta_\star\|^2$ of the label-generating vector $\theta_\star$. In the case of conditional resampling, $\theta_\star = 1$ as for pair resampling. However, in the case of residual bootstrap, $\theta_\star$ is replaced by the ERM estimator $\hat{\theta}_\lambda$, and $\rho = {}^1/_d \|\hat{\theta}_\lambda\|^2$. In the high-dimensional limit, $^1/_d \|\hat{\theta}_\lambda\|^2$ is obtained by running the equations (D.47) for full resampling, and we have $\rho = Q_{11}^{\text{fr}}$.

Ridge regression In the Ridge regression case, the state-evolution equations are given by

$$\begin{cases} \hat{\mathbf{m}} &= \alpha \mathbf{G} \mathbf{1}_B \\ \hat{\mathbf{Q}} &= \alpha \mathbf{G} \left(v_\star \mathbf{1}_{B \times B} + \Delta \mathbf{I}_B + \mathbf{B} \mathbf{Q} \mathbf{B}^\top \right) \mathbf{G}^\top \\ \hat{\mathbf{V}} &= \alpha \mathbf{G} \end{cases}, \begin{cases} \mathbf{m} &= \rho \mathbf{R}(\lambda) \hat{\mathbf{m}} \\ \mathbf{Q} &= \mathbf{R}(\lambda) \left(\rho \hat{\mathbf{m}} \hat{\mathbf{m}}^\top + \hat{\mathbf{Q}} \right) \mathbf{R}(\lambda)^\top \\ \mathbf{V} &= \mathbf{R}(\lambda) \end{cases} \tag{D.74}$$

with $\mathbf{G} = (\mathbf{I}_B + \mathbf{V})^{-1}$, $\mathbf{B} = \mathbf{1}_B \mathbf{m}^\top \mathbf{Q}^{-1} - \mathbf{I}_B$, and $\mathbf{R}(\lambda) = \left(\lambda \mathbf{I}_B + \hat{\mathbf{V}} \right)^{-1}$, and $v_\star = \rho - \mathbf{m}^\top \mathbf{Q}^{-1} \mathbf{m}$. Note that Δ is the variance of the Gaussian noise, which will be 1 for conditional resampling but not for residual bootstrap.

D.2.1 Residual bootstrap

In residual bootstrap, one uses the ERM estimator trained on the whole dataset $\mathscr{D}$ to sample new labels with fixed input data X. Then, to compute the asymptotic behaviour of residual bootstrap, the idea is to solve Equations (D.71) and (D.72) where $\theta_\star$ is replaced by $\hat{\theta}_\lambda$. Its squared norm $\|\theta_\star\|_2^2$ will be replaced by $\|\hat{\theta}_\lambda\|^2$ and, in the case of ridge regression, the noise variance is generally replaced by the training square-loss

$$\hat{\Delta} = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \hat{\theta}_\lambda^\top \mathbf{x}_i \right)^2 \tag{D.75}$$

Note that $\hat{\Delta}$ will typically underestimate Δ as $\hat{\theta}_\lambda$ is correlated to $\mathbf{x}_i$. In practice, to compute the asymptotics of residual bootstrap, we first run the state-evolution equations to compute the (scalar)

overlaps $\mathbf{m}^{\text{fr}}, \mathbf{Q}^{\text{fr}}, \mathbf{V}^{\text{fr}}$ for the ERM estimator. We then plug these overlaps in Equations (D.71) and (D.72), yielding new update equations for $\hat{\mathbf{m}}, \hat{\mathbf{Q}}, \hat{\mathbf{V}}$:

$$\begin{cases} \hat{\mathbf{m}} & = \alpha \mathbb{E}_\omega \left[\int \mathrm{d}\mathbf{y} \partial_\omega \mathcal{Z}_0(\mathbf{y}, \mu_\star(\omega), \tilde{v}_\star) \cdot \mathbf{g}_{\text{out}}(\mathbf{y}, \omega, \mathbf{V}) \right] \\ \hat{\mathbf{Q}} & = \alpha \mathbb{E}_\omega \left[\int \mathrm{d}\mathbf{y} \mathcal{Z}_0(\mathbf{y}, \mu_\star(\omega), \tilde{v}_\star) \cdot \mathbf{g}_{\text{out}}(y, \omega, \mathbf{V}) \mathbf{g}_{\text{out}}(y, \omega, \mathbf{V})^\top \right], \\ \hat{\mathbf{V}} & = -\alpha \mathbb{E}_\omega \left[\int \mathrm{d}\mathbf{y} \mathcal{Z}_0(\mathbf{y}, \mu_\star(\omega), \tilde{v}_\star) \cdot \partial_\omega \mathbf{g}_{\text{out}}(y, \omega, \mathbf{V}) \right] \end{cases} \tag{D.76}$$

where $\omega \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$. Also note that here, $\tilde{v}_\star = Q_{11}^{\text{fr}} - \mathbf{m}^\top \mathbf{Q}^{-1} \mathbf{m}$ as we replaced ρ by Q_{11}^{fr}, and for ridge regression,

$$\mathcal{Z}_0(y, \mu, v) = \int \mathrm{d}z \mathcal{N}(y|z, \tilde{\Delta}) \mathcal{N}(z|\mu, v) = \mathcal{N}(y|\mu, \tilde{\Delta} + v) \tag{D.77}$$

wherein high-dimensions, the ℓ_2 loss of $\hat{\theta}_\lambda$ on the training set $\mathscr{D}$ is $\tilde{\Delta} = \frac{1+\Delta-2m_1^{\text{fr}}+Q_{11}^{\text{fr}}}{(1+V_1^{\text{fr}})^2}$, see [154] for a proof.

D.3 Overlaps and Rates in Ridge Regression

This section is devoted to the simplification of the system of equations in eq. (D.68). Indeed, while the GAMP algorithm can be run with general $B \geq 1$, we can in fact restrict ourselves to the case $B = 2$ without loss of generality. Since our main goal is to compute the correlation between various independent bootstrap resamples and the resamples are i.i.d, the overlaps will have a simple structure that does not depend on B. Once analytical expressions for the overlaps of interest are obtained, the rates of various quantitie like bias and variance are computed in the regime $\alpha \to \infty$.

D.3.1 Solution to the State-Evolution Equations

Let us simplify the system of equations in eq. (D.68) assuming $B = 2$:

Overlaps $\mathbf{V}, \hat{\mathbf{V}}$ Note that the matrices $\mathbf{V}$ and $\hat{\mathbf{V}}$ are diagonal, so that we can denote them as $\mathbf{V} = \mathrm{Diag}(v_1, v_2)$ and $\hat{\mathbf{V}} = \mathrm{Diag}(\hat{v}_1, \hat{v}_2)$. This is due to the fact that the two estimators are independently computed. As such, combining the two equations for $\mathbf{V}$ and $\hat{\mathbf{V}}$ in eq. (D.68), one can write

$$\begin{bmatrix} v_1 & 0 \\ 0 & v_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{\lambda + \alpha \mathbb{E}_{p_1}\left[\frac{p_1}{1+p_1 v_1}\right]} & 0 \\ 0 & \frac{1}{\lambda + \alpha \mathbb{E}_{p_2}\left[\frac{p_2}{1+p_2 v_2}\right]} \end{bmatrix}. \tag{D.78}$$

Hence for $i = 1, 2$, the overlap v_i is given by the fixed-point equation

$$v_i = \frac{1}{\lambda + \alpha \mathbb{E}_{p_i}\left[\frac{p_i}{1+p_i v_i}\right]}. \tag{D.79}$$

Moreover, we have $\hat{v}_i = \alpha \mathbb{E}_{p_i}\left[\frac{p_i}{1+p_i v_i}\right] = \frac{1}{v_i} - \lambda$.

Overlaps $\mathbf{m}, \hat{\mathbf{m}}$ Next, we deduce $\mathbf{m}$ by combining the $\mathbf{m}$ and $\hat{\mathbf{m}}$ expressions from eq. (D.68):

$$\begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = \alpha \begin{bmatrix} \frac{\rho}{\lambda+\hat{v}_1} \mathbb{E}_{p_1}\left[\frac{p_1}{1+p_1 v_1}\right] \\ \frac{\rho}{\lambda+\hat{v}_2} \mathbb{E}_{p_2}\left[\frac{p_2}{1+p_2 v_2}\right] \end{bmatrix} = \begin{bmatrix} \frac{\rho \hat{v}_1}{\lambda+\hat{v}_1} \\ \frac{\rho \hat{v}_2}{\lambda+\hat{v}_2} \end{bmatrix}, \tag{D.80}$$

so that $m_i = \frac{\rho \hat{v}_i}{\lambda+\hat{v}_i} = \rho(1 - \lambda v_i)$, for $i = 1, 2$. Moreover, $\hat{m}_i = \hat{v}_i$.

Overlaps $\mathbf{Q}, \hat{\mathbf{Q}}$ One can leverage the fact that the matrices $\mathbf{Q}, \hat{\mathbf{Q}}$ are symmetric. Using the notation

$$\mathbf{Q} := \begin{bmatrix} q_1 & q_{1,2} \\ q_{1,2} & q_2 \end{bmatrix}, \quad \hat{\mathbf{Q}} := \begin{bmatrix} \hat{q}_1 & \hat{q}_{1,2} \\ \hat{q}_{1,2} & \hat{q}_2 \end{bmatrix} \quad \text{and} \quad \mathbf{Q}^{-1} := \begin{bmatrix} q'_1 & q'_{1,2} \\ q'_{1,2} & q'_2 \end{bmatrix} \tag{D.81}$$

one can rewrite the equation for $\mathbf{Q}$ from eq. (D.68) as

$$\begin{bmatrix} q_1 & q_{1,2} \\ q_{1,2} & q_2 \end{bmatrix} = \begin{bmatrix} \frac{\rho \hat{m}_1^2 + \hat{q}_1}{(\lambda+\hat{v}_1)^2} & \frac{\rho \hat{m}_1 \hat{m}_2 + \hat{q}_{1,2}}{(\lambda+\hat{v}_1)(\lambda+\hat{v}_2)} \\ \frac{\rho \hat{m}_1 \hat{m}_2 + \hat{q}_{1,2}}{(\lambda+\hat{v}_1)(\lambda+\hat{v}_2)} & \frac{\rho \hat{m}_2^2 + \hat{q}_2}{(\lambda+\hat{v}_2)^2} \end{bmatrix} \iff \begin{cases} q_i = \frac{\rho \hat{m}_i^2 + \hat{q}_i}{(\lambda+\hat{v}_i)^2} = \frac{1}{\rho} m_i^2 + v_i^2 \hat{q}_i, \quad \text{for } i = 1,2 \\ q_{1,2} = \frac{\rho \hat{m}_1 \hat{m}_2 + \hat{q}_{1,2}}{(\lambda+\hat{v}_1)(\lambda+\hat{v}_2)} = \frac{1}{\rho} m_1 m_2 + v_1 v_2 \hat{q}_{1,2} \end{cases} \tag{D.82}$$

The computations are slightly more involved for $\hat{\mathbf{Q}}$, but one can derive that

$$\mathbf{B}\mathbf{Q}\mathbf{B}^\top = (m_1^2 q'_1 + 2 m_1 m_2 q'_{1,2} + m_2^2 q'_2)\mathbf{1}_2 + Q - \begin{bmatrix} \mathbf{m}^\top \\ \mathbf{m}^\top \end{bmatrix} - \begin{bmatrix} \mathbf{m} & \mathbf{m} \end{bmatrix} \quad \text{and} \quad v_\star = \rho - (m_1^2 q'_1 + 2 m_1 m_2 q'_{1,2} + m_2^2 q'_2) \tag{D.83}$$

and consequently the equation for $\hat{\mathbf{Q}}$ from eq. (D.68) reads

$$\begin{bmatrix} \hat{q}_1 & \hat{q}_{1,2} \\ \hat{q}_{1,2} & \hat{q}_2 \end{bmatrix} = \alpha \begin{bmatrix} \mathbb{E}_{p_1}\left[(\frac{p_1}{1+p_1 v_1})^2\right](\rho + \Delta - 2m_1 + q_1) & \mathbb{E}_{p_1,p_2}\left[\frac{p_1}{1+p_1 v_1} \cdot \frac{p_2}{1+p_2 v_2}\right](\rho + \Delta - m_1 - m_2 + q_{1,2}) \\ \mathbb{E}_{p_1,p_2}\left[\frac{p_1}{1+p_1 v_1} \cdot \frac{p_2}{1+p_2 v_2}\right](\rho + \Delta - m_1 - m_2 + q_{1,2}) & \mathbb{E}_{p_2}\left[(\frac{p_2}{1+p_2 v_2})^2\right](\rho + \Delta - 2m_2 + q_2) \end{bmatrix} \tag{D.84}$$

$$\iff \begin{cases} \hat{q}_i = \alpha \mathbb{E}_{p_i}\left[\left(\frac{p_i}{1+p_i v_i}\right)^2\right](\rho + \Delta - 2m_i + q_i), \quad \text{for } i = 1,2 \\ \hat{q}_{1,2} = \alpha \mathbb{E}_{p_1,p_2}\left[\frac{p_1}{1+p_1 v_1} \cdot \frac{p_2}{1+p_2 v_2}\right](\rho + \Delta - m_1 - m_2 + q_{1,2}) \end{cases}. \tag{D.85}$$

Combining the equations for q_i and $\hat{q}_i$ just derived, one can compute q_i as

$$q_i = \frac{\frac{1}{\rho} m_i^2 + \alpha \mathbb{E}_{p_i}\left[\left(\frac{p_i v_i}{1+p_i v_i}\right)^2\right](\rho + \Delta - 2m_i)}{1 - \alpha \mathbb{E}_{p_i}\left[\left(\frac{p_i v_i}{1+p_i v_i}\right)^2\right]}, \quad \text{for } i = 1,2 \tag{D.86}$$

and similarly $q_{1,2}$ is given by

$$q_{1,2} = \frac{\frac{1}{\rho} m_1 m_2 + \alpha \mathbb{E}_{p_1,p_2}\left[\frac{p_1 v_1}{1+p_1 v_1} \cdot \frac{p_2 v_2}{1+p_2 v_2}\right](\rho + \Delta - m_1 - m_2)}{1 - \alpha \mathbb{E}_{p_1,p_2}\left[\frac{p_1 v_1}{1+p_1 v_1} \cdot \frac{p_2 v_2}{1+p_2 v_2}\right]}. \tag{D.87}$$

Let us collect these results in the following proposition:

Proposition D.3.1. *Consider two ridge estimators with sampling weights specified by p_1, p_2. The set of self-consistent equations in eq.* (D.68) *gives a characterization of their overlaps in vector/matrix form for pair resampling. Using the notation*

$$\mathbf{V} = \mathrm{Diag}(v_1, v_2), \quad \hat{\mathbf{V}} = \mathrm{Diag}(\hat{v}_1, \hat{v}_2), \quad \mathbf{Q} = \begin{bmatrix} q_1 & q_{1,2} \\ q_{1,2} & q_2 \end{bmatrix}, \quad \hat{\mathbf{Q}} = \begin{bmatrix} \hat{q}_1 & \hat{q}_{1,2} \\ \hat{q}_{1,2} & \hat{q}_2 \end{bmatrix}, \tag{D.88}$$

the overlaps of interest can be simplified as follows: each v_i is the unique solution to the fixed-point equation

$$v_i = \frac{1}{\lambda + \alpha \mathbb{E}_{p_i}\left[\frac{p_i}{1+p_i v_i}\right]}, \tag{D.89}$$

while

$$m_i = \rho(1 - \lambda v_i), \tag{D.90}$$

$$q_i = \frac{\frac{1}{\rho} m_i^2 + \alpha \mathbb{E}_{p_i}\left[\left(\frac{p_i v_i}{1+p_i v_i}\right)^2\right](\rho + \Delta - 2m_i)}{1 - \alpha \mathbb{E}_{p_i}\left[\left(\frac{p_i v_i}{1+p_i v_i}\right)^2\right]}, \tag{D.91}$$

$$q_{1,2} = \frac{\frac{1}{\rho} m_1 m_2 + \alpha \mathbb{E}_{p_1,p_2}\left[\frac{p_1 v_1}{1+p_1 v_1}\cdot\frac{p_2 v_2}{1+p_2 v_2}\right](\rho + \Delta - m_1 - m_2)}{1 - \alpha \mathbb{E}_{p_1,p_2}\left[\frac{p_1 v_1}{1+p_1 v_1}\cdot\frac{p_2 v_2}{1+p_2 v_2}\right]}, \tag{D.92}$$

where $\rho = {}^1/_d \|\theta_\star\|_2^2$ and $\Delta > 0$.

