

UNIVERSITÀ COMMERCIALE “LUIGI BOCCONI”
PHD SCHOOL

PhD program in: Statistics and Computer Science

Cycle: 37

Disciplinary Field (code): FIS/02

**Navigating Rough Landscapes:
Statistical Physics Approaches to
Machine Learning and Inference**

Advisor: Carlo Lucibello

Co-Advisor: Enrico Maria Malatesta

PhD Thesis by
Davide Straziota
ID number: 3177033

Year: 2026

Contents

1	Preliminaries	7
1.1	General Introduction to Machine Learning	7
1.1.1	Datasets	8
1.1.2	Models	11
1.1.3	Loss Functions	14
1.1.4	Optimization Algorithms	15
1.2	Tasks and Challenges in Machine Learning	16
1.2.1	Concluding Remarks	19
1.3	Statistical Physics of Machine Learning	19
1.3.1	A crash course on Statistical Physics	20
1.3.2	Components in Statistical Physics of Machine Learning	22
1.3.3	Notable Results and Open Problems	28
1.4	Statistical Physics of Machine Learning tools	29
1.4.1	Storage problem replica computation	29
1.4.2	Message Passing Algorithms	36
2	Generative diffusion for perceptron problems: statistical physics analysis and efficient algorithms	45
2.1	Related works	46
2.1.1	Diffusion models	46
2.1.2	Sampling from intractable distributions: the Stochastic Localization framework	46
2.2	Stochastic Localization Sampling Scheme	48
2.2.1	The ASL algorithm	49
2.2.2	ASL and Generative Diffusion	49
2.3	Asymptotic Analysis of ASL	49
2.4	Sampling Perceptron solutions	51
2.4.1	Implementation of ASL	52
2.4.2	General Replica computation for Perceptron models	54
2.4.3	Spherical Non-Convex Perceptron	56
2.4.4	Binary Perceptron	58
2.5	Recap of the results	62
3	Isolating the hard core of phaseless inference: the Phase Selection formulation	63
3.1	A Statistical Physics perspective	64
3.2	Introduction and Problem Setting	64

3.2.1	Phase retrieval	64
3.2.2	Phase selection problem	65
3.3	Analytical investigations	67
3.3.1	Large deviation analysis	67
3.3.2	Replica Symmetric formula for the large deviations	68
3.3.3	Results	71
3.4	Numerical Results	77
3.4.1	Problem generator	78
3.4.2	Simulated Annealing	78
3.4.3	Approximate Message Passing based solver	80
3.4.4	Langevin dynamics based solver	82
3.5	Recap of the results	84
4	On the Spectral Analysis of Neural Network Losses	85
4.1	Hessian investigation in Neural Networks	87
4.2	Analytical investigation	88
4.2.1	Theoretical background	89
4.2.2	Tree committee machine	90
4.2.3	Replica Method	91
4.2.4	Finite temperature analysis for the large width limit	94
4.2.5	Zero Temperature analysis for the large width limit	96
4.3	Preliminary analytical results	98
4.3.1	Evolution of Perceptron Hessian spectrum	98
4.3.2	Evolution of Tree Committee Machine Hessian spectrum	99
4.3.3	The role of the regularization	100
4.4	Recap of the results	101
5	Learning RBM via interaction screening optimization	103
5.1	Restricted Boltzmann Machines	105
5.1.1	Learning in Restricted Boltzmann Machines	106
5.2	Learning graphical models via interaction screening and pseudo likelihood optimization	109
5.2.1	Learning Ising Models	109
5.2.2	Learning Restricted Boltzmann Machines	111
5.3	Numerical Results	112
5.3.1	Total variation distance evolution	113
5.3.2	Dependence on Dataset Size	113
5.3.3	Dependence on the Number of GD Steps	114
5.3.4	Dependence on Regularization strength	115
5.4	Recap of the results	116
6	Conclusions and Future Directions	117
A	Stochastic Localization Appendix	119
A.1	Replica Computation for ASL on the Perceptron	119
A.1.1	General framework	119
A.1.2	Replica Symmetric Ansatz	121
A.1.3	Nishimori conditions	122

A.1.4	Replica Symmetric Free Entropy	123
A.2	Computation of the stability distribution	126
A.3	Spherical Perceptron	127
A.3.1	Free Entropy	127
A.3.2	Reference system	128
A.4	Binary Perceptron	128
A.4.1	Free Entropy	128
A.4.2	Reference system	129
A.5	Limiting behavior of the free entropy derivative	129
A.5.1	Interaction term	129
A.5.2	Energetic term	131
B	Phase Selection Appendix	135
B.1	Replica computation	135
B.1.1	Replica Symmetric Ansatz	137
B.2	Stabilities analysis	142
B.3	Replica Symmetry breaking	144
C	Hessian appendix	147
C.1	General Replica computation	147
C.1.1	Saddle point equations conjugated parameters	153
C.1.2	Large width limit	153
C.1.3	Effective order parameters	159

Abstract

Artificial Intelligence (AI) has rapidly become a central field in contemporary research, driving innovation in several disciplines, including physics, engineering, economics, and medicine. At the heart of AI lies the development of algorithms capable of autonomously learning from data and adapting to new challenges, moving beyond traditional rule-based programming. Despite remarkable progress, the underlying reason for the success of many AI models remains largely unexplained, underscoring the need for a deeper theoretical understanding of their inner mechanisms.

This thesis addresses this challenge by employing tools from statistical physics to analyze and interpret machine learning algorithms, with a particular focus on the geometry of their loss landscapes. By leveraging methods such as the replica method, random matrix theory, and message passing algorithms, the thesis investigates how the structure of the loss landscape influences inference, sampling, and optimization in high-dimensional settings.

The thesis investigates four canonical machine-learning tasks, each characterized by a non-convex loss landscape: for the negative and binary perceptron we introduce and analyse a stochastic-localization algorithm that fairly samples their solution space; for phase retrieval we recast the task within a phase-selection framework, dividing its combinatorial and continuous components; we develop a theoretical framework for the spectral analysis of neural-network loss Hessians and specialise it to the Tree Committee Machine; and for Restricted Boltzmann Machines we propose pseudo-likelihood maximisation (which can be reframed in term of loss minimization) as a stable and efficient alternative to contrastive divergence based learning.

Building on these concepts, the thesis offers new theoretical insights into the interplay between optimization, inference, and the underlying geometry of complex machine learning models.

Introduction

In recent years, *Artificial Intelligence (AI)* has rapidly emerged as one of the most influential fields of research, with applications spanning disciplines as diverse as physics, engineering, economics, and medicine. AI has become a universal tool for addressing complex challenges, thanks to its ability to develop algorithms capable of autonomously performing intricate tasks.

AI is inherently multidisciplinary, encompassing several subfields that share the common goal of enabling machines to learn from data and adapt to new challenges, moving beyond traditional rule-based programming. Among these, Machine Learning (ML) focuses on designing algorithms that extract patterns from data to make predictions and decisions [Bishop, 2006, Hastie et al., 2009]. Generative AI seeks to synthesize new data that closely resembles real-world examples [Rombach et al., 2022, Goodfellow et al., 2016, Ho et al., 2020], while Computer Vision enables machines to interpret and analyze visual information [Szeliski, 2010, Forsyth and Ponce, 2011]. Natural Language Processing (NLP) allows computers to understand and generate human language [Manning and Schütze, 1999]. These subfields, though diverse in their applications (from recommendation systems and image generation to self-driving cars and conversational agents), are unified by the search for algorithms that learn, adapt, and perform tasks traditionally requiring human intelligence.

Unlike conventional rule-based programming, which relies on explicitly coded instructions, AI algorithms learn autonomously by analyzing data, in a manner reminiscent of the human brain. The central idea of AI is both simple and transformative: *learn from past observations to effectively address future challenges* [Mitchell, 2007].

Despite their remarkable achievements, many AI models remain “black boxes,” with internal mechanisms that are difficult to interpret. This aspect generated a growing interest in understanding and explaining the principles underlying their success.

This thesis aims to contribute to this understanding by developing a theoretical framework for analyzing and interpreting machine learning algorithms, particularly through the lens of statistical physics. By leveraging tools from statistical mechanics, we can gain insights into the underlying structures and dynamics of machine learning models, clarifying their generalization properties and sampling capabilities [Mézard et al., 1987, Mézard and Montanari, 2002]. Central to this analysis is the concept of the *loss landscape*, the mathematical surface that shows how a model’s performance changes as its parameters vary, and, in particular, its geometrical properties.

Some of the machine learning models considered in this thesis are approached from the point of view of *inference*. In this context, we make the simplifying assumption that the mathematical form of the data-generating model is known, but its parameters are not. The goal is to infer these parameters from observed data, using the loss landscape to guide the inference process. A simple example of this, in the context of statistical physics, is the *teacher-student* model [Engel and Van den Broeck, 2001].

The central theme of this thesis is the investigation of rough landscapes in machine learning and inference problems in high-dimensional spaces. We aim to a dual objective: understanding how these landscapes can be effectively navigated, both to find optimal solutions and to characterize the distribution of solutions compatible with observed data; and developing a theoretical framework for analyzing the geometry of these landscapes, to understand generalization, sampling, and devising more effective optimization algorithms.

Statistical physics offers a powerful set of tools for analyzing such complex landscapes. Techniques such as the replica method [Charbonneau et al., 2023, Mézard et al., 1984, Parisi, 1979], random matrix theory [Potters and Bouchaud, 2020, Livan et al., 2017], large deviation theory [Ellis, 1985, Touchette, 2009], and message passing algorithms [Mézard et al., 1987, Mézard and Montanari, 2002, Zdeborová and Krzakala, 2016] allow us to probe the structure of the solution space, quantify the number and nature of loss minima, and understand the typical properties of solutions found during loss minimization.

In this thesis, we employ these statistical physics tools to analyze four different machine learning problems and their associated non-convex landscapes.

- Chapter 1 provides a general introduction to machine learning, its paradigms, and the statistical physics framework that provides the basis for this thesis. It sets the stage for understanding the challenges of learning in high-dimensional spaces and introduces the key concepts and tools used in the subsequent chapters.
- In Chapter 2, we address the *non-convex perceptron* problem, which involves finding optimal weight assignments for classification tasks [Gardner, 1988]. It is characterized by a loss landscape with local minima and saddle points [Baldassi et al., 2015, 2020, 2016b]. Rather than focusing solely on optimization, we reframe the problem from an inference perspective, aiming to sample from the distribution of weights that achieve perfect classification. To this end, we introduce the *Stochastic Localization (SL)* algorithm, a sampling method inspired by diffusion models, which enables sampling from complex distributions when marginal probabilities can be efficiently computed [Montanari, 2023, El Alaoui et al., 2022, Montanari and Wu, 2023]. This approach is applied to both the continuous (spherical perceptron) and discrete (binary perceptron) settings. The Approximate Message Passing (AMP) algorithm is employed for efficient marginal computation, while the replica method is used to analyze the state evolution and limitations of SL in the high-dimensional proportional regime ($N \rightarrow \infty$). Notably, this sampling scheme succeeds in uniformly sampling solutions for the spherical perceptron across a broad range of α , but encounters intrinsic obstacles in the binary case. Building on this analysis, we elucidate the reasons behind SL’s failure in the binary perceptron and propose a strategy to overcome the impossibility of uniform sampling in this context. Chapter 2 is based on the paper [Straziota et al., 2025].
- In Chapter 3, we turn to a prototypical example of non-convex continuous optimization: the real-valued *Phase Retrieval* problem, which involves reconstructing a hidden signal from incomplete (phaseless) observations [Millane, 1990, Mondelli and Montanari, 2019]. This problem is often studied as a tractable neural network model with non-linear activation functions, and its loss landscape is notably rugged due to the absence of phase information, resulting in a proliferation of local minima [Candes et al., 2015, Bandeira et al., 2014, Sarao-Mannelli et al., 2020]. After introducing the real-valued Phase Retrieval problem, we present the *Phase Selection* formulation,

which recasts the task as a two-step optimization: first, inferring the missing phase information, and second, reconstructing the signal itself via linear regression. This decomposition provides a clearer separation between the combinatorial and continuous aspects of the problem. The theoretical analysis is conducted using replica theory, while the practical performance of various algorithmic solvers (including Approximate Message Passing (AMP), simulated annealing, and Langevin dynamics) is systematically compared. Within this framework, Phase Retrieval is explored both as an optimization problem and, through the Phase Selection perspective, as a Bayesian inference task, where the objective is to reconstruct the posterior distribution over possible solutions once the missing phase has been recovered. Chapter 3 is based on the paper [Straziota and Saglietti, 2025].

- In Chapter 4, we adopt a different perspective on rough loss landscapes analysis by moving beyond optimization and sampling in specific inference and machine learning problems, and instead directly analyzing the geometry of the loss landscape itself through spectral analysis of the Hessian matrix of the loss function for a chosen model [Bonnaire et al., 2024, Sagun et al., 2016]. The Hessian, which captures the second-order derivatives of the loss function, provides crucial insights into the local curvature of the landscape, revealing directions of steepest descent [Sagun et al., 2017]. This spectral analysis not only informs the behavior of optimization algorithms but also encodes the statistical structure and complexity of the solution space, key factors for understanding generalization. While empirical studies of the Hessian can be computationally intensive, in this chapter, we introduce a novel analytical framework for investigating the Hessian spectrum during optimization. This framework leverages tools from replica theory and random matrix theory to track the evolution of the spectrum throughout the optimization procedure. The framework is then applied to study the general loss Hessian spectrum of a simple one-hidden-layer neural network, the *Tree Committee Machine*, which serves as a fundamental building block in modern deep learning architectures [Annesi et al., 2025, Engel and Van den Broeck, 2001].
- Chapter 5 focuses on generative modeling. Here, we address the problem of learning the parameters of *Restricted Boltzmann Machines (RBMs)* from observations of visible node configurations, a task often referred to as the *inverse problem* [Ackley et al., 1985]. Learning the weights of an RBM is particularly challenging because, with access only to the visible nodes, one must first infer the hidden variables before optimizing the model. The choice of loss function is crucial, as it shapes the structure and complexity of the learning landscape. Due to the intrinsic symmetries of RBMs, the loss landscape typically contains many equivalent minima. Traditionally, learning is performed by maximizing the log-likelihood using *Contrastive Divergence (CD)*, which approximates the data distribution via a k -step MCMC scheme [Decelle and Furtlehner, 2021, Tieleman, 2008]. However, in practice, the mixing times required to accurately approximate real-world data distributions often exceed k , leading to biased and potentially suboptimal solutions. To address these limitations, we propose and analyze an interaction screening-based optimization approach as an alternative to contrastive divergence [Lokhov et al., 2018, Vuffray et al., 2016]. This method aims to provide a more robust learning mechanism, better suited to handling the symmetries and complexities of RBM loss landscapes.

Chapter 1

Preliminaries

In this chapter, we introduce basic Machine Learning notions and set the stage for the theoretical developments that follow. We begin in Section 1.1 by outlining the core concepts and terminology of modern ML. Section 1.2 then examines the principal challenges and methodological paradigms that shape today’s research directions. In Section 1.3, we reinterpret these ideas through the lens of Statistical Physics, highlighting how tools from spin-glass theory can help explain learning phenomena. Finally, Section 1.4 presents the key analytical techniques, such as replica calculations and message passing, that support the algorithmic analyses in subsequent chapters.

1.1 General Introduction to Machine Learning

The observation of natural phenomena raises the question of whether they are governed by underlying laws or are simply the result of random processes. Assuming that they can be described by mathematical models, trying to uncover and formalize such laws gave rise to entire scientific disciplines. We believe that the world is not entirely chaotic, but instead exhibits regularities that can be systematically studied and understood. As elegantly stated by Tom M. Mitchell [Mitchell, 1997]:

“Learning is possible only because the world exhibits regularities.”

In the last decades, more and more complex phenomena have been considered, starting from social behaviors and arriving at the properties of protein sequences. The enormous amount of their observations (called in the framework of Machine Learning, *Data*) and of hidden characteristics to be considered for modeling them (called in the framework of Machine Learning *Parameters*), leads to an impossibility to use classical scientific techniques. In this context, Machine Learning found its spotlight. By analyzing data, Machine Learning core objective is to uncover the hidden structure within observed data and subsequently make accurate predictions or decisions on new, unseen realizations of the phenomenon. The difference with respect to classical scientific approaches is that Machine Learning does not require a priori knowledge of the underlying laws governing the data. Instead, it relies on the data itself to learn and generalize observations, making it particularly powerful for complex and high-dimensional problems where traditional methods may fail.

In this section, basic machine learning concepts are introduced. To clarify these concepts to the reader, we present simple examples of machine learning tasks. Let’s start by considering the following:

A simple example of ML task

Suppose we are given a collection of points in a Cartesian space, $\{\mathbf{x}^\mu\}_{\mu=1}^M$ with $\mathbf{x}^\mu \in \mathbb{R}^N$, each associated with a binary label $y^\mu \in \{0, 1\}$ indicating its class membership (class 0 or class 1). We want to learn a function f that maps each input point to its corresponding class, i.e., $y^\mu = f(\mathbf{x}^\mu)$. To achieve this, we use a subset of the data with known labels, referred to as the *training set*, to infer the rule f . Once f is learned, we evaluate its performance on new, previously unseen points (the *test set*). A graphical illustration of this classification task is shown in Figure 1.1.

To perform learning tasks, four essential components are always required in ML: the *Dataset*, the *Architecture/model*, the *Loss Function*, and the *Learning Algorithm*. In the following sections, these components are presented in detail.

1.1.1 Datasets

The *Dataset* is the essential building block for any learning approach: it is the set of elements containing the information necessary to perform a task. In the previous simple Cartesian classification example, the dataset is made by the collection of points and labels, which are used to learn how to classify new $\mathbf{x}^\mu$.

They contain all the information necessary to perform the task. For this reason, the choice of a specific rule for the class assignment depends on the properties of the dataset.

A simple example of the effect dataset structure

Let's consider again the Cartesian classification example: it is clear that the distribution of the points in the space influences the model used for the classification. Let's consider that we want to perform two different classification tasks: classification of linearly separable data or classification of XOR-like data, as reported respectively in Figure 1.1 left and right panels [Rumelhart et al., 1988]. Not all the models used for the first task can be used for the second task, too.

Different tasks and types of datasets have inspired different ML paradigms: the most prominent ones are *Supervised Learning* and *Unsupervised Learning*.

Supervised Learning

In *Supervised Learning* [Vapnik, 1999], we are provided with a dataset containing two types of elements: the input data (or *patterns*), denoted by $\mathbf{x}$, and the corresponding output/target values y we aim to predict. The dataset is typically expressed as:

$$\mathcal{D}_{\text{Supervised}} = \{(\mathbf{x}^\mu, y^\mu)\}_{\mu=1}^M, \quad (1.1)$$

where M is the total number of samples, μ indexes a specific element of the dataset, $\mathbf{x}^\mu \in \mathbb{R}^N$ and $y^\mu \in \mathbb{R}$ ¹. The goal is to learn a mapping f from inputs to outputs that generalizes well to new data. Typically, the dataset is divided into a *training set* (which teaches the model

¹In the thesis we assume that the output is a scalar value, however in general one can assume that $y^\mu \in \mathbb{R}^K$ for some $K > 1$.

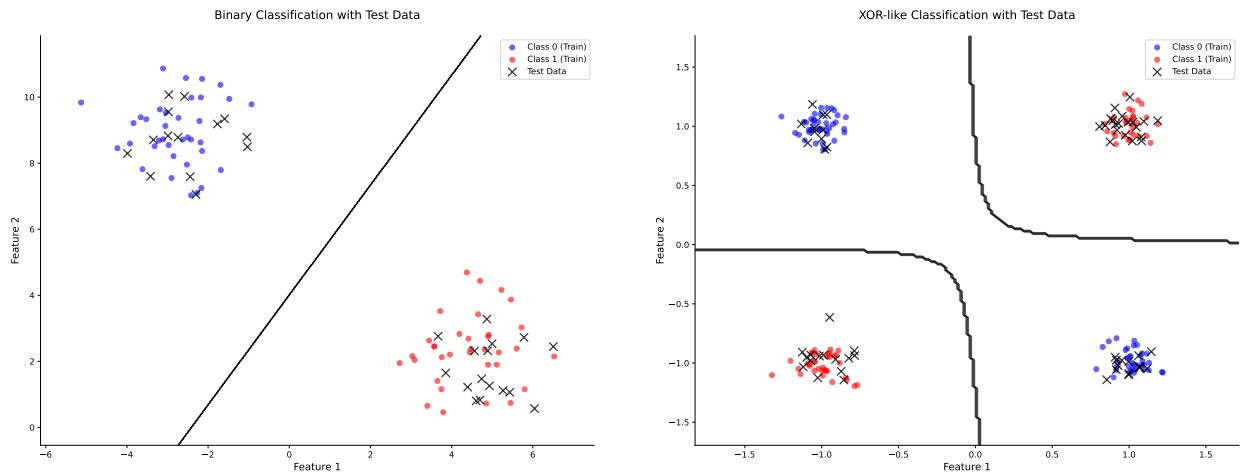


Figure 1.1: *Left panel:* Graphical representation of the classification example. The elements for the two classes of the training set are reported in red and blue, while the elements of the test set are shown as black "x". The black line represents a possible classification threshold detected by the function f . *Right panel:* same graphical representation with data distributed according to XOR structure.

to predict y^μ from $\mathbf{x}^\mu$) and a *test set*, used to evaluate the model performance² [Goodfellow et al., 2016].

Depending on the type of output, supervised learning tasks are primarily classified as:

- *Regression:* Involving continuous outputs [Vapnik, 1999]. The output y^μ can assume any real value. Regression tasks try to find a way to map any single input into an output.
- *Classification:* Involving discrete labels [Bishop, 2006]. In this case, we assume that data can be separated into classes: inside the same class, patterns show similar properties and characteristics (typically geometrical).

The Cartesian example previously introduced is an example of a Classification task.

Unsupervised Learning

In *Unsupervised Learning*, no explicit output labels are provided. Since we are not provided with specific outputs, the objective is to uncover hidden patterns or structures within the patterns through techniques such as:

- *Clustering:* Grouping similar patterns together [Jain et al., 1999]. The objective of clustering techniques is to find a way to agglomerate patterns based on an intrinsic geometrical property. Usually, clustering is performed by grouping $\mathbf{x}^\mu$ based on a distances notion.
- *Dimensionality Reduction:* Simplifying high-dimensional patterns while retaining essential features [Jolliffe, 2002]. In other words, dimensionality reduction techniques take high-dimensional patterns and remove all their useless components.

²Sometimes the test set is further divided into a *validation set*, used to fine-tune the model hyperparameters.

In this case, the dataset consists solely of patterns:

$$\mathcal{D}_{\text{Unsupervised}} = \{\mathbf{x}^\mu\}_{\mu=1}^M,$$

where again $\mathbf{x}^\mu \in \mathbb{R}^N$. As a simple example of clustering and dimensional reduction tasks, one can consider:

A simple example of Clustering Task

Given a set of Cartesian points, one can group them based on the notion of Euclidean distance.

A simple example of Dimensional reduction task

Given a set of Cartesian points in $\mathbb{R}^n$, laying on a lower dimensional manifold of $\mathbb{R}^m$, with $m < n$, a simple dimensional reduction task consists in removing the $n - m$ useless components, projecting them in a space characterized by a smaller dimension.

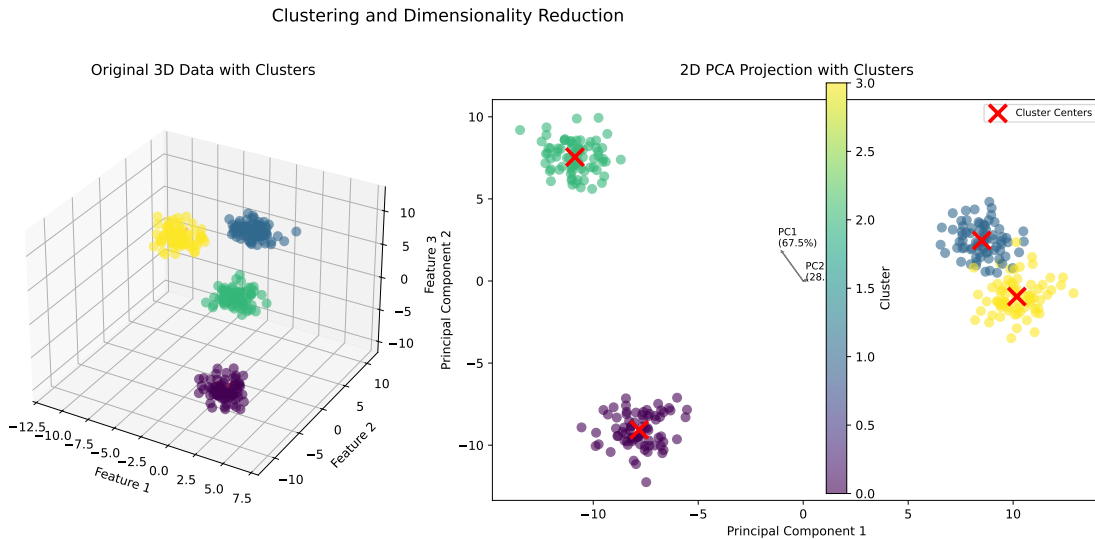


Figure 1.2: *Left panel:* Graphical representation of a clustering task. The points are grouped based on their distance (different colors represent different clusters). *Right panel:* Graphical representation of a dimensionality reduction task. The points are projected from $\mathbb{R}^3$ to $\mathbb{R}^2$, based on a technique called Principal Component Analysis (PCA) [Pearson, 1901, Jolliffe, 2002].

Additional Paradigms: Self-Supervised and Semi-Supervised Learning

In recent years, new paradigms have emerged, bridging the gap between supervised and unsupervised learning:

- *Self-Supervised Learning:* In recent years, due to the enormous quantity of data used by the Machine Learning Algorithms, a part of the data may be corrupted. Self-supervised learning faces this problem: the Machine Learning model generates its

supervisory signals (for example, by predicting missing parts of the input) to learn robust representations [Jing and Tian, 2020].

- *Semi-Supervised Learning*: Another uncomfortable situation is the following: sometimes, the technical operation to label an input (to be able to perform supervised learning) can be very expensive. To face this problem, Semi-supervised learning was introduced: the dataset is made of a small set of labeled data combined with a larger set of unlabeled data to improve model performance, thereby reducing the need for extensive manual annotation [Chapelle et al., 2006].

The main paradigms of datasets in Machine Learning have been described up to now. The theoretical investigation in this thesis mainly concentrates on *Supervised Learning tasks*, where the dataset takes the form shown in Equation 1.1.

1.1.2 Models

If on one side the data are the elements containing the information that we want to model, on the other side we need to choose how to use this information to perform the task (the shape of the law we use for describing observations). In practice, we need to choose the form of the transformation f that we want to apply to the data. The fundamental component in ML performing this transformation is called *model*. It defines the computational structure that processes input data and produces predictions, as well as the assumed functional form of the data-generating process. In machine learning, different tasks can be performed using various models. In this thesis, we consider models belonging to the class of Artificial Neural Networks (ANN), [Goodfellow et al., 2016] (called *architectures*). They have been inspired by the brain neural networks, where single computational units, called neurons, are linked through dendritic connections [McCulloch and Pitts, 1943, Rosenblatt, 1958]. From a mathematical point the architectures can be seen as parametric functions that are applied to data to transform the inputs into outputs. From the example in Figure 1.1, it is clear that each architecture is associated with a different parametric form of the transformation, and thus different architectures produce different performances on different datasets.

To give a bit of context, we now present the general structure of a Neural Network with just one computational *layer*, whose components are now outlined.

Essential components of an Artificial Neural Network

The essential components of an Artificial Neural Network include (see Figure 1.3 for a graphical representation):

1. *Input Nodes*: Representing the features of the input data. Given the dataset $\mathcal{D}$, they are denoted by $\mathbf{x}^\mu \in \mathbb{R}^N$.
2. *Weights ($\mathbf{w}$)*: Each input x_i^μ is associated with a weight w_i^μ that scales its contribution to the output.
3. *Bias (b)*: An additional parameter that shifts the weighted sum, thereby increasing the model's flexibility.
4. *Summation Function*: Computes the net input as

$$z^\mu = \sum_{i=1}^n w_i x_i^\mu + b,$$

Aggregating the contributions from all inputs.

5. *Activation Function (σ)*: Applies a nonlinear transformation to the net input, enabling the model to capture complex relationships. Common choices include the sigmoid, ReLU, and step functions.
6. *Output*: The final prediction of the output is obtained as

$$\hat{y}^\mu = \sigma(z^\mu).$$

This description outlines a single layer of an architecture, producing just one output^a. In principle, one can consider models producing several outputs, and in that case the weights are represented by a matrix $W \in \mathbb{R}^{N \times Q}$ and the biases are vectors $\mathbf{b} \in \mathbb{R}^Q$, where Q is the number of outputs. In contrast, modern neural networks are significantly more complex than a simple single-layer neural network (one example of which is the *Perceptron*), typically comprising multiple layers in which the output of the previous layer serves as the input to the next (see Figure 1.3).

^aNote that the bias, summation function, and activation function are not explicitly shown in Figure 1.3, but are intrinsically contained within node $h_i^{(j)}$.