Remark D.3.2. When p_1 and p_2 are identically distributed according to some distribution μ, we get $v_1 = v_2 \equiv v$, $m_1 = m_2 \equiv m$, and $q_1 = q_2 \equiv q$, with

$$\begin{cases} v &= \frac{1}{\lambda + \alpha \mathbb{E}_p\left[\frac{p}{1+pv}\right]} \\ m &= \rho(1 - \lambda v) \\ q &= \frac{\frac{1}{\rho} m^2 + \alpha \mathbb{E}_p\left[\left(\frac{pv}{1+pv}\right)^2\right](\rho + \Delta - 2m)}{1 - \alpha \mathbb{E}_p\left[\left(\frac{pv}{1+pv}\right)^2\right]}, \end{cases} \tag{D.93}$$

where p is a random variable distributed according to μ.

Remark D.3.3. When p_1, p_2 are independent, the overlap q_{12} can be simplified to

$$q_{1,2} = \frac{\frac{1}{\rho} m_1 m_2 + \alpha \mathbb{E}_{p_1}\left[\frac{p_1 v_1}{1+p_1 v_1}\right]\cdot\mathbb{E}_{p_2}\left[\frac{p_2 v_2}{1+p_2 v_2}\right](\rho + \Delta - m_1 - m_2)}{1 - \alpha \mathbb{E}_{p_1}\left[\frac{p_1 v_1}{1+p_1 v_1}\right]\cdot\mathbb{E}_{p_2}\left[\frac{p_2 v_2}{1+p_2 v_2}\right]} = \frac{m_1 m_2(\alpha\rho + \rho + \Delta - m_1 - m_2)}{\alpha\rho^2 - m_1 m_2}. \tag{D.94}$$

Residual Resampling The system of equations for residual resampling in eq. (D.74) is almost identical to eq. (D.68), and in fact simpler as it does not involve expectations. Hence, following the same approach and notation as above, one can solve it to determine the overlaps of interests.

Proposition D.3.4. *Consider two ridge estimators. The set of self-consistent equations in eq.* (D.74) *gives a characterization of their overlaps in vector/matrix form for residual resampling. Using*

the notation

$$\mathbf{V} = \mathrm{Diag}(v_1, v_2), \quad \hat{\mathbf{V}} = \mathrm{Diag}(\hat{v}_1, \hat{v}_2), \quad \mathbf{Q} = \begin{bmatrix} q_1 & q_{1,2} \\ q_{1,2} & q_2 \end{bmatrix}, \quad \hat{\mathbf{Q}} = \begin{bmatrix} \hat{q}_1 & \hat{q}_{1,2} \\ \hat{q}_{1,2} & \hat{q}_2 \end{bmatrix}, \tag{D.95}$$

the overlaps of interest are such that $v \equiv v_1 = v_2$, $m \equiv m_1 = m_2$, $q \equiv q_1 = q_2$. *In particular,* v *is the unique solution to the fixed-point equation*

$$v = \frac{1}{\lambda + \frac{\alpha}{1+v}}, \tag{D.96}$$

while

$$m = \rho(1 - \lambda v), \tag{D.97}$$

$$q = \frac{\frac{1}{\rho} m^2 + \alpha \left(\frac{v}{1+v}\right)^2 (\rho + \Delta - 2m)}{1 - \alpha \left(\frac{v}{1+v}\right)^2} = \frac{m^2(\alpha\rho + \rho + \Delta - 2m)}{\alpha\rho^2 - m^2}, \tag{D.98}$$

$$q_{1,2} = \frac{\frac{1}{\rho} m^2 + \alpha \left(\frac{v}{1+v}\right)^2 (\rho - 2m)}{1 - \alpha \left(\frac{v}{1+v}\right)^2} = \frac{m^2(\alpha\rho + \rho - 2m)}{\alpha\rho^2 - m^2}, \tag{D.99}$$

where $\rho = {}^1/_d \|\theta_\star\|_2^2$ *and* $\Delta > 0$.

Full Resampling Overlaps

To compute overlaps between two independent learners performing ERM on their own dataset, we consider a single dataset of size $2n$ split evenly between the learners. This is achieved by using sampling weights p_1, p_2 with joint distribution given by $\mu(p_1, p_2) = \frac{1}{2}\mathbb{1}\{p_1 = 1, p_2 = 0\} + \frac{1}{2}\mathbb{1}\{p_1 = 0, p_2 = 1\}$. Since p_1, p_2 have the same marginals, theorem D.3.2 applies. Note also that here we are in the high-dimensional regime with ${}^{2n}/_d \to 2\alpha$. With this, the fixed-point equation for v becomes $v = \frac{1}{\lambda + \frac{\alpha}{1+v}}$ and can be solved exactly. Overall, the overlaps are given by

$$\begin{cases} v & = \frac{1-\lambda-\alpha+\sqrt{(\alpha+\lambda-1)^2+4\lambda}}{2\lambda} \\ m & = \rho(1 - \lambda v) \\ q & = \frac{\frac{1}{\rho} m^2 + \alpha \left(\frac{v}{1+v}\right)^2 (\rho+\Delta-2m)}{1 - \alpha \left(\frac{v}{1+v}\right)^2} = \frac{m^2(\alpha\rho+\rho+\Delta-2m)}{\alpha\rho^2 - m^2} \\ q_{1,2} & = \frac{m^2}{\rho} \end{cases} \tag{D.100}$$

by theorem D.3.1. In the following, we refer to these overlaps as $v_i^{\mathrm{fr}}, m_i^{\mathrm{fr}}, q_i^{\mathrm{fr}}$ and $q_{1,2}^{\mathrm{fr}}$.

Residual Resampling Overlaps

The overlaps are given by theorem D.3.4:

$$\begin{cases} v & = \frac{1-\lambda-\alpha+\sqrt{(\alpha+\lambda-1)^2+4\lambda}}{2\lambda} \\ m & = \rho(1-\lambda v) \\ q & = \frac{m^2(\alpha\rho+\rho+\Delta-2m)}{\alpha\rho^2-m^2} \\ q_{1,2} & = \frac{m^2(\alpha\rho+\rho-2m)}{\alpha\rho^2-m^2} \end{cases} \tag{D.101}$$

In the following, we refer to these overlaps as $v_i^{\text{rr}}, m_i^{\text{rr}}, q_i^{\text{rr}}$ and $q_{1,2}^{\text{rr}}$.

Subsampling Overlaps

To compute overlaps between two independent learners that perform subsampling at rate r_1, r_2 of the same dataset, we must consider $p_1 \sim \text{Bern}(r_1)$ and $p_2 \sim \text{Bern}(r_2)$ with p_1 independent of p_2. The fixed-point equations for v_i become $v_i = \frac{1}{\lambda+\frac{\alpha r_i}{1+v_i}}$ and can be solved exactly to yield $v_i = \frac{1-\lambda-\alpha r_i+\sqrt{(\alpha r_i+\lambda-1)^2+4\lambda}}{2\lambda}$ for $i = 1, 2$. Note also that theorem D.3.3 applies here. By theorem D.3.1, we get

$$\begin{cases} v_i & = \frac{1-\lambda-\alpha r_i+\sqrt{(\alpha r_i+\lambda-1)^2+4\lambda}}{2\lambda} \\ m_i & = \rho(1-\lambda v_i) \\ q_i & = \frac{\frac{1}{\rho}m_i^2+\alpha r_i\left(\frac{v_i}{1+v_i}\right)^2(\rho+\Delta-2m)}{1-\alpha r_i\left(\frac{v}{1+v}\right)^2} = \frac{m_i^2(\alpha\rho r_i+\rho+\Delta-2m_i)}{\alpha\rho^2 r_i-m_i^2} \\ q_{1,2} & = \frac{m_1 m_2(\alpha\rho+\rho+\Delta-m_1-m_2)}{\alpha\rho^2-m_1 m_2}, \end{cases} \tag{D.102}$$

for $i = 1, 2$. In the following, we refer to these overlaps as $v_i^{\text{ss}}, m_i^{\text{ss}}, q_i^{\text{ss}}$ and $q_{1,2}^{\text{ss}}$.

Pairs Bootstrap Overlaps

To compute overlaps between two independent learners that perform pairs bootstrap resampling of the same dataset, we must consider $p_1, p_2 \overset{\text{i.i.d.}}{\sim} \text{Poi}(1)$, so that theorem D.3.2 and theorem D.3.3 apply. By theorem D.3.1, the overlaps are thus given by

$$\begin{cases} v & = \frac{1}{\lambda+\alpha\mathbb{E}_p\left[\frac{p}{1+pv}\right]} \\ m & = \rho(1-\lambda v) \\ q & = \frac{\frac{1}{\rho}m^2+\alpha\mathbb{E}_p\left[\left(\frac{pv}{1+pv}\right)^2\right](\rho+\Delta-2m)}{1-\alpha\mathbb{E}_p\left[\left(\frac{pv}{1+pv}\right)^2\right]} \\ q_{1,2} & = \frac{m^2(\alpha\rho+\rho+\Delta-2m)}{\alpha\rho^2-m^2}, \end{cases} \tag{D.103}$$

with $p \sim \text{Poi}(1)$.

Remark D.3.5. For $\lambda > 0$, the variance is thus equal to

$$\widehat{\text{Var}_{\text{pb}}} = q - q_{1,2} = \frac{\frac{1}{\rho} m^2 + \alpha \mathbb{E}_p \left[\left(\frac{pv}{1+pv} \right)^2 \right] (\rho + \Delta - 2m)}{1 - \alpha \mathbb{E}_p \left[\left(\frac{pv}{1+pv} \right)^2 \right]} - \frac{m^2(\alpha\rho + \rho + \Delta - 2m)}{\alpha\rho^2 - m^2}, \quad \text{(D.104)}$$

with v and m defined in eq. (D.103). Setting $\lambda = 0$ (which only makes sense for $\alpha > 1$), the variance becomes

$$\widehat{\text{Var}_{\text{pb}}} = \frac{\rho + \alpha \mathbb{E}_p \left[\left(\frac{pv}{1+pv} \right)^2 \right] (\Delta - \rho)}{1 - \alpha \mathbb{E}_p \left[\left(\frac{pv}{1+pv} \right)^2 \right]} - \frac{\alpha\rho - \rho + \Delta}{\alpha - 1} \quad \text{(D.105)}$$

$$= \Delta \left(\frac{\alpha \mathbb{E}_p \left[\left(\frac{pv}{1+pv} \right)^2 \right]}{1 - \alpha \mathbb{E}_p \left[\left(\frac{pv}{1+pv} \right)^2 \right]} - \frac{1}{\alpha - 1} \right) \quad \text{(D.106)}$$

$$= \Delta \left(\frac{1}{1 - \alpha \mathbb{E}_p \left[\left(\frac{pv}{1+pv} \right)^2 \right]} - \frac{\alpha}{\alpha - 1} \right), \quad \text{(D.107)}$$

where v is the unique solution to the fixed point equation $v = \frac{1}{\alpha \mathbb{E}_p \left[\frac{p}{1+pv} \right]}$. We thus recover Theorem 2 from [131] since this is equivalent to writing

$$\widehat{\text{Var}_{\text{pb}}} = \Delta \left(\frac{\kappa}{1 - \kappa - f(\kappa)} - \frac{1}{1 - \kappa} \right), \quad \text{(D.108)}$$

where $\kappa = \frac{1}{\alpha}$, $f(\kappa) := \mathbb{E}_p \left[\frac{1}{(1+pv)^2} \right]$, and v is the unique solution of $\mathbb{E}_p \left[\frac{1}{1+pv} \right] = 1 - \kappa$.

In the following, we refer to the overlaps as $v_i^{\text{pb}}, m_i^{\text{pb}}, q_i^{\text{pb}}$ and $q_{1,2}^{\text{pb}}$.

Residual Bootstrap Overlaps

To compute overlaps between two independent learners that perform bootstrap resampling, we follow the explanation in appendix D.2.1. It states that the overlaps for the residual bootstrap are given by those of the residual resampling, with ρ replaced by $\tilde{\rho} = q^{\text{fr}}$ and Δ replaced by

$\tilde{\Delta} = \frac{\rho+\Delta-2m^{\mathrm{fr}}+q^{\mathrm{fr}}}{(1+v^{\mathrm{fr}})^2}$. Hence, theorem D.3.4 gives

$$\begin{cases} v & = \frac{1-\lambda-\alpha+\sqrt{(\alpha+\lambda-1)^2+4\lambda}}{2\lambda} \\ m & = \tilde{\rho}(1-\lambda v) \\ q & = \frac{m^2(\alpha\tilde{\rho}+\tilde{\rho}+\tilde{\Delta}-2m)}{\alpha\tilde{\rho}^2-m^2} \\ q_{1,2} & = \frac{m^2(\alpha\tilde{\rho}+\tilde{\rho}-2m)}{\alpha\tilde{\rho}^2-m^2}. \end{cases} \tag{D.109}$$

In the following, we refer to these overlaps as $v_i^{\mathrm{rb}}, m_i^{\mathrm{rb}}, q_i^{\mathrm{rb}}$ and $q_{1,2}^{\mathrm{rb}}$.

Overlaps between Distinct Resampling Methods

Certain quantities of interest require to compute the correlation between two estimators which use different resampling methods. In the high-dimensional regime, this corresponds to the overlap $q_{1,2}$ where the sampling weights p_1, p_2 are independent. In that case, theorem D.3.3 applies and theorem D.3.1 yields

$$\begin{cases} v_i & = \frac{1}{\lambda+\alpha\mathbb{E}_{p_i}\left[\frac{p_i}{1+p_i v_i}\right]} \\ m_i & = \rho(1-\lambda v_i) \\ q_{12} & = \frac{m_1 m_2(\alpha\rho+\rho+\Delta-m_1-m_2)}{\alpha\rho^2-m_1 m_2}, \end{cases} \tag{D.110}$$

for $i = 1, 2$. In particular, the overlap between full resampling and pairs bootstrap is given by

$$q_{1,2}^{\mathrm{fr,pb}} := \frac{m^{\mathrm{fr}} m^{\mathrm{pb}}(\alpha\rho+\rho+\Delta-m^{\mathrm{fr}}-m^{\mathrm{pb}})}{\alpha\rho^2-m^{\mathrm{fr}} m^{\mathrm{pb}}}, \tag{D.111}$$

the overlap between full resampling and subsampling at rate r is given by

$$q_{1,2}^{\mathrm{fr,ss}} := \frac{m^{\mathrm{fr}} m^{\mathrm{ss}}(\alpha\rho+\rho+\Delta-m^{\mathrm{fr}}-m^{\mathrm{ss}})}{\alpha\rho^2-m^{\mathrm{fr}} m^{\mathrm{ss}}}. \tag{D.112}$$

D.3.2 Large α rates

In this section, we compute the rates of quantities of interest (variances, biases) in the $\alpha \to \infty$ limit, which are summarized in table 5.1. The approach is mathematically standard: for each overlap, we compute its series expansion at $\alpha \to \infty$ up to a desired order. Let us illustrate this with an example.

Consider the full resampling overlap v^{fr} computed in appendix D.3.1:

$$v^{\mathrm{fr}} = \frac{1-\lambda-\alpha+\sqrt{(\alpha+\lambda-1)^2+4\lambda}}{2\lambda}. \tag{D.113}$$

To compute its series expansion at $\alpha \to \infty$, we substitute α with $1/\beta$ in the equation above, and then compute its Taylor series at $\beta \to 0$. Letting

$$h(\beta) := \frac{1 - \lambda - \frac{1}{\beta} + \sqrt{(\frac{1}{\beta} + \lambda - 1)^2 + 4\lambda}}{2\lambda}, \tag{D.114}$$

one can apply this strategy and determine the Taylor expansion up to order 2 for v^{fr} by evaluating

$$\begin{aligned}
\lim_{\beta\to 0} h(\beta) &= \lim_{\beta\to 0} \frac{\beta(1-\lambda) - 1 + \sqrt{(\beta(\lambda-1)+1)^2 + 4\lambda\beta^2}}{2\lambda\beta} = 0 \\
\lim_{\beta\to 0} h'(\beta) &= \lim_{\beta\to 0} \frac{\frac{1}{\beta^2} - \frac{((\frac{1}{\beta}+\lambda-1)\frac{1}{\beta^2}}{\sqrt{(\frac{1}{\beta}+\lambda-1)^2+4\lambda}}}{2\lambda} = 1 \\
\lim_{\beta\to 0} h''(\beta) &= \lim_{\beta\to 0} \frac{-\frac{2}{\beta^3} + \frac{2\left(\frac{1}{\beta}+\lambda-1\right)}{\beta^3\sqrt{\left(\frac{1}{\beta}+\lambda-1\right)^2+4\lambda}} + \frac{1}{\beta^4\sqrt{\left(\frac{1}{\beta}+\lambda-1\right)^2+4\lambda}} - \frac{\left(\frac{1}{\beta}+\lambda-1\right)^2}{\beta^4\left(\left(\frac{1}{\beta}+\lambda-1\right)^2+4\lambda\right)^{3/2}}}{2\lambda} = 2(1-\lambda),
\end{aligned} \tag{D.115}$$

(D.116)

(D.117)

from which we conclude that for $\beta \to 0$,

$$h(\beta) = h(\beta) + h'(\beta)\beta + \frac{1}{2}h''(\beta)\beta^2 + O(\beta^3) = \beta + (1-\lambda)\beta^2 + O(\beta^3) \tag{D.118}$$

or equivalently, substituting back $\alpha = 1/\beta$,

$$v^{\text{fr}} = \frac{1}{\alpha} + \frac{1-\lambda}{\alpha^2} + O\left(\frac{1}{\alpha^3}\right) \tag{D.119}$$

for $\alpha \to \infty$. The computation of all overlaps are carried out in the same fashion, and we use the Mathematica software [241] to automate these computations.