As previously introduced, depending on the task to be performed, architectures vary widely in complexity and capability, and are tailored to tasks such as classification, regression, and data generation. The choice of architecture depends on the problem type, data characteristics, and desired computational efficiency. Common architectures include:

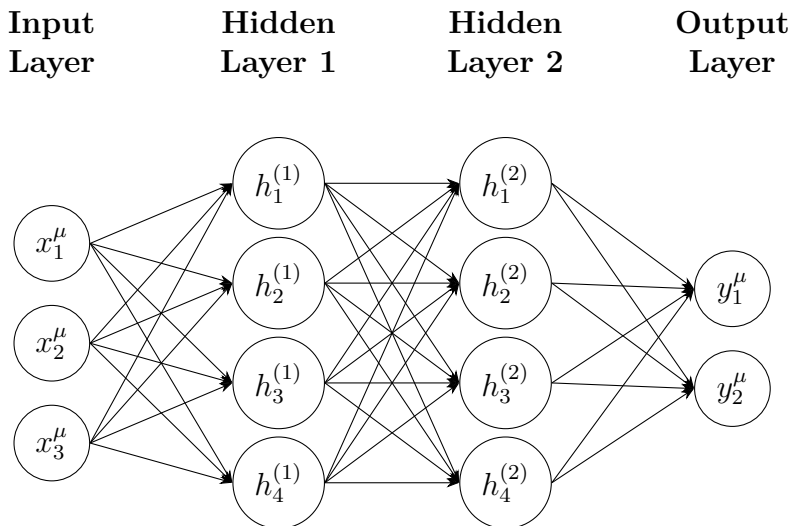


Figure 1.3: General Neural Network Architecture with 2 hidden layers. Each layer is made of several nodes, indicated by $h_i^{(l)}$, with $i \in N_l$ and $l \in L$ (L is the number of layers). For the hidden layers, we report the node as a full computational unit, which includes the bias, summation function, and activation function. The input layer is made of the input nodes x_i^μ , while the output layer is made of the output nodes y_i^μ . The connections between the nodes are represented by arrows, indicating the flow of information from one layer to another.

Common Architectures

- *Linear Models*: Which assume a linear relationship between inputs and outputs. These are among the simplest models used in ML, with examples including logistic regression and linear regression [Hastie et al., 2009].
- *Support Vector Machines (SVMs)*: They construct an optimal hyperplane to separate data points belonging to different classes, primarily used for classification tasks [Cortes and Vapnik, 1995].
- *Transformers*: Architectures specifically designed to handle sequential data (such as text or time series) using attention mechanisms [Vaswani et al., 2017]. This type of architecture is very popular in Generative AI and several contexts of LLM.

It is important to note that although these models are designed for specific tasks, they remain approximations of the underlying phenomena that generate the data. As George E. P. Box famously stated:

“All models are wrong, but some are useful.”

Given that this thesis aims to establish a rigorous mathematical and theoretical foundation for ML, the focus is primarily on simplified architectures. The main architectures presented in this thesis are the Perceptron [Rosenblatt, 1958], a simple single-layer architecture; the Tree-Committee Machine, an architecture with two layers of neurons [Engel and Van den Broeck, 2001], whose hidden nodes are connected to a disjoint set of first layer nodes; and Restricted Boltzmann Machines (RBMs) [Ackley et al., 1985], simple fully connected two-layer neural architectures. A deeper description of this architecture is performed

in the next Section.

1.1.3 Loss Functions

Once the model has been chosen, it is important to detect the best values for its weights and biases. Their initial values are chosen completely at random. But how can we get better values for them? The improvement of these parameters is performed during the *Training* procedure [Bishop, 2006].

Let's consider that we are given a training dataset $\mathcal{D}_{Train} = \{\mathbf{x}^\mu, y^\mu\}_{\mu=1,\dots,M}$ and an architecture performing a transformation, represented by a function labeled by $f_{\mathbf{w},b}(\mathbf{x})$. This transformation produces $\hat{y}^\mu = f_{\mathbf{w},b}(\mathbf{x}^\mu)$. Since weights and biases at the beginning of the training are random, not all $\hat{y}^\mu$ will match y^μ . For this reason, during the training procedure $\mathbf{w}$ and b need to be updated, so that we improve the match between $\hat{y}^\mu$ and y^μ .

Since the parameters of the model need to be updated to improve the similarity of $\hat{y}^\mu$ and y^μ , for $\mu \in \{1, \dots, M\}$, it seems reasonable to update the parameters of the architecture ($\mathbf{w}$ and b) according to the distance between the predicted and actual outputs ($\hat{y}^\mu$ and y^μ), called *loss function*, in the context of ML. Depending on the task and on the nature of the outputs, different loss functions can be devised. Some commonly used loss functions are the following:

Common Loss Functions

- *Mean Squared Error (MSE)*: Commonly used for regression tasks,

$$\mathcal{L}_{\text{MSE}}(\mathbf{w}, \mathbf{b}; \mathcal{D}) = \sum_{\mu=1}^M (y^\mu - \hat{y}^\mu)^2, \quad (1.2)$$

where $\hat{y}^\mu$ is the prediction corresponding to the μ^{th} example.

- *Cross-Entropy Loss*: Frequently adopted for classification tasks,

$$\mathcal{L}_{\text{XENT}}(\mathbf{w}, \mathbf{b}; \mathcal{D}) = - \sum_{\mu=1}^M \sum_{i=1}^C y_i^\mu \log(\hat{y}_i^\mu), \quad (1.3)$$

where y_i^μ equals 1 if the μ^{th} sample belongs to class i (and 0 otherwise), and $\hat{y}_i^\mu$ denotes the predicted probability to belong to class i for the input $\mathbf{x}^\mu$.

- *Theta Loss*: Occasionally used in binary classification, this loss function indicates whether the prediction matches the true label. It is defined as:

$$\mathcal{L}_\theta(\mathbf{w}, \mathbf{b}; \mathcal{D}) = \sum_{\mu=1}^M \theta(-y^\mu \hat{y}^\mu), \quad (1.4)$$

where $\theta(\cdot)$ is the Heaviside step function^a.

^aNote that due to its non-differentiability, this loss is rarely used in gradient-based optimization.

While numerous loss functions exist for a variety of tasks, here we focus on those most pertinent to our analysis.

Regularization

Loss functions are instrumental in learning optimal model parameters; however, it is often desirable to enforce specific properties on these parameters. Regularization techniques are incorporated into the loss function to enforce constraints on the model parameters³. The most commonly used regularization [Tikhonov and Arsenin, 1977, Hastie et al., 2009] methods include:

- *L₂-regularization*: Penalizes large weights by adding a term proportional to the sum of squared weights. This is the most commonly used regularization technique, and it is usually used for preventing overfitting of the model.

$$\mathcal{L}_2(\mathbf{w}) = \|\mathbf{w}\|_2 \quad (1.5)$$

- *L₁-regularization*: Encourages sparsity by adding a term proportional to the sum of the absolute values of the weights. It is usually used for enforcing sparsity in the model, i.e., to reduce the number of parameters of the architecture.

$$\mathcal{L}_1(\mathbf{w}) = \|\mathbf{w}\|_1 \quad (1.6)$$

1.1.4 Optimization Algorithms

As introduced in the previous section, the definition of the loss function clarifies how the architecture's estimate of the output is distant from the real one. However, to properly update the values of the weights and biases, we need a procedure reducing the value of the loss function. The learning procedure of parameters values is defined through the *Optimization Algorithms*. A large number of these optimization (learning) algorithms are elements of a class of algorithms called *Empirical risk minimization*. In this class of algorithms, once a risk function is defined (in the case of machine learning, we deal with a loss function), several steps are taken to modify slightly the parameters of the architecture and reduce the distance between the real output and estimated ones.

The father of the majority of nowadays optimization algorithms is *Gradient Descent* [Bottou, 2010, Nesterov, 2013]. A brief description of gradient descent is now reported.

³They are a function of the only parameters of the architecture. For maintaining a clear notation, we present their definition for the only weights, and not for the biases.

Gradient Descent

Given a dataset $\mathcal{D} = \{\mathbf{x}^\mu, y^\mu\}_{\mu=1, \dots, M}$ ^a, an architecture parameterized by weights $\mathbf{w}$ and an activation function $\sigma(\cdot)$, and a loss function $\mathcal{L}(\cdot)$, the weights are updated iteratively^b as follows:

$$w_i^{(t)} = w_i^{(t-1)} - \eta \frac{\partial \mathcal{L}(\mathbf{w}, \mathbf{b}; \mathcal{D})}{\partial w_i}, \quad (1.7)$$

where η represents the learning rate (a parameter controlling the step size of the update). The essential idea is to follow the gradient direction that most effectively decreases the loss. Training is carried out for T iterations. A modern version of GD is Stochastic GD, at each iteration, a different subset of the data (called a *batch*) is shown. Although initial iterations may yield imperfect predictions, exposing the model to the data in different batches leads to improved accuracy.

^aI consider, for simplicity the case of a single output, however everything can be generalized to multiple outputs case.

^bNote that the bias can be treated in the same way as the weights, and thus is not explicitly reported in the following equations.

Modern optimization algorithms

Common first order optimization techniques (techniques using only the gradient of the loss for optimizing architecture weights) include *Stochastic Gradient Descent (SGD)* [Robbins and Monro, 1951, Bottou, 2010]; *Adam* [Kingma and Ba, 2015], which add in the optimization procedure memory term, concerning previous iteration of the algorithm (The effect of the momentum term is to reduce the oscillation of the optimizer); *Proximal Gradient Descent* [Combettes and Wajs, 2007], helping in implementing a gradients on losses having a non differentiable regularization terms; and *Mirror Descent* [Nemirovski and Yudin, 2004], an algorithm working in the same direction of Proximal Gradient Descent. Recently, new optimization techniques, using second-order information of the loss, i.e., the Hessian matrix of the loss, have been implemented for reducing the dependence on the choice of the learning rate. A simple example is the so-called *Second Order Newton Method* [Nocedal and Wright, 2006].

It is important to stress that modern deep learning models, perform Gradient computation typically employing the *Backpropagation algorithm* [Werbos, 1974, Rumelhart et al., 1986], a specific application of the chain rule from calculus, allowing efficient computation of gradients for all weights in the network [Goodfellow et al., 2016].

1.2 Tasks and Challenges in Machine Learning

Modern machine learning focuses on extracting meaningful structure from high-dimensional data. Two different strategies are commonly distinguished: rich, task-specific feature-learning models, which adapt their internal filters during training; and 'lazy' or kernel-like methods, whose parameters remain "essentially fixed". Training either kind of model entails navigating highly non-convex loss landscapes and striking a delicate balance between fitting

the training set and generalising to unseen samples. In the following, a quick review of these concepts is presented.

Representation Learning: In machine learning, we work with three main ingredients. Data are the raw signals we observe (images, speech, text, and so on). Model parameters are the adjustable weights that tell the model how to turn inputs into outputs; they are not features or hidden variables. As the data flows through a neural network, each layer produces an activation—a vector or tensor whose numbers bring useful patterns to the surface. Every neuron is like a tiny pattern detector: it “fires” when it spots something simple, like a short edge in a picture or a brief sound in speech. By stacking many such detectors, later layers start to pick out larger, more abstract concepts—for example, a whole face instead of just an edge. Teaching the network to discover these detectors on its own, rather than hand-crafting them, is precisely the goal of representation learning, in which one aims to reconstruct useful representations.

The representation learning approach contrasts with classical kernel methods, where a fixed, human-designed feature map is chosen a priori. The ability to learn both parameters and internal representations has propelled recent breakthroughs in vision, speech, and natural-language processing [Goodfellow et al., 2016].

Lazy Training Regime and Kernel Limits: Recent theory highlights a “lazy” training regime where a neural network’s parameters barely move from their initialization, making the model behave like a fixed-kernel method. For infinitely wide networks, Jacot et al. showed that, during gradient descent training, the network function evolves linearly in parameter space [Jacot et al., 2018]. In this regime, the network essentially performs kernel regression with a data-dependent kernel. Chizat et al. further explain that lazy training arises from specific parameter scalings: the model is then equivalent to its linearization around initialization, converging fast but without meaningful feature adaptation [Chizat et al., 2019]. These works imply that over-parameterized networks (networks with a huge amount of parameters) can achieve zero training loss very quickly by staying close to initialization. However, experiments show that forcing a lazy regime often *degrades* performance on complex tasks, suggesting that the success of deep learning comes from *rich* representation learning where representations evolve significantly [Huang and Yau, 2020].

Non-Convex Optimization and Loss Landscapes: Training deep networks requires optimizing a highly non-convex loss function. Classical theory might predict that many poor local minima exist, but modern understanding is more nuanced. It is now believed that in large networks, most critical points are saddle points rather than bad minima [Dauphin et al., 2014]. Choromanska et al. mapped certain neural loss landscapes to models from statistical physics, showing that for wide networks the vast majority of local minima lie in a narrow band of near-optimal loss, and that high-loss minima become exponentially rare as width grows. Consequently, gradient descent is likely to find one of these “good” minima and achieve low training error [Choromanska et al., 2015].

Recent statistical physics analyses reveal complex hierarchical structures in the loss and suggest that flatter minima often correlate with better generalization [Sagun et al., 2017, Goyal et al., 2018, Baldassi et al., 2023, 2016b]. Random initialization and large parameter counts tend to yield similarly low-loss solutions, mitigating concerns about getting stuck in a

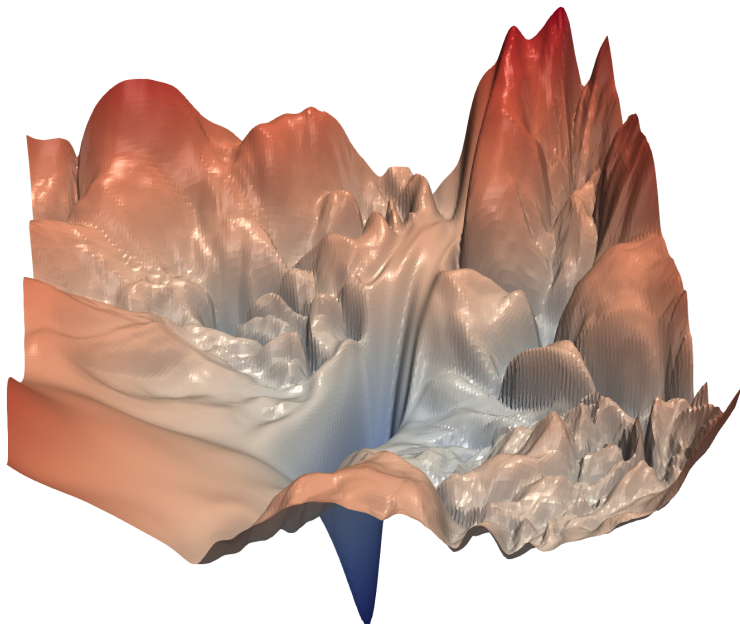


Figure 1.4: Tridimensional example of a non-convex loss landscape with minima, saddle points, and flat regions [Li et al., 2018].

“bad” valley. The main object of investigation of this thesis is high-dimensional non-convex loss landscapes.

Overfitting and Generalization: Overfitting, fitting noise or spurious patterns in training data, instead of learning the underlying data distribution, is a central concern in ML. Classical wisdom based on VC theory and structural risk minimization holds that model complexity must be controlled to generalize well [Vapnik, 1999]. Regularization techniques like weight decay, early stopping, and dropout [Bishop, 2006] are widely used to counter overfitting. However, modern over-parameterized models complicate this picture [Srivastava et al., 2014]. Zhang et al. demonstrated that deep networks can achieve *zero training error even on random labels*, revealing vast capacity beyond what classical complexity measures would predict [Zhang et al., 2021].

A simple example of overfitting

Consider a dataset $\{(x^\mu, y^\mu)\}_{\mu=1}^M$ arranged roughly along a line with some added noise. A linear model would capture the true relation. However, fitting a high-degree polynomial may perfectly match the training points, including noise, leading to poor generalization on new data.

Bias–Variance Trade-Off and Double Descent: The bias–variance trade-off offers a classical lens for generalization. A too simple model, a model with few parameters, underfits (produces high bias); while a too complex model, a model with too many parameters, overfits (produces high variance). Neither situation is optimal for learning performance. The value of the loss on the test set is minimized at a sweet spot between these extremes. Mathematically, the expected prediction error can be decomposed as:

$$\mathbb{E}[(y - \hat{y})^2] = \text{Bias}^2 + \text{Variance}.$$

However, this is not the complete scenario. Belkin et al. showed that test error can decrease after perfect training accuracy is achieved, leading to a “double descent” curve [Belkin et al., 2019, Geman et al., 1992].

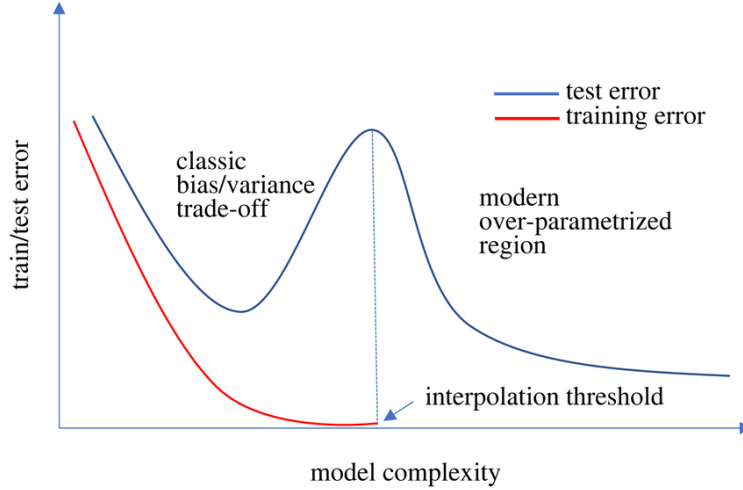


Figure 1.5: Graphical representation of the classical and modern (double descent) bias-variance trade-off [Chawla et al., 2022].

1.2.1 Concluding Remarks

Representation learning, lazy regimes, non-convex optimization, overfitting, and bias-variance are interlinked tasks and challenges in modern machine learning. In this thesis, we mainly focus on the analysis of the geometrical structure of the loss landscape through the lens of statistical physics. The next section details this approach and presents the fundamental techniques of statistical physics of machine learning.

1.3 Statistical Physics of Machine Learning

The techniques used for analyzing ML loss landscapes are borrowed from Statistical Physics.

Statistical Physics is a field of theoretical physics, whose aim is to infer macroscopic properties of a system made of an extensive number of interacting particles. Such systems cannot be investigated by considering every single interaction among the particles forming it, but instead, it is worth investigating its macroscopic properties. By employing probabilistic methods and central concepts, such as free energy, phase transitions, and the interplay between order and disorder, statistical physics provides a powerful framework for interpreting emergent phenomena [Landau and Lifshitz, 1980]. A key advantage of this approach is its ability to bridge the gap between rigorous mathematical formulations and the pragmatic efficiency of engineering solutions, which often sacrifice theoretical depth. However, a significant challenge of this methodology lies in its need for simplifications in the models considered.

Modern Machine Learning techniques make use of very big architectures and huge datasets. It is not possible to analyze them by considering separately each connection

among different layers. Using statistical physics tools, one can overcome these dimensional issues by finding analogies among concepts of both disciplines.

To develop analytical models for Machine Learning, certain simplifications are necessary. The statistical physics approach intentionally abstracts away complicated or irrelevant details to focus on the fundamental elements of the problem. In this chapter, we review the key components of machine learning (presented in the previous section) from a statistical physics perspective and introduce the essential tools required to construct the theoretical framework underlying modern machine learning.

1.3.1 A crash course on Statistical Physics

In this section, we introduce key concepts from Statistical Physics and highlight their formal analogies with components of Machine Learning systems.

Consider a system composed of N interacting units, whose microscopic state is defined by a configuration vector $\boldsymbol{\sigma} = \{\sigma_i\}_{i=1,\dots,N}$. Each configuration is assigned an energy via a function known as the *Hamiltonian*, $H(\boldsymbol{\sigma})$, which encodes the interaction structure of the system.

The central goal of Statistical Physics is to determine the probability of observing a configuration $\boldsymbol{\sigma}$ in equilibrium. In the *canonical ensemble*, where the temperature is fixed and the system exchanges energy with an external reservoir, this probability is given by the *Boltzmann distribution*:

$$P(\boldsymbol{\sigma}) = \frac{e^{-\beta H(\boldsymbol{\sigma})}}{Z}, \quad (1.8)$$

where $\beta = 1/(k_B T)$ is the *inverse temperature*, and Z is the *partition function*:

$$Z = \sum_{\boldsymbol{\sigma}} e^{-\beta H(\boldsymbol{\sigma})}. \quad (1.9)$$

The partition function is not merely a normalization constant; it encodes important macroscopic thermodynamic information about the system. For instance, one can compute the expected values of physical observables and thermodynamic potentials from it.

Among these, a central role is played by the *Helmholtz free energy*, defined as:

$$F = -\frac{1}{\beta} \log Z. \quad (1.10)$$

This quantity bridges microscopic and macroscopic descriptions by determining the most probable behavior of the system at a given temperature.

Although the system can explore many configurations due to thermal fluctuations, it is overwhelmingly likely to be found in configurations minimizing the *free energy*. In the *low-temperature limit* $\beta \rightarrow \infty$, the Boltzmann measure concentrates on the configurations that minimize the Hamiltonian, i.e., the ground states. In contrast, in the *thermodynamic limit* $N \rightarrow \infty$, fluctuations around the typical energy become negligible due to the law of large numbers (self-averaging).

Let us now consider the probability of observing a given energy value E . Since multiple configurations may share the same energy, we define the *density of states* $\Omega(E)$, which counts the number of configurations with energy E . The probability of observing such energy is then:

$$P(E) \propto \Omega(E) e^{-\beta E} = e^{S(E) - \beta E} = e^{-\beta [E - TS(E)]} = e^{-\beta N f(E)}, \quad (1.11)$$

where $S(E) = \log \Omega(E)$ is the *entropy*, and $f(E) = \frac{1}{N}[E - TS(E)] = \frac{F(E)}{N}$ is the *free energy density*. The most probable energy corresponds to the minimum of $f(E)$, justifying the central role of free energy in determining equilibrium behavior.

Analogy with Machine Learning

This framework provides a powerful analogy for analyzing Machine Learning systems from a statistical physics perspective. In this analogy:

- The *loss function* $\mathcal{L}(\mathbf{w}; \mathcal{D})$ plays the role of the *Hamiltonian*: it assigns an “energy” to each configuration $\mathbf{w}$ of the model’s parameters (from now on we consider just $\mathbf{w}$ and we assume the bias to be null).
- The *parameter space* corresponds to the configuration space.
- The *posterior distribution* over weights, often written as $P(\mathbf{w}|\mathcal{D}) \propto e^{-\beta \mathcal{L}(\mathbf{w}; \mathcal{D})}$, is formally equivalent to the Boltzmann distribution.

The parameter β controls the concentration of this distribution: at high temperature (low β), the system explores the parameter space broadly; at low temperature (high β), it concentrates around minima of the loss. In the zero-temperature limit $\beta \rightarrow \infty$, the model behaves deterministically, converging to configurations that minimize the loss function.

This perspective enables the use of powerful tools from Statistical Physics to study the *typical properties* of learning systems, particularly in high-dimensional settings.

Quenched Disorder and Typical Behavior

A crucial aspect of learning problems is the role of the *training dataset*, which is typically modeled as a realization of a random process. From a statistical physics viewpoint, this corresponds to the notion of *quenched disorder*: a random variable that is fixed during the optimization process but over which we must average when computing macroscopic quantities.

Let $\mathcal{D}$ be a dataset sampled from some distribution. The loss function depends explicitly on $\mathcal{D}$, and so does the partition function $Z(\mathcal{D})$. To compute meaningful, dataset-independent predictions, one introduces *quenched averages*, such as the *quenched free energy*:

$$F(\beta, P) = -\frac{1}{\beta} \langle \log Z(\mathcal{D}) \rangle_{\mathcal{D}}, \quad (1.12)$$

where $\langle \cdot \rangle_{\mathcal{D}}$ denotes the expectation over the dataset distribution. This contrasts with *annealed* averages, where the average of $\log Z$ is replaced by $\log \langle Z \rangle_{\mathcal{D}}$, which generally yields incorrect results in systems with strong disorder, which is often the case in the context of Statistical Physics of Machine Learning.

Training and Generalization Errors

Using these tools, one can define key performance metrics in statistical terms for the performance of a learning algorithm. The *training error* ϵ_t and the *generalization error* ϵ_g ⁴ are

⁴The generalization error is usually defined when the considered data have a structure, as in the case of teacher student framework, defined in the next Section.

defined as:

$$\epsilon_t = \left\langle \int d\mathbf{w} P(\mathbf{w}|\mathcal{D}) \frac{1}{M} \sum_{\mu=1}^M \ell(\mathbf{w}; \mathbf{x}^\mu, y^\mu) \right\rangle_{\mathcal{D}}, \quad (1.13)$$

$$\epsilon_g = \left\langle \int d\mathbf{w} P(\mathbf{w}|\mathcal{D}) \int P(d\mathbf{x}, dy) \ell(\mathbf{w}; \mathbf{x}, y) \right\rangle_{\mathcal{D}}, \quad (1.14)$$

Where:

- $P(\mathbf{w}|\mathcal{D}) \propto e^{-\beta \mathcal{L}(\mathbf{w}; \mathcal{D})}$ is the posterior over weights,
- $\ell(\mathbf{w}; \mathbf{x}, y)$ is the per-sample loss function,
- $\langle \cdot \rangle_{\mathcal{D}}$ averages over the dataset distribution.

These expressions help in studying the theoretical performance of the model, but their computation is notoriously hard in realistic architectures. Analytical results are usually obtainable only under simplifying assumptions (e.g., linear models, Gaussian inputs, shallow architectures) using the *replica method*.

1.3.2 Components in Statistical Physics of Machine Learning

In this section, the four main components of ML, presented in Section 1.1, are revisited in the optics of statistical physics. The datasets, architectures, loss functions, and learning algorithms used in statistical physics of Machine Learning are introduced, so to provide the reader a solid background for understanding the main chapters of this thesis.

Data

In Section 1.1.1, we discussed how data constitute the starting point for any learning task, as they carry the essential information needed for the learning process. In the language of statistical physics, data play the role of quenched disorder, i.e., a fixed realization over which averages of observable quantities are computed. However, our objective is to characterize general properties of learning algorithms independently of any specific realization of the disorder. This requires a shift in perspective: rather than focusing on a particular dataset, we aim to compute average quantities over the distribution of datasets. Consequently, real-world datasets such as MNIST are not suitable for our theoretical analysis, as their underlying data-generating distribution is unknown and uncontrolled. For this reason, throughout the thesis, we work with artificial datasets sampled from known distributions. Specifically, we consider datasets composed of independent and identically distributed (i.i.d.) samples (with either binary or continuous patterns) drawn from a prescribed probability distribution. Under the i.i.d. assumption, we assume that both features within each pattern and the patterns themselves are drawn independently, leading to a fully factorized joint distribution (this assumption is far from being realistic, but it is a common assumption in statistical physics of machine learning, as it allows for analytical tractability):

$$P(\{x_1^\mu, \dots, x_N^\mu\}) = \prod_{i=1}^N P(x_i^\mu). \quad (1.15)$$

Within Statistical Physics frameworks, we primarily deal with Gaussian data, i.e., $x_i \sim \mathcal{N}(0, 1)$ for continuous examples, or Bernoulli distributed data, i.e., $x_i \sim \frac{1}{2}\delta(x_i, 1) + \frac{1}{2}\delta(x_i, -1)$ for binary patterns. These assumptions facilitate the application of general theorems, such as the Central Limit Theorem.

Regarding label distributions, this thesis primarily addresses binary classification tasks with randomly generated labels $y^\mu \in \{-1, +1\}$. In Statistical Physics frameworks, labels can be generated by two main schemes:

Labels generating schemes

- *Random Labels:* Here, the dataset $\mathcal{D} = \{\mathbf{x}^\mu, y^\mu\}_{\mu=1, \dots, M}$ is entirely random; labels are sampled with equal probability (Bernoulli distribution). In this unstructured setting, patterns and labels are independent, and the overall distribution factorizes. The classification problem of a completely random dataset is called in the literature: *Storage Capacity*. This setup corresponds to the most unstructured learning scenario, where no correlation exists between input and output. Although this problem does not capture the complexity of real-world ML tasks, it is important for its combinatorial properties—specifically, it enables the computation of the probability that a random dataset is linearly separable as a function of $\alpha = M/N$. In other words, one calculates the fraction of weight space volume that correctly classifies the dataset [Vapnik and Chervonenkis, 1971, Engel and Van den Broeck, 2001].
- *Structured Labels:* In this case, data are not sampled randomly but are generated according to a specific law. The generative model (the *teacher*) is assumed to be known in its structural form, although its parameters, denoted by $\mathbf{w}^*$, remain unknown:

$$y^\mu = f(\mathbf{w}^*, b^*, \mathbf{x}^\mu) \quad \mu \in \{1, \dots, M\}. \quad (1.16)$$

The learning task (the *Teacher-Student Problem*) is to infer the teacher's weights and biases by using a network (the *student*) with the same architecture but randomly initialized parameters. Unlike the storage problem, the teacher-student scenario deals with a structured dataset and focuses on learning the underlying rule and allows the study of generalization performance, as the true labeling function is known. [Gardner, 1988, Engel and Van den Broeck, 2001].

It is important to underline that the teacher-student scenario is not a realistic representation of real-world ML tasks, as it assumes the existence of a known generative model. However, it serves as a useful theoretical framework for understanding the learning dynamics and generalization properties of machine learning algorithms. Furthermore, under the teacher-student setting, learning tasks can be reframed as inference tasks, where the goal is to infer the parameters of the teacher model from the data. This perspective allows for a deeper understanding of the learning process and the role of different components in machine learning systems. In two chapters of this thesis, we focus on ML problems from the point of view of inference.

Once the dataset is defined, it is crucial to quantify the amount of information it contains. This measure is given by the ratio [Vapnik and Chervonenkis, 1971, Engel and Van den

Broeck, 2001]:

$$\alpha = \frac{M}{N} = \frac{\text{Number of examples}}{\text{Dimension of the examples}}, \quad (1.17)$$

which plays a fundamental role in Statistical Physics analysis. In the present thesis, we primarily focus on the case where α is a constant, i.e., the number of examples grows proportionally to the dimension of the patterns, and $N \rightarrow \infty$. These assumptions are crucial for ensuring that the dataset remains analytically treatable as the system size increases.