Full Resampling Rates

From the overlaps computed in appendix D.3.1, we retrieve the limiting behaviors

$$\begin{cases} v^{\text{fr}} & \overset{\alpha\to\infty}{\simeq} \frac{1}{\alpha} + \frac{1-\lambda}{\alpha^2} + O\left(\frac{1}{\alpha^3}\right) \\ m^{\text{fr}} & \overset{\alpha\to\infty}{\simeq} \rho - \frac{\rho\lambda}{\alpha} + \frac{\rho\lambda(\lambda-1)}{\alpha^2} + O\left(\frac{1}{\alpha^3}\right) \\ q^{\text{fr}} & \overset{\alpha\to\infty}{\simeq} \rho + \frac{\Delta-2\lambda\rho}{\alpha} + \frac{\Delta(1-2\lambda)+\rho\lambda(3\lambda-2)}{\alpha^2} + O\left(\frac{1}{\alpha^3}\right) \\ q^{\text{fr}}_{1,2} & \overset{\alpha\to\infty}{\simeq} \rho - \frac{2\rho\lambda}{\alpha} + \frac{\rho\lambda(3\lambda-2)}{\alpha^2} + O\left(\frac{1}{\alpha^3}\right), \end{cases} \tag{D.120}$$

so that the variance is given by

$$\text{Var}_{\mathscr{D}}(\hat{\theta}_\lambda) = q^{\text{fr}} - q^{\text{fr}}_{1,2} \overset{\alpha\to\infty}{\simeq} \frac{\Delta}{\alpha} + O\left(\frac{1}{\alpha^2}\right) \tag{D.121}$$

and the bias is

$$\text{Bias}^2_{\mathscr{D}}(\hat{\theta}_\lambda) = \rho + q^{\text{fr}}_{1,2} - 2m^{\text{fr}} \overset{\alpha\to\infty}{\simeq} \frac{\rho\lambda^2}{\alpha^2} + O\left(\frac{1}{\alpha^3}\right). \tag{D.122}$$

Residual Resampling Rates

From the overlaps computed in appendix D.3.1, we retrieve the limiting behaviors

$$\begin{cases} v^{\text{rr}} & \overset{\alpha\to\infty}{\simeq} \frac{1}{\alpha} + \frac{1-\lambda}{\alpha^2} + O\left(\frac{1}{\alpha^3}\right) \\ m^{\text{rr}} & \overset{\alpha\to\infty}{\simeq} \rho - \frac{\rho\lambda}{\alpha} + \frac{\rho\lambda(\lambda-1)}{\alpha^2} + O\left(\frac{1}{\alpha^3}\right) \\ q^{\text{rr}} & \overset{\alpha\to\infty}{\simeq} \rho + \frac{\Delta-2\rho\lambda}{\alpha} + \frac{\Delta(1-2\lambda)+\lambda(3\lambda-2)}{\alpha^2} + O\left(\frac{1}{\alpha^3}\right) \\ q^{\text{rr}}_{1,2} & \overset{\alpha\to\infty}{\simeq} \rho - \frac{2\rho\lambda}{\alpha} + \frac{\rho\lambda(3\lambda-2)}{\alpha^2} + O\left(\frac{1}{\alpha^3}\right), \end{cases} \tag{D.123}$$

so that the variance is given by

$$\text{Var}_{\mathscr{D}|\mathbf{X}}(\hat{\theta}_\lambda) = q^{\text{rr}} - q^{\text{rr}}_{1,2} \overset{\alpha\to\infty}{\simeq} \frac{\Delta}{\alpha} + O\left(\frac{1}{\alpha^2}\right) \tag{D.124}$$

and the bias is

$$\text{Bias}^2_{\mathscr{D}|\mathbf{X}}(\hat{\theta}_\lambda) = \rho + q^{\text{rr}}_{1,2} - 2m^{\text{rr}} \overset{\alpha\to\infty}{\simeq} \frac{\rho\lambda^2}{\alpha^2} + O\left(\frac{1}{\alpha^3}\right). \tag{D.125}$$

Rates of Overlaps between Distinct Resampling Methods

From the overlaps computed in appendix D.3.1, we retrieve the limiting behaviors

$$\begin{cases} q^{\text{fr,ss}}_{1,2} & \overset{\alpha\to\infty}{\simeq} \rho + \frac{r\Delta-\rho\lambda(r+1)}{r\alpha} + \frac{r^2\Delta+\rho\lambda(\lambda+r(\lambda+(\lambda-1)r)-1)-r\Delta\lambda(r+1)}{r^2\alpha^2} + O\left(\frac{1}{\alpha^3}\right) \\ q^{\text{fr,pb}}_{1,2} & \overset{\alpha\to\infty}{\simeq} \rho + \frac{\Delta-2\lambda\rho}{\alpha} + \frac{\Delta(1-2\lambda)+3\rho\lambda(\lambda-1)}{\alpha^2} + O\left(\frac{1}{\alpha^3}\right). \end{cases} \tag{D.126}$$

Subsampling and Jackknife Rates

From the overlaps computed in appendix D.3.1, we retrieve the limiting behaviors

$$\begin{cases} v_i^{\text{ss}} & \overset{\alpha\to\infty}{\simeq} \frac{1}{r_i\alpha} + \frac{1-\lambda}{r_i^2\alpha^2} + O\left(\frac{1}{\alpha^3}\right) \\ m_i^{\text{ss}} & \overset{\alpha\to\infty}{\simeq} \rho - \frac{\rho\lambda}{r_i\alpha} + \frac{\rho\lambda(\lambda-1)}{r_i^2\alpha^2} + O\left(\frac{1}{\alpha^3}\right) \\ q_i^{\text{ss}} & \overset{\alpha\to\infty}{\simeq} \rho + \frac{\Delta-2\rho\lambda}{r_i\alpha} + \frac{\Delta(1-2\lambda)+\rho\lambda(3\lambda-2)}{r_i^2\alpha^2} + O\left(\frac{1}{\alpha^3}\right) \\ q_{1,2}^{\text{ss}} & \overset{\alpha\to\infty}{\simeq} \rho + \frac{\Delta r_1 r_2 r - 2\rho\lambda}{r_1 r_2 \alpha} + \frac{\Delta + \frac{(\lambda-1)\lambda\rho}{r_1^2} + \frac{\lambda(\lambda\rho-\Delta r_2)}{r_1 r_2} + \frac{(\lambda-1)\lambda\rho}{r_2^2} - \frac{\Delta\lambda}{r_2}}{\alpha^2} + O\left(\frac{1}{\alpha^3}\right), \end{cases} \tag{D.127}$$

so that the variance when subsampling at rate $r_1 = r_2 \equiv r$ is given by

$$\widehat{\text{Var}_{\text{ss}}} = \frac{q^{\text{ss}} - q_{1,2}^{\text{ss}}}{1-r} \overset{\alpha\to\infty}{\simeq} \frac{\Delta}{\alpha r} + O\left(\frac{1}{\alpha^2}\right). \tag{D.128}$$

and the bias is

$$\widehat{\text{Bias}_{\text{ss}}^2} = \frac{q_{1,2}^{\text{ss}} + q^{\text{fr}} - 2q_{1,2}^{\text{fr,ss}}}{(1-r)^2} \overset{\alpha\to\infty}{\simeq} = \frac{\rho\lambda^2}{\alpha^2 r^2} + O\left(\frac{1}{\alpha^3}\right). \tag{D.129}$$

The Jackknife variances and biases are computed by taking the limit $r \to 1$, and we get

$$\widehat{\text{Var}_{\text{jk}}} = \lim_{r\to 1} \frac{q^{\text{ss}} - q_{1,2}^{\text{ss}}}{1-r} \overset{\alpha\to\infty}{\simeq} \frac{\Delta}{\alpha} + O\left(\frac{1}{\alpha^2}\right). \tag{D.130}$$

and

$$\widehat{\text{Bias}_{\text{jk}}^2} = \lim_{r\to 1} \frac{q_{1,2}^{\text{ss}} + q^{\text{fr}} - 2q_{1,2}^{\text{fr,ss}}}{(1-r)^2} \overset{\alpha\to\infty}{\simeq} = \frac{\rho\lambda^2}{\alpha^2} + O\left(\frac{1}{\alpha^3}\right). \tag{D.131}$$

Pairs Bootstrap Rates

The computation of rates in this case are less straightforward given that the overlaps depend on the evaluation of various expectations (see appendix D.3.1). Let us consider v^{pb} first, which is given by the fixed-point equation

$$v^{\text{pb}} = \frac{1}{\lambda + \alpha\mathbb{E}_p\left[\frac{p}{1+pv^{\text{pb}}}\right]}. \tag{D.132}$$

We use the Ansatz that v^{pb} behaves as $1/\alpha$ in the $\alpha \to \infty$ limit, and hence write it as $v^{\text{pb}} = \frac{\tilde{v}}{\alpha}$. Since $\frac{1}{1+x} = 1 - x + O(x^2)$ for $x \to 0^+$, we get

$$\tilde{v} = \frac{\alpha}{\lambda + \alpha \mathbb{E}_p\left[\frac{p}{1+\frac{p\tilde{v}}{\alpha}}\right]} \approx \frac{\alpha}{\lambda + \alpha \mathbb{E}_p\left[p(1-\frac{p\tilde{v}}{\alpha})\right]} = \frac{\alpha}{\lambda + \alpha - 2\tilde{v}}. \tag{D.133}$$

This can be solved exactly and

$$\tilde{v} = \frac{\alpha + \lambda - \sqrt{(\alpha+\lambda)^2 - 8\alpha}}{4} \Rightarrow v^{\text{pb}} = \frac{\alpha + \lambda - \sqrt{(\alpha+\lambda)^2 - 8\alpha}}{4\alpha} \overset{\alpha\to\infty}{\simeq} \frac{1}{\alpha} + \frac{2-\lambda}{\alpha^2} + O\left(\frac{1}{\alpha^3}\right). \tag{D.134}$$

Overlaps m^{pb} and $q_{1,2}^{\text{pb}}$ are thus given by

$$m^{\text{pb}} \overset{\alpha\to\infty}{\simeq} \rho - \frac{\rho\lambda}{\alpha} + \frac{\rho\lambda(\lambda-2)}{\alpha^2} + O\left(\frac{1}{\alpha^3}\right) \tag{D.135}$$

$$q_{1,2}^{\text{pb}} \overset{\alpha\to\infty}{\simeq} \rho + \frac{\Delta - 2\rho\lambda}{\alpha} + \frac{\Delta(1-2\lambda) + \rho\lambda(3\lambda-4)}{\alpha^2} + O\left(\frac{1}{\alpha^3}\right). \tag{D.136}$$

Overlap q^{pb} involves the evaluation of $\mathbb{E}_p\left[\left(\frac{pv^{\text{pb}}}{1+pv^{\text{pb}}}\right)^2\right]$, which can be computed using the same approximation as in eq. (D.133):

$$\mathbb{E}_p\left[\left(\frac{pv^{\text{pb}}}{1+pv^{\text{pb}}}\right)^2\right] \approx \mathbb{E}_p\left[\left(pv^{\text{pb}}(1-pv^{\text{pb}})\right)^2\right] \tag{D.137}$$

$$= \mathbb{E}_p\left[(pv^{\text{pb}})^2 - 2(pv^{\text{pb}})^3 + (pv^{\text{pb}})^4\right] \tag{D.138}$$

$$= 2(v^{\text{pb}})^2 - 10(v^{\text{pb}})^3 + 15(v^{\text{pb}})^4, \tag{D.139}$$

where the last equality is obtained since $p \sim \text{Pois}(1)$. This yields

$$q^{\text{pb}} \overset{\alpha\to\infty}{\simeq} 1 + \frac{2(\Delta - \rho\lambda)}{\alpha} + \frac{2\Delta(1-2\lambda) + \rho\lambda(3\lambda-4)}{\alpha^2} + O\left(\frac{1}{\alpha^3}\right). \tag{D.140}$$

so that the variance in the $\alpha \to \infty$ limit is thus given by

$$\widehat{\text{Var}_{\text{pb}}} = q^{\text{pb}} - q_{1,2}^{\text{pb}} \overset{\alpha\to\infty}{\simeq} \frac{\Delta}{\alpha} + O\left(\frac{1}{\alpha^2}\right) \tag{D.141}$$

and the bias is

$$\widehat{\text{Bias}^2_{\text{pb}}} = q_{1,2}^{\text{pb}} + q^{\text{fr}} - 2q_{1,2}^{\text{fr,pb}} \overset{\alpha\to\infty}{\simeq} = \frac{\rho\lambda^2}{\alpha^4} + O\left(\frac{1}{\alpha^5}\right). \tag{D.142}$$

Residual Bootstrap Rates

From the overlaps computed in appendix D.3.1, we retrieve the limiting behaviors

$$\begin{cases} v^{\mathrm{rb}} & \overset{\alpha\to\infty}{\simeq} \frac{1}{\alpha} + \frac{1-\lambda}{\alpha^2} + O\left(\frac{1}{\alpha^3}\right) \\ m^{\mathrm{rb}} & \overset{\alpha\to\infty}{\simeq} \rho + \frac{\Delta-3\rho\lambda}{\alpha} + \frac{\Delta(1-3\lambda)+3\rho\lambda(2\lambda-1)}{\alpha^2} + O\left(\frac{1}{\alpha^3}\right) \\ q^{\mathrm{rb}} & \overset{\alpha\to\infty}{\simeq} \rho + \frac{2(\Delta-2\lambda\rho)}{\alpha} + \frac{\Delta(1-6\lambda)+2\rho\lambda(5\lambda-2)}{\alpha^2} + O\left(\frac{1}{\alpha^3}\right) \\ q^{\mathrm{rb}}_{1,2} & \overset{\alpha\to\infty}{\simeq} \rho + \frac{\Delta-4\rho\lambda}{\alpha} + \frac{\Delta(1-4\lambda)+2\rho\lambda(5\lambda-2)}{\alpha^2} + O\left(\frac{1}{\alpha^3}\right), \end{cases} \tag{D.143}$$

so that the variance is

$$\widehat{\mathrm{Var}_{\mathrm{rb}}} = q^{\mathrm{rb}} - q^{\mathrm{rb}}_{1,2} \overset{\alpha\to\infty}{\simeq} \frac{\Delta}{\alpha} + O\left(\frac{1}{\alpha^2}\right) \tag{D.144}$$

and the bias is

$$\widehat{\mathrm{Bias}^2_{\mathrm{rb}}} = q^{\mathrm{rb}}_{1,2} + q^{\mathrm{fr}} - 2m^{\mathrm{rb}} \overset{\alpha\to\infty}{\simeq} \frac{\rho\lambda^2}{\alpha^2} + O\left(\frac{1}{\alpha^3}\right). \tag{D.145}$$

Differences between Rates

Recall that pairs bootstrap and subsampling aim to estimate bias and variance with respect to the joint distribution $p_\theta(y,\mathbf{x})$, while residual bootstrap seeks to estimate the bias and variance with respect to the conditional distribution $p_\theta(y|\mathbf{x})$. To understand how good each estimate of the bias and variance is, we compute for each resampling method the difference between their estimate and the true value. For the variances, this results in

$$\begin{aligned} \left|\widehat{\mathrm{Var}_{\mathrm{ss}}} - \mathrm{Var}_{\mathscr{D}}(\hat{\theta}_\lambda)\right| &\overset{\alpha\to\infty}{\simeq} \frac{\Delta(1-r)}{\alpha r} + \frac{\Delta\left((1-2\lambda)(1-r^2)+r\right)}{\alpha^2 r^2} + O\left(\frac{1}{\alpha^3}\right) \\ \left|\widehat{\mathrm{Var}_{\mathrm{jk}}} - \mathrm{Var}_{\mathscr{D}}(\hat{\theta}_\lambda)\right| &\overset{\alpha\to\infty}{\simeq} \frac{\Delta}{\alpha^2} + O\left(\frac{1}{\alpha^3}\right) \\ \left|\widehat{\mathrm{Var}_{\mathrm{pb}}} - \mathrm{Var}_{\mathscr{D}}(\hat{\theta}_\lambda)\right| &\overset{\alpha\to\infty}{\simeq} \frac{\Delta(4\lambda+7)}{\alpha^3} + O\left(\frac{1}{\alpha^4}\right) \\ \left|\widehat{\mathrm{Var}_{\mathrm{rb}}} - \mathrm{Var}_{\mathscr{D}|\mathbf{X}}(\hat{\theta}_\lambda)\right| &\overset{\alpha\to\infty}{\simeq} \frac{\Delta}{\alpha^2} + O\left(\frac{1}{\alpha^3}\right) \end{aligned}$$

while the biases are given by

$$\left|\widehat{\text{Bias}^2_{\text{ss}}} - \text{Bias}^2_{\mathscr{D}}(\hat{\theta}_\lambda)\right| \overset{\alpha\to\infty}{\simeq} \frac{\rho\lambda^2(r^2-1)}{r^2\alpha^2} + \frac{\lambda^2\left(\rho\left(2\lambda - 2(\lambda-1)r^3 - (3-2\lambda)r - 2\right) - \Delta r\right)}{r^3\alpha^3} + O\left(\frac{1}{\alpha^4}\right)$$

$$\left|\widehat{\text{Bias}^2_{\text{jk}}} - \text{Bias}^2_{\mathscr{D}}(\hat{\theta}_\lambda)\right| \overset{\alpha\to\infty}{\simeq} \frac{\lambda^2(\rho(2\lambda-3)-\Delta)}{\alpha^3} + O\left(\frac{1}{\alpha^4}\right)$$

$$\left|\widehat{\text{Bias}^2_{\text{pb}}} - \text{Bias}^2_{\mathscr{D}}(\hat{\theta}_\lambda)\right| \overset{\alpha\to\infty}{\simeq} \frac{\rho\lambda^2}{\alpha^2} + O\left(\frac{1}{\alpha^3}\right)$$

$$\left|\widehat{\text{Bias}^2_{\text{rb}}} - \text{Bias}^2_{\mathscr{D}|\mathbf{X}}(\hat{\theta}_\lambda)\right| \overset{\alpha\to\infty}{\simeq} \frac{\lambda^2(2\lambda\rho-\Delta)}{\alpha^3} + O\left(\frac{1}{\alpha^4}\right).$$

D.4 Asymptotics of prediction variance

The focus of our work is the variance of estimators with respect to the resampling of the training set. However, one can also be interested in computing the *prediction variance*, often defined as

$$\mathrm{Var}_{\mathbf{x},y}\left(y - \hat{y}(\mathbf{x})\right) \tag{D.146}$$

where now the training set is fixed, and the variance is taken with respect to the new test sample $\mathbf{x}, y$. In a linear model where $\hat{y} = \hat{\theta}_\lambda^\top \mathbf{x}$ and in our setting defined in eq. (5.1), the prediction variance is equal to the test error of the ERM estimator. Indeed :

$$\begin{aligned}\mathrm{Var}_{\mathbf{x},y}\left(y - \hat{y}(\mathbf{x}) \,|\mathscr{D}\right) &= \mathbb{E}\left[(y - \hat{\theta}_\lambda^\top \mathbf{x})^2\right] + \mathbb{E}\left[(y - \hat{\theta}_\lambda^\top \mathbf{x})\right]^2 &\text{(D.147)}\\ &= \mathbb{E}\left[(y - \hat{\theta}_\lambda^\top \mathbf{x})^2\right] = \varepsilon_g &\text{(D.148)}\end{aligned}$$

because $\mathbb{E}\left[(y - \hat{\theta}_\lambda^\top \mathbf{x})\right]^2 = 0$. In the case of Ridge regression,

$$\varepsilon_g = \rho - 2m^{\mathrm{fr}} + Q_{11}^{\mathrm{fr}} + \sigma^2. \tag{D.149}$$

Note that at optimal $\lambda = \sigma^2$ ($\lambda = 1$ in our case), the performance of the ERM estimator is equal the posterior variance of the Bayes-optimal, as

$$\begin{aligned}\mathrm{Var}_{\mathrm{bo}} &= \rho - q^{\mathrm{bo}} &\text{(D.150)}\\ &= \rho - 2m^{\mathrm{bo}} + q^{\mathrm{bo}} &\text{(D.151)}\\ &= \rho - 2m^{\mathrm{fr}} + Q_{11}^{\mathrm{fr}}, &\text{(D.152)}\end{aligned}$$

where eq. (D.151) follows from the *Nishimori condition* $m^{\mathrm{bo}} = q^{\mathrm{bo}}$, and eq. (D.152) is due to the fact that $\hat{\theta}_\lambda = \mathbb{E}[\theta|\mathscr{D}]$ for optimal λ.

D.5 Additional Details for Numerical Experiments

The state evolution equations for the resampling methods are written in the Julia language [36] and are available on the Github repository `https://github.com/SPOC-group/BootstrapAsymptotics` that also contains the code used to reproduce the plots. The code leverages libraries such as `NLSolvers.jl` for optimization [175], `QuadGK.jl` and `HCubature.jl` for integration [117, 118, 92], `MLJLinearModels.jl` for estimation of GLMs [120], as well as various utilities for statistical functions [123, 122], performance [121] and plotting [39]. The code to compute the posterior variance of the Bayes-optimal estimator is written in Rust and is available at `https://github.com/spoc-group/double_descent_uncertainty`. All the experiments were run on a computer with the following specifications: 16 GB RAM, Apple M1 Pro CPU.

D.5.1 Effects of finite B

In section 5.4, we studied the behavior of resampling methods in the limit $B \to \infty$. However, in practice B is usually not very large, and the finiteness of B has an impact on the estimated bias and variances. Indeed :

$$\widehat{\mathrm{Var}} = \frac{1}{dB}\sum_{b=1}^{B}\left\|\hat{\theta}_b - \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}_b\right\|^2 = \frac{1}{dB}\sum_{b=1}^{B}\|\hat{\theta}_b - \mathbb{E}_{\mathscr{D}^\star}\left[\hat{\theta}\right]\|^2 + \frac{1}{d}\|\mathbb{E}_{\mathscr{D}^\star}\left[\hat{\theta}\right] - \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}_b\|^2$$

where second term vanishes as $B \to \infty$. Note that our framework allows us to compute the $\widehat{\mathrm{Var}}(B)$ for a finite number of Bootstrap resamples B, as we get asymptotically

$$\widehat{\mathrm{Var}}(B) = \frac{B-1}{B}\lim_{B\to\infty}\widehat{\mathrm{Var}}$$

where $\widehat{\mathrm{Var}}$ is the variance plotted in fig. 5.1 and fig. 5.3.

Likewise, the estimator of the bias with finite B can be computed and equates

$$\widehat{\mathrm{Bias}}(B) = \widehat{\mathrm{Bias}} + \frac{1}{B}\widehat{\mathrm{Var}}$$

where $\frac{1}{B}\widehat{\mathrm{Var}}$ is due to finite sampling and vanishes as $B \to \infty$. Note that the overlaps computed with our state-evolution equations allow us to compute $\widehat{\mathrm{Bias}}(B)$ at any B.

E Appendix of chapter 7

E.1 Approximate Message chgr to approximate leave-one-out residuals

E.1.1 Introduction of relaxed-Belief Propagation and Approximate Message Passing

In this section, we explain how AMP can be used to compute the leave-one-out residuals used in eq. (7.4). The naive way to compute these residuals is to fit the leave-one-out estimators $\hat{\theta}_{-i}(y)$ for each sample $1 \leqslant i \leqslant n$ and each possible label y, which requires $n \times |\mathscr{Y}|$ fits, with $\mathscr{Y}$ the set of candidate labels, typically a discretization of $\mathbb{R}$. We will first see that AMP can be used to compute all the $\hat{\theta}_{-i}$

To introduce AMP, we first consider the following problem. Consider a dataset $\mathscr{D} = \left(\mathbf{x}_i, y_i\right)_{i=1}^n$ of size n. Assume that the data is generated from the model (7.17), where the input $\mathbf{x}_i \in \mathbb{R}^d$ are sampled according to $\mathcal{N}(\mathbf{0}, I_d/d)$, and the labels are generated from a *teacher* as $y \sim p(y|{\theta_\star}^\top \mathbf{x})$. Our goal is to sample the following distribution

$$p(\theta) = \frac{1}{Z} \prod_{i=1}^{n} P_{out}\left(y_i|\theta^\top \mathbf{x}_i\right) \prod_{\mu=1}^{d} P_\theta(\theta_\mu) \tag{E.1}$$

The empirical risk minimization problem (7.2) introduced in section 7.2 is a particular instance of eq. (E.1) where

$$P_{out}(y|z) \propto e^{-\beta \ell(y,z)}, \qquad P_\theta(z) \propto e^{-\beta r(z)} \tag{E.2}$$

in the limit $\beta \to \infty$. The starting point of approximate message passing is the writing of the belief-propagation algorithm for the graph associated with eq. (E.1), where the variable-nodes of the graph are the coordinates θ_μ and the factor nodes, representing the interaction between the variable-nodes, are the observations y_i. The message passing consists in iterating messages

$m_{\mu\to i}$ from variable to factor-nodes and $m_{i\to\mu}$ from factor to variable-nodes. These messages read

$$m_{\mu\to i}(\theta_\mu) = \frac{1}{z_{i\to\mu}} P_\theta(\theta_\mu) \prod_{j\neq i} m_{j\to\mu}(\theta_\mu) \tag{E.3}$$

$$m_{i\to\mu}(\theta_\mu) = \frac{1}{z_{\mu\to i}} \int \prod_{\nu\neq\mu} \mathrm{d}\theta_\nu m_{\nu\to i} P_{out}\left(y_i | \sum_\nu \mathbf{x}_{i\nu}\theta_\nu\right) \tag{E.4}$$

This messages give access to the distribution $p(\theta)$ and in particular this marginals : indeed, the marginal distribution $p(\theta_\mu)$ is given by

$$p(\theta_\mu) = \frac{1}{z_\mu} P_\theta(\theta_\mu) \prod_{i=1}^{n} m_{i\to\mu}(\theta_\mu) \tag{E.5}$$

where z_μ is a normalization constant. Iterating eq. (E.4) is not tractable, especially in high-dimensions as it involves $(d-1)$ integrals to update each $m_{i\to\mu}$. To make these equations tractable, one can use relaxed-Belief Propagation (rBP), which relies on the central limit theorem and the projection of the messages on their first two moments. We thus define the *cavity mean* $\hat{\theta}_{\mu\to i}$ and *cavity variance* $\hat{v}_{\mu\to i}$ as

$$\hat{\theta}_{\mu\to i} = \int \mathrm{d}\theta_\mu \theta_\mu m_{\mu\to i}(\theta_\mu) \tag{E.6}$$

$$\hat{v}_{\mu\to i} = \int \mathrm{d}\theta_\mu \theta_\mu^2 m_{\mu\to i}(\theta_\mu) - \hat{\theta}^2_{\mu\to i} \tag{E.7}$$

In particular, the vector $\left(\hat{\theta}_{\mu\to i}\right)_{\mu=1}^{d}$ represents the mean of the marginals of distribution (E.1) in the absence of the i-th sample. In the context of empirical risk minimization, this is exactly the leave-one-out estimator $\hat{\theta}_{-i}$ defined as

$$\hat{\theta}_{-i} = \arg\min_\theta \sum_{j\neq i} \ell\left(y_j, \theta^\top \mathbf{x}_j\right) + \sum_{\mu=1}^{d} r(\theta_\mu) \tag{E.8}$$

Our goal is thus to compute efficiently the cavity means and use them to compute the leave-one-out residuals.

rBP The main idea behind rBP is to iteratively compute the cavity means and variances, to obtain the desired marginal mean and variance of θ. We define $\omega_{i\to\mu}, V_{i\to\mu}$ the mean and variance of the messages $m_{i\to\mu}$ and $\hat{\theta}_{\mu\to i}, \hat{v}_{\mu\to i}$ the mean and variance of $m_{\mu\to i}$.

We detail rBP in algorithm 9, and refer to [247, Chapter VI, Section C] for a detailed explanation of the algorithm. In particular, the algorithm makes use of the *channel* and *denoising* functions $\mathbf{g}_{\text{out}}$ and $\mathbf{f}_a$ functions, defined respectively as

$$\mathbf{g}_{\text{out}}(y,\omega,V) = \frac{\partial \log \mathcal{Z}_y(y,\omega,V)}{\partial\omega}, \qquad \mathcal{Z}_y(y,\omega,V) = \int \mathrm{d}z P_{out}(y|z) e^{-\frac{1}{2V}(z-\omega)^2} \tag{E.9}$$

and

$$\mathbf{f}_a(b,A) = \frac{\partial \log \mathcal{Z}_w(b,A)}{\partial b}, \qquad \mathcal{Z}_w(b,A) = \int \mathrm{d}x P_\theta(x) e^{bx - \frac{A}{2}x^2} \tag{E.10}$$

In the case of empirical risk minization (7.2), using the prior and likelihood from eq. (E.2) into the definitions(E.9) and (E.10) and taking the limit $\beta \to \infty$ yields Equation (7.11).

From rBP to AMP Note that in rBP, we iterate over $n \times d$ means and variances $\omega_{i\to\mu}, V_{i\to\mu}, \hat{\theta}_{\mu\to i}, \hat{v}_{\mu\to i}$, which scales quadratically with the dimension in the high-dimensional limit where $n, d \to \infty$ with a constant sampling ratio $n/d = \alpha$. However, a key observation is that the quantities $\hat{\theta}_{\mu\to i}, \hat{v}_{\mu\to i}$ only weakly depend on μ, and similarly $\omega_{i\to\mu}, V_{i\to\mu}$ weakly depend on μ. Hence, let us define

$$\begin{cases} \omega_i &= \sum_\mu \mathbf{x}_{i\mu} \hat{\theta}_{\mu\to i} \\ V_i &= \sum_\mu \mathbf{x}_{i\mu}^2 \hat{v}_{\mu\to i} \end{cases}, \qquad \begin{cases} A_\mu &= -\sum_{i=1}^n \partial_\omega g_{out}\left(y_i, \omega_i, V_i\right) \mathbf{x}_{i\mu}^2 \\ b_\mu &= \sum_{i=1}^n g_{out}\left(y_i, \omega_{i\to\mu}, V_{i\to\mu}\right) \mathbf{x}_{i\mu} \end{cases} \tag{E.11}$$

note that for all μ and all i, in the high-dimensonal limit considered here we have

$$\omega_i = \omega_{i\to\mu} + \mathbf{x}_{i\mu}\hat{\theta}_{\mu\to i} = \omega_{i\to\mu} + O(1/\sqrt{n}) \tag{E.12}$$

$$V_i = V_{i\to\mu} + \mathbf{x}_{i\mu}^2 \hat{v}_{\mu\to i} = V_{i\to\mu} + O(1/n) \tag{E.13}$$

As a consequence, we have for all μ and all i

$$A_\mu = -\sum_{j=1}^n \mathbf{x}_{j\mu}^2 \partial_\omega g_{out}\left(y_j, \omega_j, V_j\right) = \sum_{j=1}^n \mathbf{x}_{j\mu}^2 \left[\partial_\omega g_{out}\left(y_j, \omega_{j\to\mu}, V_{j\to\mu}\right) + O(1/\sqrt{n})\right] \tag{E.14}$$

$$= -\sum_{j=1}^n \mathbf{x}_{j\mu}^2 \partial_\omega g_{out}\left(y_j, \omega_{j\to\mu}, V_{j\to\mu}\right) + O(1/\sqrt{n}) \tag{E.15}$$

$$= -\sum_{j\neq i}^n \mathbf{x}_{j\mu}^2 \partial_\omega g_{out}\left(y_j, \omega_{j\to\mu}, V_{j\to\mu}\right) + O(1/\sqrt{n}) \tag{E.16}$$

$$= -A_{\mu\to i} + O(1/\sqrt{n}) \tag{E.17}$$

Similarly, we get

$$b_\mu = b_{\mu\to i} + O(1/\sqrt{n}) \tag{E.18}$$

So that one can simply compute the estimator $\theta = \mathbf{f}_a(\mathbf{b}, \mathbf{A})$. The challenge is to compute the vectors $\omega, \mathbf{V}, \mathbf{b}$. To do so, we note that

$$\mathbf{g}_{\text{out}}\left(y_i, \omega_{i\to\mu}, V_{i\to\mu}\right) = \mathbf{g}_{\text{out}}\left(y_i, \omega_i, V_i\right) - \mathbf{x}_{i\mu}\hat{\theta}_{\mu\to i}\partial_\omega \mathbf{g}_{\text{out}}\left(y_i, \omega_{i\to\mu}, V_{i\to\mu}\right) + O(1/n) \tag{E.19}$$

$$\tag{E.20}$$

such that

$$b_\mu = \sum_{i=1}^{n} \mathbf{x}_{i\mu} \mathbf{g}_{\text{out}}\left(y_i, \omega_i, V_i\right) - \sum_i \mathbf{x}_{i\mu}^2 \hat{\theta}_\mu \partial_\omega \mathbf{g}_{\text{out}}\left(y_i, \omega_i, V_i\right) + O\left(1/\sqrt{n}\right) \tag{E.21}$$

$$\tag{E.22}$$

Moreover,

$$\omega_i = \sum_{\mu=1}^{d} \mathbf{x}_{i\mu} \hat{\theta}_{\mu \to i} = \sum_\mu \mathbf{x}_{i\mu} \left(\hat{\theta}_\mu - \mathbf{x}_{i\mu} v_\mu \mathbf{g}_{\text{out}}\left(y_i, \omega_i, V_i\right)\right) + O(1/n) \tag{E.23}$$

$$\tag{E.24}$$

These iterative equations are, in the leading order, the same as those shown in algorithm 5. In the high-dimensional regime, these iteratives coincide with rBP. Going from rBP to AMP, we have reduced the number of variables to iterate on from $O(n \times d)$ to $O(n + d)$, and can still recover the marginal distribution by

$$\hat{\theta}_\mu = \mathbf{f}_a\left(b_\mu, A_\mu\right) \tag{E.25}$$

E.1.2 Recovering the leave-one-out estimators from AMP

For each sample i, computing the leave-one-out estimator $\hat{\theta}_{-i}$ means computing the marginals of the distribution

$$p(\theta) = \frac{1}{Z} \prod_{j \neq i} P_{out}\left(y_j | \theta^\top \mathbf{x}_j\right) \prod_{\mu=1}^{d} P_\theta\left(\theta_\mu\right) \tag{E.26}$$

with P_{out} and P_θ defined in eq. (E.2) and where the sample $(\mathbf{x}_i, y_i)$ is removed from the data. Our method leverages the fact that these marginals are computed iteratively by relaxed-BP and stored in the variables $\hat{\theta}_{\mu \to i}$. Indeed, each $\hat{\theta}_{\mu \to i}$ stores the posterior mean of θ_{mu} when the interaction node i is removed from the graph associated to eq. (E.1), which corresponds exactly to the distribution of eq. (E.26). While rBP explicitly computes these quantities, its computational complexity makes it unusable. Instead, we will recover these estimators from AMP. Indeed, at the leading order we have :

$$\hat{\theta}_{\mu \to i} = \mathbf{f}_a\left(b_{\mu \to i}, A_{\mu \to i}\right) = \mathbf{f}_a\left(b_{\mu \to i}, A_\mu\right) + O\left(1/n\right) \tag{E.27}$$

$$= \mathbf{f}_a\left(b_\mu, A_\mu\right) - b_{i \to \mu} \partial_b \mathbf{f}_a\left(b_\mu, A_\mu\right) + O\left(1/n\right) = \hat{\theta}_\mu - \mathbf{g}_{\text{out}}(y_i, \omega_i, V_i) \mathbf{x}_{i\mu} \hat{\mathbf{v}}_\mu + O\left(1/n\right) \tag{E.28}$$

The expression on the right-hand side corresponds to the approximation of the leave-one-out estimators $\hat{\theta}_{-i,\text{gamp}}$ used in algorithm 5.