Architecture

As presented in Section 1.1.2, the architectures represent the functional form one assumes when modeling the relation between labels and patterns. As shown in the introduction to Statistical Physics of Machine Learning, the presence of many layers makes computations intractable. Simplifying assumptions are necessary. In this work, we avoid the complications of multi-layer architectures because the main theorems and techniques in Statistical Physics of ML have not been rigorously established for deep networks, or they require computational resources beyond practical limits. Consequently, the architectures considered in this thesis are primarily architectures containing one or two layers.

- *Perceptron*: The simplest architecture for binary classification, consisting of a single layer of neurons with weights w_i for $i \in \{1, \dots, N\}$. Introduced in [Rosenblatt, 1958], the perceptron has been shown to correctly classify linearly separable data via a hyperplane. Given a set of patterns $\{\mathbf{x}^\mu\}_{1, \dots, M}$ the labels produced by the perceptron network are:

$$\hat{y}^\mu = \sigma(\mathbf{w} \cdot \mathbf{x}^\mu + b) \quad (1.18)$$

where usually in theoretical investigation the activation σ is assumed to be the $\text{sign}(\cdot)$ function, since it implements a deterministic decision rule that assigns label +1 if the linear combination is positive, and -1 otherwise. From this analysis, it is clear that the classification performed by the perceptron works as follows: the weights and bias define the classification hyperplane, the plane dividing patterns classified as positive from the ones classified as negative, and the activation chooses which class to assign to each side. The graphical representation of the perceptron network is reported in Figure 1.6.

In this thesis, the perceptron is mainly considered in the context of optimization: one seeks the optimal values of the weights and bias, minimizing the loss function. In this context, it can be considered to be part of the class of Constraint Satisfaction Problems (CSPs) [Tsang, 1993].

Many of the fundamental results in statistical physics of Machine Learning have been obtained using the perceptron network. However, while the perceptron is effective for linearly separable data, it fails to classify non-linearly separable cases (e.g., the XOR problem) [Engel and Van den Broeck, 2001, Gardner and Derrida, 1988]. To address this limitation, architectures inspired by the perceptron are augmented with hidden layers. One elementary example is the *Tree Committee Machine* [Seung et al., 1992].

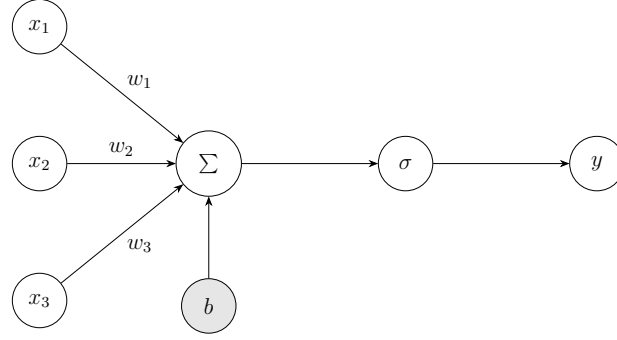


Figure 1.6: Single-layer perceptron: inputs are linearly combined with weights and a bias term, passed through an activation function σ , and produce output y .

- *Tree Committee Machine*: It is one of the simplest possible architectures with one hidden layer: in contrast to modern fully connected networks, the hidden neurons in a tree-committee machine are not fully connected, i.e., each second layer node is not linked with every input. The second layer nodes are connected to disjoint subsets of first layer ones, a property that greatly simplifies theoretical analysis. In Figure 1.7, for example, neuron c_1 connects only to the first two neurons of the first layer⁵. If the number of hidden units is fixed to K , the output of the layer can be expressed as:

$$\hat{y}^\mu = \sigma_C \left(\sqrt{\frac{1}{K}} \sum_{k=1}^K c_k \sigma_k \left(\sqrt{\frac{K}{N}} \mathbf{w}_k \cdot \mathbf{x}_k + b_k \right) + b_C \right) \quad (1.19)$$

where σ_k is the activation function of the k -th hidden unit, $\mathbf{w}_k$ is the vector of weights connecting the k -th hidden unit to the first layer ($\mathbf{w}_k \in \mathbb{R}^{N/K}$), $\mathbf{x}_k$ is the vector of patterns connected with the k -th hidden unit to the first layer ($\mathbf{x}_k \in \mathbb{R}^{N/K}$), b_k is the bias of the k -th second layer unit, and c_k is a constant weight associated with the k -th hidden unit. The output layer is a linear combination of the outputs of the hidden units, followed by an activation function σ_C . The Tree Committee Machine can be seen as a simple form of ensemble learning, where multiple decision trees (or committee members) are combined to produce a final output.

The transformation performed by the Tree Committee Machine is not linear, and it can be used to perform classification in more complex situations, where the Perceptron fails.

⁵In the graphical representation of the tree committee machine, for simplicity, the diagram omits biases and activation functions, which are nonetheless present and essential in the actual model. The biases need to be added to each layer summation, and the activations need to be applied to the result of the linear transformation.

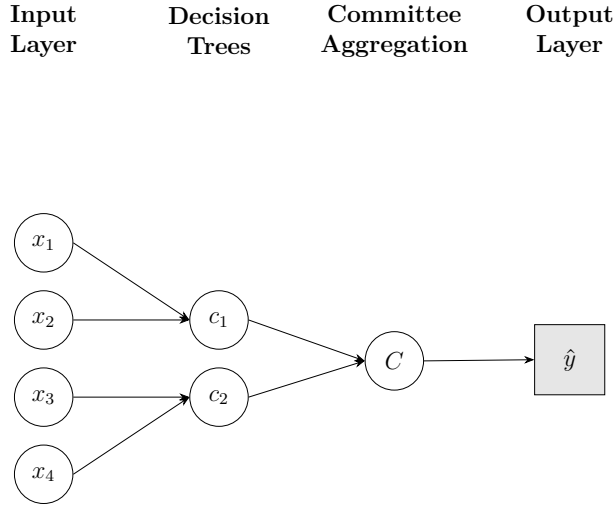


Figure 1.7: Tree-Committee Machine Architecture

- *Restricted Boltzmann Machines (RBM)*: In contrast concerning Perceptron and Tree Committee Machine, RBMs are one of the original architectures used for performing unsupervised learning and generative tasks. They are architectures made of two layers: a *visible* layer $\mathbf{v} \in \mathbb{R}^{N_v}$, made of N_v nodes called *visible nodes*, and a single *hidden* layer $\mathbf{h} \in \mathbb{R}^{N_h}$, made of N_h hidden units, called *hidden nodes*. In RBMs, no weights are present between nodes of the same type, but only between hidden and visible units. Biases can act on each type of node. A graphical representation of Restricted Boltzmann Machines is reported in Figure 1.8.

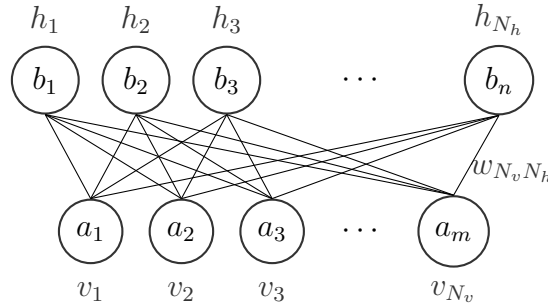


Figure 1.8: Graphical representation of a Restricted Boltzmann Machine, where $\mathbf{a}$ is the vector of biases acting on the visible layer, $\mathbf{b}$ is the vector of biases acting on the hidden layer, and W is the matrix of weights connecting visible and hidden units. The visible layer is composed of N_v visible units, while the hidden layer is composed of N_h hidden units.

RBM's architecture is simple; however, one can devise more complex architectures by stacking multiple RBMs together, forming the so-called Deep Boltzmann Machines. The investigation of the properties of the Deep Boltzmann Machines is out of the scope of this thesis.

Since the main corpus of the thesis will deal with supervised learning paradigms, and only the last chapter investigates RBM-related tasks, we postpone the structural presentation of this architecture to Chapter 5.

Loss Function

Once the architecture and dataset are defined, the loss function becomes the observable of interest in the statistical physics of machine learning. The main loss functions considered in this thesis are *Error Counting Loss*, *Squared Loss*, and the *Cross-Entropy loss*, together with the regularization $\mathcal{L}_2$ and $\mathcal{L}_1$, which have already been described in Section 1.1.3. Depending on the specific problem and on the dataset considered, the loss function takes on a particular form.

Loss function for random and structure labels

- *Random Labels*: For a completely random dataset $\mathcal{D} = \{\mathbf{x}^\mu, y^\mu\}_{\mu=1, \dots, M}$ and a chosen architecture represented by the function $f(\mathbf{w}, b, \mathbf{x}^\mu)$, the loss is given by

$$\mathcal{L}_{\text{storage}}(\mathbf{w}, b; \mathcal{D}) = \frac{1}{M} \sum_{\mu=1}^M \ell(y^\mu, f(\mathbf{w}, b, \mathbf{x}^\mu)). \quad (1.20)$$

- *Teacher-Student Problem*: When the data-generating architecture $f(\mathbf{w}^*, b^*, \mathbf{x}^\mu)$ is known in structure but not in its parameters, the loss can be reformulated as

$$\mathcal{L}_{\text{TS}}(\mathbf{w}^*, b^*, \mathbf{w}, b; \mathcal{D}) = \frac{1}{M} \sum_{\mu=1}^M \ell(f(\mathbf{w}^*, b^*, \mathbf{x}^\mu), f(\mathbf{w}, b, \mathbf{x}^\mu)). \quad (1.21)$$

The main difference between these two label-generation schemes emerges when computing the average training error. For random labels, the average is taken over a distribution of i.i.d. samples, which typically allows for an analytically tractable formulation. In contrast, in the teacher-student scenario, the average must take into account the distribution of the teacher's weights, introducing additional complexity and making the calculation generally non-trivial.

Learning Algorithm

As discussed earlier, various learning algorithms have been proposed. In this thesis, we focus on methods based on *Empirical Risk Minimization* (ERM) [Vapnik, 1999], namely

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{w}, b} \mathcal{L}(\mathbf{w}, b; \mathcal{D}) + \lambda \|\mathbf{w}\|^2, \quad (1.22)$$

where $\mathcal{L}$ is the training loss on dataset $\mathcal{D}$.

To embed ERM in a statistical-physics framework, we introduce the associated *Gibbs distribution* over parameters:

$$P_\beta(\mathbf{w}, b \mid \mathcal{D}) = \frac{1}{Z(\beta)} \exp\left[-\beta(\mathcal{L}(\mathbf{w}, b; \mathcal{D}) + \lambda \|\mathbf{w}\|^2)\right]. \quad (1.23)$$

Here β (inverse temperature) tunes how sharply the measure concentrates on low-loss configurations. For any finite β , P_β has support on the *entire* parameter space, assigning higher probability to configurations of smaller “energy” $\mathcal{L} + \lambda \|\mathbf{w}\|^2$. In the limit $\beta \rightarrow \infty$, the Gibbs measure *concentrates* on the set of *global* minimizers of $\mathcal{L} + \lambda \|\mathbf{w}\|^2$ (in proportion to their volume), though one must not assume a priori that this concentration is strictly uniform

unless the minima are symmetry-equivalent.

The factor $\exp(-\beta\lambda\|\mathbf{w}\|^2)$ can be viewed as a regularization-induced “prior” on the weights, while the biases b are typically endowed with a flat (constant) prior if unpenalized.

Note on the “static” nature of the approach: This statistical-physics treatment is *static*: for each fixed β , one studies the thermodynamic properties of the Gibbs ensemble (e.g. overlaps, typical training and generalization errors), but *does not* model the time-dependent trajectory of any specific learning dynamics (e.g. gradient descent or stochastic Langevin updates). Although the Gibbs distribution coincides with the stationary state of an overdamped Langevin process at temperature $1/\beta$, the formalism itself does not describe the algorithmic path followed during training.

1.3.3 Notable Results and Open Problems

In Section 1.2, we introduced the fundamental paradigms and challenges of Machine Learning. Here we summarize some of the most significant results obtained via statistical-physics methods, and highlight several important open problems.

Feature Learning and the NTK Regime A central question is when neural networks learn *meaningful features* of the data, as opposed to behaving in the “lazy” limit described by the *Neural Tangent Kernel* (NTK) [Jacot et al., 2018]. This problem has been studied in various ways, particularly in the context of the *teacher–student framework* (in the case of two layer Neural Network, with ReLU activation and Gaussian teacher noise) [Engel and Van den Broeck, 2001, Goldt et al., 2020], where one measures feature learning by the *overlap* $\rho = \frac{\mathbf{w}^* \cdot \hat{\mathbf{w}}}{\|\mathbf{w}^*\| \|\hat{\mathbf{w}}\|}$ between the teacher weights $\mathbf{w}^*$ and the student solution $\hat{\mathbf{w}}$. Statistical-physics analyses show a sharp transition in ρ as the sample ratio $\alpha = M/N$ crosses a critical value. The NTK regime (limiting regime for $N \rightarrow \infty$) corresponds to $\rho \approx 0$, i.e. no feature learning, whereas beyond the transition one enters a *rich feature-learning* phase with $\rho > 0$. These results hold in the infinite-dimensional limit, but for finite N the architectures show an intermediate behavior.

Loss Landscapes and Spin-Glass Techniques The geometry of nonconvex loss landscapes has been studied using methods borrowed from the theory of spin glasses and random fields. *Replica method* gives the exponential *complexity* of critical points at a given loss level [Mézard et al., 1987]. The *Franz–Parisi potential*, defined via the overlap between two replicas, quantifies the ruggedness of the landscape [Silvio Franz and Giorgio Parisi, 1995]. Random Matrix Theory applied to the Hessian predicts the density of its eigenvalues around minima [Sagun et al., 2017, Liao and Mahoney, 2021]. Empirical and theoretical studies suggest that *flat minima*, those with few large Hessian eigenvalues, tend to generalize better, although this connection remains debated [Keskar et al., 2017, Dinh et al., 2017]. In [Dinh et al., 2017], an explicit construction of sharp minima with good generalization is presented. In simplified nonconvex models, one finds a multitude of narrow, suboptimal minima coexisting with a smaller set of wide, “high-entropy” minima that achieve near-optimal generalization [Baldassi et al., 2016b, 2015].

Capacity, Overfitting and Double Descent The *storage capacity* of a perceptron, namely the critical ratio $\alpha_c = M/N$ below which random labels are linearly separable

with high probability, was computed by Gardner and Derrida via the replica method in the $N \rightarrow \infty$ limit [Gardner and Derrida, 1988]. Subsequent analyses for other shallow architectures (e.g. tree committee machines) reveal distinct critical capacities $\alpha_c^{(\text{model})}$, each obtained through a dedicated replica calculation. In modern overparameterized models, as the number of parameters surpasses the number of samples, one observes the *double descent* phenomenon: the test error first peaks near the interpolation threshold, then decreases again as model size grows [Belkin et al., 2019, Mei and Montanari, 2021].

Open Problems Despite these advances, several major challenges remain:

- *Realistic data distributions.* Most analyses assume i.i.d. Gaussian patterns. Extending results to structured, non-Gaussian inputs (e.g. images, graphs) remains a major challenge.
- *Deep architectures.* Replica and cavity methods are well developed for shallow models. A comprehensive theory for truly deep networks, beyond infinite-width or NTK limits, is still lacking.
- *Modern generative models.* While RBMs admit a mature statistical-physics treatment, state-of-the-art generative approaches (diffusion models, autoregressive transformers) resist a full analytic description. In particular, understanding the boundary between memorization and genuine data generation is an active frontier [Biroli and Mézard, 2023, Achilli et al., 2025, Biroli et al., 2024, Achilli et al., 2024, Ventura et al., 2024].

1.4 Statistical Physics of Machine Learning tools

In this section, we introduce the principal analytical tools from the statistical physics of machine learning that are employed throughout this thesis. To provide a concrete foundation, we begin with the classic problem of computing the storage capacity of the spherical perceptron network (assuming absence of bias, $b = 0$), which serves as a natural excuse for introducing the replica method, a cornerstone technique for analyzing high-dimensional inference and optimization problems. Building on this, we turn to the teacher-student scenario for the perceptron network (still in the absence of biases), which allows for a systematic exploration of message passing algorithms. In particular, we discuss three key schemes: *Belief Propagation* (BP), *Relaxed Belief Propagation* (rBP), and *Approximate Message Passing* (AMP). These algorithms provide both theoretical insight and practical tools for performing inference and optimization in factor graph models.

1.4.1 Storage problem replica computation

To illustrate the replica method and how it is performed, one can consider the task of computing the capacity for the storage problem in the simplest architecture: the perceptron with normally distributed patterns [Engel and Van den Broeck, 2001], and weights that are spherically distributed. It is important to stress that the results of this replica computation are used in the next chapters, and with slight modifications are addressed as *reference system replica computation* in Chapter 2, and as the *equilibrium* replica computation, in Chapter 4.

In the spherical perceptron, one assumes that the weights are lying on an N -dimensional sphere, thus their prior assumes the following form

$$P(\mathbf{w}) = \delta\left(\|\mathbf{w}\|_2^2 - N\right), \quad (1.24)$$

Before entering in the details of the replica computation, the reader can be wondering what is the reason for which one should assume that the weights lie on a sphere. Historically, the spherical perceptron was one of the statistical physics approaches to constrain the weight norms (modern approaches smooth this assumption by adding regularization terms to the loss function). Although this is a simplification, it is sufficient for our theoretical analysis.

The loss function used in the original paper, [Gardner and Derrida, 1988], for this case is the simplest possible: the *Error Counting Loss*, which simply counts the number of misclassifications made by the perceptron:

$$\mathcal{L}(\mathbf{w}; \mathcal{D}) = \sum_{\mu=1}^M \Theta\left(-y^\mu \frac{\mathbf{w} \cdot \mathbf{x}^\mu}{\sqrt{N}}\right), \quad (1.25)$$

where the product $s^\mu = y^\mu \frac{\mathbf{w} \cdot \mathbf{x}^\mu}{\sqrt{N}}$ is known as the *stability* of the example. The Gibbs measure for a fixed β , and its infinite limit, are defined as

$$P_\beta(\mathbf{w}) = \frac{P(\mathbf{w})}{Z_\beta} e^{-\beta \sum_{\mu=1}^M \Theta\left(-y^\mu \frac{\mathbf{w} \cdot \mathbf{x}^\mu}{\sqrt{N}}\right)} \quad P_\infty(\mathbf{w}) = \frac{P(\mathbf{w})}{Z_\infty} \prod_{\mu=1}^M \Theta\left(y^\mu \frac{\mathbf{w} \cdot \mathbf{x}^\mu}{\sqrt{N}}\right), \quad (1.26)$$

where we have removed the conditioning on the dataset $\mathcal{D}$ for the sake of clarity, in the Gibbs distribution; and we define the normalization constants (partition functions) as

$$Z_\beta = \int dP(\mathbf{w}) e^{-\beta \sum_{\mu=1}^M \Theta\left(-y^\mu \frac{\mathbf{w} \cdot \mathbf{x}^\mu}{\sqrt{N}}\right)}, \quad Z_\infty = \int dP(\mathbf{w}) \prod_{\mu=1}^M \Theta\left(y^\mu \frac{\mathbf{w} \cdot \mathbf{x}^\mu}{\sqrt{N}}\right). \quad (1.27)$$

The finite inverse temperature problem investigates the number of patterns correctly classified when the learning is not finalized, while in the zero-temperature limit ($\beta \rightarrow \infty$), the Gibbs measure becomes uniform over all weight vectors on the sphere that correctly classify the entire training dataset. These zero-error vectors are referred to as solutions to the storage perceptron problem.

Since statistical physics aims to study typical macroscopic properties independently of the specific realization of the dataset, while fixing its distribution. The partition function, which depends on the specific examples, must be averaged over all possible datasets. Denoting this average by

$$\overline{Z} = \mathbb{E}_{\mathcal{D}}[Z], \quad (1.28)$$

one can in principle compute this *annealed average*. The problem with this computation consists in the fact that the partition function is an element scaling exponentially with the size of the system. As already presented in Section 1.3, the partition function of a statistical physics system is computed by summing over all the possible configurations the system can assume, which are exponential. However, the annealed computation doesn't provide the correct capacity; one needs that the observable we are considering to be *self-averaging*, if we consider an observable O_N , depending on the size of the system N , it is said to be self-averaging if:

$$\frac{\text{Var}_{\mathcal{D}}(O_N)}{\mathbb{E}_{\mathcal{D}}(O_N)^2} \rightarrow 0 \quad (1.29)$$

where typical instances of the problem behave as the average over the disorder. It is clear by looking at Equation (1.28) that the partition function is not self-averaging, since its volume is exponential in N . A more appropriate quantity is the logarithm of the partition function, known as the (zero-temperature) *free entropy*:

$$\Phi = \mathbb{E}_{\mathcal{D}} \log Z. \quad (1.30)$$

Averaging the logarithm (a quenched average) is expected to yield the correct result in the high-dimensional limit. The computation of this quenched average is performed through the famous *Replica Method* [Mézard et al., 1987].

The replica method is the central technique in statistical mechanics for computing the *free entropy* (or the negative free energy). In this section, the replica computation for the solution volume of the storage perceptron problem (known as the *Gardner Volume* [Gardner and Derrida, 1988]) is presented, in the infinite-dimensional limit $N \rightarrow \infty$ and in the proportional regime $M = \alpha N$.⁶

The Free entropy in the infinite-dimensional regime is defined as:

$$\phi = \lim_{N \rightarrow \infty} \frac{1}{N} \mathbb{E}_{\mathcal{D}} \log Z. \quad (1.31)$$

Where the partition function has been scaled by the size of the system. This computation requires the definition of mathematical identity, addressed as the *replica trick*:

$$\log Z = \lim_{n \rightarrow 0} \frac{Z^n - 1}{n}. \quad (1.32)$$

Thus, the free entropy can be written as

$$\phi = \lim_{N \rightarrow \infty} \lim_{n \rightarrow 0} \frac{\mathbb{E}_{\mathcal{D}} Z^n - 1}{Nn}. \quad (1.33)$$

Although the partition function's complex form makes the computation difficult for general n , assuming $n \in \mathbb{N}$ allows us to view the system as consisting of n interacting⁷ copies, or "replicas." In other words, one can write the n^{th} -power of the partition function as:

$$Z^n = \int \prod_{a=1}^n dP(\mathbf{w}^a) \prod_{a=1}^n \prod_{\mu=1}^M \Theta\left(\frac{\mathbf{w}^a \cdot \mathbf{x}^\mu}{\sqrt{N}}\right), \quad (1.34)$$

where the index $a \in \{1, \dots, n\}$ labels each replica, and the labels y^μ , without loss of gener-

⁶From this point onward, the index referring to the inverse temperature in the partition function is omitted for simplicity. We work in the limit $\beta \rightarrow \infty$.

⁷At the moment, it is not clear why these replicas are interacting among themselves, but after performing the average over the disorder it will.

ality, have been fixed to 1. By introducing the variable

$$\lambda_\mu^a = \frac{\mathbf{w}^a \cdot \mathbf{x}^\mu}{\sqrt{N}}, \quad (1.35)$$

via the Dirac delta representation,

$$\begin{aligned} 1 &= \int \prod_{\mu,a} d\lambda_{\mu,a} \delta\left(\lambda_\mu^a - \frac{1}{\sqrt{N}} \sum_i x_i^\mu w_i^a\right) \\ &= \int \prod_{\mu,a} d\lambda_\mu^a d\hat{\lambda}_\mu^a \frac{1}{2\pi} \exp\left\{i \sum_{\mu,a} \lambda_\mu^a \hat{\lambda}_\mu^a - i \sum_{\mu,a} \hat{\lambda}_\mu^a \frac{1}{\sqrt{N}} \sum_i x_i^\mu w_i^a\right\}, \end{aligned}$$

the expected value $\mathbb{E}_X Z^n$ becomes

$$\mathbb{E}_X Z^n = \mathbb{E}_X \int \prod_a dP(\mathbf{w}^a) \prod_{\mu,a} \frac{d\lambda_\mu^a d\hat{\lambda}_\mu^a}{2\pi} \exp\left\{i \sum_{\mu,a} \lambda_\mu^a \hat{\lambda}_\mu^a - i \sum_{\mu,a} \hat{\lambda}_\mu^a \frac{1}{\sqrt{N}} \sum_i x_i^\mu w_i^a\right\} \Theta(\lambda_\mu^a).$$

Averaging over the dataset yields

$$\mathbb{E}_X Z^n = \int \prod_a dP(\mathbf{w}^a) \prod_{\mu,a} \frac{d\lambda_\mu^a d\hat{\lambda}_\mu^a}{2\pi} \exp\left\{i \sum_{\mu,a} \lambda_\mu^a \hat{\lambda}_\mu^a - \frac{1}{2N} \sum_\mu \sum_{a,b} \hat{\lambda}_\mu^a \hat{\lambda}_\mu^b \sum_i w_i^a w_i^b\right\} \Theta(\lambda_\mu^a).$$

where it is now clear how different replicas of the system are interacting among themselves. Since the data are i.i.d., this expression can be recast as

$$\mathbb{E}_X Z^n = \int \prod_a dP(\mathbf{w}^a) \left(\prod_a \frac{d\lambda^a d\hat{\lambda}^a}{2\pi} \exp\left\{i \sum_a \lambda^a \hat{\lambda}^a - \frac{1}{2N} \sum_{a,b} \hat{\lambda}^a \hat{\lambda}^b \sum_i w_i^a w_i^b\right\} \Theta(\lambda^a) \right)^{\alpha N}.$$

At this point, it is important to quantify the interaction between the different replicas. One can address the scalar products between two copies of weights, from which one can understand how similar two replicas are. One can then define the *Overlap Parameter* [Engel and Van den Broeck, 2001]

$$Q_{ab} = \frac{\mathbf{w}^a \cdot \mathbf{w}^b}{N}, \quad (1.36)$$

and insert this definition into the computation via an appropriate Dirac delta in the definition of the free entropy. Ultimately, the expression for the averaged partition function

becomes

$$\mathbb{E}_{\mathcal{D}} Z^n = \int \prod_{a,b} dQ_{ab} d\hat{Q}_{ab} \exp\left\{N \phi(Q_{ab}, \hat{Q}_{ab})\right\}, \quad (1.37)$$

$$\phi(Q_{ab}, \hat{Q}_{ab}) = G_I + G_S + \alpha G_E, \quad (1.38)$$

$$G_I = i \sum_{a,b} Q_{ab} \hat{Q}_{ab}, \quad (1.39)$$

$$G_S = \frac{1}{N} \log \left\{ \prod_a \int dP(\mathbf{w}^a) \exp \left[-i \sum_{a,b} \hat{Q}_{ab} \sum_i w_i^a w_i^b \right] \right\}, \quad (1.40)$$

$$G_E = \log \left\{ \prod_a \int \frac{d\lambda^a d\hat{\lambda}^a}{2\pi} \exp \left[i\lambda^a \hat{\lambda}^a - \frac{1}{2} \sum_{a,b} \hat{\lambda}^a \hat{\lambda}^b Q_{ab} \right] \prod_a \Theta(\lambda^a) \right\}. \quad (1.41)$$

where the three terms are called respectively Interaction term G_I , entropic term G_S and energetic term G_E . The interaction term represents the coupling among different replicas, and it comes from the average performed over the disorder. The entropic term derive from the computation of the assignments of the weights that are compatible with a specific overlap, thus it is higher for overlap values that are more represented. The energetic term couples the replicas as a function of the loss function. Thus the system of replicas is characterized by the classical tradeoff of energy and entropy of statistical physics systems. From the point of view of machine learning, this is related to the tradeoff between optimization of the loss function and generalization of the solution.

To compute the free entropy, we evaluate the above integral in the $N \rightarrow \infty$ limit using the Laplace (saddle point) method, in the $N \rightarrow \infty$ limit.

In general, this computation remains intractable unless further assumptions, called *Ansatz*, are introduced regarding the structure of the overlap matrix Q_{ab} , specifying how replicas interact among themselves. Common choices include the Replica Symmetric (RS) Ansatz, 1-step Replica Symmetry Breaking (1RSB), and Full Replica Symmetry Breaking (full-RSB). In the simplest case, we adopt the *Replica Symmetric Ansatz* [Gardner and Derrida, 1988, Mézard et al., 1987]:

$$Q_{ab} = \begin{cases} 1 & \text{if } a = b, \\ q & \text{if } a \neq b, \end{cases} \quad \hat{Q}_{ab} = \begin{cases} \hat{Q} & \text{if } a = b, \\ \hat{q} & \text{if } a \neq b. \end{cases} \quad (1.42)$$

The RS Ansatz assumes uniform interaction among different replicas (characterized by q), i.e., each replica weight vector has uniform scalar product with the others, given by q . While it has been shown [Gardner and Derrida, 1988] that the RS ansatz is valid for the storage problem in the spherical perceptron, there are cases (discussed later in the Stochastic Localization and Phase Selection chapter) where replica symmetry breaking becomes necessary, since replicas are not interacting uniformly.

Subsequently, the RS ansatz is substituted into the expressions for the interaction, entropic, and energetic terms.