Algorithm 9 relaxed-Belief Propagation

repeat

Input: Dataset $\mathscr{D} = \left(\mathbf{x}_i, y_i\right)_{i=1}^n$

$$\begin{cases} V_{i\to\mu}^t & = \sum_{\nu\neq\mu} \mathbf{x}_{i\mu}^2 v_{\nu\to i}^{t-1} \\ \omega_{i\to\mu}^t & = \sum_{\nu\neq\mu} \mathbf{x}_{i\mu} \hat{\theta}_{\nu\to i}^{t-1} \end{cases} \tag{E.29}$$

$$\begin{cases} A_{\mu\to i}^t & = -\sum_{j\neq i} \partial_\omega \mathbf{g}_{\text{out}}\left(y_j, \omega_{j\to\mu}^t, V_{j\to\mu}\right) \mathbf{x}_{j\mu}^2 \\ b_{\mu\to i}^t & = \sum_{j\neq i} \mathbf{g}_{\text{out}}\left(y_j, \omega_{j\to\mu}^t, V_{j\to\mu}\right) \mathbf{x}_{j\mu} \end{cases} \tag{E.30}$$

$$\hat{\theta}_{\mu\to i}^t = \mathbf{f}_a\left(b_{\mu\to i}^t, A_{\mu\to i}^t\right) \tag{E.31}$$

$$\hat{v}_{\mu\to i}^t = \partial_b \mathbf{f}_a\left(b_{\mu\to i}^t, A_{\mu\to i}^t\right) \tag{E.32}$$

until Convergence of $\hat{\theta}_{\mu\to i}, \hat{v}_{\mu\to i}$

Return $\hat{\theta}, \hat{\mathbf{v}}$ **such that :**

$$\hat{\theta}_\mu = \mathbf{f}_a\left(\sum_i b_{\mu\to i}, \sum_i A_{\mu\to i}\right) \tag{E.33}$$

$$\hat{v}_\mu = \partial_b \mathbf{f}_a\left(\sum_i b_{\mu\to i}, \sum_i A_{\mu\to i}\right) \tag{E.34}$$

Convergence of the leave-one-out residuals in high-dimensions Under the assumptions (7.17), we see from eq. (E.28) thatin the high-dimensional limit the leave-one-out estimators computed by AMP will converge to the exact ones at a $O(1/n)$ rate. As such, for a given test sample $\mathbf{x}, y$ the approximated residuals $y - \mathbf{x}^\top \theta_{-i,\text{gamp}}$ will converge to $y - \mathbf{x}^\top \theta_{-i}$ at a $O(1/\sqrt{n})$ rate. This implies that asymptotically the prediction intervals built using the AMP leave-one-out converge to the prediction intervals with the exact residuals.

Applying AMP without Gaussian assumptions We thus see that from AMP, we get an approximation of the leave-one-out estimator that can be used to compute the residuals in eq. (7.4). The derivations performed in this section were done under the assumption that the input data are Gaussian with i.i.d. covariance and $1/d$ variance. However, AMP can be applied on any data, with no guarantee a priori on its performance.

E.2 Derivation ot Taylor-AMP

In this section, we derive the Taylor-AMP algorithm. Our starting point is AMP, derived in Appendix E.1. In what follow, we consider a dataset $\mathscr{D}$ of size $n+1$ to stay consistent with the notation of the main text. Our goal is to compute the variation of the $\hat{\theta}_{-i}$ to the first order with respect to the last label y_{n+1}. To this end, we will write the vectors defined in AMP $\hat{\theta}(y), \hat{\mathbf{v}}(y), \mathbf{g}(y), \partial\mathbf{g}(y), \mathbf{b}(y), \mathbf{A}(y), \omega(y), \mathbf{V}(y)$ as functions of $y_{n+1} = y$

For the sake of conciseness, let us define the vector

$$\Omega\left(y\right) = \left(\hat{\theta}(y), \hat{\mathbf{v}}(y), \omega(y), \mathbf{V}(y), \mathbf{g}(y), \partial\mathbf{g}(y), \mathbf{b}(y), \mathbf{A}(y)\right) \in \mathbb{R}^{4\times(d+n)} \tag{E.35}$$

Then, $\Omega(y)$ is the fixed point of the equation

$$\Omega(y) = \mathbf{f}_{\text{gamp}}(\Omega(y), y)$$

where the function $\mathbf{f}_{\text{gamp}}(\Omega) = \left(f^{\hat{\theta}}_{\text{gamp}}, f^{\hat{\mathbf{v}}}_{\text{gamp}}, f^{\omega}_{\text{gamp}}, f^{\mathbf{V}}_{\text{gamp}}, f^{\mathbf{g}}_{\text{gamp}}, f^{\partial\mathbf{g}}_{\text{gamp}}, f^{\mathbf{b}}_{\text{gamp}}, f^{\partial\mathbf{A}}_{\text{gamp}}\right)$ is defined as

$$\begin{cases} \mathbf{f}^{\hat{\theta}}_{\text{gamp}} &= \mathbf{f}_a(\mathbf{b}, \mathbf{A}) \\ \mathbf{f}^{\hat{\mathbf{v}}}_{\text{gamp}} &= \partial_b \mathbf{f}_a(\mathbf{b}, \mathbf{A}) \\ \mathbf{f}^{\omega}_{\text{gamp}} &= X\hat{\theta} - \mathbf{V} \odot \mathbf{g} \\ \mathbf{f}^{\mathbf{V}}_{\text{gamp}} &= X^2\hat{\mathbf{v}} \\ \mathbf{f}^{\mathbf{g}}_{\text{gamp}} &= \mathbf{g}_{\text{out}}\left(\mathbf{y}, \omega, \mathbf{V}\right) \\ \mathbf{f}^{\partial\mathbf{g}}_{\text{gamp}} &= \partial_\omega \mathbf{g}_{\text{out}}\left(\mathbf{y}, \omega, \mathbf{V}\right) \\ \mathbf{f}^{\mathbf{b}}_{\text{gamp}} &= X^\top \mathbf{g} + \mathbf{A} \odot \hat{\theta} \\ \mathbf{f}^{\partial\mathbf{A}}_{\text{gamp}} &= -X^{2\top}\partial\mathbf{g} \end{cases} \tag{E.36}$$

Equivalently, we have $\Omega(y) - \mathbf{f}_{\text{gamp}}(\Omega(y), y) = \mathbf{0}$. Under the assumption that the function $\Omega(y)$ is differentiable, one can use the implicit function theorem around a value $\hat{y}$ to write

$$\frac{\partial\Omega}{\partial y}\left(\hat{y}\right) = \left(\mathbf{I} - Jac\left(\mathbf{f}_{\text{gamp}}\right)\right)^{-1} \frac{\partial\mathbf{f}_{\text{gamp}}}{\partial y}\left(\hat{y}\right) \tag{E.37}$$

$$\Leftrightarrow \frac{\partial\Omega}{\partial y}\left(\hat{y}\right) = Jac\left(\mathbf{f}_{\text{gamp}}\right)\left(\frac{\partial\Omega}{\partial y}\left(\hat{y}\right)\right) + \frac{\partial\mathbf{f}_{\text{gamp}}}{\partial y}\left(\hat{y}\right) \tag{E.38}$$

From the last equality we find that we can compute the derivative $\frac{\partial\Omega}{\partial y}\left(\hat{y}\right)$ by iterating the following system of linear equations over a vector $\Delta\Omega^t$:

$$\Delta\Omega^{t+1} = Jac\left(\mathbf{f}_{\text{gamp}}\right)\left(\Delta\Omega^t\right) + \frac{\partial\mathbf{f}_{\text{gamp}}}{\partial y}\left(\hat{y}\right) \tag{E.39}$$

The jacobian of the function $\mathbf{f}_{\text{gamp}}$ is written

$$\begin{cases} Jac f^{\hat{\theta}}_{\text{amp}} &= (\mathbf{0},\mathbf{0},\mathbf{0},\mathbf{0},\partial_b \mathbf{f}_a(\mathbf{b},\mathbf{A}),\partial_A \mathbf{f}_a(\mathbf{b},\mathbf{A})) \\ Jac f^{\hat{\mathbf{v}}}_{\text{amp}} &= (\mathbf{0},\mathbf{0},\mathbf{0},\mathbf{0},\partial_b\partial_b \mathbf{f}_a(\mathbf{b},\mathbf{A}),\partial_A\partial_b \mathbf{f}_a(\mathbf{b},\mathbf{A})) \\ Jac f^{\omega}_{\text{amp}} &= \big(X,\mathbf{0},\mathbf{0},-Diag(\mathbf{g}),-Diag(\mathbf{V}),\mathbf{0},\mathbf{0},\mathbf{0}\big) \\ Jac f^{\mathbf{V}}_{\text{amp}} &= \big(\mathbf{0},X^2,\mathbf{0},\mathbf{0},\mathbf{0},\mathbf{0}\big) \\ Jac f^{\mathbf{g}}_{\text{amp}} &= \big(\mathbf{0},\mathbf{0},\partial_\omega \mathbf{g},\partial_V \mathbf{g},\mathbf{0},\mathbf{0},\mathbf{0},\mathbf{0}\big) \\ Jac f^{\partial\mathbf{g}}_{\text{amp}} &= \big(\mathbf{0},\mathbf{0},\partial_\omega\partial_\omega \mathbf{g},\partial_V\partial_\omega \mathbf{g},\mathbf{0},\mathbf{0}\big) \\ Jac f^{\mathbf{b}}_{\text{amp}} &= \big(Diag(\mathbf{A}),\mathbf{0},\mathbf{0},\mathbf{0},X^\top,\mathbf{0},\mathbf{0},Diag(\mathbf{w})\big) \\ Jac f^{\mathbf{A}}_{\text{amp}} &= \big(\mathbf{0},\mathbf{0},\mathbf{0},\mathbf{0},-X^{2\top},\mathbf{0},\mathbf{0},\mathbf{0}\big) \end{cases} \tag{E.40}$$

and the derivative $\frac{\partial \mathbf{f}_{\text{gamp}}}{\partial y}$ with respect to the last label is

$$\begin{cases} \partial_y f^{\theta}_{\text{amp}} &= \mathbf{0} \\ \partial_y f^{\mathbf{v}}_{\text{amp}} &= \mathbf{0} \\ \partial_y f^{\omega}_{\text{amp}} &= \mathbf{0} \\ \partial_y f^{\mathbf{V}}_{\text{amp}} &= \mathbf{0} \\ \partial_y f^{\mathbf{g}}_{\text{amp}} &= \big(0,\cdots,0,\partial_y g(y_n,\omega_n,V_n)\big) \\ \partial_y f^{\partial\mathbf{g}}_{\text{amp}} &= \big(0,\cdots,0,\partial_y\partial_\omega g(y_n,\omega_n,V_n)\big) \\ \partial_y f^{\mathbf{b}}_{\text{amp}} &= \mathbf{0} \\ \partial_y f^{\mathbf{A}}_{\text{amp}} &= \mathbf{0} \end{cases}$$

When writing Equation (E.39) with the expression of the Jacobian of Equation (E.40), one obtains the iterations of Taylor-AMP in Algorithm 6.

E.2.1 Justification of Taylor-AMP

As stated in the previous subsection, Taylor-AMP is based on the assumption that the function $y \to \Omega\big)y\big)$ is differentiable. Our underlying assumption behind Taylor-AMP is that the leave-one-out residuals only weakly depend on the last label in high-dimensions. We numerically justify this assumption in fig. E.1. In this Figure, we compare the leave-one-out residuals obtained by computing the estimators $\hat{\theta}_{-i}$ exactly and with Taylor-AMP for different settings. To do so, we sample a dataset $\mathscr{D}$ at random. We use algorithm 5 and algorithm 6 to compute the $\hat{\theta}_{-i,\text{gamp}}(y_n)$ and $\Delta\hat{\theta}_{-i,\text{gamp}}(y)$ as prescribed above. Then, we change the last label $y_n \to y_n + \delta y$ with $\delta y = 5$. After this change we compute the leave-one-estimators exactly $\hat{\theta}_{-i}(y_n + \delta y)$ and use our linear approximation $\hat{\theta}_{-i,\text{gamp}}(y + \delta y) = \hat{\theta}_{-i}(y) + \delta y \Delta\hat{\theta}_{-i.\text{gamp}}(y)$. We then compare $\hat{\theta}_{-i}(y_n + \delta y)^\top \mathbf{x}_i$ and our approximation $\hat{\theta}_{-i,\text{gamp}}(y_n + \delta y)^\top \mathbf{x}_i$ that is used to compute our conformity scores. As

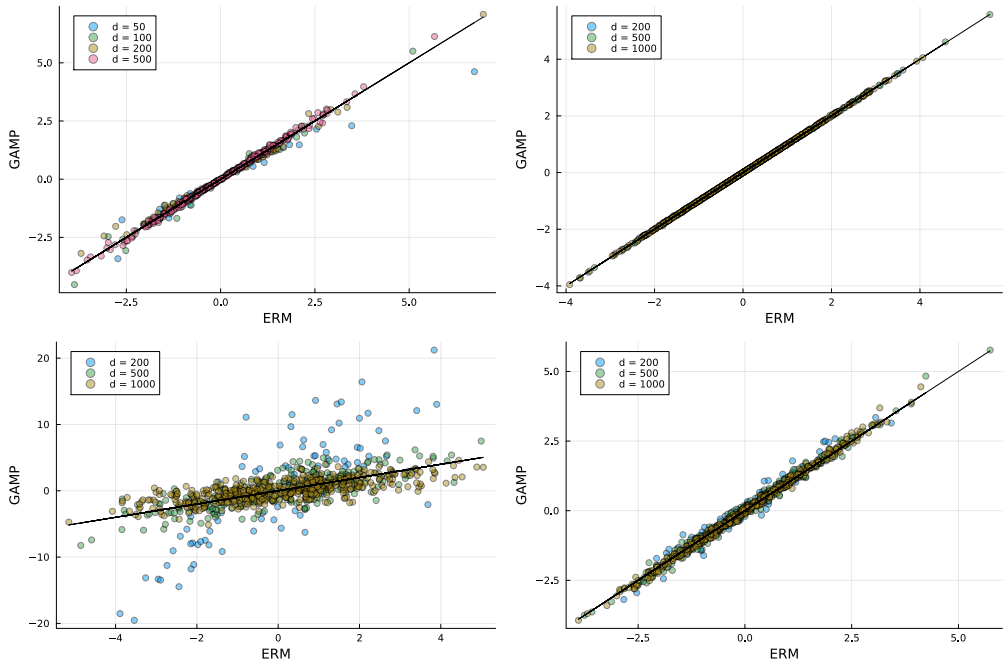


Figure E.1: Comparison of the leave-one-out estimators computed exactly by solving eq. (7.3) and by Taylor-AMP , for Ridge (top row) and Lasso (bottom row), as $\lambda = 0.01$ (left column) and $\lambda = 1$ (right column). All plots are at $n/d = 0.5$

we observe in the figure, at high dimensions $d = 1000$, our approximations are very close to the true values, meaning that Taylor-AMP will accurately estimate the scores (hence the prediction intervals) of FCP.