Interaction Term Substituting the RS ansatz yields:

$$G_I = \frac{1}{2}n(n-1)q\hat{q} + \frac{1}{2}n\hat{Q}.$$

Taking the limit $n \rightarrow 0$ and dividing by n results in:

$$\mathcal{G}_I = \lim_{n \rightarrow 0} \frac{G_I}{n} = -\frac{1}{2}q\hat{q} + \frac{1}{2}\hat{Q}. \quad (1.43)$$

Entropic Term Inserting the RS ansatz into Equation (1.40) and noting that the perceptron weights are i.i.d. yields:

$$G_S = \log \left\{ \int \prod_a dw^a \exp \left[-\frac{1}{2}\hat{q} \left(\sum_a w^a \right)^2 - \left(\frac{\hat{Q} - \hat{q}}{2} \right) \sum_a (w^a)^2 \right] \right\}.$$

To linearize the quadratic term in the sum of weights, we apply the Hubbard-Stratonovich transformation:

$$e^{\frac{1}{2}A^2} = \int Dx e^{xA}, \quad \text{with} \quad Dx = \frac{dx}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}.$$

This allows us to rewrite the entropic term as

$$G_S = \log \left\{ \int Dx \prod_a \int dw^a e^{i\sqrt{\hat{q}}xw^a - \left(\frac{\hat{Q} - \hat{q}}{2} \right) (w^a)^2} \right\}.$$

Since all replicas are identical, we obtain:

$$G_S = \log \left\{ \int Dx \left[\int dw e^{i\sqrt{\hat{q}}xw - \left(\frac{\hat{Q} - \hat{q}}{2} \right) w^2} \right]^n \right\}.$$

Taking the limit $n \rightarrow 0$ gives the final expression for the entropic term:

$$\mathcal{G}_S = \lim_{n \rightarrow 0} \frac{G_S}{n} = \int Dx \log \left\{ \int dw e^{i\sqrt{\hat{q}}xw - \left(\frac{\hat{Q} - \hat{q}}{2} \right) w^2} \right\}. \quad (1.44)$$

Energetic Term Following similar steps and substituting the RS ansatz and applying the Hubbard-Stratonovich transformation, we obtain:

$$G_E = \log \left\{ \int Dx \left[\int \frac{d\lambda d\hat{\lambda}}{2\pi} \exp \left(i\lambda\hat{\lambda} - ix\sqrt{q}\hat{\lambda} - \frac{1}{2}(1-q)(\hat{\lambda})^2 \right) \Theta(\lambda) \right]^n \right\}.$$

Taking the limit $n \rightarrow 0$ results in:

$$\mathcal{G}_E = \lim_{n \rightarrow 0} \frac{G_E}{n} = \int Dx \log \left[\int \frac{d\lambda d\hat{\lambda}}{2\pi} \exp \left(i\lambda\hat{\lambda} - ix\sqrt{q}\hat{\lambda} - \frac{1}{2}(1-q)(\hat{\lambda})^2 \right) \Theta(\lambda) \right].$$

After performing the integrals, we can express the energetic term in terms of the auxiliary function

$$H(x) = \int_x^\infty Dy = \frac{1}{2} \operatorname{erfc}\left(\frac{x}{\sqrt{2}}\right).$$

Thus, the final expression for the energetic term is:

$$\mathcal{G}_E = \int Dx \log H\left(-\frac{\sqrt{q}}{\sqrt{1-q}} x\right). \quad (1.45)$$

Saddle Point Equations

Finally, the free entropy is given by

$$\phi = \lim_{N \rightarrow \infty} \frac{1}{N} \mathbb{E}_X [\log Z] = \lim_{n \rightarrow 0} \lim_{N \rightarrow \infty} \frac{1}{nN} (\mathbb{E}_X Z^n - 1).$$

Interchanging the limits (with the usual caveats regarding mathematical rigor) yields:

$$\phi = \lim_{n \rightarrow 0} \phi(q^*, \hat{q}^*, \hat{Q}^*),$$

where q^* , $\hat{q}^*$, and $\hat{Q}^*$ are the saddle point values determined by the following equations:

$$\frac{\partial \phi}{\partial q} = 0, \quad \frac{\partial \phi}{\partial \hat{q}} = 0, \quad \frac{\partial \phi}{\partial \hat{Q}} = 0.$$

These saddle point equations must be solved to obtain the final expression for the free entropy. The saddle point method is a method used for optimizing functions of multiple variables. It consists of deriving the function to optimize, considering all the parameters, and fixing the derivative values to 0. In general, in Statistical Physics systems the form of such derivatives contains not analytically solvable integrals, thus the optimization is performed by repeating the optimization of each single derivative by itself, and plugging the temporal optimizer value in the other derivative expressions.

On the perceptron capacity and storage problem phase transitions

In this section, we briefly describe the behavior observed when solving the saddle point equations for different values of α . For low values of α , the perceptron can easily classify the examples; however, as α increases, the task becomes progressively more challenging. It can be shown that increasing the ratio of examples to dimension also causes the overlap q to increase, indicating that different replicas of the system tend to converge toward similar weight configurations. Eventually, the overlap reaches the value of one at $\alpha = \alpha_C = 2.0$, marking the threshold beyond which no solutions to the storage problem exist [Engel and Van den Broeck, 2001]. This phenomenon is known as the *SAT-UNSAT transition* and was first discovered by Cover in [Cover, 1965]. The SAT-UNSAT transition represents the threshold beyond which no solutions to the storage optimization problem can be found, signaling a change in the solution space structure. By introducing a negative margin κ , presented in the next Section, different structural changes of the solution space can be observed.

It is clear that different models exhibit distinct transitions; for instance, variations arise when considering the binary perceptron storage problem. When we encounter a phase transition, it could be the case that the considered ansatz is not valid, and a replica symmetry breaking (RSB) is required [Gardner and Derrida, 1988, Engel and Van den Broeck, 2001, Nishimori, 2001]. The proper definition of a general phase transition is out of the scope of this thesis. Some specific problems considered in this thesis undergo phase transitions, they will be described whenever encountered in the discussion.

Margin

In machine learning, one often introduces a margin parameter κ to allow the model to admit a limited number of classification errors during training. In the perceptron, using the zero-one (error-counting) loss, this margin enters the objective as

$$\mathcal{L}(\mathbf{w}; \mathcal{D}) = \sum_{\mu=1}^M \Theta\left(\kappa - y^\mu \frac{\mathbf{w} \cdot \mathbf{x}^\mu}{\sqrt{N}}\right). \quad (1.46)$$

At first glance, adding a margin may seem to offer no practical advantage in terms of empirical performance. However, from a theoretical perspective, it is indispensable: incorporating κ not only induces new phase transitions in the perceptron’s behavior but also renders the underlying optimization problem non-convex. Although the margin’s impact is chiefly conceptual rather than empirical, its inclusion provides a valuable framework for investigating the fundamental paradigms introduced in Section 1.2 of this Chapter.

1.4.2 Message Passing Algorithms

The replica method provides a theoretical framework to compute macroscopic observables in the thermodynamic limit $N \rightarrow \infty$ and in the proportional regime $\alpha = M/N$. However, this approach presents two main limitations: first, using replica theory one can analyze only problems in the $N \rightarrow \infty$ limit; second, it enables the investigation of just the averaged (typical) properties of the system. In particular, it cannot be used to study single-instance realizations of a statistical-physics problem.

In the context of statistical mechanics, a special class of algorithms, *Message Passing Algorithms*, has been devised to infer properties of individual instances of optimization problems. These algorithms are designed for inference on models with a factor graph structure. In this section, after introducing the notion of a *factor graph*, we briefly present several message-passing schemes. we start from *Belief Propagation (BP)* (the common ancestor of most message-passing algorithms) [Pearl, 1988, Mézard et al., 1987], then discuss *Relaxed Belief Propagation (rBP)* [Mézard et al., 2002], and conclude with *Approximate Message Passing (AMP)*, a message-passing framework tailored to high-dimensional, dense problems [Donoho et al., 2009, Bayati and Montanari, 2011].

In Chapters 2 and 3, we present problems related to the teacher-student setup for the perceptron architecture. Their instances are solved using AMP; for this reason, message passing algorithms are presented in the context of teacher-student problems, for the perceptron network.

Consider a set of teacher-student instances for a perceptron: the “Teacher” generates labels y^μ for patterns $\mathbf{x}^\mu$ (Normally distributed), $\mu \in \{1, \dots, M\}$, using a weight vector $\mathbf{w}^*$.

Given this instance, the “Student” is trained to classify the examples correctly and therefore to reconstruct $\mathbf{w}^*$. To evaluate the student’s performance, one computes the overlap between the teacher’s weight vector $\mathbf{w}^*$ and the student’s weight vector $\mathbf{w}$. By inferring the posterior structure of the student’s solutions, it is possible to recover the overlaps of interest. For simplicity of treatment, we assume the weights prior to be factorized $P(\mathbf{w}) = \prod_{i=1}^N P(w_i)$, where $P(w_i)$ is left unspecified. The pseudo-code of AMP for the spherical prior is presented in Chapter 2.

Factor Graphs

Message passing algorithms are usually defined on a specific graphical structure called *Factor Graphs*. A *Factor Graph* is a bipartite graphical representation that expresses the factorization of a global function over multiple variables. In the cases of interest for the thesis, we are interested in estimating the posterior probability of a vector of perceptron weights $\mathbf{w}$, so as to be able to infer the teacher’s weight vector $\mathbf{w}^*$ ⁸:

$$P(\mathbf{w}|\{\mathbf{x}^\mu, y^\mu\}_{\mu=1,\dots,M}) \propto P(\mathbf{w}) \prod_{\mu=1}^M P_\mu(y^\mu|\mathbf{x}^\mu, \mathbf{w}). \quad (1.47)$$

where $P(\mathbf{w}) = \prod_{i=1}^N P(w_i)$ is the factorized prior of the weights of the perceptron and $P_\mu(y^\mu|\mathbf{x}^\mu, \mathbf{w}) = \Theta\left(y^\mu \sum_i \frac{x_i^\mu w_i}{\sqrt{N}}\right)$ is the likelihood factor (or its smoothed version, for enabling the integral definition of the channels). The corresponding factor graph contains one *variable node* for each variable w_i and one *factor node* for each local function P_μ . An undirected edge connects variable node w_i to factor node P_μ if and only if w_i appears in the arguments of P_μ . Variable nodes are also connected with prior factor nodes $P(w_i)$.

The resulting factor graph is bipartite, with variable nodes corresponding to the weights and factor nodes representing the training constraints. Each factor connects several variables (the factor nodes of the perceptron network connect all variables due to the fully connected nature of the network). Optimizing the weights thus becomes an inference problem over this graph: we seek a configuration of variables w_i maximizing the posterior. Message-passing algorithms such as belief propagation, or their approximations like AMP, can be employed to solve this inference problem.

A graphical representation of the factor graph for the perceptron model is shown in Figure 1.9. In this figure, each variable node represents a weight w_i to be inferred, while the factor node encodes the aggregation (activation) function of the perceptron. The edges between variable nodes and factor nodes represent the dependencies between weights and the activation function.

Belief Propagation

Let’s first introduce the simplest message-passing algorithm: *Belief Propagation (BP)*. BP is an iterative algorithm for performing inference on factor graph models by passing a form of information on the variable nodes, under the form of *messages* between them.

⁸For simplifying the Bayesian formulation of the posterior distribution of student weights, we have substituted the notation $P(\mathbf{w}|\mathcal{D})$ with $P(\mathbf{w}|\{\mathbf{x}^\mu, y^\mu\}_{\mu=1,\dots,M})$

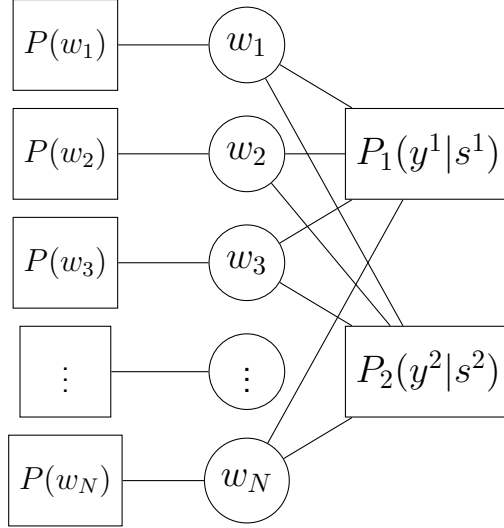


Figure 1.9: Factor graph representation of the perceptron model. Variable nodes represent the weights to be inferred, while the factor node encodes the M factors associated with the likelihood distribution and the N factors associated with the prior distribution.

Given a specific teacher-student instance, the posterior distribution for the teacher-student perceptron problem is given by [Zdeborová and Krzakala, 2016]:

$$P(\mathbf{w} \mid \{y^\mu, \mathbf{x}^\mu\}_\mu) = \frac{1}{Z} \prod_{\mu=1}^M P_\mu(y^\mu \mid s^\mu) \prod_{i=1}^N P(w_i),$$

where the quantity

$$s^\mu = \sum_{i=1}^N \frac{x_i^\mu w_i}{\sqrt{N}}$$

represents the stability of example μ and assuming a factorized prior. The scaling factor $1/\sqrt{N}$ is crucial for ensuring the correctness of the message passing scheme. In the case of the perceptron, y^μ denotes the labels generated by a teacher network with weights $\mathbf{w}^*$, and $\mathbf{x}^\mu$ are the corresponding examples.

Once the factor graph is constructed for the perceptron model, as in Figure 1.9, the BP algorithm can be applied to compute the marginal distributions of the weights w_i given the dataset $\{y^\mu, \mathbf{x}^\mu\}_{\mu=1, \dots, M}$. The goal is to compute the marginal posterior distribution for each weight w_i . Information is exchanged between these nodes via two types of messages: messages from variable nodes to factor nodes, denoted by $m_{i \rightarrow \mu}(w_i)$, and messages from factor nodes to variable nodes, denoted by $\hat{m}_{\mu \rightarrow i}(w_i)$. By iteratively exchanging these messages, the algorithm converges to a set of *beliefs* (marginal distributions) that approximate the true posterior [Mézard and Montanari, 2002].

The BP update equations for the messages are

$$m_{i \rightarrow \mu}(w_i) = \frac{1}{z_{i \rightarrow \mu}} P(w_i) \prod_{\gamma \neq \mu} \hat{m}_{\gamma \rightarrow i}(w_i), \quad (1.48)$$

$$\hat{m}_{\mu \rightarrow i}(w_i) = \frac{1}{\hat{z}_{\mu \rightarrow i}} \int \prod_{j \neq i} dw_j m_{j \rightarrow \mu}(w_j) P_{\text{out}} \left(y^\mu \middle| \frac{1}{\sqrt{N}} \sum_j w_j x_j^\mu \right), \quad (1.49)$$

where $z_{i \rightarrow \mu}$ and $\hat{z}_{\mu \rightarrow i}$ are normalization factors.

The convergence of the BP algorithm is guaranteed under certain conditions, such as when the factor graph is a tree or has a locally tree-like structure. In practice, BP can be applied to graphs with loops, but convergence is not guaranteed and may require additional techniques to ensure stability. A key limitation of standard BP is that it becomes intractable on densely connected graphs or graphs with many loops (which is often the case for models like the perceptron), since BP assumes a locally tree-like structure. In loopy graphs, the messages can become correlated, leading to convergence issues or excessive computational cost. For this reason, BP is often used as a heuristic method in practice, and its performance can be improved by using approximations or modifications such as Relaxed Belief Propagation (r-BP) or Approximate Message Passing (AMP).

Relaxed Belief Propagation

To overcome the limitations of standard BP on dense graphs, one can use a simplified version of the message-passing scheme known as *Relaxed Belief Propagation (r-BP)*. This approach leverages the Central Limit Theorem (CLT) to approximate the messages by projecting them onto a few of their moments, typically the mean and variance. The idea is to simplify the message updates while retaining the essential structure of the problem. In the following derivation of the r-BP algorithm, we follow the notation and conventions introduced in [Zdeborová and Krzakala, 2016], removing the time index (which is then reinserted in the pseudocode).

In the context of the perceptron model, the r-BP algorithm approximates the messages as Gaussian distributions. The messages from factor nodes to variable nodes need to distinguish between the contributions from the variable node itself and the contributions from other variable nodes. In particular, one can first rewrite the stability of example μ as

$$s^\mu = \frac{x_i^\mu w_i}{\sqrt{N}} + \sum_{j \neq i} \frac{x_j^\mu w_j}{\sqrt{N}},$$

where the second term can be proven to be approximately Gaussian due to the CLT, with mean and variance given by

$$\omega_{\mu \rightarrow i} = \frac{1}{\sqrt{N}} \sum_{j \neq i} x_j^\mu a_{j \rightarrow \mu}, \quad V_{\mu \rightarrow i} = \sum_{j \neq i} \left(\frac{x_j^\mu}{\sqrt{N}} \right)^2 v_{j \rightarrow \mu},$$

where we have denoted

$$a_{j \rightarrow \mu} = \int dw_j w_j m_{j \rightarrow \mu}(w_j), \quad v_{j \rightarrow \mu} = \int dw_j w_j^2 m_{j \rightarrow \mu}(w_j) - a_{j \rightarrow \mu}^2.$$

The messages from factor nodes to variable nodes can then be approximated as Gaussian distributions:

$$\hat{m}_{\mu \rightarrow i}(w_i) = \frac{1}{Z_{\mu \rightarrow i}} \int ds^\mu P_{\text{out}}(y^\mu | s^\mu) \exp \left[-\frac{(s^\mu - \omega_{\mu \rightarrow i} - \frac{x_i^\mu w_i}{\sqrt{N}})^2}{2V_{\mu \rightarrow i}} \right]. \quad (1.50)$$

We have replaced a complicated multi-dimensional integral involving full probability distributions by a single scalar one that involves only the first two moments. Since the interactions scale as $O(1/\sqrt{N})$, one can expand the exponent to first order in $x_i^\mu w_i/\sqrt{N}$ (retaining terms up to $O(1/N)$). After some algebra, the message from factor to variable nodes takes the Gaussian form

$$\hat{m}_{\mu \rightarrow i}(w_i) = \sqrt{\frac{A_{\mu \rightarrow i}}{2\pi N}} \exp \left[-\frac{A_{\mu \rightarrow i} w_i^2}{2N} + B_{\mu \rightarrow i} \frac{w_i}{\sqrt{N}} - \frac{B_{\mu \rightarrow i}^2}{2A_{\mu \rightarrow i}} \right], \quad (1.51)$$

where

$$B_{\mu \rightarrow i} = g_{\text{out}}(\omega_{\mu \rightarrow i}, y^\mu, V_{\mu \rightarrow i}) \left(\frac{x_i^\mu}{\sqrt{N}} \right), \quad A_{\mu \rightarrow i} = -\partial_\omega g_{\text{out}}(\omega_{\mu \rightarrow i}, y^\mu, V_{\mu \rightarrow i}) \left(\frac{x_i^\mu}{\sqrt{N}} \right)^2,$$

and the function

$$g_{\text{out}}(\omega, y, V) = \frac{\int dz P_{\text{out}}(y | z) (z - \omega) e^{-\frac{(z - \omega)^2}{2V}}}{V \int dz P_{\text{out}}(y | z) e^{-\frac{(z - \omega)^2}{2V}}}$$

represents the *output channel*, and it is a function of the mean $\omega_{\mu \rightarrow i}$ and variance $V_{\mu \rightarrow i}$ of the message, as well as the label y^μ . The output channel encodes how the input patterns are transformed into outputs, and it is crucial for the inference process [Mézard and Montanari, 2002, Zdeborová and Krzakala, 2016, Krzakala et al., 2015].

The messages from factor to variable have been written as a Gaussian distribution; we can now similarly approximate the messages from variable to factor as Gaussian distributions. The messages from variable nodes to factor nodes are defined as

$$m_{i \rightarrow \mu}(w_i) \propto P(w_i) \exp \left[-\frac{(w_i - R_{i \rightarrow \mu})^2}{2\Sigma_{i \rightarrow \mu}} \right].$$

Using this notation, the messages from variable nodes to factor nodes are the product of the factorized prior of the weights with a Gaussian distribution with mean $R_{i \rightarrow \mu}$ and variance $\Sigma_{i \rightarrow \mu}$. The parameters $R_{i \rightarrow \mu}$ and $\Sigma_{i \rightarrow \mu}$ are computed as

$$R_{i \rightarrow \mu} = \Sigma_{i \rightarrow \mu} \sum_{\nu \neq \mu} B_{\nu \rightarrow i}, \quad \Sigma_{i \rightarrow \mu} = \left(\sum_{\nu \neq \mu} A_{\nu \rightarrow i} \right)^{-1}, \quad (1.52)$$

where $B_{\nu \rightarrow i}$ and $A_{\nu \rightarrow i}$ are the parameters of the messages from factor nodes to variable nodes, as defined above. The messages from variable nodes to factor nodes are then updated iteratively until convergence. The final means and variances for the weights' marginal

distributions can be computed as

$$a_i = g_{in} \left(\sum_{\nu} \frac{1}{A_{\nu \rightarrow i}}, \frac{\sum_{\nu} B_{\nu \rightarrow i}}{\sum_{\nu} A_{\nu \rightarrow i}} \right), \quad v_i = f_v \left(\sum_{\nu} \frac{1}{A_{\nu \rightarrow i}}, \frac{\sum_{\nu} B_{\nu \rightarrow i}}{\sum_{\nu} A_{\nu \rightarrow i}} \right),$$

where g_{in} and f_v are defined as follows:

$$g_{in}(\Sigma, R) = \frac{\int dw w P(w) \exp \left[-\frac{(w-R)^2}{2\Sigma} \right]}{\int dw P(w) \exp \left[-\frac{(w-R)^2}{2\Sigma} \right]},$$

$$f_v(\Sigma, R) = \Sigma \partial_R g_{in}(\Sigma, R).$$

The pseudo-code below outlines the iterative update scheme of the Relaxed Belief Propagation (r-BP) algorithm.

Algorithm 1 Relaxed Belief Propagation (r-BP)**Input:** $\{\mathbf{x}^\mu, y^\mu\}_{\mu=1,\dots,M}$ **Initialization:** Set $a_{i \rightarrow \mu}^{(0)}, v_{i \rightarrow \mu}^{(0)}$; $t \leftarrow 1$ **while** not converged on $\{a_{i \rightarrow \mu}^{(t)}\}$ and $\{v_{i \rightarrow \mu}^{(t)}\}$ **do** Update $\omega_{\mu \rightarrow i}^{(t)}$ and $V_{\mu \rightarrow i}^{(t)}$:

$$V_{\mu \rightarrow i}^{(t)} \leftarrow \sum_{j \neq i} \frac{(x_j^\mu)^2}{N} v_{j \rightarrow \mu}^{(t-1)} \quad (1.53)$$

$$\omega_{\mu \rightarrow i}^{(t)} \leftarrow \sum_{j \neq i} \frac{x_j^\mu}{\sqrt{N}} a_{j \rightarrow \mu}^{(t-1)} \quad (1.54)$$

 Update $B_{\mu \rightarrow i}^{(t)}$ and $A_{\mu \rightarrow i}^{(t)}$:

$$B_{\mu \rightarrow i}^{(t)} \leftarrow g_{\text{out}}(\omega_{\mu \rightarrow i}^{(t)}, y^\mu, V_{\mu \rightarrow i}^{(t)}) \frac{x_i^\mu}{\sqrt{N}}, \quad (1.55)$$

$$A_{\mu \rightarrow i}^{(t)} \leftarrow -\partial_\omega g_{\text{out}}(\omega_{\mu \rightarrow i}^{(t)}, y^\mu, V_{\mu \rightarrow i}^{(t)}) \frac{(x_i^\mu)^2}{N} \quad (1.56)$$

 Update $R_{i \rightarrow \mu}^{(t)}$ and $\Sigma_{i \rightarrow \mu}^{(t)}$:

$$\Sigma_{i \rightarrow \mu}^{(t)} \leftarrow \left(\sum_{\nu \neq \mu} A_{\nu \rightarrow i}^{(t)} \right)^{-1}, \quad (1.57)$$

$$R_{i \rightarrow \mu}^{(t)} \leftarrow \Sigma_{i \rightarrow \mu}^{(t)} \sum_{\nu \neq \mu} B_{\nu \rightarrow i}^{(t)} \quad (1.58)$$

Update the messages from variable nodes to factor nodes:

$$a_{i \rightarrow \mu}^{(t)} \leftarrow g_{\text{in}}(\Sigma_{i \rightarrow \mu}^{(t)}, R_{i \rightarrow \mu}^{(t)}), \quad (1.59)$$

$$v_{i \rightarrow \mu}^{(t)} \leftarrow f_v(\Sigma_{i \rightarrow \mu}^{(t)}, R_{i \rightarrow \mu}^{(t)}) \quad (1.60)$$

 $t \leftarrow t + 1$ **end while****Return:** Estimated marginals:

$$a_i = g_{\text{in}}\left(\sum_\nu \frac{1}{A_{\nu \rightarrow i}}, \sum_\nu \frac{B_{\nu \rightarrow i}}{A_{\nu \rightarrow i}}\right), \quad v_i = f_v\left(\sum_\nu \frac{1}{A_{\nu \rightarrow i}}, \sum_\nu \frac{B_{\nu \rightarrow i}}{A_{\nu \rightarrow i}}\right)$$

Approximate Message Passing

One of the main problems of Belief Propagation and Relaxed Belief Propagation is that, in the case in which the factor graph is not a tree, by iterating the message updates, one introduces correlations between the patterns and the weights, due to the presence of the cycles. This correlation, on factor graphs with small enough cycles, can cause the divergence of the message update procedure. This effect is usually observed when one considers dense models, like the ones considered in the proportional regime investigation for $N \rightarrow \infty$. By adding the so-called Onsager reaction term to the expansion and using the law of large

numbers [Zdeborová and Krzakala, 2016], one can further improve r-BP.

One can notice that in the dense model, messages depend weakly on the target node; for this reason, by following [Zdeborová and Krzakala, 2016], one can first define

$$\omega_\mu \equiv \frac{1}{\sqrt{N}} \sum_j x_j^\mu a_{j \rightarrow \mu} \quad V_\mu \equiv \frac{1}{N} \sum_j (x_j^\mu)^2 v_{j \rightarrow \mu}, \quad (1.61)$$

$$R_i \equiv \Sigma_i \sum_\nu B_{\nu \rightarrow i}, \quad \Sigma_i \equiv \left(\sum_\nu A_{\nu \rightarrow i} \right)^{-1}, \quad (1.62)$$

and notice that the correction to V are small in the large- N limit, since V_μ is a sum of N terms, each of which is $O(1/N)$:

$$V_\mu \approx \frac{1}{N} \sum_j (x_j^\mu)^2 v_j. \quad (1.63)$$

Using the definition of A and B , together with:

$$g_{\text{out}}(\omega_{\mu \rightarrow i}, y^\mu, V_{\mu \rightarrow i}) \approx g_{\text{out}}(\omega_\mu, y^\mu, V_\mu) - \frac{x_i^\mu}{\sqrt{N}} a_{i \rightarrow \mu} \partial_\omega g_{\text{out}}(\omega_\mu, y^\mu, V_\mu), \quad (1.64)$$

$$\approx g_{\text{out}}(\omega_\mu, y^\mu, V_\mu) - \frac{x_i^\mu}{\sqrt{N}} a_i \partial_\omega g_{\text{out}}(\omega_\mu, y^\mu, V_\mu), \quad (1.65)$$

one can also notice that:

$$\Sigma_i = \left(-\frac{1}{N} \sum_\mu (x_i^\mu)^2 \partial_{\omega_\mu} g_{\text{out}}(y_\mu, \omega_\mu, V_\mu) \right)^{-1},$$

$$R_i = a_i + \Sigma_i \frac{1}{\sqrt{N}} \sum_\mu x_i^\mu g_{\text{out}}(y_\mu, \omega_\mu, V_\mu).$$

Using these findings together with the definition of a , one can conclude:

$$a_{i \rightarrow \mu} \approx g_{\text{in}}(R_i, \Sigma_i) - B_{\mu \rightarrow i} f_v(R_i, \Sigma_i) \approx a_i - g_{\text{out}}(y_\mu, \omega_\mu, V_\mu) \frac{x_i^\mu}{\sqrt{N}} v_i.$$

By plugging this definition of $a_{i \rightarrow \mu}$ into the expression for ω_μ , we can conclude:

$$\omega_\mu = \frac{1}{\sqrt{N}} \sum_j x_j^\mu a_j - V_\mu g_{\text{out}}(y_\mu, \omega_\mu, V_\mu), \quad (1.66)$$

where the second term in Equation (1.66) is the Onsager reaction term, which accounts for the correlations introduced by the cycles in the factor graph. This term is crucial for ensuring that the messages remain stable and do not diverge during the updates. The pseudocode for the AMP algorithm for the perceptron model is shown in Algorithm 2. The algorithm iteratively updates the messages until convergence, allowing for the estimation of the marginals of the weights w_i .

Algorithm 2 Approximate Message Passing (AMP)

Input: Data X and observations $\mathbf{y}$;

Initialize: Set $\mathbf{a}^{(0)}$, $\mathbf{v}^{(0)}$, $\mathbf{V}^{(0)}$, $\boldsymbol{\omega}^{(0)}$, g_{out} , g_{in} , $\mathbf{R}^{(0)}$, $\boldsymbol{\Sigma}^{(0)}$, and $t \leftarrow 1$
while not converged on $\mathbf{a}^{(t)}$ and $\mathbf{v}^{(t)}$ **do**

 Update of $\omega_\mu^{(t)}$ and $V_\mu^{(t)}$:

$$V_\mu^{(t)} \leftarrow \frac{1}{N} \sum_j \left(x_j^\mu\right)^2 v_j^{(t-1)} \quad (1.67)$$

$$\omega_\mu^{(t)} \leftarrow \frac{1}{\sqrt{N}} \sum_j x_j^\mu a_j^{(t-1)} - V_\mu^{(t)} g_{out}\left(y^\mu, \omega_\mu^{(t-1)}, V_\mu^{(t-1)}\right) \quad (1.68)$$

 Update of $\Sigma_i^{(t)}$ and $R_i^{(t)}$:

$$\Sigma_i^{(t)} \leftarrow \left(-\frac{1}{N} \sum_\mu \left(x_i^\mu\right)^2 \partial_{\omega_\mu} g_{out}\left(y^\mu, \omega_\mu^{(t)}, V_\mu^{(t)}\right)\right)^{-1} \quad (1.69)$$

$$R_i^{(t)} \leftarrow a_i^{(t-1)} + \Sigma_i^{(t)} \frac{1}{\sqrt{N}} \sum_\mu x_i^\mu g_{out}\left(y^\mu, \omega_\mu^{(t)}, V_\mu^{(t)}\right) \quad (1.70)$$

 Update of $a_i^{(t)}$ and $v_i^{(t)}$:

$$a_i^{(t)} \leftarrow g_{in}\left(R_i^{(t)}, \Sigma_i^{(t)}\right) \quad (1.71)$$

$$v_i^{(t)} \leftarrow \Sigma_i^{(t)} \partial_{R_i} g_{in}\left(R_i^{(t)}, \Sigma_i^{(t)}\right) \quad (1.72)$$

 $t \leftarrow t + 1$
end while
Return: Converged marginal mean and variances $\mathbf{a}_i$ and $\mathbf{v}_i$

In summary, by incorporating the Onsager reaction term in the expression of $\omega_\mu^{(t)}$ and exploiting the weak dependence of messages on the target node, AMP provides a computationally efficient way to estimate the marginals directly. This methodology is beneficial in high-dimensional inference problems such as the teacher-student perceptron scenario.

AMP algorithm, now presented in a general form for the teacher-student perceptron problem with factorizable prior $P(\mathbf{w})$, is tuned in the next chapters to match the specific problem we are addressing.

Chapter 2

Generative diffusion for perceptron problems: statistical physics analysis and efficient algorithms

A central challenge in modern machine learning is the efficient sampling from complex, high-dimensional probability distributions. This issue is fundamental in several domains: in generative modeling, where methods such as variational autoencoders, autoregressive transformers, and diffusion models aim to approximate real-world data distributions; and in Bayesian inference, where the objective is to explore posterior distributions over model parameters.

In this chapter, based on the work [Straziota et al., 2025], we address the sampling problem using the Stochastic Localization (SL) scheme, which is closely related to diffusion models [Montanari and Wu, 2023, Montanari, 2023, El Alaoui et al., 2022]. After introducing the SL framework and its algorithmic implementation, referred to as Algorithmic Stochastic Localization (ASL), we investigate its performance in sampling the solution space of random instances of two non-convex perceptron problems: the spherical perceptron with negative margin and the binary-weight perceptron, both previously introduced in Chapter 1.

The spherical perceptron with negative margin, originally proposed as a simple model of glassy behavior [Franz and Parisi, 2016], can be addressed as a non-convex constraint satisfaction problem whose solution space exhibits a complex, star-shaped geometry [Annesi et al., 2023]. As shown in Section 2.4.3, it is possible to sample uniformly from the solution space of this model using ASL.

The scenario is markedly different for the binary-weight perceptron. The statistical physics literature has extensively studied this problem [Krauth and Mézard, 1989, Engel and Van den Broeck, 2001], revealing that, under the uniform distribution, most solutions are isolated and thus conjectured to be algorithmically inaccessible to a broad class of polynomial-time algorithms due to the overlap gap property [Gamarnik, 2021, Gamarnik et al., 2022, Huang and Kabashima, 2014, Baldassi et al., 2023]. The hardness of sampling has been rigorously established for the symmetric binary perceptron [Alaoui and Gamarnik, 2024].

Despite these challenges, efficient algorithms for finding solutions do exist [Braunstein and Zecchina, 2006, Baldassi et al., 2016a, Abbe et al., 2022], owing to the presence of a subdominant but algorithmically accessible dense cluster of solutions [Baldassi et al., 2015].

The structure of this chapter is as follows. Section 2.1 reviews the relevant literature on

diffusion models and sampling from intractable distributions, emphasizing the connections between these approaches. Section 2.2 introduces the Stochastic Localization process as a generative diffusion scheme for sampling from a target distribution $p(\mathbf{w})$ known up to an intractable normalization constant. Section 2.3 presents a theoretical analysis of SL in the high-dimensional limit using the replica method, providing criteria for the feasibility of efficient sampling. Finally, Section 2.4 applies SL to the perceptron problems, evaluating its effectiveness in sampling from their solution spaces.

2.1 Related works

In this Section, at first, we review the relevant literature on diffusion models and sampling from intractable distributions, and then we focus the attention on Stochastic Localization, highlighting the connection and the fundamental differences with generative diffusion.

2.1.1 Diffusion models

Diffusion models [Sohl-Dickstein et al., 2015, Ho et al., 2020] have recently emerged as a powerful class of generative models, capable of capturing highly complex data distributions across a wide range of domains, while maintaining relatively low sample complexity. In typical applications, the target distribution is not known in closed form, but access to samples from it is assumed. The generative task consists of producing new samples from this unknown distribution through a two-step procedure.

In the first phase, known as the *forward* or *noising process*, noise is progressively added to an observed data sample $\mathbf{w}_0$, yielding a sequence $\{\mathbf{w}_1, \dots, \mathbf{w}_T\}$. At each time step $t \in \{1, \dots, T\}$, a controlled amount of noise is introduced, such that the final state $\mathbf{w}_T$ follows a known prior distribution, typically a standard Gaussian.

The second phase, the *reverse* or *denoising process*, aims to invert the noising dynamics. A neural network is trained to approximate the reverse transition $\mathbf{w}_{t-1}$ given $\mathbf{w}_t$, effectively learning how to recover cleaner samples from noisy inputs. Starting from $\mathbf{w}_T \sim \mathcal{N}(0, I)$, the network iteratively refines the sample one step at a time, from $t = T$ down to $t = 1$, gradually transforming noise into a data-like sample.

This denoising procedure can be interpreted as integrating a stochastic differential equation (SDE), specifically a diffusion process, where the key quantity to be learned is the *score function*, i.e., the gradient of the log-density of the data distribution. In practice, a neural network is trained to approximate this score function using the *denoising score matching* objective [Vincent, 2011], which enables effective sampling from the learned generative model. A graphical intuition of the diffusion models scheme is reported in Figure 2.1.

2.1.2 Sampling from intractable distributions: the Stochastic Localization framework

Stochastic Localization (SL) setting is slightly different from the one of diffusion models. In the SL setting, one is interested in sampling from a target probability density, known up to an intractable normalization factor:

$$p(\mathbf{w}) = \frac{\psi(\mathbf{w})}{Z}, \quad (2.1)$$

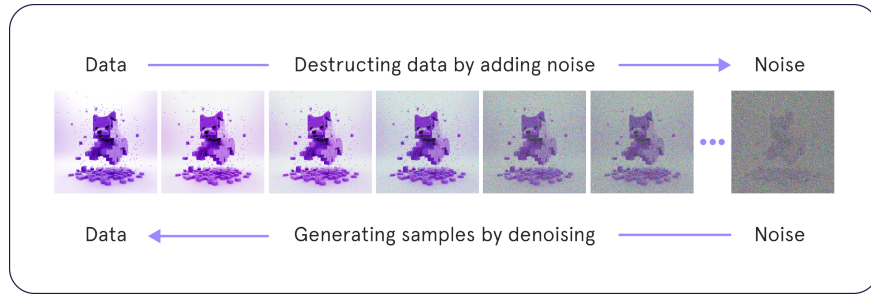


Figure 2.1: Graphical representation of noising and denoising processes in diffusion models, from [Parti, 2024].

as is often the case in Bayesian inference or in the large-sample regime. Typical sampling algorithms belong to the Markov Chain Monte Carlo (MCMC) family [Brooks et al., 2011]. This class of algorithms can suffer from slow mixing times, and their theoretical analysis is generally challenging. In contrast, for generative diffusion, the continuous stochastic process can be well approximated by a small number of discrete steps, and the Bayesian structure induced by the noising/denoising process facilitates theoretical analysis. Recent efforts have explored hybrid approaches combining generative diffusion with MCMC methods [Grenioux et al., 2024, Noble et al., 2024, Vargas et al., 2023a,b].

SL [Eldan, 2013, Montanari, 2023] provides a principled framework to sample from distributions of the form in Eq. (2.1) using a diffusion-like process, with an exact score function. As demonstrated in [El Alaoui et al., 2022], the stochastic localization procedure can successfully sample from the target distribution, provided that the score function can be accurately approximated at any time t . This raises a fundamental question: can we approximate the score function efficiently at any time t , using a polynomial-time algorithm with access to the unnormalized density $\psi(\mathbf{w})$?

In this Chapter, a theoretical framework to address this question, in the high-dimensional limit, is proposed, considering random instances of the distribution p itself (i.e., assuming quenched disorder, in statistical physics terminology). Our framework relies on the non-rigorous but extensively validated replica method from spin glass theory [Mézard et al., 1987, Charbonneau et al., 2023]. After performing the theoretical analysis, a sampling algorithm (associated with the replica analysis) employing Approximate Message Passing (AMP) [Donoho et al., 2009, Barbier et al., 2019] as the score approximator within the diffusion process, is introduced. The new sampling scheme is referred to as *Algorithmic Stochastic Localization (ASL)*¹.

Algorithmic Stochastic Localization (ASL) was first introduced in [El Alaoui et al., 2022], where it was shown to generate unbiased samples from the Sherrington-Kirkpatrick spin glass model at sufficiently high temperatures. Beyond spin glasses, ASL has also been successfully applied to statistical inference problems, such as spiked matrix models and high-dimensional regression [Montanari and Wu, 2023].

A work closely related to this Chapter is [Ghio et al., 2024], which analyzes an analogue of ASL within the stochastic interpolant framework [Albergo et al., 2023] using replica and cavity methods. Their analysis provides asymptotic thresholds for diffusion-based sampling across a range of problems in statistical physics, statistics, and combinatorial optimiza-

¹AMP is conjectured to be optimal among polynomial-time algorithms for the denoising task involved in computing the score function.

tion, though it is restricted to so-called "planted" target distributions. Additionally, [Ricci-Tersenghi and Semerjian, 2009] performed a similar analysis for the Belief Propagation-guided decimation scheme in random SAT problems.

2.2 Stochastic Localization Sampling Scheme

In this Section, we present the Stochastic Localization sampling scheme. Let's suppose that we are given a probability density $p(\mathbf{w})$, with $\mathbf{w} \in \mathbb{R}^N$, referred to as the *target distribution*, and we want to sample from p . We assume p to be known, possibly up to a hard-to-compute normalization factor, and write it as $p(\mathbf{w}) = \psi(\mathbf{w})/Z$, where we call *partition function* the normalization $Z = \int d\mathbf{w} \psi(\mathbf{w})$ and we call $\psi(\mathbf{w})$ *unnormalized density*.

Given the target density p , we define a vector $\mathbf{h}_t \in \mathbb{R}^N$, which we call the time-dependent *field*, evolving according to a stochastic differential equation (SDE) that goes from time $t = 0$ to $t = +\infty$. The field $\mathbf{h}_t$ is initialized at $t = 0$ to the zero vector, $\mathbf{h}_0 = \mathbf{0}$, and evolves according to the following SDE:

$$d\mathbf{h}_t = \mathbf{m}_t(\mathbf{h}_t) dt + d\mathbf{b}_t, \quad (2.2)$$

where $(\mathbf{b}_t)_{t \geq 0}$ is the standard Wiener process in N dimensions. The drift term $\mathbf{m}_t(\mathbf{h}_t)$ is computed as the expectation

$$\mathbf{m}_t(\mathbf{h}) = \mathbb{E}_{\mathbf{w} \sim p_{\mathbf{h},t}} [\mathbf{w}], \quad (2.3)$$

over what we call the *tilted distribution* $p_{\mathbf{h},t}(\mathbf{w})$. The tilted distribution is proportional to the target distribution $p(\mathbf{w})$, but with an additional term that depends on the field $\mathbf{h}_t$ and on the time t :

$$p_{\mathbf{h},t}(\mathbf{w}) = \frac{\psi(\mathbf{w}) e^{\langle \mathbf{h}, \mathbf{w} \rangle - \frac{t}{2} \|\mathbf{w}\|^2}}{Z_{\mathbf{h},t}}. \quad (2.4)$$

The key feature of the SL process is that as $t \rightarrow +\infty$ the field diverges and $p_t := p_{\mathbf{h}_t,t}$ peaks around a single configuration $\mathbf{w}^*$ that is statistically distributed (over the realizations of the process) as a sample from the target distribution p ,

$$p_t \xrightarrow{t \rightarrow +\infty} \delta_{\mathbf{w}^*}, \quad \mathbf{w}^* \sim p. \quad (2.5)$$

Therefore, a sample from the target distribution p can be obtained by evaluating $\mathbf{h}_t$, or analogously $\mathbf{m}_t(\mathbf{h}_t)$, at large times.

The Stochastic Localization procedure admits a natural Bayesian interpretation, which highlights its relevance for inference problems. As shown in [Montanari, 2023], at any time $t \geq 0$, the solution $\mathbf{h}_t$ of Equation (2.2) is distributed as

$$\mathbf{h}_t = t\mathbf{w}^* + \sqrt{t}\mathbf{g}, \quad (2.6)$$

where $\mathbf{w}^* \sim p$ and $\mathbf{g} \sim \mathcal{N}(\mathbf{0}, I_N)$. This is analogous to the backward process in denoising diffusion models. In this inference framework, the distribution $p_{\mathbf{h}_t,t}$ can be interpreted as the posterior distribution of $\mathbf{w}^*$ given the noisy observation $\mathbf{h}_t$, and the function $\mathbf{m}_t(\mathbf{h}_t)$ acts as the Bayesian denoiser.

2.2.1 The ASL algorithm

The Algorithmic Stochastic Localization that we use for a given sampling task consists of discretizing the SDE, in Equation (2.2), running it up to some large but finite times, and decoding the solution from $\mathbf{h}_t$ or $\mathbf{m}_t(\mathbf{h}_t)$. For the problems considered in this thesis, the denoiser $\mathbf{m}_t(\mathbf{h}_t)$ is obtained using Approximate Message Passing [Donoho and Tanner, 2009]. Therefore, we have a double loop algorithm where at each step of the discretized SDE, the inner AMP loop has to be iterated until convergence. While AMP has limited use on real-world tasks and datasets, due to its failure in handling correlated or low-dimensional data, in the synthetic settings addressed in this work, it provides a fast and efficient algorithm for the Bayesian denoising tasks, conjecturally outperforming any other polynomial algorithm when working on large system sizes. Moreover, the fixed points of AMP are in one-to-one relation with stationary points of the replica free entropy discussed in the next section, so that the behavior of AMP is amenable to a simple asymptotic theoretical analysis [Zdeborová and Krzakala, 2016].

2.2.2 ASL and Generative Diffusion

In this section, we recap the connection between ASL and diffusion models.

Both ASL and diffusion models involve a stochastic process for sampling from a target distribution. While in diffusion models, the target distribution is typically unknown and only accessible through samples, in ASL, the target distribution is known up to a normalization constant, but is intractable to sample from directly.

In both the processes, we use SDE to add progressive noise²: while in generative diffusion, starting from data, we reach a known prior distribution (Forward Process), and then we learn to revert the forward noising process to generate new samples, in principle, following the data distribution (Backward Process); in ASL, the forward process is implicit, and the reverse process is defined by the SDE in Equation (2.2), which progressively localizes the tilted distribution around a single sample from the target measure.

It is important to stress that the diffusion-based formulation of Stochastic Localization (SL) [Eldan, 2013] can be mapped to standard denoising diffusion processes via appropriate transformations of time and space, of Equation (2.2) [Montanari, 2023, El Alaoui and Montanari, 2022]. This mapping allows us to interpret the ASL process as a specific instance of a diffusion model, where the score function is exact and computed, via AMP on the tilted distribution $p_{\mathbf{h},t}$.

2.3 Asymptotic Analysis of ASL

We devise a formalism to investigate the asymptotic behavior of the ASL sampling process in the large system size limit $N \rightarrow +\infty$. The formalism relies on the non-rigorous but well-established [Mézard et al., 1987] replica method. Our main result consists in a criterion for the feasibility of fair sampling involving the evaluation of a time-dependent free entropy. What follows is a generalization of the scheme in [Ghio et al., 2024] that allows for handling unnormalized densities, as the ones we deal with in the next sections.

Given the target distribution p , which we assume to be stochastic and drawn from an ensemble of distributions (quenched disorder), the solution $\mathbf{h}_t$ of the SDE, in Equation (2.2)

²relatively to data and fields.

(assuming the drift term is correctly estimated) is distributed as $\mathbf{h}_t = t\mathbf{w}^* + \sqrt{t}\mathbf{g}$, where $\mathbf{w}^*$ is a sample from p , referred to as the *reference sample*, and $\mathbf{g}$ is a standard Gaussian noise (see Equation (2.6)). A relevant parameter for tracking the dynamics is the overlap

$$q(t) = \frac{\mathbf{w}^* \cdot \mathbf{w}_t}{N}, \quad (2.7)$$

where $\mathbf{w}^*$, is not planted, but represent a sample of the target distribution.

We argue that this quantity, while fluctuating over the realization of p and of the SDE path, concentrates for large N to a deterministic quantity. This quantity emerges naturally as an order parameter in the replica computation that we now describe.

As introduced in Chapter 1, the free entropy of the system is a central object of investigation, since it provides access to several quantities of interest. In our case, for a given time t , and considering the presence of disorder, we define the asymptotic average free entropy (or just free entropy) as

$$\phi_t = \lim_{N \rightarrow +\infty} \frac{1}{N} \mathbb{E}_{\psi, \mathbf{g}, \mathbf{w}^*} \log Z_{\mathbf{h}_t, t}, \quad (2.8)$$

where

$$Z_{\mathbf{h}_t, t} = \int d\mathbf{w} \psi(\mathbf{w}) e^{\langle \mathbf{h}_t, \mathbf{w} \rangle - \frac{t}{2} \|\mathbf{w}\|^2},$$

and the expectation is computed over the realization of the target unnormalized density ψ over the reference sample $\mathbf{w}^*$ and over the Gaussian noise $\mathbf{g}$ of Equation (2.6). Averaging over the realization of the reference sample $\mathbf{w}^*$ gives

$$\phi_t = \lim_{N \rightarrow +\infty} \frac{1}{N} \mathbb{E}_{\psi, \mathbf{g}} \int \frac{\psi(d\mathbf{w}^*)}{Z} \log Z_{\mathbf{h}_t, t}, \quad (2.9)$$

where $\mathbf{h}_t$ depends on $\mathbf{w}^*$ as $\mathbf{h}_t = t\mathbf{w}^* + \sqrt{t}\mathbf{g}$. To handle the usually intractable expectation of a logarithm and the sampling from an unnormalized distribution, we employ the replica method twice:

$$\log Z = \lim_{n \rightarrow 0} \partial_n Z^n \quad \text{and} \quad Z^{-1} = \lim_{s \rightarrow 0} Z^{s-1}. \quad (2.10)$$

We introduce s replicas of the target distribution and n replicas for the tilted distribution and obtain

$$\phi_t = \lim_{N \rightarrow +\infty} \frac{1}{N} \lim_{s \rightarrow 0} \lim_{n \rightarrow 0} \partial_n \mathbb{E}_{\psi, \mathbf{g}} \int \prod_{\alpha=1}^s \psi(d\mathbf{w}_\alpha^*) \prod_{a=1}^n \psi(d\mathbf{w}_a) e^{\langle \mathbf{h}_t, \mathbf{w}_a \rangle - \frac{t}{2} \|\mathbf{w}_a\|^2}, \quad (2.11)$$

where $\mathbf{h}_t$ is computed from $\mathbf{w}_1^*$, the first of the s replicas associated to the target distribution: $\mathbf{h}_t = t\mathbf{w}_1^* + \sqrt{t}\mathbf{g}$. The structure of this computation is closely related to the one presented in the seminal work of [Silvio Franz and Giorgio Parisi, 1995]. Notice that the n replicas $\mathbf{w}_a$ have symmetric roles, while in the set of s replicas $\mathbf{w}_\alpha^*$, the $\alpha = 1$ one has a special role. Carrying out the computation, in dense systems as the ones we consider in this chapter, all the possible pairwise overlaps among the $n+s$ replicas emerge naturally as order parameters subject to saddle point optimization. We then assume the Replica Symmetric (RS) ansatz [Mézard et al., 1987, Mezard and Montanari, 2009], and we restrict the saddle point evaluation to the most symmetric overlap structure under replica exchanges compat-

ible with the symmetries of Equation (2.11). Moreover, thanks to the Bayesian structure of the problem, one can apply the Nishimori conditions [Nishimori, 1980, Iba, 1999, Ghio et al., 2024] to further simplify the overlap structure, by imposing additional symmetries (see Appendix A.1.3). Under these hypotheses, the overlap $q(t)$ of Equation (2.7) can be computed from the overlaps of replicas involved in Equation (2.11) as

$$q(t) = \frac{\mathbf{w}_1^* \cdot \mathbf{w}_a}{N} = \frac{\mathbf{w}_a \cdot \mathbf{w}_b}{N}, \quad \forall a, b \in [n] \text{ and } b \neq a. \quad (2.12)$$

At time $t = 0$, all the $n + s$ replicas become equivalent to each other, therefore $q(t = 0) = r$, where r is the overlap between two distinct samples of the target distribution: $r = \frac{\mathbf{w}_\alpha^* \cdot \mathbf{w}_\beta^*}{N}$ for $\alpha, \beta \in [s]$ and $\alpha \neq \beta$. We further mention that while the Nishimori conditions found for the planted distributions analyzed in [Ghio et al., 2024] guarantee the correctness of the RS ansatz, in our more general setting they guarantee only that the analysis of the tilted system doesn't involve further replica symmetry breaking compared to the standard replica analysis of the reference systems. Therefore, our RS analysis is exact only in the RS region.

In general, the replica computation will arrive at an expression where the free entropy ϕ_t is expressed as the saddle point of a function of several order parameters, one of which is q :

$$\phi_t = \max_q \phi_t(q), \quad \phi_t(q) := \text{extr}_\theta \phi_t(q, \theta), \quad (2.13)$$

where we denoted with θ all the remaining order parameters except q . The value $q_{\max}$ that maximizes $\phi_t(q)$ represents the typical value of $q(t)$.

Success and Failure of ASL

As shown in [Ghio et al., 2024], the study of $\phi_t(q)$ can reveal whether the ASL sampling scheme can recover samples from the target distribution, in the large N limit. More specifically, the success of ASL is based on the unimodality of $\phi_t(q)$ as a function of q . If it has a single maximum at any time t , moving smoothly from low q to $q = 1$ (assuming $\|\mathbf{w}\|^2 = N$), then the AMP messages correctly recover (with high probability in the limit of large N) the value $q(t) = \arg \max_q \phi_t(q)$ and the algorithm successfully samples from the target distribution. Conversely, if $\phi_t(q)$ becomes multimodal, it could be the case that AMP doesn't return the correct estimate of $\mathbf{m}_t(\mathbf{h}_t)$ and the SDE integration fails. The mechanism is as follows: at $t = 0$, there is always a maximum located at low q , which is the one found by AMP. As t increases, this maximum moves toward higher q , and AMP follows it; however, if at any t there is a second, higher maximum and at higher q , it should be the correct one that solves the saddle point equations, but AMP misses it. As we show below, this can happen both if the global maximum exists from the outset or if it develops over time. See Figure 2.2 (Left) for a setting in which a high q maximum first appears and then becomes the global one.

2.4 Sampling Perceptron solutions

In this section, we revisit the Stochastic Localization sampling scheme in the context of two perceptron problems: the spherical perceptron with negative margin and the binary-weight perceptron.

A general introduction to perceptron models and their significance in machine learning and statistical physics is provided in Chapter 1. Here, rather than adopting the usual optimization perspective, we follow a constraint satisfaction perspective [Engel and Van den Broeck, 2001] and define a family of probability distributions over the solution space of the perceptron.

Specifically, we reformulate the problem as follows. Given a dataset X consisting of M patterns $\mathbf{x}^\mu \in \mathbb{R}^N$ with unit labels $y^\mu = 1$ and a margin parameter κ , a *weight configuration* $\mathbf{w} \in \mathbb{R}^N$ is considered a *solution* if the associated stabilities, defined as $s^\mu = \frac{\mathbf{x}^\mu \cdot \mathbf{w}}{\sqrt{N}}$, satisfy $s^\mu \geq \kappa$ for all μ , meaning that all examples are correctly classified.

We introduce a family of distributions over the solution space, characterized by the unnormalized density (see Equation (2.1)):

$$\psi(\mathbf{w}) = P(\mathbf{w}) \prod_{\mu=1}^M \Theta(s^\mu - \kappa) e^{-\frac{1}{T}U(s^\mu - \kappa)}, \quad \text{with } s^\mu = \frac{\mathbf{x}^\mu \cdot \mathbf{w}}{\sqrt{N}}, \quad (2.14)$$

where $\Theta(s)$ is the Heaviside step function, $\Theta(s) = 1$ if $s > 0$ and 0 otherwise. The parameter $T \geq 0$ is referred to as the temperature, and $U(s)$ is called the *potential*. For $U(s) = 0$ (or any constant) or $T \rightarrow +\infty$, the distribution $p(\mathbf{w}) = \psi(\mathbf{w})/Z$ reduces to the uniform distribution over the solution space. Otherwise, the distribution becomes non-uniform and depends on the specific form of the potential $U(s)$. The function $P(\mathbf{w})$ denotes a prior over the weights, which may be unnormalized. Two priors are considered in this chapter: for the spherical perceptron (a continuous weights model), $P(\mathbf{w}) = \delta(\|\mathbf{w}\|^2 - N)$, and for the binary perceptron (a discrete weights model), $P(\mathbf{w}) = \prod_i P(w_i)$ with $P(w_i) = \delta(w_i - 1) + \delta(w_i + 1)$.

The patterns are i.i.d. samples of a normal distribution or a uniform distribution over $\{1, -1\}$, $\mathbf{x}^\mu \sim \mathcal{N}(0, I_N)$ or $\mathbf{x}^\mu \sim \text{Unif}(\{-1, +1\}^N)$. For our asymptotic analysis, these two settings are equivalent. The labels are uniformly sampled. We focus on the high-dimensional limit in the proportional regime $\alpha = M/N$. This statistical setting has been extensively studied in the statistical physics literature; see [Engel and Van den Broeck, 2001] and [Gabri   et al., 2023] for comprehensive reviews.

2.4.1 Implementation of ASL

To adapt the ASL sampling scheme, as discussed in Section 2.2, to the target distribution in Equation (2.14), the corresponding AMP algorithm has been implemented. Since Equation (2.14) can be seen as a specific type of generalized linear model, we can adapt the GAMP algorithm from [Rangan, 2011] for our purposes. The AMP framework relies on ϕ_{in} and ϕ_{out} , commonly referred to as the input- and output-channel free entropies, given by:

$$\phi_{\text{in}}(\Sigma, R) = \log \int P(dw) e^{wR - \frac{1}{2}w^2\Sigma}, \quad (2.15)$$

$$\phi_{\text{out}}(\omega, V) = \log \int ds \Theta(s - \kappa) \frac{e^{-\frac{1}{T}U(s - \kappa) - \frac{(s - \omega)^2}{2V}}}{\sqrt{2\pi V}} = \log \tilde{H}_V(-\omega), \quad (2.16)$$

where $\tilde{H}_a(b) = \int Dz \Theta(-b - \kappa + \sqrt{a}z) e^{-\frac{1}{T}U(-b - \kappa + \sqrt{a}z)}$ and $Dz = \frac{dz}{\sqrt{2\pi}} e^{-z^2/2}$. For the case $U(s) = 0$, $\tilde{H}$ simplifies to $\tilde{H}_a(b) = H(-\frac{b + \kappa}{\sqrt{a}})$, with $H(x) = \frac{1}{2} \text{erfc}\left(\frac{x}{\sqrt{2}}\right)$. In the binary weights case, one can substitute $\int P(dw) = \sum_{w=\pm 1}$. The AMP algorithm, reported in Algorithm 3, is derived for the perceptron target distribution in Equation (2.14), tilted by the observation

of $\mathbf{h}_t$, as described in Equation (2.4).

Even if the AMP pseudocode has been presented in Section 1.3, we report its modified version here for completeness.

Algorithm 3 AMP for ASL on the Perceptron model

Input: Data $X = \{\mathbf{x}^\mu\}_{\mu=1,\dots,M} \in \mathbb{R}^{M \times N}$, $\mathbf{y} \in \mathbb{R}^M$, ASL time and field $t \geq 0$ $\mathbf{h}_t \in \mathbb{R}^N$, $\epsilon > 0$, $K > 0$.

Initialize: Set $\mathbf{a}^{(0)}$, $\mathbf{v}^{(0)}$, $\mathbf{V}^{(0)}$, $\boldsymbol{\omega}^{(0)}$, ϕ_{out} , ϕ_{in} , $\mathbf{R}^{(0)}$, $\boldsymbol{\Sigma}^{(0)}$, Δ_{iter} , and $k \leftarrow 0$

while not converged on $\mathbf{a}^{(t)}$ and $\mathbf{v}^{(t)}$ (number of iters == K or $\Delta_{\text{iter}} < \epsilon$) **do**

Updating mean and variance estimates ω_μ, V_μ

$$V_\mu^{(k)} \leftarrow \sum_i \frac{(x_i^\mu)^2}{N} \quad (2.17)$$

$$\omega_\mu^{(k)} \leftarrow \sum_i \frac{x_i^\mu}{\sqrt{N}} a_i^{(k-1)} - V_\mu^{(k)} g_\mu^{(k-1)} \quad (2.18)$$

Updating estimates Σ_i, R_i, g_μ

$$g_\mu^{(k)} \leftarrow \partial_{\omega} \phi_{\text{out}} \left(y^\mu, \omega_\mu^{(k)}, V_\mu^{(k)} \right) \quad (2.19)$$

$$\Sigma_i^{(k)} \leftarrow - \sum_\mu \frac{(x_i^\mu)^2}{N} \partial_{\omega}^2 \phi_{\text{out}} \left(y^\mu, \omega_\mu^{(k)}, V_\mu^{(k)} \right) \quad (2.20)$$

$$R_i^{(k)} \leftarrow \sum_\mu \frac{x_i^\mu}{\sqrt{N}} g_\mu^{(k)} + a_i^{(k-1)} \Sigma_i^{(k)} \quad (2.21)$$

Only for spherical case: enforce norm constraint solving (2.24) for γ .