We note however from the lower-left plot that at moderate dimension, Taylor-AMP does not precisely approximates the leave-one-out residuals for the LASSO, which partly explains the mediocre results obtained by Taylor-AMP on real data in table 7.4 in the main.

E.3 Coverage guarantee for AMP

First, we show that AMP is symmetric : indeed, consider a permutation $s : [1, n] \to [1, n]$ and S the corresponding permutation matrix defined as $S_{ij} = \delta\left(j = s(i)\right)$. Then, consider running AMP on the permutated data $\tilde{X} = SX$ and labels $\tilde{y} = SY$. At each iteration t, the channel vectors $\tilde{\mathbf{g}}^t, \tilde{\partial \mathbf{g}}^t$ $\tilde{\mathbf{g}}^t = S\mathbf{g}^t$ and $\tilde{\partial \mathbf{g}}^t = S\partial \mathbf{g}^t$. Then, the vectors $\mathbf{b}^t, \mathbf{A}^t$ now become

$$\begin{cases} \tilde{\mathbf{A}}^t &= -\tilde{X^2}^\top \tilde{\partial \mathbf{g}}^t = -X^{2\top} S^T S \mathbf{g}^t = \mathbf{A}^t \\ \tilde{\mathbf{b}}^t &= \tilde{X}^\top \tilde{\mathbf{g}}^t + \tilde{\mathbf{A}}^t \otimes \hat{\theta}^t = X^\top S^T S \mathbf{g}^t + \tilde{\mathbf{A}}^t \otimes \hat{\theta}^t = \mathbf{b}^t \end{cases} \tag{E.41}$$

and by recursion we deduce that the estimator of AMP $(\hat{\theta}, \hat{\mathbf{v}})$ given after convergence is invariant under permutation. Then, the scores computed from eq. (7.13) are symmetric. Then, under the

assumption that the data $(\mathbf{x}_i, y_i)$ is exchangeable, we obtain theorem 7.3.1 : in expectation over the training and test data

$$\mathbb{P}_{\mathscr{D},\mathbf{x}}\left(y \in \mathscr{S}(\mathbf{x})\right) \geqslant 1-\kappa \tag{E.42}$$

E.4 Details on real datasets

In this section, we provide details on the datasets used in table 7.4. We use the Boston housing dataset (availa containing 506 samples at dimension 14, and the Riboflavin dataset [42] of 71 samples at dimension 4088. Both datasets were normalized and split randomly into training and test data, with 80 percent of the data being in the training set. The coverage, average time and interval size are obtained by computing the average on the test set, and averaging over 20 random train / test splits.

For table 7.4, approximate homotopy was used with the default parameters provided by the authors. The same data preprocessing was applied as for our method to compare fairly the size of the prediction intervals.

E.5 Full conformal prediction for classification

In chapter 7, we focused on the case of regression, however conformal prediction has been successfully applied for classification tasks [9, 10]. Consider a classification task with k classes, where a predictor estimate the probabilities $p_1(\mathbf{x}), \cdots, p_n(\mathbf{x})$. Then, the conformity scores are defined as

$$\sigma_i = \sum_{k=1}^{\pi^{-1}(y)} p_{\pi(1)} \tag{E.43}$$

where π is a permutation that ranks the classes by decreasing order of probability, i.e $p_{\pi(1)} > \cdots > p_{\pi(K)}$. In words, the score is the sum of the probability of all the classes whose p_i is higher of equal to the true observed class.

Binary classification In the context of generalized linear model, one might train an estimator using the cross entropy loss with an L_2 regularizer. For $K = 2$ classes, this is logistic regression

$$\hat{\theta} = \arg\min_{\theta} -\sum_{i=1}^{n} \log\left(1 + e^{-y_i \times \mathbf{x}_i^\top \theta}\right) + \lambda/2 \|\theta\|^2 \tag{E.44}$$

As for regression, one can use AMP and Taylor-AMP with the adequate channel and denoising function to estimate $\hat{\theta}$, and compute the leave-one-out estimators using eq. (7.13). For the logistic

loss, the channel function is defined as

$$\mathbf{g}_{\text{out}}(y,\omega,V)=\frac{\text{prox}\ell_{\omega,V}(y,\cdot)-\omega}{V},\qquad \text{prox}\ell_{\omega,V}(y,\cdot)=\arg\min_{z}\ell(y,z)+\frac{1}{2V}(z-\omega)^2 \quad \text{(E.45)}$$

Bibliography

[1] Ehsan Abbasi, Fariborz Salehi, and Babak Hassibi. "Universality in Learning from Linear Measurements". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[2] Moloud Abdar et al. "A review of uncertainty quantification in deep learning: Techniques, applications and challenges". In: *Information Fusion* 76 (2021), pp. 243–297.

[3] Ben Adlam and Jeffrey Pennington. "Understanding Double Descent Requires A Fine-Grained Bias-Variance Decomposition". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 11022–11032.

[4] Ben Adlam, Jasper Snoek, and Samuel L. Smith. "Cold Posteriors and Aleatoric Uncertainty". In: *ArXiv* abs/2008.00029 (2020).

[5] Laurence Aitchison. "A statistical theory of cold posteriors in deep neural networks". In: *International Conference on Learning Representations*. 2021.

[6] Michael Aizenman, Robert Sims, and Shannon L Starr. "Mean-Field Spin Glass models from the Cavity–ROSt Perspective". In: *arXiv preprint math-ph/0607060* (2006).

[7] Antonios Alexos, Alex J Boyd, and Stephan Mandt. "Structured Stochastic Gradient MCMC". In: *Proceedings of the 39th International Conference on Machine Learning*. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 414–434.

[8] Ryo Ando and Fumiyasu Komaki. "On High-Dimensional Asymptotic Properties of Model Averaging Estimators". In: 2023.

[9] Anastasios N Angelopoulos et al. "Learn then test: Calibrating predictive algorithms to achieve risk control". In: *arXiv preprint arXiv:2110.01052* (2021).

[10] Anastasios N. Angelopoulos and Stephen Bates. "A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification". In: (2022).

[11] Gabriel Arpino, Xiaoqi Liu, and Ramji Venkataramanan. "Inferring Change Points in High-Dimensional Linear Regression via Approximate Message Passing". In: *Proceedings of the 41st International Conference on Machine Learning*. Vol. 235. Proceedings of Machine Learning Research. PMLR, 2024, pp. 1841–1864.

[12] Benjamin Aubin et al. “Exact asymptotics for phase retrieval and compressed sensing with random generative priors”. In: *Proceedings of The First Mathematical and Scientific Machine Learning Conference*. Vol. 107. Proceedings of Machine Learning Research. PMLR, 2020, pp. 55–73.

[13] Benjamin Aubin et al. “Generalization error in high-dimensional perceptrons: approaching Bayes error with convex optimization”. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS’20. Vancouver, BC, Canada: Curran Associates Inc., 2020.

[14] Benjamin Aubin et al. “The committee machine: computational to statistical gaps in learning a two-layers neural network”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2019.12 (2019), p. 124023.

[15] Benjamin Aubin et al. “The Spiked Matrix Model With Generative Priors”. In: *IEEE Transactions on Information Theory* 67.2 (2021), pp. 1156–1181.

[16] Y. Bai et al. “Don’t Just Blame Over-parametrization for Over-confidence: Theoretical Analysis of Calibration in Binary Classification”. In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. PMRL. PMLR, 2021, pp. 566–576.

[17] Yu Bai et al. “Understanding the Under-Coverage Bias in Uncertainty Estimation”. In: *arXiv:2106.05515 [cs.LG]* 34 (2021), pp. 18307–18319.

[18] Zhidong Bai and Wang Zhou. “Large sample covariance matrices without independence structures in columns”. In: *Statist. Sinica* 18.2 (2008), pp. 425–442. ISSN: 1017-0405.

[19] Rina Foygel Barber et al. “Predictive inference with the jackknife+”. In: *The Annals of Statistics* (2019).

[20] Jean Barbier and Nicolas Macris. “The adaptive interpolation method: a simple scheme to prove replica formulas in Bayesian inference”. In: *Probability theory and related fields* 174 (2019).

[21] Jean Barbier, Dmitry Panchenko, and Manuel Sáenz. “Strong replica symmetry for high-dimensional disordered log-concave Gibbs measures”. In: *Information and Inference: A Journal of the IMA* 11.3 (2021), pp. 1079–1108. ISSN: 2049-8772.

[22] Jean Barbier et al. “Optimal Errors and Phase Transitions in High-Dimensional Generalized Linear Models”. In: *Proceedings of the National Academy of Sciences* 116.12 (Mar. 19, 2019), pp. 5451–5460. ISSN: 0027-8424, 1091-6490.

[23] Jean Barbier et al. “Performance of Bayesian linear regression in a model with mismatch”. In: *arXiv preprint arXiv:2107.06936* (2021).

[24] Jean Barbier et al. “The mutual information in random linear estimation beyond iid matrices”. In: *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE. 2018, pp. 1390–1394.

[25] Peter L. Bartlett et al. “Benign overfitting in linear regression”. In: *Proceedings of the National Academy of Sciences* 117.48 (2020), pp. 30063–30070.

[26] Mohsen Bayati and Andrea Montanari. "The dynamics of message passing on dense graphs, with applications to compressed sensing". In: *IEEE Transactions on Information Theory* 57.2 (2011), pp. 764–785.

[27] Mohsen Bayati and Andrea Montanari. "The LASSO risk for gaussian matrices". In: *IEEE Transactions on Information Theory* (2015).

[28] D. Bean et al. "Optimal M-estimation in high-dimensional regression". In: *Proc. Natl. Acad. Sci. U.S.A.* 110.36 (2013), pp. 14563–14568.

[29] Mikhail Belkin et al. "Reconciling modern machine-learning practice and the classical bias–variance trade-off". In: *Proceedings of the National Academy of Sciences* 116.32 (2019), pp. 15849–15854.

[30] P.C. Bellec. "Out-of-sample error estimate for robust M-estimators with convex penalty". In: *arXiv:2008.11840* (2023).

[31] P.C. Bellec, Y. Shen, and C.-H. Zhang. "Asymptotic normality of robust M-estimators with convex penalty". In: *Electron. J. Stat.* 16.2 (2022), pp. 5591–5622.

[32] P.C. Bellec and C.-H. Zhang. "Debiasing convex regularized estimators and interval estimation in linear models". In: *Ann. Stat.* 51.2 (2023), pp. 391–436.

[33] Pierre Bellec and Arun Kuchibhotla. "First order expansion of convex regularized estimators". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[34] Lucas Benigni and Sandrine Péché. "Eigenvalue distribution of some nonlinear models of random matrices". In: *Electronic Journal of Probability* 26.none (2021), pp. 1–37.

[35] Raphaël Berthier, Andrea Montanari, and Phan-Minh Nguyen. "State evolution for approximate message passing with non-separable functions". In: *Information and Inference: A Journal of the IMA* 9.1 (2019), pp. 33–79. ISSN: 2049-8772.

[36] Jeff Bezanson et al. "Julia: A Fresh Approach to Numerical Computing". In: *SIAM Review* 59.1 (2017), pp. 65–98. ISSN: 0036-1445, 1095-7200.

[37] Filippo Bigi et al. "A prediction rigidity formalism for low-cost uncertainties in trained neural networks". In: *Machine Learning: Science and Technology* 5.4 (2024), p. 045018.

[38] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN: 0387310738.

[39] Tom Breloff. "Plots.Jl". In: (2024). (Visited on 02/01/2024).

[40] Nicolas Brosse et al. "On Last-Layer Algorithms for Classification: Decoupling Representation from Uncertainty Estimation". In: (2020).

[41] A D Bruce and D Saad. "Statistical mechanics of hypothesis evaluation". In: *Journal of Physics A: Mathematical and General* 27.10 (1994), pp. 3355–3363.

[42] Peter Bühlmann, Markus Kalisch, and Lukas Meier. "High-Dimensional Statistics with a View Toward Applications in Biology". In: 2014.

[43] Philippe Carmona and Yueyun Hu. “Universality In Sherrington-Kirkpatrick’s Spin Glass Model”. In: (2004).

[44] Michael Celentano, Andrea Montanari, and Yuchen Wu. “The estimation error of general first order methods”. In: *Conference on Learning Theory*. PMLR. 2020, pp. 1078–1141.

[45] Sourav Chatterjee. “A simple invariance theorem”. In: (2005). eprint: math/0508213.

[46] Xin Chen et al. “Sketched Ridgeless Linear Regression: The Role of Downsampling”. In: *Proceedings of the 40th International Conference on Machine Learning*. Vol. 202. Proceedings of Machine Learning Research. PMLR, 2023, pp. 5296–5326.

[47] Giovanni Cherubin, Konstantinos Chatzikokolakis, and Martin Jaggi. “Exact Optimization of Conformal Predictors via Incremental and Decremental Learning”. In: *International Conference on Machine Learning*. 2021.

[48] Sanggyu Chong et al. “Prediction rigidities for data-driven chemistry”. In: *Faraday Discuss.* 256 (0 2025), pp. 322–344.

[49] Clément Chouard. “Quantitative deterministic equivalent of sample covariance matrices with a general dependence structure”. In: *arXiv:2211.13044* (2022).

[50] Lucas Clarté et al. “Analysis of Bootstrap and Subsampling in High-dimensional Regularized Regression”. In: *Uncertainty in Artificial Intelligence*. 2024.

[51] Lucas Clarté et al. “Expectation consistency for calibration of neural networks”. In: *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*. Vol. 216. Proceedings of Machine Learning Research. PMLR, 2023, pp. 443–453.

[52] Lucas Clarté et al. “On double-descent in uncertainty quantification in overparametrized models”. In: *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*. Vol. 206. Proceedings of Machine Learning Research. PMLR, 2023, pp. 7089–7125.

[53] Lucas Clarté et al. “Theoretical characterization of uncertainty in high-dimensional linear classification”. In: *Machine Learning: Science and Technology* 4.2 (2023), p. 025029.

[54] Elisabetta Cornacchia et al. “Learning curves for the multi-class teacher-student perceptron”. In: (2022).

[55] M Cover Thomas and A Thomas Joy. “Elements of information theory”. In: *New York: Wiley* 3 (1991).

[56] Hugo Cui et al. “A phase transition between positional and semantic learning in a solvable model of dot-product attention”. In: *NeurIPS* (2024).

[57] Hugo Cui et al. “Analysis of learning a flow-based generative model from limited sample complexity”. In: (2024).

[58] Stéphane D’Ascoli et al. “Double Trouble in Double Descent: Bias and Variance(s) in the Lazy Regime”. In: *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 2280–2290.

[59] Yatin Dandi et al. “Universality laws for Gaussian mixtures in generalized linear models”. In: *arXiv.org:2302.08933* (2023).

[60] A. C. Davison and D. V. Hinkley. *Bootstrap Methods and their Application*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1997.

[61] Erik Daxberger et al. “Laplace Redux - Effortless Bayesian Deep Learning”. In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 20089–20103.

[62] Aurelien Decelle et al. “Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications”. In: *Physical Review E* 84.6 (2011), p. 066106.

[63] Arthur P Dempster, Nan M Laird, and Donald B Rubin. “Maximum likelihood from incomplete data via the EM algorithm”. In: *Journal of the royal statistical society: series B (methodological)* 39.1 (1977), pp. 1–22.

[64] Zeyu Deng, Abla Kammoun, and Christos Thrampoulidis. “A model of double descent for high-dimensional binary linear classification”. In: *Information and Inference: A Journal of the IMA* 11.2 (2022), pp. 435–495.

[65] Al Depope, Marco Mondelli, and Matthew R. Robinson. “Inference of Genetic Effects via Approximate Message Passing”. In: *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2024, pp. 13151–13155.

[66] Oussama Dhifallah and Yue M Lu. “A precise performance analysis of learning with random features”. In: *arXiv:2008.11904 [cs.IT]* (2020).

[67] Xiaohan Ding et al. “Repvgg: Making vgg-style convnets great again”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 13733–13742.

[68] Edgar Dobriban and Stefan Wager. “High-dimensional asymptotics of prediction: Ridge regression and classification”. In: *The Annals of Statistics* 46.1 (2018), pp. 247–279.

[69] David Donoho and Andrea Montanari. “High dimensional robust m-estimation: Asymptotic variance via approximate message passing”. In: *Probability Theory and Related Fields* 166.3 (2016), pp. 935–969.

[70] David Donoho and Jared Tanner. “Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 367.1906 (2009), pp. 4273–4293.

[71] David L. Donoho, Adel Javanmard, and Andrea Montanari. “Information-Theoretically Optimal Compressed Sensing via Spatial Coupling and Approximate Message Passing”. In: *IEEE Transactions on Information Theory* 59.11 (2013), pp. 7434–7464.

[72] David L. Donoho, Arian Maleki, and Andrea Montanari. “Message-passing algorithms for compressed sensing”. In: *Proceedings of the National Academy of Sciences* 106.45 (2009), pp. 18914–18919. ISSN: 1091-6490.

[73] Jin-Hong Du, Pratik Patil, and Arun K. Kuchibhotla. "Subsample Ridge Ensembles: Equivalences and Generalized Cross-Validation". In: *Proceedings of the 40th International Conference on Machine Learning*. Vol. 202. Proceedings of Machine Learning Research. PMLR, 2023, pp. 8585–8631.

[74] B. Efron. "Bootstrap Methods: Another Look at the Jackknife". In: *The Annals of Statistics* 7.1 (1979), pp. 1–26.

[75] Bradley Efron and Charles Stein. "The jackknife estimate of variance". In: *The Annals of Statistics* (1981), pp. 586–596.

[76] Ahmed El Alaoui, Andrea Montanari, and Mark Sellke. "Sampling from the Sherrington-Kirkpatrick Gibbs measure via algorithmic stochastic localization". In: *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2022, pp. 323–334.

[77] N. El Karoui et al. "On robust regression with high-dimensional predictors". In: *Proc. Natl. Acad. Sci. U.S.A.* 110.36 (2013), pp. 14557–14562.

[78] Melikasadat Emami et al. "Generalization error of generalized linear models in high dimensions". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 2892–2901.

[79] Laszlo Erdos, Benjamin Schlein, and Horng-Tzer Yau. "Universality of Random Matrices and Local Relaxation Flow". In: (2009).

[80] Laszlo Erdos, Horng-Tzer Yau, and Jun Yin. "Bulk universality for generalized Wigner matrices". In: (2010).

[81] Yunzhen Feng et al. "Attacking Bayes: On the Adversarial Robustness of Bayesian Neural Networks". In: (2024).

[82] Edwin Fong and Chris C Holmes. "Conformal Bayesian Computation". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 18268–18279.

[83] D. A. Freedman. "Bootstrapping Regression Models". In: *The Annals of Statistics* 9.6 (1981), pp. 1218–1228.

[84] Lior Frenkel and Jacob Goldberger. "Network Calibration by Class-based Temperature Scaling". In: *2021 29th European Signal Processing Conference (EUSIPCO)*. 2021, pp. 1486–1490.

[85] Marylou Gabrié et al. "Entropy and mutual information in models of deep neural networks". In: *Advances in Neural Information Processing Systems* 31 (2018).

[86] Yarin Gal and Zoubin Ghahramani. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: *arXiv:1506.02142 [cs, stat]* (Oct. 4, 2016).

[87] E Gardner and B Derrida. "Three unfinished works on the optimal storage capacity of networks". In: *Journal of Physics A: Mathematical and General* 22.12 (1989), p. 1983.

[88] Jakob Gawlikowski et al. “A Survey of Uncertainty in Deep Neural Networks”. In: *arXiv:2107.03342 [cs, stat]* (Jan. 18, 2022).

[89] Mario Geiger et al. “Jamming transition as a paradigm to understand the loss landscape of deep neural networks”. In: *Phys. Rev. E* 100 (1 2019), p. 012115.

[90] Mario Geiger et al. “Scaling description of generalization with number of parameters in deep learning”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2020.2 (2020), p. 023401.

[91] Stuart Geman, Elie Bienenstock, and René Doursat. “Neural networks and the bias/variance dilemma”. In: *Neural computation* 4.1 (1992), pp. 1–58.

[92] A. C. Genz and A. A. Malik. “Remarks on Algorithm 006: An Adaptive Algorithm for Numerical Integration over an N-dimensional Rectangular Region”. In: *Journal of Computational and Applied Mathematics* 6.4 (1980), pp. 295–302. ISSN: 0377-0427. (Visited on 02/01/2024).

[93] Federica Gerace et al. “Gaussian Universality of Linear Classifiers with Random Labels in High-Dimension”. In: (2022).

[94] Federica Gerace et al. “Generalisation error in learning with random features and the hidden manifold model”. In: *International Conference on Machine Learning*. PMLR. 2020.

[95] Cedric Gerbelot, Alia Abbara, and Florent Krzakala. “Asymptotic errors for teacher-student convex generalized linear models (or: How to prove Kabashima’s replica formula)”. In: *IEEE Transactions on Information Theory* 69.3 (2022), pp. 1824–1852.

[96] Cédric Gerbelot, Alia Abbara, and Florent Krzakala. “Asymptotic Errors for High-Dimensional Convex Penalized Linear Regression beyond Gaussian Matrices”. In: *Proceedings of Thirty Third Conference on Learning Theory*. Vol. 125. Proceedings of Machine Learning Research. PMLR, 2020, pp. 1682–1713.

[97] Cédric Gerbelot and Raphaël Berthier. “Graph-based approximate message passing iterations”. In: *Information and Inference: A Journal of the IMA* 12.4 (2023), pp. 2562–2628.

[98] Sebastian Goldt et al. “Modeling the Influence of Data Structure on Learning in Neural Networks: The Hidden Manifold Model”. In: *Phys. Rev. X* 10 (4 2020), p. 041044.

[99] Sebastian Goldt et al. “The Gaussian equivalence of generative models for learning with shallow neural networks”. In: *arXiv: 2006.14709* (2021).

[100] Alex Graves. “Practical Variational Inference for Neural Networks”. In: *Advances in Neural Information Processing Systems*. Vol. 24. Curran Associates, Inc., 2011.

[101] Chuan Guo et al. “On Calibration of Modern Neural Networks”. In: Proceedings of Machine Learning Research 70 (2017), pp. 1321–1330.

[102] Chirag Gupta, Aleksandr Podkopaev, and Aaditya Ramdas. "Distribution-Free Binary Classification: Prediction Sets, Confidence Intervals and Calibration". In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS'20. Vancouver, BC, Canada: Curran Associates Inc., 2020.

[103] Walid Hachem, Philippe Loubaton, and Jamal Najim. "Deterministic equivalents for certain functionals of large random matrices". In: (2007).

[104] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. "The Elements of Statistical Learning". In: Springer Series in Statistics (2001).

[105] Trevor Hastie et al. "Surprises in high-dimensional ridgeless least squares interpolation". In: *The Annals of Statistics* 50.2 (2022), pp. 949–986.

[106] K. He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, 2016, pp. 770–778.

[107] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. "Why ReLU Networks Yield High-Confidence Predictions Far Away From the Training Data and How to Mitigate the Problem". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[108] Dan Hendrycks and Thomas G. Dietterich. "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations". In: (2019).

[109] Hong Hu and Yue M. Lu. "Universality Laws for High-Dimensional Learning with Random Features". In: (2020).

[110] G. Huang et al. "Densely Connected Convolutional Networks". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, 2017, pp. 2261–2269.

[111] Xinmeng Huang et al. "Uncertainty in Language Models: Assessment through Rank-Calibration". In: (2024).

[112] Yukito Iba. "The Nishimori line and Bayesian statistics". In: *Journal of Physics A: Mathematical and General* 32.21 (1999), p. 3875.

[113] Gareth James et al. "Resampling methods". In: *An Introduction to Statistical Learning: with Applications in Python*. Springer, 2023, pp. 201–228.

[114] Adel Javanmard and Andrea Montanari. "Confidence Intervals and Hypothesis Testing for High-Dimensional Regression". In: *Journal of Machine Learning Research* 15.82 (2014), pp. 2869–2909.

[115] Adel Javanmard and Andrea Montanari. "State evolution for general approximate message passing algorithms, with applications to spatial coupling". In: *Information and Inference: A Journal of the IMA* 2.2 (2013), pp. 115–144.

[116] Lei Jing et al. "Distribution-Free Predictive Inference for Regression". In: *Journal of the American Statistical Association* 113.523 (2018), pp. 1094–1111.

[117] Steven G. Johnson. "QuadGK.Jl: Gauss–Kronrod Integration in Julia". In: (2013).

[118] Steven G. Johnson. "The HCubature.Jl Package for Multi-Dimensional Adaptive Integration in Julia". In: (2017).

[119] Laurent Valentin Jospin et al. "Hands-On Bayesian Neural Networks—A Tutorial for Deep Learning Users". In: *IEEE Computational Intelligence Magazine* 17.2 (2022), pp. 29–48.

[120] "JuliaAI/MLJLinearModels.Jl". In: (2023). (Visited on 02/01/2024).

[121] "JuliaArrays/StaticArrays.Jl". In: (2024). (Visited on 02/01/2024).

[122] "JuliaStats/LogExpFunctions.Jl". In: (2023). (Visited on 02/01/2024).

[123] "JuliaStats/StatsFuns.Jl". In: (2024). (Visited on 02/01/2024).

[124] Y. Kabashima and D. Saad. "Statistical mechanics of error-correcting codes". In: *Europhysics Letters* 45.1 (1999), p. 97.

[125] Y. Kabashima and S. Shinomoto. "Learning Curves for Error Minimum and Maximum Likelihood Algorithms". In: *Neural Computation* 4.5 (1992), pp. 712–719.

[126] Yoshiyuki Kabashima et al. "Phase transitions and sample complexity in Bayes-optimal matrix factorization". In: *IEEE Transactions on Information Theory* 62.7 (July 2016), pp. 4228–4265. ISSN: 0018-9448, 1557-9654.

[127] Sanyam Kapoor et al. "On Uncertainty, Tempering, and Data Augmentation in Bayesian Classification". In: (2022).

[128] Noureddine El Karoui. "Asymptotic behavior of unregularized and ridge-regularized high-dimensional robust regression estimators : rigorous results". In: (2013).

[129] Noureddine El Karoui. "Concentration of measure and spectra of random matrices: Applications to correlation matrices, elliptical distributions and beyond". In: *The Annals of Applied Probability* 19.6 (2009), pp. 2362–2405.

[130] Noureddine El Karoui. "The spectrum of kernel random matrices". In: *The Annals of Statistics* 38.1 (2010), pp. 1–50.

[131] Noureddine El Karoui and Elizabeth Purdom. "Can We Trust the Bootstrap in High-dimensions? The Case of Linear Models". In: *J. Mach. Learn. Res.* 19 (2018), 5:1–5:66.

[132] Noureddine El Karoui et al. "On robust regression with high-dimensional predictors". In: *Proceedings of the National Academy of Sciences* 110.36 (2013), pp. 14557–14562.

[133] Alex Kendall and Yarin Gal. "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Curran Associates Inc., 2017.

[134] Antti Knowles and Jun Yin. "Anisotropic local laws for random matrices". In: *Probability Theory and Related Fields* 169.1 (2017), pp. 257–352.

[135] Satish Babu Korada and Andrea Montanari. "Applications of Lindeberg Principle in Communications and Statistical Learning". In: (2010).

[136] Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. "Being Bayesian, Even Just a Bit, Fixes Overconfidence in ReLU Networks". In: Proceedings of Machine Learning Research 119 (2020), pp. 5436–5446.

[137] Alex Krizhevsky. "Learning Multiple Layers of Features from Tiny Images". In: *Technical report, University of Toronto*. 2009.

[138] Anders Krogh and John Hertz. "A Simple Weight Decay Can Improve Generalization". In: *Advances in Neural Information Processing Systems*. Vol. 4. Morgan-Kaufmann, 1991.

[139] Anders Krogh and Peter Sollich. "Statistical mechanics of ensemble learning". In: *Phys. Rev. E* 55 (1 1997), pp. 811–825.

[140] F. Krzakala et al. "Statistical-Physics-Based Reconstruction in Compressed Sensing". In: *Phys. Rev. X* 2 (2 2012), p. 021005.

[141] Florent Krzakala et al. "Probabilistic reconstruction in compressed sensing: algorithms, phase diagrams, and threshold achieving matrices". In: *Journal of Statistical Mechanics: Theory and Experiment* 2012.08 (2012), P08009.

[142] Meelis Kull et al. "Beyond Temperature Scaling: Obtaining Well-Calibrated Multiclass Probabilities with Dirichlet Calibration". In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[143] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles". In: *Advances in Neural Information Processing Systems*. Vol. 30. 2017.

[144] Jing Lei. "Fast Exact Conformalization of Lasso using Piecewise Linear Homotopy". In: *Biometrika*. 2017.

[145] Daniel LeJeune, Hamid Javadi, and Richard Baraniuk. "The Implicit Regularization of Ordinary Least Squares Ensembles". In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Vol. 108. Proceedings of Machine Learning Research. PMLR, 2020, pp. 3525–3535.

[146] Tengyuan Liang and Pragya Sur. "A precise high-dimensional asymptotic theory for boosting and minimum-L1-norm interpolated classifiers". In: *The Annals of Statistics* 50.3 (2022).

[147] Zhenyu Liao and Romain Couillet. "On the Spectrum of Random Features Maps of High Dimensional Data". In: *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 3063–3071.

[148] Zhenyu Liao and Michael W Mahoney. "Hessian Eigenspectra of More Realistic Nonlinear Models". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 20104–20117.

[149] Licong Lin and Edgar Dobriban. "What Causes the Test Error? Going Beyond Bias-Variance via ANOVA". In: *Journal of Machine Learning Research* 22.155 (2021), pp. 1–82.

[150] Jeremiah Zhe Liu et al. "Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness". In: *Advances in Neural Information Processing Systems* (2020).

[151] Cosme Louart, Zhenyu Liao, and Romain Couillet. "A random matrix approach to neural networks". In: *Ann. Appl. Probab.* 28.2 (2018), pp. 1190–1248.

[152] Bruno Loureiro et al. "Fluctuations, Bias, Variance & Ensemble of Learners: Exact Asymptotics for Convex Losses in High-Dimension". In: *Proceedings of the 39th International Conference on Machine Learning*. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 14283–14314.

[153] Bruno Loureiro et al. "Learning curves of generic features maps for realistic datasets with a teacher-student model". In: 34 (2021), pp. 18137–18151.

[154] Bruno Loureiro et al. "Learning Gaussian Mixtures with Generalized Linear Models: Precise Asymptotics in High-dimensions". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 10144–10157.

[155] David J. C. MacKay. "Bayesian Interpolation". In: *Neural Computation* 4.3 (1992), pp. 415–447. ISSN: 0899-7667.

[156] David J. C. MacKay. "Hyperparameters: Optimize, or Integrate Out?" In: *Maximum Entropy and Bayesian Methods: Santa Barbara, California, U.S.A., 1993*. Dordrecht: Springer Netherlands, 1996, pp. 43–59.

[157] Wesley J. Maddox et al. "A Simple Baseline for Bayesian Uncertainty in Deep Learning". In: *NeurIPS*. 2019.

[158] Xiaoyi Mai, Zhenyu Liao, and Romain Couillet. "A Large Scale Analysis of Logistic Regression: Asymptotic Performance and New Insights". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019, pp. 3357–3361.

[159] Antoine Maillard et al. "Phase retrieval in high dimensions: Statistical and computational phase transitions". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 11071–11082.

[160] Andrey Malinin, Bruno Mlodozeniec, and Mark Gales. "Ensemble Distribution Distillation". In: *International Conference on Learning Representations* (2020).

[161] Dörthe Malzahn and Manfred Opper. "A Statistical Mechanics Approach to Approximate Analytical Bootstrap Averages". In: *Advances in Neural Information Processing Systems*. Vol. 15. MIT Press, 2002.

[162] Dörthe Malzahn and Manfred Opper. "Approximate Analytical Bootstrap Averages for Support Vector Classifiers". In: *Advances in Neural Information Processing Systems*. Vol. 16. MIT Press, 2003.

[163] G Marion and D Saad. "A statistical mechanical analysis of a Bayesian inference scheme for an unrealizable rule". In: *Journal of Physics A: Mathematical and General* 28.8 (1995), pp. 2159–2171.

[164] Glenn Marion and David Saad. "Hyperparameters Evidence and Generalisation for an Unrealisable Rule". In: *Advances in Neural Information Processing Systems*. Vol. 7. MIT Press, 1994.

[165] Glenn Marion and David Saad. "Hyperparameters Evidence and Generalisation for an Unrealisable Rule". In: *Advances in Neural Information Processing Systems*. Vol. 7. MIT Press, 1994.

[166] Pierre-Alexandre Mattei. "A Parsimonious Tour of Bayesian Model Uncertainty". In: (2019).

[167] Cyril Méasson et al. "The generalized area theorem and some of its consequences". In: *IEEE Transactions on Information Theory* 55.11 (2009), pp. 4793–4821.

[168] Song Mei and Andrea Montanari. "The Generalization Error of Random Features Regression: Precise Asymptotics and the Double Descent Curve". In: *Communications on Pure and Applied Mathematics* 75.4 (2022), pp. 667–766.