Updating marginals m_i and Δ_i

$$a_i^{(k)} \leftarrow \partial_{R_i} \phi_{\text{in}} \left(\Sigma_i^{(k)} + t, R_i^{(k)} + h_{t,i} \right) \quad (2.22)$$

$$v_i^{(k)} \leftarrow \partial_{R_i}^2 \phi_{\text{in}} \left(\Sigma_i^{(k)} + t, R_i^{(k)} + h_{t,i} \right) \quad (2.23)$$

$\Delta_{\text{iter}} \leftarrow \|\mathbf{a}^{(k)} - \mathbf{a}^{(k-1)}\|^2 / N$ and $k \leftarrow k + 1$

end while

Return: Converged marginals $\mathbf{a}^{(k)}$ and $\mathbf{v}^{(k)}$.

In the case of spherical perceptron, the algorithm enforces (in expectation) the norm constraint $\|\mathbf{w}\|^2 = N$ by assuming a unnormalized prior $P(w) = e^{-\frac{1}{2}\gamma w^2}$, and then solving for γ the equation

$$-2 \sum_{i=1}^N \partial_\gamma \phi_{\text{in}}(\Sigma_i^{(k)} + t, R_i^{(k)} + h_{t,i}) = N, \quad (2.24)$$

in each AMP iteration, using a root-finding algorithm. Notice that γ appears implicitly in ϕ_{in} defined in Equation (2.15).

The relaxed prior is equivalent to the hard one in the limit of large N , but AMP can handle only factorized priors. As a consequence, the integral in ϕ_{in} can always be computed in closed form. Furthermore, for some choices of the potential $U(s)$, notably $U(s) = s^2$, $U(s) = 0$ and $U(s) = -\log(s)$, the integration in ϕ_{out} can also be carried out analytically³. Therefore, the AMP used in the settings discussed is quite fast since it does not contain any integrals.

³However the only two relevant cases, for our analysis, are $U(s) = 0$ and $U(s) = -\log(s)$

Once the AMP algorithm converges, the output $\mathbf{a}^{(k)}$ is used to estimate the drift term $\mathbf{m}_t(\mathbf{h}_t)$ in Equation (2.2) as reported in Algorithm 4. The evolution of the tilting field is performed using the Euler-Maruyama scheme, on Equation (2.6). The process is repeated for L iterations, where L is a parameter that controls the number of steps in the SDE integration. At the end of the process, the final weights are obtained by applying the sign function to the last AMP output.

Algorithm 4 Sampling via stochastic localization

Input: Data: $X \in \mathbb{R}^{N \times M}$, $\mathbf{y} \in \{-1, 1\}^N$, ϵ , L , δ , $\tilde{T}$

Initialization: Σ^0 , $\mathbf{R}^0$, $t \leftarrow 0$

while $l \leq L$ **do**

AMP updates of the marginals:

$$\mathbf{w}^l, \Delta^l = \text{AMP}(X, \mathbf{y}, \Sigma^{l-1}, \mathbf{R}^{l-1}, \epsilon, \tilde{T})$$

Update the fields using the Euler-Maruyama scheme:

$$t^l \leftarrow t^{l-1} + \delta \tag{2.25}$$

$$(\Sigma^l)_i \leftarrow t \tag{2.26}$$

$$(\mathbf{R}^l)_i \leftarrow (\mathbf{R}^{l-1})_i + w_i^{l-1} \delta + \sqrt{\delta} z_i \text{ with } \mathbf{z} \sim \mathcal{N}(0, \mathbb{I}_N), \forall i \in \{1, \dots, N\} \tag{2.27}$$

$l \leftarrow l + 1$

end while

$$\mathbf{w}_i^L = \text{sign}(\mathbf{w}_i^L), \forall i \in \{1 \dots N\}$$

return a solution sample $\mathbf{w}^L$

In Algorithm 4, the field term is incorporated into the AMP input channel by shifting its mean R_i^l with $w_i^{l-1} \delta$, as shown in Equation (2.27).

2.4.2 General Replica computation for Perceptron models

In this Section, the fundamental steps of the replica calculation used to derive the time-dependent free-entropy ϕ_t of the Stochastic Localization process are presented. The formalism is applied to the general perceptron model given in Equation (2.14), and then specialized on the spherical and binary case. For the sake of conciseness, we report only the fundamental results of the computation; the full computation is reported in Appendix A.1. The computation is performed, under the assumption that the dataset $X = \{\mathbf{x}^\mu\}_{\mu=1}^M$, is made of patterns $\mathbf{x}^\mu \sim \mathcal{N}(\mathbf{0}, I_N)$. The labels are assumed to be unitary, $y^\mu = 1$. The weights have a factorized prior $P(\mathbf{w}) = \prod_i P(w_i)$, for the binary perceptron case. The case of the non-factorizable global spherical prior is handled naturally by constraining the diagonal overlap to unity.

For the sake of notation, starting from Equation (2.11), one can define the replicated partition function

$$Z_t^{n,s} = \int \prod_{\alpha=1}^s \psi(d\mathbf{w}_\alpha^*) \prod_{a=1}^n \psi(d\mathbf{w}_a) e^{\langle \mathbf{h}_t, \mathbf{w}_a \rangle - \frac{t}{2} \|\mathbf{w}_a\|^2}. \tag{2.28}$$

In the case of the general perceptron model, the expectation over the replicated partition

function $Z_t^{n,s}$ become:

$$\mathbb{E} Z_t^{n,s} = \mathbb{E}_{X,g} \int \prod_{\alpha=1}^s P(d\mathbf{w}_\alpha^*) \prod_{a=1}^n P(d\mathbf{w}_a) \prod_{\mu\alpha} \tilde{\Theta} \left(\sum_i \frac{w_{\alpha i}^* x_i^\mu}{\sqrt{N}} \right) \prod_{\mu a} \tilde{\Theta} \left(\sum_i \frac{w_{a i} x_i^\mu}{\sqrt{N}} \right) \times \quad (2.29)$$

$$\times e^{t \sum_{ai} w_{1i}^* w_{ai} + \sqrt{t} \sum_{ai} g_i w_{ai} - \frac{1}{2} t \sum_{ai} w_{ai}^2}. \quad (2.30)$$

where the realization of ψ depends from the realization of the dataset X , and $\tilde{\Theta}$.

One can now introduce the overlap parameters definition

$$q_{ab} = \frac{\mathbf{w}_a \cdot \mathbf{w}_b}{N}, \quad r_{\alpha\beta} = \frac{\mathbf{w}_\alpha^* \cdot \mathbf{w}_\beta^*}{N}, \quad p_{\alpha a} = \frac{\mathbf{w}_\alpha^* \cdot \mathbf{w}_a}{N}, \quad (2.31)$$

where q_{ab} is the overlap between replicas of the tilted distribution, $r_{\alpha\beta}$ is an overlap in the reference system, and $p_{\alpha a}$ is a cross-overlap between reference and tilted system. We also introduce the conjugate parameters $\hat{q}_{ab}, \hat{r}_{\alpha\beta}, \hat{p}_{\alpha a}$ to enforce their definitions. The overlaps q_{aa} and $r_{\alpha\alpha}$ are fixed to unity, since $\|\mathbf{w}_b\|^2 = N$ in both the spherical and binary case. For the spherical case, the diagonal multipliers $\hat{q}_{aa}$ have to be optimized, while their value is irrelevant in the binary case.

After some mathematical manipulation, the expectation of the replicated partition function becomes:

$$\mathbb{E} Z_t^{n,s} = \int \prod_{\alpha \leq \beta} dr_{\alpha\beta} d\hat{r}_{\alpha\beta} \prod_{a \leq b} dq_{ab} d\hat{q}_{ab} \prod_{\alpha a} dp_{\alpha a} d\hat{p}_{\alpha a} e^{N\Phi_t}, \quad (2.32)$$

where

$$\Phi_t = G_I + G_S + \alpha G_E \quad (2.33)$$

$$G_I = -\frac{1}{2} \sum_{\alpha\beta} \hat{r}_{\alpha\beta} r_{\alpha\beta} - \frac{1}{2} \sum_{ab} \hat{q}_{ab} q_{ab} - \sum_{\alpha a} \hat{p}_{\alpha a} p_{\alpha a} + t \sum_a p_{1a} + \frac{1}{2} t \sum_{ab} q_{ab} - \frac{t}{2} \sum_a q_{aa}, \quad (2.34)$$

$$G_S = \log \int \prod_{\alpha=1}^s P(d\mathbf{w}_\alpha^*) \prod_{a=1}^n P(d\mathbf{w}_a) e^{+\frac{1}{2} \sum_{\alpha\beta} \hat{r}_{\alpha\beta} w_\alpha^* w_\beta^* + \frac{1}{2} \sum_{ab} \hat{q}_{ab} w_a w_b + \sum_{\alpha a} \hat{p}_{\alpha a} w_\alpha^* w_a}, \quad (2.35)$$

$$G_E = \log \int \prod_\alpha \frac{d\lambda_\alpha d\hat{\lambda}_\alpha}{2\pi} \prod_a \frac{du_a d\hat{u}_a}{2\pi} \prod_\alpha \tilde{\Theta}(\lambda_\alpha) \prod_a \tilde{\Theta}(u_a) \times e^{-i \sum_\alpha \hat{\lambda}_\alpha \lambda_\alpha - i \sum_a \hat{u}_a u_a - \frac{1}{2} \sum_{\alpha\beta} \hat{\lambda}_\alpha \hat{\lambda}_\beta r_{\alpha\beta} - \frac{1}{2} \sum_{ab} \hat{u}_a \hat{u}_b q_{ab} - \sum_{\alpha a} \hat{\lambda}_\alpha \hat{u}_a p_{\alpha a}}. \quad (2.36)$$

The computation is now performed under the Replica Symmetric Ansatz:

$$q_{ab} = \begin{cases} q_d = 1 & \text{if } a = b \\ q & \text{if } a \neq b \end{cases}, \quad r_{\alpha\beta} = \begin{cases} r_d = 1 & \alpha = \beta \\ r & \alpha \neq \beta \end{cases}, \quad p_{\alpha a} = \begin{cases} p^* & \alpha = 1 \\ p & \alpha \neq 1 \end{cases}. \quad (2.37)$$

The Nishimori conditions, discussed in Appendix A.1, guarantee that if the RS ansatz is correct for the reference system, then the RS ansatz above gives a correct prediction for ϕ_t . More generally, the correct Ansatz for the tilted system has the same number of steps of symmetry breaking as the correct one for the reference system. The Bayesian Optimality

condition used in this context are the following:

$$p^* = q \quad p = r \quad \hat{p}^* = \hat{q} \quad \hat{p} = \hat{r}. \quad (2.38)$$

At $t = 0$, the tilted system coincides with the reference one, i.e. $q = r$. However, for $t > 0$, the overlap q can differ from r , in this case the field term induces a bias in the sampling process, favoring the alignment of tilted measure weights with the reference ones, $\mathbf{w}_1^*$. Only the first reference system replica drives the generative process of the tilted system, i.e. the overlap between tilted replicas and the first reference replica differs from the other possible overlaps ($p^* \neq p$).

Using these simplifications, the expression of ϕ_t can be computed for both the binary and spherical perceptron models. Their analytical expressions are reported in the following sections.

2.4.3 Spherical Non-Convex Perceptron

In this section, we present the asymptotic analysis of ASL, considering the spherical case $\|\mathbf{w}\|^2 = N$ and uniform distribution, i.e., $U(s) = 0$ in Equation (2.14). In particular, we're interested in the case $\kappa < 0$, since in that case the space of the solutions (if non-empty, i.e., for small enough α) is non-convex on the sphere [Franz and Parisi, 2016, Annesi et al., 2023]. As discussed in Section 2.3, we characterize the sampling behavior by studying the free entropy ϕ_t as a function of q . The replica calculation yields (full derivation in Appendix A.1):

$$\phi_t(q, \hat{q}) = -\frac{1}{2}(\hat{r}_d + \hat{q}q) + \hat{r}r + \frac{1}{2}tq + \mathcal{G}_S(\hat{q}) + \alpha \mathcal{G}_E(q), \quad (2.39)$$

$$\mathcal{G}_S(\hat{q}) = -\frac{1}{2}\log(\hat{q} - \hat{r}_d) + \frac{1}{2(\hat{q} - \hat{r}_d)} \left(\hat{q} + 2\hat{r} \frac{(\hat{q} - \hat{r})}{(\hat{r} - \hat{r}_d)} + \frac{(\hat{q} - \hat{r})^2}{(\hat{r} - \hat{r}_d)} \left(\frac{\hat{r}}{\hat{r} - \hat{r}_d} + 1 \right) \right), \quad (2.40)$$

$$\mathcal{G}_E(q) = \int Dz D\gamma \frac{\tilde{H}_{1-q}(\gamma\sqrt{r} + z\sqrt{q-r})}{\tilde{H}_{1-r}(\gamma\sqrt{r})} \log \tilde{H}_{1-q}(\gamma\sqrt{r} + z\sqrt{q-r}). \quad (2.41)$$

The saddle point equations for the order parameters $q, \hat{q}$ follows:

$$\hat{q} = t + 2\alpha \frac{\partial}{\partial q} \mathcal{G}_E(q), \quad (2.42)$$

$$q = 2 \frac{\partial}{\partial \hat{q}} \mathcal{G}_S(\hat{q}). \quad (2.43)$$

The parameters for the reference system, $r, \hat{r}$ and $\hat{r}_d$, are independent of t, q and $\hat{q}$. They are determined upfront, using the saddle point equation obtained from the reference system free entropy (see Appendix A.1.4), which is the one of the storage problems introduced in Section 1.3.2. They play the role of external parameters in the above expressions.

The above expressions allow us to derive $\phi_t(q) = \text{extr}_{\hat{q}} \phi_t(q, \hat{q})$ and gain direct insight into whether the SL sampling scheme can recover samples from the target distribution asymptotically. The left panel of Figure 2.2 shows $\phi_t(q)$ as a function of q for different values of t , with parameters $\alpha = 278$ and $\kappa = -2.5$. Initially, $\phi_t(q)$ exhibits a single global maximum. However, a second local maximum emerges as t increases, eventually becoming

the global optimizer of ϕ_t . As discussed in Section 2.3, this transition marks the onset of multimodality in the free entropy landscape. The central panel of Figure 2.2 presents the phase diagram for the ASL scheme for a fixed margin κ , delineating the different sampling regimes. For each point in the t - α plane, we test for the presence or absence of distinct local maxima by initializing the optimization of $\phi_t(q, \hat{q})$ in Equation (2.39) at two different values of q , low and high, and checking whether the results coincide.

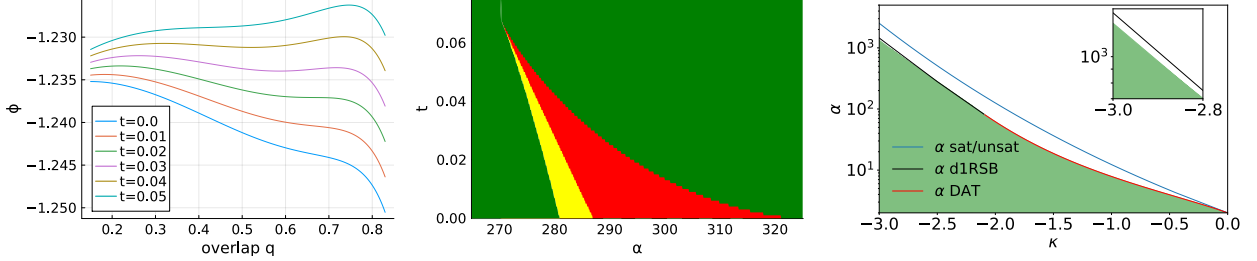


Figure 2.2: Asymptotic Analysis of ASL sampling for the Spherical Perceptron with uniform distribution. *Left panel:* Free entropy function $\phi_t(q)$ for different values of t and for $\alpha = 278, \kappa = -2.5$. Initially, $\phi_t(q)$ has a single maximum, but as t increases, a second maximum appears, eventually becoming the global one. *Center panel:* Phase diagram of ASL in the t -vs- α plane for $\kappa = -2.5$. *Green region:* $\phi_t(q)$ has a single optimizer, meaning the AMP succeeds at denoising. *Yellow region:* $\phi_t(q)$ has two optimizers, but the global maximum corresponds to the smaller overlap q . AMP still succeeds. *Red region:* $\phi_t(q)$ has two optimizers, but the global maximum corresponds to a larger overlap q . In this case, AMP fails the denoising task. For ASL to succeed at sampling, a vertical line at the corresponding α should lie entirely in the green region. *Right panel:* Phase diagram delineating the samplable and non-samplable regions for ASL in α -vs- κ plane. Transition lines predicted from replica theory are taken from [Baldassi et al., 2023]. The green region can be sampled by ASL. The zoom in the inset shows the failure of ASL at reaching the d1RSB line.

The right panel of Figure 2.2 shows the region in the α - κ plane where ASL succeeds. The figure also shows replica symmetry-breaking transition lines and the 1RSB prediction for the sat/unsat transition line reported in [Baldassi et al., 2023]. Notably, the sampling threshold of ASL coincides with the De Almeida-Thouless (DAT) line and comes slightly short of the dynamical 1-step Replica Symmetry Breaking (1RSB) transition. This result highlights a fundamental limitation: the SL algorithm fails to sample from the target distribution when Replica Symmetry Breaking (RSB) occurs, which signals the fragmentation of the solution space into disconnected clusters. In other words, ASL stops working as soon as ergodicity is broken. In the case of the DAT transition, ergodicity is continuously broken, and ASL reaches the transition line. In the presence of a discontinuous (d1RSB) transition, failure of ASL happens earlier. See [Ghio et al., 2024] for more details of phase transitions in similar contexts and [Mézard et al., 1987, Mezard and Montanari, 2009] for the generic RSB picture.

To validate numerically the uniformity of the ASL sampling, we compare the empirical distribution of stabilities s^μ obtained via sampling to the asymptotic theoretical prediction, which can also be obtained by the replica method (Figure 2.3). The derivation is reported in Appendix A.2. The empirical results match perfectly with the predictions in the whole region where ASL successfully samples a solution to the constraint satisfaction problem. This strongly suggests that the obtained samples are distributed according to the target distribution, i.e., uniformly over the solution space in this case, as expected.

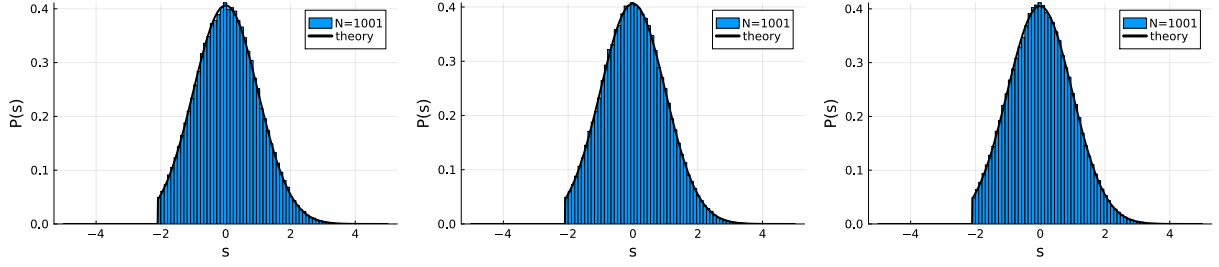


Figure 2.3: Distribution of the stabilities s^μ of a spherical perceptron with negative margin, where $s^\mu = \frac{w \cdot x^\mu}{\sqrt{N}}$. For number of variables $N = 1001$, $\kappa = -2.1$ and $\alpha = 5$ (Left), $\alpha = 20$ (Center) and $\alpha = 80$ (Right). The empirical distribution of the stabilities of the ASL samples (blue) coincides for different parameter regimes with the distribution computed through the replica method (black solid line).

2.4.4 Binary Perceptron

In this section, we analyze the ability to sample binary perceptron solutions. We focus on the zero margin case, $\kappa = 0$. The replica analysis is reported in Appendix A.4; the final expression for the free entropy $\phi_t(q, \hat{q})$ at time t is the same as Equation (2.39), as well as the saddle point equations, except for the Entropic term, which becomes

$$\mathcal{G}_S(\hat{q}) = \sum_{w^*=\pm 1} \int Dz D\gamma \frac{e^{\gamma\sqrt{\hat{r}}w^*}}{2 \cosh(\gamma\sqrt{\hat{r}})} \log \left(2e^{-\frac{1}{2}(\hat{q}-\hat{r}d)} \cosh \left(z\sqrt{\hat{q}-\hat{r}} + \gamma\sqrt{\hat{r}} + (\hat{q}-\hat{r})w^* \right) \right). \quad (2.44)$$

The expression for the free entropy of the reference system is reported in Appendix A.4.

In the following, the possibility of sampling binary solutions for the binary perceptron models is investigated, for two classes of potentials $U(x)$: non-diverging potentials and a specific choice of diverging one.

Sampling fails for non-diverging potentials

As anticipated in Section 2.3, ASL fails in the presence of a second peak in the free entropy $\phi_t(q)$ that becomes the global maximum at some time. Let's first consider the ability to sample binary perceptron solutions under the uniform distribution, i.e., $U(s) = 0$.

The left panel of Figure 2.4 shows the free entropy $\phi_t(q)$ as a function of the overlap q for different values of t , with parameters $\alpha = 0.5$ and $\kappa = 0$. The phenomenology observed in this case is completely different from that observed for the spherical case: for all α and small t , there is a second maximum at $q = 1$. As shown in the figure, the maximum at $q = 1$ becomes the global one for sufficiently large t , while there is still a lower- q maximum. As explained in Section 2.3, this implies that during the sampling procedure, the information on the target is not correctly reconstructed, leading to the failure of ASL. The phase diagram shown in the right panel of Figure 2.4 shows that this phenomenology is present for all $\alpha > 0$ ⁴.

This observation is related to the frozen-1RSB nature of the binary perceptron [Krauth and Mézard, 1989], implying that most solutions are isolated, and it is consistent with the

⁴The figure does not report values of $\alpha \approx 0$ since in this case numerical imprecision occurs

known hardness of sampling from the uniform distribution [Huang and Kabashima, 2014, Alaoui and Gamarnik, 2024]. More surprisingly, this difficulty cannot be removed even by using a non-zero potential, unless it diverges at the origin, as we now discuss.

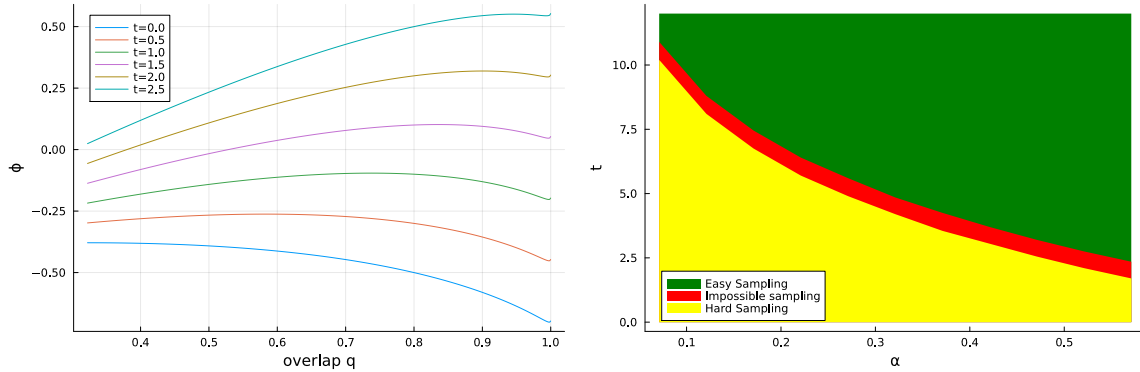


Figure 2.4: *Left panel:* Free entropy $\phi_t(q)$ for the binary perceptron problem for several times. At least for small t , the curves have two local maxima: one at low q that grows with t , and a persistent one at $q = 1$. By increasing t , the persistent one becomes global, while the lower- q one is still present, which leads to the failure of ASL. *Right panel:* Phase diagram of SL sampling for the binary perceptron. At all α , the situation is like the one in the left panel: there is a region of t where the lower- q maximum is not the global maximum (red regions) and sampling fails, even though eventually the lower- q maximum eventually disappears (green regions).

To understand the origin of the peak at $q = 1$, we compute $\frac{d\phi_t(q)}{dq}$, for $q = 1 - \epsilon$, with $\epsilon \ll 1$ [Huang and Kabashima, 2014]. The computations are reported in Appendix A.5. For the binary case, independently of the choice of the potential, the expression of the free entropy derivative takes the form:

$$\left. \frac{d\phi_t(q)}{dq} \right|_{q=1-\epsilon} = \frac{1}{2} \log(\epsilon) + \alpha C(\epsilon) \epsilon^{-\frac{1}{2}} + O(1) \quad (2.45)$$

where $C(\epsilon)$ is a function whose scaling with ϵ depends on the explicit form of the potential $U(s)$. As it turns out, unless $U(s)$ diverges at 0, $C(\epsilon) = O(1)$ and is always positive, and as a consequence the free entropy unavoidably exhibits a peak at $q = 1$ at all $t > 0$ and all $\alpha > 0$. The second peak becomes dominant at a large enough time, before the first peak disappears, leading to the failure of ASL sampling.

Diffusion with Log-potential

The only way to avoid the peak at $q = 1$ is to let the potential $U(s)$ diverge at the origin. The choice $U(s) = -\log(s)$ is particularly simple to analyze: in that case, $C(\epsilon) = O(\epsilon^{\frac{1}{2T}})$ and therefore the corresponding term $C(\epsilon) \epsilon^{-\frac{1}{2}}$ in Equation (2.45) becomes negligible for small ϵ when $T < 1$. The derivative of the free entropy at $q = 1$ becomes dominated by the logarithmic term and thus $\phi_t(q)$ always has a local minimum at $q = 1$.

In this case, the phenomenology becomes similar to the one observed for the spherical perceptron case, as shown in Figure 2.5 (Left): with the choice $T = 0.5$, there is a whole range of α up to about $\alpha_{\text{alg}} \approx 0.65$ that is samplable. The transition is close to, although

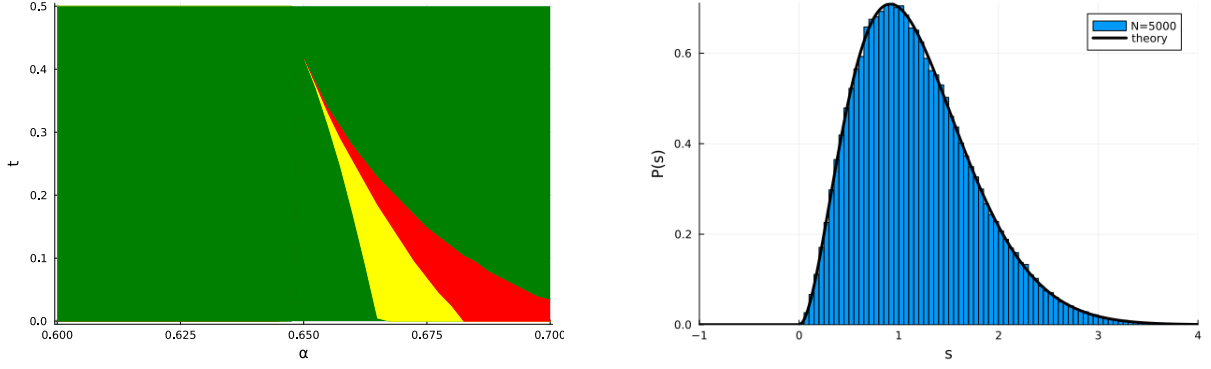


Figure 2.5: ASL sampling for the Binary Perceptron with $U(s) = -\log(s)$ potential. *Left panel:* Phase diagram of ASL in the t -vs- α plane for $\kappa = 0$ and $T = 0.5$. The color scheme is as explained for the central panel of Figure 2.2. *Right panel:* Empirical distribution of the stabilities s^μ for a configuration obtained by ASL in the case of binary perceptron with the log-potential, $N = 5000$, $\kappa = 0$, $T = 0.5$, and $\alpha = 0.3$. The black line is the asymptotic theoretical prediction. The excellent agreement shows that ASL produces fair samples from the target distribution.

slightly lower than, the best known algorithmic thresholds from heuristic solvers [Braunstein and Zecchina, 2006, Baldassi and Braunstein, 2015, Baldassi et al., 2016b]. In our case, though, and contrary to all other solver algorithms we are aware of, the solutions found are fair samples from a target distribution which is fully under control analytically. This is corroborated by the comparison between the empirical stabilities from the ASL sampler and the theoretical predictions, shown in Figure 2.5 (Right).

Theoretical findings have been validated against numerical simulations of the ASL algorithm. We considered 250 random instances of the binary perceptron, $\mathcal{D} = \{X, \mathbf{y}\}$ for several values of α , with margin $\kappa = 0$ and system sizes

$$N \in \{1000, 2000, 5000, 8000\}.$$

We use ASL for sampling from a tuned binary perceptron measure with a logarithmic potential at temperature $T = 0.5$. Figure 2.6 shows the probability of sampling a solution that satisfies all constraints as a function of α for the different system sizes N . The numerical results deviate from the replica-based prediction (guaranteeing sampling up to $\alpha \approx 0.65$), with ASL fairly sampling only up to $\alpha \approx 0.6$. While this confirms that ASL can fairly sample binary perceptron solutions over a wide range of α , the origin of this discrepancy is attributed to finite size effect.

τ -annealed MCMC

Using the ASL algorithm in combination with the potential $U(s) = -\log(s)$, we can sample solutions of random instances of the binary perceptron problem. ASL, however, inherits the well-known limitations of AMP, generally failing to converge in the presence of structured data, and therefore, ASL should not be considered a practical sampling algorithm for generic perceptron problems. On the other hand, direct use of the log-potential in an MCMC algorithm is infeasible, since the potential is not defined on negative stabilities s , which means that one should initialize the MC chain from a solution. As a workaround, we

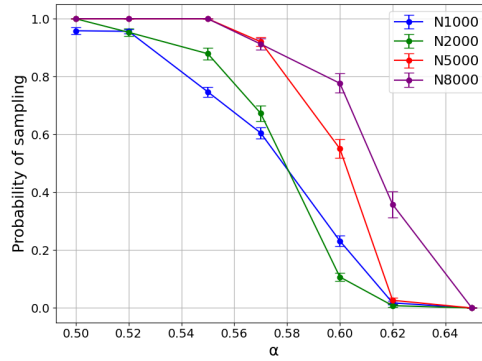


Figure 2.6: Probability of sampling a solution satisfying all the constraints for the binary perceptron problem with the log-potential tuned distribution, as a function of constraint density α and for different system sizes N . The results are averaged over 250 random instances. The error bars represent the empirical standard deviation for a Bernoulli distribution.

propose a reshaping of the potential inspired by the replica trick, which, given a parameter $\tau > 0$, is defined by

$$U_\tau(s) = \begin{cases} \frac{1}{\tau}(1 - s^\tau) & s > 0, \\ \frac{1}{\tau}(1 - s) & s \leq 0. \end{cases} \quad (2.46)$$

Notice that $\lim_{\tau \rightarrow 0} U_\tau(s) = -\log(s)$ for $s > 0$ and $+\infty$ for $s \leq 0$. We set $p_\tau(\mathbf{w}) \propto e^{-\frac{1}{\tau} \sum_\mu U_\tau(s^\mu)}$ as the moving target density of a Metropolis-Hasting algorithm, where at each MC sweep we decrease linearly τ , starting with initial value 1 and down to 0. Keeping T fixed and with a sufficiently slow annealing, we should be able to sample from the solution space weighted by the log-potential. We call this procedure τ -annealing. In Figure 2.7, we compare it with a standard Simulated Annealing (SA) on the potential $U(s) = -s\Theta(-s)$, where the temperature T is decreased linearly at each MC sweep from 1 down to 0, so that final samples at $T = 0$ should be distributed according to the uniform measure for a slow enough annealing. While SA fails quickly when increasing N , the τ -annealing scheme remains very effective at finding solutions. It is possible to prove that by scaling the number of sweeps as $n_{\text{sweeps}} = N$, we solve up to $\alpha \approx 0.5$ using τ -annealing, while temperature annealing keeps struggling at large system sizes (by increasing the number of MCMC sweeps, keeping it proportional to N , it is possible to solve binary perceptron instances up to $\alpha \approx 0.55$).

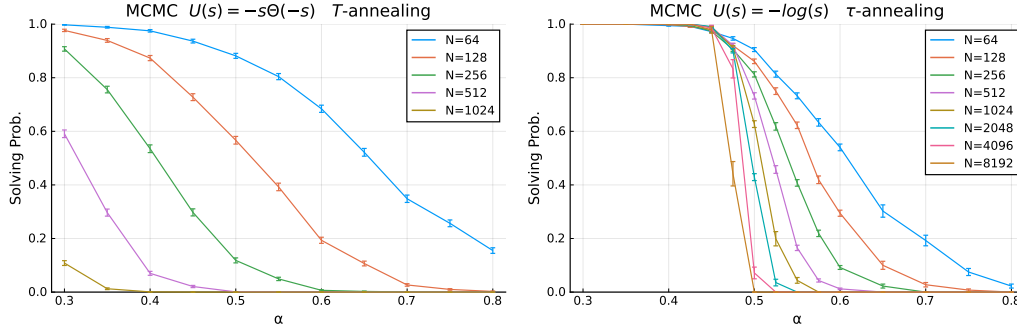


Figure 2.7: Results for the Binary Perceptron problem, showing the probability of finding a solution as a function of constraint density α and for different system sizes N , after 100 sweeps of MCMC. Simulated Annealing on temperature T (*left panel*) is compared to our proposed and much more effective τ -annealing scheme (*right panel*).

2.5 Recap of the results

In this Chapter,

- We investigated the feasibility of sampling solutions to perceptron problems via the diffusion scheme based on Approximate Message Passing, known as Algorithmic Stochastic Localization (ASL). We analyze its asymptotic behavior in the limit of large system size, focusing on two perceptron models: the spherical perceptron and the binary perceptron.
- For the spherical perceptron, we showed that ASL can sample from the target distribution as long as the free entropy landscape remains unimodal along the trajectory, which holds for constraint density $\alpha = M/N$ below a threshold depending on the margin κ .
- For the binary perceptron, we found that the uniform distribution is always unsampleable. To address this, we introduced a potential $U(s) = -\log(s)$ to bias the distribution, enabling efficient sampling over a broader range of α values.
- We proposed a robust MCMC scheme, τ -annealing, that overcomes ASL's limitations (inherited from AMP) on structured instances of the problem.

Chapter 3

Isolating the hard core of phaseless inference: the Phase Selection formulation

A central challenge in supervised machine learning is to optimize models so that they accurately predict outcomes from a set of input features. This task is often complicated by the presence of non-convex optimization landscapes, where local minima can trap optimization algorithms and lead to suboptimal algorithmic performances.

A paradigmatic and analytically tractable example of such a scenario is the Phase Retrieval problem, which involves reconstructing a signal from the magnitudes of its linear measurements, in the absence of phase information, i.e. the signal is reconstructed starting from signless outputs. The lack of phase information promotes the emergence of spurious local minima, resulting in a highly non-convex landscape where standard algorithms frequently struggle to achieve good sample efficiency.

The real-valued version of Phase Retrieval is one of the simplest instances of non-convex continuous optimization and is often studied as a prototypical neural network model with non-linear activation functions [Saglietti et al., 2020, Sarao-Mannelli et al., 2020, Mondelli and Montanari, 2019]. In this analogy, the measurement vectors correspond to input patterns, the hidden signal represents the optimal configuration of network weights, and the measured magnitudes play the role of output labels. This perspective provides a fertile ground for exploring key topics in learning theory, such as optimization in non-convex landscapes, the influence of initialization and optimization heuristics on generalization, and the impact of training dataset size [Mignacco et al., 2020, Arnaboldi et al., 2023, Mignacco et al., 2021].

In this chapter, a statistical physics perspective on Phase Retrieval is discussed in Section 3.1, and a new framing of the MAP setting is proposed, called the Phase Selection problem. Section 3.2 provides a mathematical definition of the Phase Retrieval and Phase Selection problems, highlighting the common properties of their solutions. In Section 3.3, we present the large deviation and replica analysis of the system in the infinite-size limit, within the so-called proportional regime. Finally, Section 3.4 introduces three different meta-heuristics for solving Phase Selection instances, based on Simulated Annealing, Approximate Message Passing, and Langevin Dynamics, and compares their performance.

This chapter is based on the work [Straziota and Saglietti, 2025].

3.1 A Statistical Physics perspective

From a statistical physics perspective, phase retrieval is an example of high-dimensional signal estimation, characterized by an intricate energy-entropy competition where the true signal is concealed within a vast landscape, rugged with local minima. Such "crystal hunting" problems, as they are informally denoted in the physics of disordered systems [Krzakala et al., 2015], exhibit an algorithmically hard phase where all known polynomial-time algorithms struggle, becoming trapped in local minima with low overlap with the true signal [Ricci-Tersenghi et al., 2019, Antenucci et al., 2019a].

An interesting observation, however, is that from the information-theoretical point of view, Phase retrieval becomes possible as soon as the number of measurements equals the dimensionality of the signal, as in a simple linear system [Barbier et al., 2019, Bandeira et al., 2014, Saglietti et al., 2020]. The loss of phase information strongly complicates the inference performance of different local-search algorithms, making the problem difficult to solve efficiently without additional constraints or prior knowledge. For example, gradient descent was shown to struggle in the signal retrieval task [Candes et al., 2015] and therefore requires good initialization, e.g., through spectral methods [Candès et al., 2006, Candes et al., 2015] or large amounts of data. Exploiting full knowledge of the data generative process, the Bayes-optimal approximate message-passing (AMP) approach to this problem was conjectured to achieve the best possible algorithmic performance in the large size limit.

Here, a new framing of the maximum-a-posteriori (MAP) setting of Phase retrieval is proposed, where one potentially deals with imperfect information on the signal prior. Inspired by [Obuchi et al., 2018], presenting a statistical mechanical analysis of Compressed sensing framed as a variable selection problem, we aim to investigate whether it is possible to isolate the hard combinatorial component of Phase retrieval, i.e. the correct selection of the missing phases (or signs, in the real-valued setting), and elucidate how strategic problem formulation can improve both our understanding of the problem and the inference performance. Along this line, *Phase selection* is introduced. It consists of a closely related two-level optimization, where the hidden signal is to be retrieved through a standard linear regression after a specific choice for the signs of the measurements.

A detailed statistical physics analysis of this novel framework is provided under standard simplifying assumptions about the data and signal distributions. At the end of this analysis, different meta-heuristics are applied to Phase Selection, investigating the advantages of disentangling the two sub-tasks within Phase retrieval.

3.2 Introduction and Problem Setting

3.2.1 Phase retrieval

In the noiseless real-valued Phase retrieval problem we are given a measurement matrix $X = \{\mathbf{x}^\mu\}_{\mu=1,\dots,M} \in \mathbb{R}^{M \times N}$ and a measurement vector $\mathbf{y} = \{y^\mu\}_{\mu=1,\dots,M} \in \mathbb{R}^M$, whose entries are assumed to be produced as:

$$y^\mu = \left| \frac{\sum_i x_i^\mu w_i^*}{\sqrt{N}} \right|, \quad (3.1)$$

where $\mathbf{w}^* \in \mathbb{R}^N$ is a hidden signal to be inferred from the observations (where the dataset is $\mathcal{D} = \{X, \mathbf{y}\}$) and $|\cdot|$ denotes the element-wise modulus operation.

In our analysis, we are going to make the standard simplifying assumption that the measurement matrix X has i.i.d. Gaussian entries, i.e. $x_i^\mu \sim \mathcal{N}(0, 1)$, and consider a factorized normal prior also for the hidden signal, i.e. $w_i^* \sim \mathcal{N}(0, 1)$. Note that it is possible to relax these unrealistic assumptions to more complicated distributions [Maillard et al., 2020], but introducing correlations leads to an unnecessarily complicated analysis for this study. Moreover, we consider the so-called proportional scaling regime, where N and M grow to infinity at a fixed rate $\alpha = M/N$, with $\alpha > \alpha_{IT} = 1$ [Bandeira et al., 2014], i.e. a sufficiently large dataset for the signal to be information-theoretically detectable.

In the maximum-a-posteriori setting, the signal recovery problem is mapped to the optimization of a loss function:

$$\begin{aligned} \hat{\mathbf{w}}(\mathcal{D}) &= \arg \min_{\mathbf{w}} \mathcal{L}(\mathbf{w}; \mathcal{D}) \\ \mathcal{L}(\mathbf{w}; \mathcal{D}) &= \|\mathbf{y} - |X\mathbf{w}|\|_2^2 + \frac{\lambda}{2} \|\mathbf{w}\|_2^2, \end{aligned} \tag{3.2}$$

which computes the Mean Squared Error (MSE) between the measurement vector and the output of the inferred model. The L_2 penalty on the model weights was shown to help recover the signal in previous work [Ma et al., 2018]. In the absence of the modulus operator, the optimization task is convex.

It is important to underline that the modulus operation in Equation (3.2) introduces a degeneracy in the optimization landscape, as it does not distinguish between positive and negative values of the inferred signal. This degeneracy leads to an optimization problem characterized by a highly non-convex loss landscape, where multiple local minima exist, corresponding to different sign configurations for the measurements.

Note that often one considers a formally equivalent, but slightly less nefarious, formulation where the loss entails the MSE between the squared measurements of true and inferred signals, avoiding the singular absolute value operation and simplifying the landscape for gradient-based optimization. However, we are interested in maintaining a connection to linear regression. The reason for this is explained in the next sections.

3.2.2 Phase selection problem

The core idea of the Phase selection formulation is to resolve the degeneracy introduced by the modulus operation, and explicit the signs, i.e., real phases to be associated with the measurements via an auxiliary binary variable $\mathbf{S} \in \{-1, 1\}^M$. Then, the loss function, in Equation (3.2), can be reframed as:

$$\mathcal{L}(\mathbf{w}, \mathbf{S}; \mathcal{D}) = \|\mathbf{S} \odot \mathbf{y} - X\mathbf{w}\|_2^2 + \frac{\lambda}{2} \|\mathbf{w}\|_2^2, \tag{3.3}$$

where $\odot : \mathbb{R}^M \times \mathbb{R}^M \rightarrow \mathbb{R}^M$ is the element-wise multiplication. Thus, the original signal estimation problem is decomposed into two optimization levels, the *external* combinatorial problem of choosing the phases, $\mathbf{S}$, and the *internal* nested convex problem of estimating the signal, $\mathbf{w}$. For each sign selection, the *internal* problem reduces to a simple linear regression

(convex problem), solved in closed form by the Ridge estimator:

$$\hat{\mathbf{w}}(\mathbf{S}; \mathcal{D}) = \arg \min_{\mathbf{w}} \mathcal{L}(\mathbf{w}, \mathbf{S}; \mathcal{D}) = \left(X^\top X + \lambda \mathbb{I} \right)^{-1} X^\top \mathbf{S} \odot \mathbf{y}. \quad (3.4)$$

where $\mathbb{I}$ is the identity matrix. The solution to this problem is unique and well-defined, as long as the regularization parameter λ is strictly positive. On the other hand, the loss achieved in correspondence of this minimum induces a probability distribution over the sign configurations. It is clear that the two optimization levels are coupled, as the optimal signal estimate depends on the choice of signs, and the optimal choice of signs depends on the signal estimate.

When the regularization is completely switched off, the unique global minimum of the generalized loss function Equation (3.2) still achieves exactly 0 mean squared error, with a perfect sign selection and retrieval of the hidden signal. The presence of a non-zero L_2 -regularization could bias MSE by explicitly reducing the norm of the inferred signal. It is important to underline, that however in this context the minimum is still hard to find, due to the fractured nature of the loss landscape. A reasonable learning scheme should start with a large regularization parameter, and then anneal it to zero, as discussed in Section 3.3.3. Remarkably, the complete structure of local minima of Phase retrieval is preserved, since each minimum in the original problem is in one-to-one correspondence with a specific gauge for the signs, made explicit by removing the modulus operation after local convergence.

It is important to note that the Phase selection formulation can introduce new local minima compared to Phase retrieval, since in some cases the Ridge estimator might not be able to yield a configuration that matches all the selected signs $\mathbf{S}$. Since Phase selection only introduces additional sub-optimal local minima, the information-theoretic threshold for exact recovery remains unchanged from that of the Phase retrieval problem [Bandeira et al., 2014].

Given a locally optimal choice of signs $\hat{\mathbf{S}}$, stable against single spin flips, the two natural measures of the inference performance are given by:

$$\begin{aligned} MSE_y &= \frac{1}{M} \left\| \hat{\mathbf{S}} \odot \mathbf{y} - \frac{X \hat{\mathbf{w}}(\hat{\mathbf{S}}; \mathcal{D})}{\sqrt{N}} \right\|_2^2 \\ MSE_w &= \frac{1}{N} \left\| \mathbf{w}^* - \hat{\mathbf{w}}(\hat{\mathbf{S}}; \mathcal{D}) \right\|_2^2, \end{aligned} \quad (3.5)$$

respectively representing the mean squared error between the measurements and the mean squared error associated with the hidden signal reconstruction. Closely connected to those metrics, one can also measure the two overlaps:

$$\begin{aligned} m &= \frac{\mathbf{w}^* \cdot \hat{\mathbf{w}}(\hat{\mathbf{S}}; \mathcal{D})}{N} \\ O &= \frac{\text{sign}(X \mathbf{w}^*) \cdot \hat{\mathbf{S}}}{M}, \end{aligned} \quad (3.6)$$

which should approach unity in the limit of perfect recovery. The goal of our theoretical analysis, presented in Section 3.3, is to obtain an exact estimate of the inference performance as a function of the above-defined overlaps, making use of the Replica method [Parisi, 1979,

1983].

A Bayesian perspective

It is possible to reformulate the Phase Retrieval problem from a Bayesian perspective: the estimate $\mathbf{w}$ is drawn from a posterior measure $p(\mathbf{w}|\mathcal{D})$ where the missing phases do not appear explicitly. Using the marginal probability theorem, we can explicit the dependency on $\mathbf{S}$, rewriting the posterior probability for $\mathbf{w}$ as

$$p(\mathbf{w}|\mathcal{D}) = \int_{\Omega_S} p(\mathbf{w}|\mathcal{D}, \mathbf{S})p(\mathbf{S}|\mathcal{D})d\mathbf{S} \quad (3.7)$$

where Ω_S is the set of all possible sign assignments, $p(\mathbf{S}|\mathcal{D})$ is the probability of selecting $\mathbf{S}$ given the dataset $\mathcal{D}$, and $p(\mathbf{w}|\mathcal{D}, \mathbf{S})$ is the likelihood of $\mathbf{w}$ given a realization of the signs $\mathbf{S}$. From Equation (3.7), the two-level optimization structure of Phase Retrieval becomes clear: we first infer the missing phases, and then we use them to obtain an estimate for the model.

3.3 Analytical investigations

In this section, we present the analytical investigation of the Phase selection problem, focusing on the large deviation analysis of the energy landscape associated with the optimization task. The goal is to understand how the energy landscape behaves as a function of the number of correct signs selected, and how this influences the performance of local search algorithms in finding low-energy configurations. The analytical perspective presented in this chapter is slightly different from the one presented in Chapter 1, here, we do not work in the canonical ensemble, but rather we work fixing the fraction of correctly guessed signs.

3.3.1 Large deviation analysis

We use the large deviation principle to characterize the distribution of energy levels, denoted by e , associated with the local minima of the loss function in Equation (3.3), as a function of the different possible sign assignments. As it is standard in high-dimensional statistical physics settings [Dotsenko et al., 1994], we assume that the probability distribution of local minima satisfies a large deviation principle: for a given measurement ratio $M/N = \alpha = \mathcal{O}(1)$ and a fixed fraction of correctly assigned signs O , there exist exponentially many sign configurations yielding a given energy e (remember the analogy with the loss function reported in Section 1.3). The number of such configurations, $e^{N\Sigma(O,e)}$, is governed by the rate function $\Sigma(O, e)$, commonly referred to as complexity.

The large deviation principle implies that, for a random selection of signs with overlap O to the correct assignment, the resulting energy, with high probability, is close to the *typical* value $\bar{e}(O) = \arg \max_e \Sigma(O, e)$. In contrast, lower-energy local minima, those of interest in the Phase Selection problem, are exponentially rare [Cui et al., 2021]. Therefore, analyzing the complexity function provides valuable insight into the performance of local-search algorithms, as it helps to understand how likely it is for such algorithms to find these rare, low-energy configurations.

For technical reasons, in the following, we access the complexity via the computation of its Legendre transform [Cui et al., 2021], the so-called free entropy $\phi(\nu, \zeta)$, where ν is an inverse temperature biasing towards lower energy levels, and ζ is a chemical potential for correct sign choices. In the thermodynamic limit $N \rightarrow \infty$, the saddle-point method yields:

$$\phi(\nu, \zeta) = \text{extr}_{e, O} \{ \Sigma(O, e) + \nu e + \zeta O \}, \quad (3.8)$$

where $e = \partial_\nu \phi(\nu, \zeta)$ and $O = \partial_\zeta \phi(\nu, \zeta)$. This relation implies that, for any value of ν and ζ , an interplay of entropy and energy causes the probability measure to concentrate on local minima with specific energy and sign overlap, which can be obtained by extremizing the free entropy. Then, by inverting the Legendre transform, one can recover the sought complexity:

$$\Sigma(O, e) = [\phi(\nu, \zeta) - \nu e - \zeta O]_{\partial_\nu \phi = e, \partial_\zeta \phi = O}. \quad (3.9)$$

In the next section, we present a concise summary of the analytical computation for the quenched free entropy, employing well-established techniques from the physics of disordered systems [Mezard and Montanari, 2009].

3.3.2 Replica Symmetric formula for the large deviations

To compute the quenched free-entropy for Phase selection, the starting point is the definition of the partition function Z , associated with the probability distribution over the sign configurations. Introducing an inverse temperature ν and a chemical potential ζ , we can write:

$$Z(\nu, \zeta; X, \mathbf{w}^*) = \sum_{\{\mathbf{S}\}} e^{\zeta \sum_{\mu=1}^M S^\mu - \nu \mathcal{L}(\mathbf{S}; X, \mathbf{w}^*)} \quad (3.10)$$

where the energy of the internal system, $\mathcal{L}(\mathbf{S}; X, \mathbf{w}^*)^1$, is recovered as the maximum-a-posteriori, i.e. in the zero temperature limit of the posterior, for the estimated signal:

$$\mathcal{L}(\mathbf{S}; X, \mathbf{w}^*) = \lim_{\rho \rightarrow \infty} -\frac{1}{\rho} \log \int \prod_{i=1}^N dw_i P(w_i) \frac{e^{-\frac{\rho}{2} \left(\sum_{\mu=1}^M \left(S^\mu \frac{\mathbf{x}^\mu \cdot \mathbf{w}^*}{\sqrt{N}} - \frac{\mathbf{x}^\mu \cdot \mathbf{w}}{\sqrt{N}} \right)^2 + \frac{\lambda}{2} \|\mathbf{w}\|_2^2 \right)}}{\sqrt{2\pi/\rho}}, \quad (3.11)$$

and where $\mathbf{x}^\mu$ represents a row of the measurement matrix. Note that, for convenience of notation, compared to the definition Equation (3.5) for the MSE_y , we have considered the gauge transformation:

$$\mathbf{S} \odot \left| \frac{X\mathbf{w}^*}{\sqrt{N}} \right| \rightarrow \mathbf{S} \odot \frac{X\mathbf{w}^*}{\sqrt{N}}, \quad (3.12)$$

which corresponds to reabsorbing in $\mathbf{y}$ the correct signs for the measurements.

This reparametrization does not affect the evaluation of the complexity function but turns out to make the final expression easily interpretable: $S_\mu = 1$ now corresponds to a correct sign selection for the corresponding measurement, while $S_\mu = -1$ corresponds to an incorrect one. This also allows the simplified expression for the chemical potential term in Equation (3.10): a large ζ pushes towards positive — and thus correct — signs.

¹Here we represent the energy function of the internal system using $\mathcal{L}$ to maintain the connection with the ML perspective of the Phase Retrieval Problem.

In this type of analysis, the goal is to average over the specific realizations of the dataset, referred to as quenched disorder, as well as over the particular choice of the target signal, and to compute the typical value of the free entropy:

$$\phi(\nu, \zeta) = \lim_{N \rightarrow \infty} \frac{1}{N} \mathbb{E}_{X, \mathbf{w}^*} \ln Z(\nu, \zeta; X, \mathbf{w}^*).$$

Following [Obuchi et al., 2018], we employ two separate instances of the replica trick to estimate this average. First, one can remove the intractable expectation of the logarithm by using:

$$\mathbb{E}_{X, \mathbf{w}^*} \ln Z = \mathbb{E}_{X, \mathbf{w}^*} \lim_{s \rightarrow 0} \frac{Z^s}{s}, \quad (3.13)$$

since the disorder average can be easily evaluated for integer values of s , and the result can be analytically continued to $s \rightarrow 0$. As for the second replica trick, we assume the ratio between the two inverse temperatures in the problem, external, ν , and internal, ρ , to be integer, $\nu/\rho = \beta \in \mathbb{N}$ and then extrapolate the limit $\beta \rightarrow 0$. The replicated partition function can thus be rewritten as:

$$\begin{aligned} Z^s(\nu, \zeta; X, \mathbf{w}^*, \beta) &= \sum_{\{S^a\}} \prod_a e^{\zeta \sum_{\mu} S^{\mu, a}} \times \\ &\times \left[\int \prod_{a=1}^s \prod_{c=1}^{\beta} d\mathbf{w}_{ac} P(\mathbf{w}_{ac}) \prod_{\mu=1}^{\alpha N} e^{\frac{-\frac{\nu}{2\beta} \sum_{ac} \left(S^{\mu, a} \frac{\mathbf{x}^{\mu} \cdot \mathbf{w}^*}{\sqrt{N}} - \frac{\mathbf{x}^{\mu} \cdot \mathbf{w}_{ac}}{\sqrt{N}} \right)^2 - \sum_{ac} \frac{\lambda \nu}{2\beta} \|\mathbf{w}_{ac}\|_2^2}}{\sqrt{2\pi\beta/\nu}} \right], \end{aligned}$$

introducing two replica indices, $a \in \{1, \dots, s\}$ and $c \in \{1, \dots, \beta\}$.

After switching expectation and limit operations, one can recover the free entropy as:

$$\phi(\nu, \zeta) = \lim_{N \rightarrow \infty} \frac{1}{N} \lim_{s \rightarrow 0} \frac{1}{s} \lim_{\beta \rightarrow 0} \mathbb{E}_{X, \mathbf{w}^*} Z^s(\nu, \zeta; X, \mathbf{w}^*, \beta). \quad (3.14)$$

Definition of the Order parameters

Once the disorder average in $\mathbb{E}_{X, \mathbf{w}^*} Z^s(\nu, \zeta; X, \mathbf{w}^*, \beta)$ is performed, it is natural to identify the overlap order parameters of the model, or summary statistics [Auffinger et al., 2013], which represent a set of sufficient descriptors of the system in high dimensions:

$$\begin{aligned} \tilde{q} &= \frac{\mathbf{w}^* \cdot \mathbf{w}^*}{N}, \\ q_{ac, bd} &= \frac{\mathbf{w}_{ac} \cdot \mathbf{w}_{bd}}{N}, \\ m_{ac} &= \frac{\mathbf{w}^* \cdot \mathbf{w}_{ac}}{N}, \end{aligned} \quad (3.15)$$

where $\tilde{q}$ represents the norm of the hidden signal, $q_{ac, bd}$ the overlap between two signal estimates sampled from the above-defined measure, and m_{ac} is the magnetization, i.e., the overlap between hidden and estimated signal. Through these order parameters, and the associated Lagrange multipliers (denoted with a hat symbol), one can express the free-

entropy as a composition of an interaction, an entropic, and an energetic part:

$$\phi(\tilde{q}, q, m) = \lim_{s, \beta \rightarrow 0} \frac{1}{s} [G_I + \ln G_S + \alpha \ln G_E], \quad (3.16)$$

defined as:

$$\begin{aligned} G_I &= i\hat{q}\tilde{q} + i \sum_{\{a,b,c,d\} \in D} \hat{q}_{ac,bd} q_{ac,bd} + i \sum_{ac} \hat{m}_{ac} m_{ac}, \\ G_S &= \int dw^* P(w^*) \int \prod_{ac} dw_{ac} P(w_{ac}) e^{-\sum_{ac} \frac{\lambda\nu}{2\beta} w_{ac}^2 - i\hat{q}(w^*)^2 - i \sum_{\{a,b,c,d\} \in D} \hat{q}_{ac,bd} w_{ac} w_{bd} - i \sum_{ac} \hat{m}_{ac} w^* w_{ac}}, \\ G_E &= \sum_{\{S^a\}} e^{\zeta \sum_a S^a} \int \frac{du_0 d\hat{u}_0}{2\pi} \prod_{ac} \frac{du_{ac} d\hat{u}_{ac}}{2\pi} \times \\ &\quad \times e^{i\hat{u}_0 u_0 + \sum_{ac} i\hat{u}_{ac} u_{ac} - \frac{1}{2} \left((\hat{u}_0)^2 \tilde{q} + \sum_{ac,bd} \hat{u}_{ac} \hat{u}_{bd} q_{ac,bd} + 2\hat{u}_0 \sum_{ac} \hat{u}_{ac} m_{ac} \right)} \frac{e^{-\frac{\nu}{2\beta} \sum_{ac} (S^a u_0 - u_{ac})^2}}{\sqrt{2\pi\beta/\nu}}, \end{aligned}$$

where $P(w^*)$ is the factorized prior for the hidden signal (that we assume normal), $P(w_{ac})$ is the factorized distribution of w (that we assume flat) and $D = \{a, b, c, d | (a = b) \wedge (c \geq d)\} \vee \{a, b, c, d | (a > b) \wedge c, d\}$.

In the $N \rightarrow \infty$ limit, Equation (3.14) can be evaluated using a saddle-point approximation over the order parameters. This requires imposing stationarity (extremum) conditions for the overlaps that appear in the free entropy expression. To make the problem tractable, a parametric ansatz for the order parameters is adopted; here, we employ the replica symmetric (RS) ansatz, which assumes symmetry under permutation of the replica indices and significantly reduces the complexity of the extremization. The overlap $q^{ac,bd}$ becomes

$$q^{ac,bd} = \begin{cases} q_2 & \text{if } a = b, c = d \\ q_1 & \text{if } a = b, c > d \\ q_0 & \text{if } a > b, \forall c, d \end{cases}, \quad (3.17)$$

and a replica-index-independent magnetization with the signal m . Additionally, in the zero temperature limit $\rho \rightarrow \infty$ entailed in the MAP of Equation (3.11), we impose the standard rescaling [Obuchi et al., 2018] for the order parameters:

$$\begin{aligned} q_2 &= q_1 + \frac{\delta q}{\rho}; & \hat{q}_1 - \hat{q}_2 &= \frac{\rho}{\nu} \delta \hat{q}; \\ \hat{q}_{1/0} &= \left(\frac{\rho}{\nu}\right)^2 \hat{q}_{1/0}; & \hat{m} &= \frac{\rho}{\nu} \hat{m}. \end{aligned} \quad (3.18)$$

The saddle-point values of $\hat{q}$ and $\tilde{q}$ can be determined independently of the other order parameters, since one can require $\zeta(s=0) = 0$. In this setting, we have $\tilde{q} = \int dw^* P(w^*) (w^*)^2$ and $\hat{q} = 0$, and with a Gaussian prior for the true signal $\tilde{q} = \int Dw^* (w^*)^2 = 1$. Since the saddle point value for $\hat{q} = 0$, we have the following expression for the interaction term in the free entropy:

$$\mathcal{G}_I = \lim_{\beta \rightarrow 0} \lim_{s \rightarrow 0} \frac{G_I}{s} = \frac{1}{2} \left(\delta \hat{q} q_1 - \frac{\hat{q}_1 \delta q}{\nu} - \hat{q}_1 q_1 + \hat{q}_0 q_0 - 2\hat{m} m \right). \quad (3.19)$$

The final expression for the free entropy, after taking the $s \rightarrow 0$ and $\beta \rightarrow 0$ limit is simply $\phi = \mathcal{G}_i + \mathcal{G}_S + \alpha \mathcal{G}_E$, where the entropic and energetic terms read:

$$\mathcal{G}_S = \lim_{\beta \rightarrow 0} \lim_{s \rightarrow 0} \frac{\ln G_S}{s} = -\frac{1}{2} \log \left(\frac{(\delta \hat{q} + \lambda \nu) - (\hat{q}_1 - \hat{q}_0)}{(\delta \hat{q} + \lambda \nu)} \right) + \frac{\hat{q}_0 + \hat{m}^2}{2((\delta \hat{q} + \lambda \nu) - (\hat{q}_1 - \hat{q}_0))}, \quad (3.20)$$

$$\begin{aligned} \mathcal{G}_E = \lim_{\beta \rightarrow 0} \lim_{s \rightarrow 0} \frac{\ln G_E}{s} = & \frac{1}{2} \log \left(\frac{1 + 2\delta q}{1 + 2\delta q + 2\nu(q_1 - q_0)} \right) - \nu \frac{\tilde{q} + q_0}{(1 + 2\delta q) + 2\nu(q_1 - q_0)} \\ & + \int Dz_0 \int Du_0 \log \left[2 \cosh \left(\zeta + 2\nu \frac{\sqrt{\tilde{q}}(mu_0 + \sqrt{q_0 - m^2}z_0)u_0}{(1 + 2\delta q) + 2\nu(q_1 - q_0)} \right) \right], \end{aligned} \quad (3.21)$$

where we denoted the standard Gaussian measure with $Dx = \frac{e^{-x^2/2}}{\sqrt{2\pi}} dx$, as reported in Chapter 1.