[169] Nicolai Meinshausen and Peter Bühlmann. "Stability selection". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 72.4 (2010), pp. 417–473. DOI: https://doi.org/10.1111/j.1467-9868.2010.00740.x.

[170] Marc Mezard and Andrea Montanari. *Information, physics, and computation*. Oxford University Press, 2009. ISBN: 9780198570837.

[171] Marc Mézard, Giorgio Parisi, and Miguel Angel Virasoro. *Spin glass theory and beyond: An Introduction to the Replica Method and Its Applications*. Vol. 9. World Scientific Publishing Company, 1987.

[172] Francesca Mignacco et al. "The Role of Regularization in Classification of High-dimensional Noisy Gaussian Mixture". In: *Proceedings of the 37th International Conference on Machine Learning*. Proceedings of Machine Learning Research. PMLR, 2020.

[173] Charles Millard et al. "An Approximate Message Passing Algorithm For Rapid Parameter-Free Compressed Sensing MRI". In: *IEEE International Conference on Image Processing, ICIP 2020, Abu Dhabi, United Arab Emirates, October 25-28, 2020*. IEEE, 2020, pp. 91–95.

[174] Matthias Minderer et al. "Revisiting the Calibration of Modern Neural Networks". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 15682–15694.

[175] Patrick K. Mogensen and Asbjørn N. Riseth. "Optim: A Mathematical Optimization Package for Julia". en. In: *Journal of Open Source Software* 3.24 (2018), p. 615. ISSN: 2475-9066. (Visited on 02/01/2024).

[176] Christopher Mohri and Tatsunori Hashimoto. "Language models with conformal factuality guarantees". In: ICML'24 (2024).

[177] Andrea Montanari and Basil N. Saeed. "Universality of empirical risk minimization". In: *Proceedings of Thirty Fifth Conference on Learning Theory*. Vol. 178. Proceedings of Machine Learning Research. PMLR, 2022, pp. 4310–4312.

[178] Andrea Montanari et al. "The generalization error of max-margin linear classifiers: High-dimensional asymptotics in the overparametrized regime". In: *arXiv:1911.01544 [math.ST]* (2020).

[179] Jishnu Mukhoti et al. "Calibrating Deep Neural Networks using Focal Loss". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 15288–15299.

[180] Félix Musil et al. "Fast and Accurate Uncertainty Estimation in Chemical Machine Learning". In: *Journal of Chemical Theory and Computation* 15.2 (2019), pp. 906–915.

[181] Preetum Nakkiran et al. "Deep Double Descent: Where Bigger Models and More Data Hurt". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[182] Preetum Nakkiran et al. "Optimal Regularization can Mitigate Double Descent". In: *International Conference on Learning Representations*. 2021.

[183] Eugene Ndiaye and Ichiro Takeuchi. "Computing Full Conformal Prediction Set with Approximate Homotopy". In: vol. 32. Curran Associates, Inc., 2019.

[184] Yuval Netzer et al. "Reading Digits in Natural Images with Unsupervised Feature Learning". In: *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*. 2011.

[185] Anh Nguyen, Jason Yosinski, and Jeff Clune. "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 427–436.

[186] Alexandru Niculescu-Mizil and Rich Caruana. "Predicting Good Probabilities with Supervised Learning". In: ICML '05. Bonn, Germany: Association for Computing Machinery, 2005, pp. 625–632.

[187] Jeremy Nixon and Dustin Tran. "Why Aren't Bootstrapped Neural Networks Better?" In: Neurips 2020 ICBINB Workshop, 2020.

[188] Tomoyuki Obuchi and Yoshiyuki Kabashima. "Cross validation in LASSO and its acceleration". In: *Journal of Statistical Mechanics: Theory and Experiment* 2016.5 (2016), p. 053304.

[189] Tomoyuki Obuchi and Yoshiyuki Kabashima. "Semi-Analytic Resampling in Lasso". In: *Journal of Machine Learning Research* 20.70 (2019), pp. 1–33.

[190] M Opper et al. “On the ability of the optimal perceptron to generalise”. In: *Journal of Physics A: Mathematical and General* 23.11 (1990), p. L581.

[191] Ashkan Panahi and Babak Hassibi. “A Universal Analysis of Large-Scale Regularized Least Squares Solutions”. In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017.

[192] Harris Papadopoulos et al. “Inductive confidence machines for regression”. In: *Machine Learning: ECML 2002: 13th European Conference on Machine Learning Helsinki, Finland, August 19–23, 2002 Proceedings 13*. Springer. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 345–356. ISBN: 978-3-540-36755-0.

[193] Pratik Patil, Jin-Hong Du, and Arun Kumar Kuchibhotla. “Bagging in overparameterized learning: Risk characterization and risk monotonization”. In: *Journal of Machine Learning Research* 24.319 (2023), pp. 1–113.

[194] Pratik Patil and Daniel LeJeune. “Asymptotically free sketched ridge ensembles: Risks, cross-validation, and tuning”. In: (2023).

[195] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1988.

[196] Jeffrey Pennington and Pratik Worah. “Nonlinear random matrix theory for deep learning”. In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017.

[197] Giovanni Piccioli, Emanuele Troiani, and Lenka Zdeborová. “Gibbs sampling the posterior of neural networks”. In: *Journal of Physics A: Mathematical and Theoretical* 57.12 (2024), p. 125002.

[198] John Platt. “Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods”. In: *Adv. Large Margin Classif.* 10 (2000).

[199] Konstantin Posch, Jan Steinbrener, and Jürgen Pilz. “Variational Inference to Measure Model Uncertainty in Deep Neural Networks”. In: *arXiv:1902.10189 [stat.ML]* (2019).

[200] M. H. Quenouille. “Notes on Bias in Estimation”. In: *Biometrika* 43.3/4 (1956), pp. 353–360. (Visited on 01/23/2024).

[201] Ali Rahimi and Benjamin Recht. “Random Features for Large-Scale Kernel Machines”. In: *Advances in Neural Information Processing Systems*. Vol. 20. Curran Associates, Inc., 2007.

[202] Sundeep Rangan. “Generalized approximate message passing for estimation with random linear mixing”. In: *2011 IEEE International Symposium on Information Theory Proceedings*. IEEE. 2011.

[203] Sundeep Rangan, Philip Schniter, and Alyson K. Fletcher. “Vector Approximate Message Passing”. In: *IEEE Transactions on Information Theory* 65.10 (2019), pp. 6664–6684.

[204] Maria Refinetti, Alessandro Ingrosso, and Sebastian Goldt. “Neural networks trained with SGD learn distributions of increasing complexity*”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2025.2 (2025), p. 024001.

[205] Hippolyt Ritter, Aleksandar Botev, and David Barber. "A Scalable Laplace Approximation for Neural Networks". In: *International Conference on Learning Representations*. 2018.

[206] Yaniv Romano, Evan Patterson, and Emmanuel Candes. "Conformalized Quantile Regression". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[207] Saharon Rosset, Ji Zhu, and Trevor Hastie. "Margin Maximizing Loss Functions". In: *Advances in Neural Information Processing Systems*. Vol. 16. MIT Press, 2003.

[208] Robert E. Schapire. "A brief introduction to boosting". In: IJCAI'99. Stockholm, Sweden: Morgan Kaufmann Publishers Inc., 1999, pp. 1401–1406.

[209] Dominik Schröder et al. "Deterministic equivalent and error universality of deep random features learning*". In: *Journal of Statistical Mechanics: Theory and Experiment* 2024.10 (2024).

[210] Mohamed El Amine Seddik et al. "Random matrix theory proves that deep learning representations of gan-data behave as gaussian mixtures". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 8573–8582.

[211] H. S. Seung, H. Sompolinsky, and N. Tishby. "Statistical mechanics of learning from examples". In: *Phys. Rev. A* 45 (8 1992), pp. 6056–6091.

[212] Glenn Shafer and Vladimir Vovk. "A Tutorial on Conformal Prediction". In: *J. Mach. Learn. Res.* 9 (2008), pp. 371–421. ISSN: 1532-4435.

[213] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *CoRR* abs/1409.1556 (2014).

[214] Peter Sollich. "Gaussian Process Regression with Mismatched Models". In: *Advances in Neural Information Processing Systems*. Vol. 14. MIT Press, 2001.

[215] Peter Sollich. "Learning Curves for Gaussian Processes". In: *Advances in Neural Information Processing Systems*. Vol. 11. MIT Press, 1998.

[216] Peter Sollich and Anders Krogh. "Learning with ensembles: How overfitting can be useful". In: *Advances in Neural Information Processing Systems*. Vol. 8. MIT Press, 1995.

[217] S Spigler et al. "A jamming transition from under- to over-parametrization affects generalization in deep learning". In: *Journal of Physics A: Mathematical and Theoretical* 52.47 (2019), p. 474001.

[218] Mihailo Stojnic. "A framework to characterize performance of LASSO algorithms". In: (2013). arXiv: 1303.7291.

[219] Pragya Sur and Emmanuel J. Candès. "A modern maximum-likelihood theory for high-dimensional logistic regression". In: *Proceedings of the National Academy of Sciences* 116.29 (2019), pp. 14516–14525.

[220] Pragya Sur and Emmanuel J. Candès. “The phase transition for the existence of the maximum likelihood estimate in high-dimensional logistic regression”. In: *The Annals of Statistics* 48.1 (2020), pp. 27–42.

[221] Pragya Sur, Yuxin Chen, and Emmanuel J Candès. “The likelihood ratio test in high-dimensional logistic regression is asymptotically a rescaled chi-square”. In: *Probability theory and related fields* 175 (2019), pp. 487–558.

[222] Hossein Taheri, Ramtin Pedarsani, and Christos Thrampoulidis. “Sharp Asymptotics and Optimal Performance for Inference in Binary Models”. In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Proceedings of Machine Learning Research. PMLR, 2020.

[223] Takashi Takahashi. “A replica analysis of under-bagging”. In: (2024). arXiv: 2404.09779.

[224] Takashi Takahashi. “Role of Bootstrap Averaging in Generalized Approximate Message Passing”. In: *2023 IEEE International Symposium on Information Theory (ISIT)* (2023), pp. 767–772.

[225] Takashi Takahashi and Yoshiyuki Kabashima. “Macroscopic Analysis of Vector Approximate Message Passing in a Model-Mismatched Setting”. In: *IEEE Transactions on Information Theory* 68.8 (2022), pp. 5579–5600.

[226] Takashi Takahashi and Yoshiyuki Kabashima. “Replicated Vector Approximate Message Passing For Resampling Problem”. In: (2019).

[227] Terence Tao and Van Vu. “Random Matrices: universal properties of eigenvectors”. In: *Random Matrices: Theory and Applications* 01.01 (2012), p. 1150001.

[228] Christos Thrampoulidis, Ehsan Abbasi, and Babak Hassibi. “Precise error analysis of regularized M-estimators in high dimensions”. In: *IEEE Transactions on Information Theory* 8 (2018).

[229] Christos Thrampoulidis, Samet Oymak, and Babak Hassibi. “Regularized Linear Regression: A Precise Analysis of the Estimation Error”. In: *Proceedings of The 28th Conference on Learning Theory*. Proceedings of Machine Learning Research. PMLR, 2015.

[230] Christos Thrampoulidis, Samet Oymak, and Babak Hassibi. “Regularized Linear Regression: A Precise Analysis of the Estimation Error”. In: *Proceedings of The 28th Conference on Learning Theory*. Vol. 40. Proceedings of Machine Learning Research. Paris, France: PMLR, 2015, pp. 1683–1709.

[231] Sunil Thulasidasan et al. “On Mixup Training: Improved Calibration and Predictive Uncertainty for Deep Neural Networks”. In: *Advances in Neural Information Processing Systems* (2019).

[232] Robert J Tibshirani and Bradley Efron. “An introduction to the bootstrap”. In: *Monographs on statistics and applied probability* 57.1 (1993).

[233] Emanuele Troiani et al. “Fundamental limits of learning in sequence multi-index models and deep attention networks: High-dimensional asymptotics and sharp thresholds”. In: (2025).

[234] John W. Tukey. "Bias and confidence in not quite large samples (abstract)". In: *The Annals of Mathematical Statistics* 29.2 (1958), pp. 614–623.

[235] Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Berlin, Heidelberg: Springer-Verlag, 2005.

[236] Deng-Bao Wang, Lei Feng, and Min-Ling Zhang. "Rethinking Calibration of Deep Neural Networks: Do Not Be Afraid of Overconfidence". In: *Neural Information Processing Systems*. 2021.

[237] L. Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. Springer Texts in Statistics. Springer New York, 2013.

[238] Max Welling and Yee Whye Teh. "Bayesian Learning via Stochastic Gradient Langevin Dynamics". In: *Proceedings of the 28th International Conference on International Conference on Machine Learning*. ICML'11. Bellevue, Washington, USA: Omnipress, 2011, pp. 681–688.

[239] Florian Wenzel et al. "How Good is the Bayes Posterior in Deep Neural Networks Really?" In: *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 10248–10259.

[240] Andrew Gordon Wilson. "The Case for Bayesian Deep Learning". In: (2020).

[241] Wolfram Research, Inc. "Mathematica, Version 13.3". In: (). Champaign, IL, 2023.

[242] C. F. J. Wu. "Jackknife, Bootstrap and Other Resampling Methods in Regression Analysis". In: *The Annals of Statistics* 14.4 (1986), pp. 1261–1295.

[243] Junwen Yao and Miles Lopes. "Rates of Bootstrap Approximation for Eigenvalues in High-Dimensional PCA". In: *Statistica Sinica* (Jan. 2024). DOI: 10.5705/ss.202021.0158.

[244] Bianca Zadrozny and Charles Elkan. "Obtaining Calibrated Probability Estimates from Decision Trees and Naive Bayesian Classifiers". In: *Proceedings of the Eighteenth International Conference on Machine Learning*. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 609–616.

[245] Bianca Zadrozny and Charles Elkan. "Transforming Classifier Scores into Accurate Multiclass Probability Estimates". In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2002).

[246] Margaux Zaffran et al. "Adaptive Conformal Predictions for Time Series". In: *Proceedings of the 39th International Conference on Machine Learning*. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 25834–25866.

[247] Lenka Zdeborová and Florent Krzakala. "Statistical physics of inference: Thresholds and algorithms". In: *Advances in Physics* 65.5 (2016), pp. 453–552. ISSN: 1460-6976.

[248] Ji Zhu et al. "Multi-class AdaBoost". In: *Statistics and its interface* 2 (2006).

Lucas Clarté

Last year PhD Student in machine learning at EPFL (Lausanne), working on the theory of machine learning and uncertainty quantification. Expected graduation date : 09/2025

clarte.lucas@gmail.com | lclarte | linkedin.com/in/lucas-clarté | Nationality : French

Education

EPFL *Lausanne, Switzerland*

PhD in Computer Science *2021 - 2025 (Expected)*

- Advised by Prof. Lenka Zdeborová in the Statistical Physics of Computation laboratory and partially funded by a fellowship of the computer science doctoral school.
- Research on high-dimensional statistics and uncertainty quantification in neural networks.

Master 2 ICFP – ENS Paris *Paris, France*

Master's Degree in Theoretical Physics *2020 - 2021*

- Final grade : 14.2 / 20 (Magna cum Laude)

Master 2 MVA – ENS Paris-Saclay *Cachan, France*

Master's Degree in Mathematics, Computer Vision and Machine Learning *2019 - 2020*

- Final grade : 17.7 / 20 (Summa cum Laude)

École Polytechnique *Palaiseau, France*

Master of engineering, Majored in applied mathematics *2016 - 2020*

- Final GPA : 3.95 / 4

Publications

Clarté, Zdeborova, *Building Conformal Prediction Intervals with Approximate Message Passing*, 2024, in preparation

Clarté, Vandenbroucque, Dalle, Loureiro, Krzakala, Zdeborova, *Analysis of Bootstrap and Subsampling in High-dimensional Regularized Regression*, UAI 2024 (**Spotlight**)

Clarté, Loureiro, Krzakala, Zdeborova, *Expectation consistency for calibration of neural networks*, presented at UAI 2023

Clarté, Loureiro, Krzakala, Zdeborova, *On double-descent in uncertainty quantification in overparametrized models*, presented at AISTATS 2023

Clarté, Loureiro, Krzakala, Zdeborova, *Theoretical characterization of uncertainty in high-dimensional linear classification*, published in *Machine Learning, Science and Technology*, 2023

Experience

ENSAE *Palaiseau, France*

Research Intern *April - August 2020*

- Research on Bayesian inverse reinforcement learning with Prof. Vianney Perchet

Shadow *Mountain View, USA*

Research Intern *April- August 2019*

- Prototyped in Python an image compression algorithm for a video streaming software.
- Patent filing : WO2021123617A1

Proxem *Paris, France*

Software engineering Intern *June - August 2018*

- Implemented in C# a transfer learning algorithm for a NLP library.

Skills

Programming

- Python : experience with machine learning libraries (scipy, scikit-learn, pytorch), C++

Languages

- French (Mother tongue), English (Fluent), German (Intermediate), Malagasy (Beginner)