It is important to point out that the replica symmetric (RS) assumption, a common simplification in similar analyses [Cui et al., 2021], may fail to capture the full complexity of the system. This occurs if the system transitions to a replica symmetry-broken (RSB) phase. Previous studies [Cui et al., 2021, Barbier et al., 2019] suggest that an RSB ansatz is likely unnecessary near the peak of the complexity curves. However, in more extreme regimes with large values of ν , the system may enter a highly frustrated state where the complexity approaches zero, indicating that the system's phase space is sparsely populated with solutions and the RS assumption might be violated. Pursuing a complete RSB analysis in a large deviation context is technically challenging due to the largely increased computational cost of converging the saddle-point equations. In Appendix B.3, we provide the analytical form of these equations, in the case of a symmetry breaking for the replica index associated with the external optimization problem (associated with the reconstruction of the signs). We, in fact, expect the internal optimization task to be convex and the external one to be non-convex, but a thorough investigation of the RSB effects is left for future work.

3.3.3 Results

We can now seek the stationary values for the order parameters, and then use the inverse Legendre transform to recover the complexity function, Σ , for different settings of $\alpha, \nu, \zeta, \lambda$.

$\lambda = 0$, fixed sign overlap O

We start by exploring the unregularized case, at an intermediate value $\alpha = 1.7$. We recover the full large deviation of the complexity at fixed values of O , allowing us to study the impact of a strategic selection of the signs compared to random sampling. O is fixed implicitly, by finding the conjugate value of the chemical potential ζ .

The left panel of Figure 3.1 illustrates the relationship between the complexity Σ and the mean squared error MSE_y , for different values of the sign overlap, $O \in \{0.2, 0.4, 0.6, 0.8\}$, $\alpha = 1.7$. The curves are spanned by varying the inverse temperature ν , i.e., low values of ν are associated with a more uniform choice of signs, achieving higher complexity and higher MSE_y . As the overlap O increases, the system's MSE_y decreases. Notably, the maximum

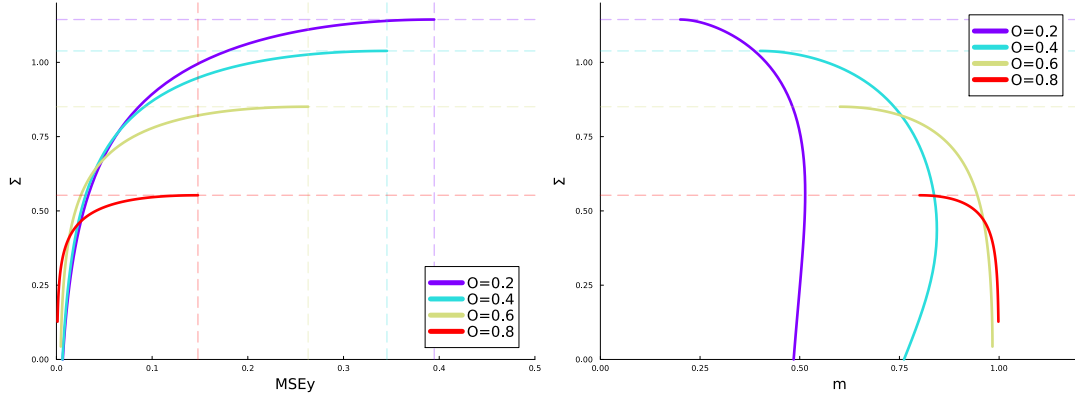


Figure 3.1: *Left panel:* The complexity Σ is represented as a function of the MSE_y . The reported curves have different values of the overlap among sign $O \in \{0.2, 0.4, 0.6, 0.8\}$, and are obtained with null regularization and fixed $\alpha = 1.7$. The horizontal dashed line reports the expected maximum value for the complexity, given by the binomial distribution, while the vertical one outlines the largest MSE_y value. *Right panel:* The complexity Σ is reported as a function of the overlap m , i.e. the number of binary configurations providing an estimate $\mathbf{w}$, having an overlap m with $\mathbf{w}^*$. The parameters are the same used in the left panel. The horizontal dashed line reports the maximum complexity value.

of the entropy curve, at $\nu = 0$, achieves $\Sigma = -\alpha(\frac{1+O}{2} \log \frac{1+O}{2} + \frac{1-O}{2} \log \frac{1-O}{2})$ as one would expect from a completely random selection of the signs.

The right panel of Figure 3.1 displays the achieved $\mathbf{w}$ overlap with the signal, while the sign selections range from random to strategic. Interestingly, at low sign overlap O , the curves are characterized by a non-monotonic behavior of m while Σ decreases, which indicates that the loss is not necessarily minimized when the magnetization is maximum.

$\lambda = 0, \zeta = 0$ (unconstrained O)

In a real optimization scenario, the correct phases are unknown, making it unfeasible to bias toward a correct sign selection with the potential ζ . We thus set $\zeta = 0$ in the following, allowing the overlap O to vary with ν . To assess the recoverability of the hidden signal, one can study the free entropy ϕ as a function of the inverse temperature ν .

At large values of ν and sufficiently small α , two distinct branches of free entropy coexist: the *uninformed* branch, characterized by low magnetization with the true signal, $m \sim 0$, and the *informed* branch, reaching signal recovery $m \rightarrow 1$ in the limit $\nu \rightarrow \infty$. The informed branch appears above the so-called spinodal transition ν_{SP} , where the free entropy develops a maximum in correspondence of the signal, and then becomes the dominant branch at ν_F , where said maximum becomes the global maximum of the free energy.

The analysis implies that a ν -annealing procedure from a random (low magnetization) initialization retrieves the true phases and signal only when α is sufficiently large, since the informed branch merges with the uninformed one in this setting. Instead, at lower values of α , the two branches are detached, and the ν -annealing is bound to get trapped in the uninformed branch, preventing recovery.

The left panel of Figure 3.2 displays the coexisting free entropy branches (full lines for dominant branches, dashed lines for sub-dominant ones), and shows that the merging happens for $\alpha = 1.7$. For low values of α the informed branch becomes dominant for

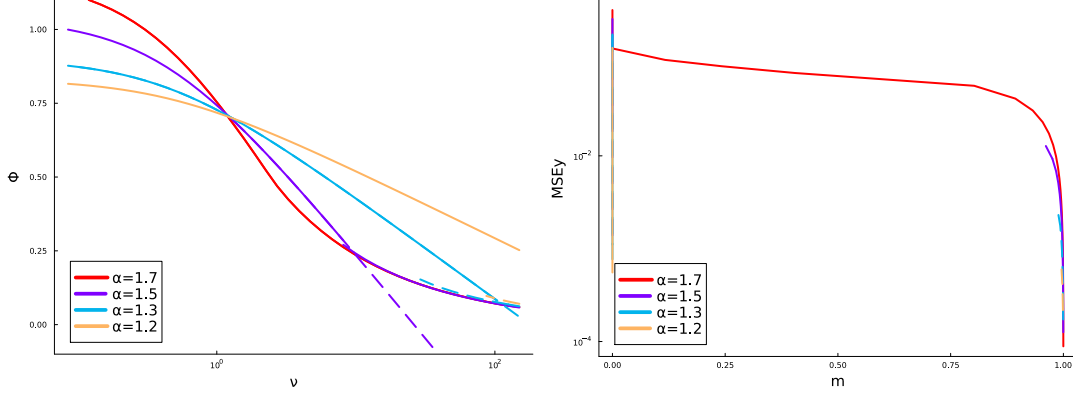


Figure 3.2: *Left panel:* We report the Free Entropy ϕ as a function of ν , for various values of $\alpha \in \{1.2, 1.3, 1.4, 1.5, 1.6, 1.7\}$ and null regularization. The informed branch and the uninformed one are reported. For each value of the inverse temperature ν , the stable branch (the one having higher Free Entropy) is represented with a solid line, while the unstable one is reported with a dashed line. *Right panel:* We report the mean square error on the labels MSE_y as a function of m , for different value of $\alpha \in \{1.2, 1.3, 1.4, 1.5, 1.6, 1.7\}$. The informed and uninformed branches are reported. At fixed inverse temperature, the branch having the highest ϕ is drawn with a solid line, the other with a dashed one.

$\nu \rightarrow \infty$, while for smaller values of the inverse temperature, the dominant branch is the uninformed one. In the right panel of Figure 3.2, one can see the corresponding behavior of the magnetization, plotted as a function of the fitting error MSE_y .

In principle, the merging of the two branches in this analysis should signal a hard-easy algorithmic transition for Phase Selection. However, as we will describe in Section 3.4, we find a discrepancy between the predicted threshold and the algorithmic behavior of a simulated annealing-based solver, which can solve the problem only for higher values $\alpha > 1.7$. We attribute this discrepancy to the presence of Replica symmetry-breaking effects, signaled by the local instability of the RS results at high values of ν . However, we also show that the qualitative insights provided by this incorrect analysis are still informative of the underlying properties of the optimization problem, and lead to the development of a novel efficient heuristic for solving Phase retrieval.

Note also that the predicted threshold $\alpha = 1.7$ is still much larger than the Bayes-Optimal threshold of Phase retrieval, $\alpha_{BO} = 1.12$. To bridge this gap, it is necessary to incorporate regularization, as we discuss in the next paragraphs.

Analysis of the stabilities, impact of ν and λ

We have shown how annealing the inverse temperature ν may be sufficient to recover the hidden signal $\mathbf{w}^*$, provided α is large enough. We now turn to understanding exactly what happens to the signs' selection as ν increases, and whether switching on the regularization λ can help to achieve a lower recovery threshold. Some insight about their impact can be derived from the analysis of the so-called *stabilities*:

$$s^\mu = \frac{\mathbf{w} \cdot \mathbf{x}^\mu}{\sqrt{N}},$$

representing the measurement amplitudes obtained with the signal estimate $\mathbf{w}$. In particular, for each stability, the inferred model also estimates a sign S^μ , which needs to be compared to the missing phase produced by the true signal $\mathbf{w}^*$. To highlight the correlation between the amplitude of the stabilities and the correctness of the sign choices, as a function of ν and λ , we derive the analytical expression for the distribution of the stabilities $P(s)$, and the distribution of the stabilities conditioned to a correct sign choice $P_*(s)$:

$$P(s) = \int Dz_0 \int Du_0 \frac{\sum_{S \in \{-1,1\}} (1+2\delta q) e^{-\frac{z_+^*(s, \delta q, q_1, q_0, m, S)^2}{2}} e^{-S\zeta} e^{-\frac{(z_+^*(s, \delta q, q_1, q_0, m, S) + u_0)^2}{1+2\delta q}}}{\sqrt{2\pi(q_1 - q_0)} e^{-\nu \frac{(\sqrt{q_0 - m^2} z_0 + m u_0)^2 + u_0^2}{(1+2\delta q) + 2\nu(q_1 - q_0)}} 2 \cosh \left(\zeta + 2\nu \frac{(\sqrt{q_0 - m^2} z_0 + m u_0) u_0}{(1+2\delta q) + 2\nu(q_1 - q_0)} \right)}, \quad (3.22)$$

$$P_*(s) = \int Dz_0 \int Du_0 \frac{(1+2\delta q) e^{-\frac{z_+^*(s, \delta q, q_1, q_0, m, 1)^2}{2}} e^{-\zeta} e^{-\frac{(z_+^*(s, \delta q, q_1, q_0, m, 1) + u_0)^2}{1+2\delta q}}}{\sqrt{2\pi(q_1 - q_0)} e^{-\nu \frac{(\sqrt{q_0 - m^2} z_0 + m u_0)^2 + u_0^2}{(1+2\delta q) + 2\nu(q_1 - q_0)}} 2 \cosh \left(\zeta + 2\nu \frac{(\sqrt{q_0 - m^2} z_0 + m u_0) u_0}{(1+2\delta q) + 2\nu(q_1 - q_0)} \right)}. \quad (3.23)$$

where $z_+^*(s, \delta q, q_1, q_0, m, S) = s \frac{1+2\delta q}{\sqrt{q_1 - q_0}} - \frac{m u_0 + \sqrt{q_0 - m^2} z_0 - 2S \delta q u_0}{\sqrt{q_1 - q_0}}$. The expression of $P_*(s)$ can be obtained from Equation (3.22) by fixing $S = 1$. Details on the derivation of Equation (3.22) are reported in Appendix B.2.

In Figure 3.3, we display the distributions obtained with $\alpha = 1.7$ and $O = 0.5$, and different values of ν and λ . In particular, we show two perturbations of the same initial setting (left panel): increased ν (center), and increased λ (right). Notably, the effect of increasing the inverse temperature mirrors that of increasing the regularization: both changes focus the mass of $P_*(s)$ on stabilities with large magnitude, inducing a bimodal distribution, while the overall distribution $P(s)$ squeezes around $s = 0$. An intuitive explanation of this observation is that a wrong sign of a large magnitude stability has a larger impact on the signal reconstruction, as can be deduced from Equation (3.4), and on the MSE_y , so correcting this type of errors first is the most efficient way of optimizing the loss.

The role of λ in signal recovery

From the results of Section 3.3.3, it is clear that the introduction of regularization can be beneficial since it has some similar effects to annealing ν . Driven by this finding, we repeat the analysis of the ν -annealing in the presence of a fixed $\lambda > 0$.

The top left panel of Figure 3.4 shows again the MSE_y as a function of magnetization m while ν is increased, this time with a regularization $\lambda = 0.005$. By comparing the result with Figure 3.2, it is evident that the merging of the branches happens above a smaller alpha threshold, allowing a random initialization to reach high magnetization for large values of ν . However, as the inverse temperature increases, the magnetization m initially rises, reaches a peak, and can also decrease. This non-monotonic behavior is attributed to the biasing

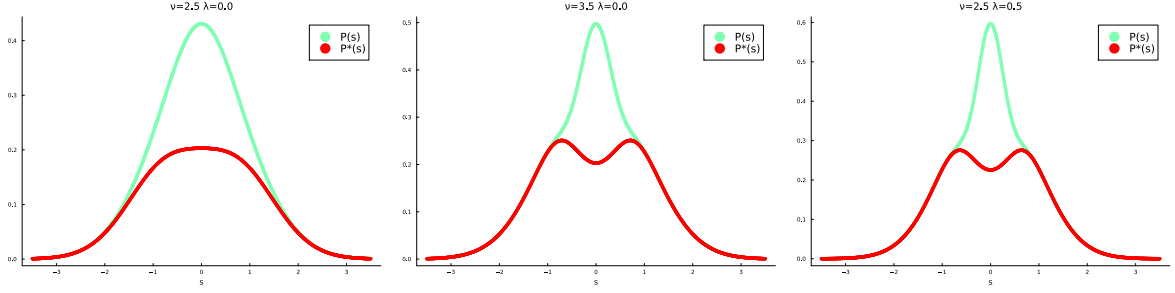


Figure 3.3: We present the histogram of stabilities for $O = 0.5$, $\alpha = 1.7$, for three possible combinations of λ and μ values. *Left panel:* The inverse temperature is fixed to $\nu = 2.5$ and the regularization parameter to $\lambda = 0.0$. *Central panel:* The inverse temperature is $\nu = 3.5$ and the regularization parameter is $\lambda = 0.0$. *Right panel:* The inverse temperature is $\nu = 2.5$ and the regularization parameter is $\lambda = 0.5$.

effect of regularization, which constrains the norm of the signal. A non-zero λ prevents $\mathbf{w}$ from adequately aligning with $\mathbf{w}^*$, forcing $\mathbf{w}$ to lose the direction of the hidden signal. For $\alpha = 1.3$, we observe a range of m in which both branches are unstable. This phenomenon is associated with the behavior of $\phi(\nu)$: as ν increases beyond $\nu_{Spinodal}$, the uninformed branch initially remains subdominant but eventually merges with the informed branch.

In the top right panel of Figure 3.4, we show the phase diagram of the problem and the effect of L_2 regularization. In particular, we identify the hard algorithmic region where the two free entropy branches coexist and are detached, characterized by distinct values of ν for the spinodal and the first-order transitions. This hard window shrinks in the presence of regularization, which means that a $\lambda > 0$ can be useful to escape the uninformed initialization trap.

However, the bottom left and bottom right panels of Figure 3.4 show that the regularization has a dual effect. On the one hand, on the left, we show that increasing λ can anticipate the temperature at which the estimator starts to build up an overlap with the signal (e.g., in the figure we show the value of ν at which m reaches 0.1). On the other hand, the right panel shows that excessive regularization can impair complete signal reconstruction since the low norm bias will also reduce the maximum achieved overlap between $\mathbf{w}$ and $\mathbf{w}^*$. Note that $\mathbf{w}^*$ is the global minimum of the Phase Selection loss only for $\lambda = 0$.

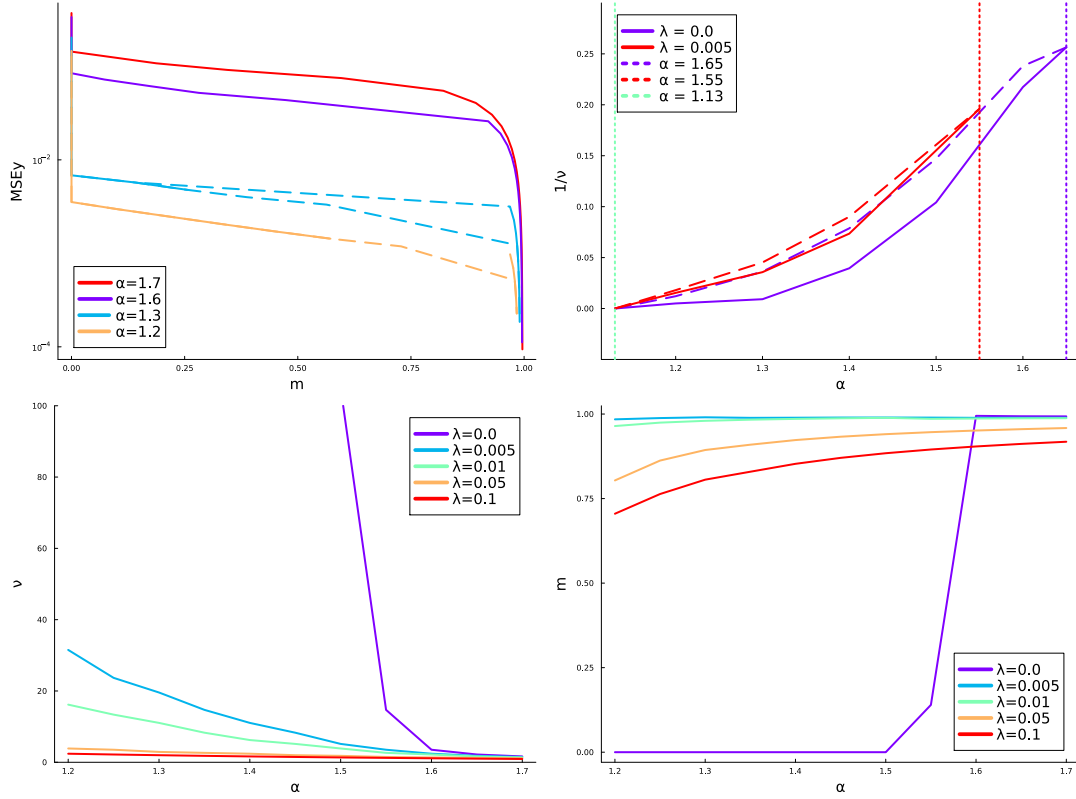


Figure 3.4: *Top-left panel:* We present the mean square error on the labels MSE_y vs m for $\alpha \in \{1.2, 1.3, 1.4, 1.5, 1.6, 1.7\}$, at non-null regularization. Both the informed and the uninformed branches are reported. The dashed lines are associated with the branch having lower free entropy at a specific value of ν . *Top-right panel:* The phase diagram is reported: for different values of α , the temperature at which first-order transition (the dashed line) and spinodal transitions (the solid line) occur, for fixed λ . The green vertical line represents the information-theoretic threshold for the Phase Selection problem, and the purple and red vertical lines represent the solvability threshold for the different values of λ . *Bottom-left panel:* the plot presents the inverse temperature at which the uninformed branch reaches $m = 0.1$ as a function of α , for different values of the regularization parameter. The maximum ν considered is fixed to 100. *Bottom-right panel:* the plot reports the maximum m obtained in the dynamics as a function of α , for different values of λ , for the uninformed branch.

Simultaneous annealing of ν and λ

The shown results suggest that, while the regularization can be beneficial at low inverse temperatures since it anticipates the alignment with the true signal, it can hinder the recovery process at larger values of ν , where it can prevent full recovery. This suggests the implementation of a simultaneous annealing procedure, where the product $\nu\lambda$ is kept fixed as $\nu \rightarrow \infty$ and $\lambda \rightarrow 0$.

The effect of the simultaneous annealing at constant $\nu\lambda$ is presented in the left panel of Figure 3.5. Compared to Figure 3.2 and Figure 3.4, an early strong regularization allows the optimization to flow toward the perfect retrieval solution even at very low values of alpha $\alpha \sim \alpha_{BO}$. The right panel of Figure 3.5 shows that it is crucial to keep the product $\nu \cdot \lambda$ small enough to ensure an appropriate value of λ for each ν . If λ becomes too large at high

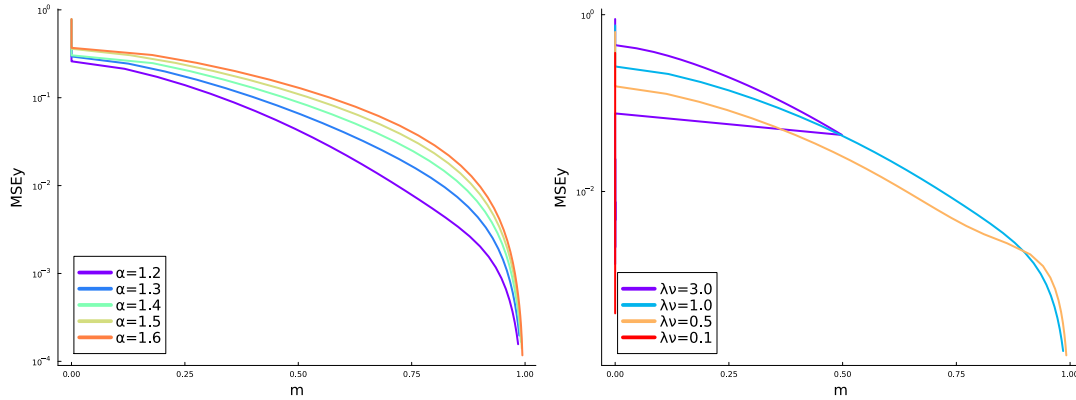


Figure 3.5: In the *left panel*, we report the mean squared error MSE_y as a function of the magnetization, for the uninformed branch. We fixed the product $\lambda \cdot \nu = 1.0$ and we consider different values of $\alpha \in \{1.2, 1.3, 1.4, 1.5, 1.6, 1.7\}$. In the *right panel*, we report the same quantities, for a fixed $\alpha = 1.2$, varying the product value $\lambda \cdot \nu \in \{0.1, 0.5, 1.0, 3.0\}$.

ν , the bias effect introduced by the regularization becomes evident. At the same time, if λ is too small, the optimization may not be able to escape the uninformed branch, leading to a failure in signal recovery, for α small enough.

3.4 Numerical Results

The presented replica analysis suggests that partitioning the Phase retrieval problem into two nested optimization levels can benefit optimization. In practice, the Phase selection formulation is also convenient since the internal problem, the optimization of $\mathbf{w}$ given a choice for $\mathbf{S}$, is convex and the solution can be obtained in closed form, Equation (3.4). The external level, the sign selection problem, is instead combinatorial and can be tackled with standard heuristic solvers, such as Simulated annealing [Kirkpatrick et al., 1983] and Approximate message passing [Mezard and Montanari, 2009]. In this study, we also consider Langevin dynamics on the continuous relaxation of the signs, which is defined below.

Let us first analyze the behavior of the solvers when the regularization parameter is kept constant. In Section 3.3.3, we observed that, for example, at $\alpha = 1.3$ there are ranges of m where both free energy branches are subdominant, making it impossible for a randomly initialized configuration to reconstruct the signal. Indeed, in this regime, Simulated Annealing (SA) fails to reconstruct $\mathbf{w}^*$. However, the recovery threshold of the algorithm is found to be located at values of λ and α higher than the ones predicted by the merging of informed and uninformed branches in the RS analysis, likely due to the onset of Replica Symmetry Breaking. In Appendix B.3, we report the result of the replica analysis under a 1RSB assumption. Exploring the full large deviation analysis in this more complex case is however, numerically challenging, and is left for future work.

In the following sections, we explore the retrieval performance of the three algorithms when the simultaneous annealing procedure of ν and λ , described in the previous section, is implemented. The pseudo-code for the problem instance generator is reported in the following Section.

3.4.1 Problem generator

In order to deal with numerical investigation, a Phase Selection instances generator has been implemented. It follows in Algorithm 5.

Algorithm 5 Phase Selection Instances Generator

Input: Structure parameters: $N, \alpha, \Delta_0, P_0(\cdot)$

Initialization: $M \leftarrow \alpha N$

Sample the hidden signal $\mathbf{w}^* \sim P_0(\cdot)$

Construct the instances: generate $X, \mathbf{y}$ and test set

$$\begin{aligned} X &\sim \frac{1}{\sqrt{N}} \mathcal{N}_{N \times M}(\mathbf{0}, I) \\ \mathbf{v} &\sim \mathcal{N}_M(\mathbf{0}, I) \\ \mathbf{y} &= |X \cdot \mathbf{w}^* + \sqrt{\Delta_0} \cdot \mathbf{v}| \end{aligned}$$

Return: $X, \mathbf{y}, \mathbf{w}^*$

3.4.2 Simulated Annealing

Simulated Annealing (SA) is a heuristic solver that leverages the Markov Chain Monte Carlo (MCMC) method and a temperature annealing process to control the exploration of the configuration space [Kirkpatrick et al., 1983]. Note that the application of SA to Phase selection is closely inspired by [Obuchi et al., 2018], where a different variable selection problem was considered. The pseudo-code for the SA-based solver is outlined in Algorithm 6. We consider a multiplicative annealing of the inverse temperature:

$$\nu_t = \nu_{t-1}(1 + \epsilon),$$

where ϵ is a hyper-parameter controlling the annealing rate. At each temperature, we perform T_A Monte Carlo steps, and then proceed in ν annealing. It is theoretically established that a “slow enough” cooling schedule should allow SA to focus on the global minima of the loss, thereby leading to signal recovery in the retrieval phase of our model. Simultaneously, we consider an exponential decay of the regularization parameter λ , from an initial value λ_0 to 0, keeping the product $\nu\lambda$ constant.

Note that performing the annealing only after T_A steps of MCMC is convenient since it reduces the computational cost of the evaluation of the closed-form optimizer $\hat{\mathbf{w}}$ at each iteration. One can rewrite:

$$\hat{\mathbf{w}} = \left(XX^T + \lambda I_N \right)^{-1} X^T (\mathbf{S} \odot \mathbf{y}) = M_{ols}(\mathbf{S} \odot \mathbf{y}), \quad (3.24)$$

where M_{ols} needs an update –with a $\mathcal{O}(M^3)$ cost because of the inversion– only when the regularization intensity changes. Note that the matrix multiplication in the right term of Equation (3.24) has a $\mathcal{O}(M^2)$ cost. Moreover, the impact of the loss of a change of the ν -th sign can be evaluated efficiently after rewriting:

$$MSE_y = \frac{1}{M} \|(\mathbb{I} - X^T M_{ols})(\mathbf{S} \odot \mathbf{y})\|_2^2 = \frac{1}{M} \|\Delta_{\mathbf{w}}\|_2^2, \quad (3.25)$$

and by considering the shift to the vector of errors Δ_w induced by the sign flip. The matrix $(\mathbb{I} - X^T M_{ols})$ can be computed only when the regularization value is updated.

Algorithm 6 Simulated Annealing-Based Solver

Input: Phase selection instance $(X, \mathbf{y})$ and the parameters $\epsilon, \nu_0, \lambda_0, T_A$

Initialization: Sample $\mathbf{S} \sim \text{Uniform}\{-1, 1\}^N$, set $\nu \leftarrow \nu_0, \lambda \leftarrow \lambda_0, t \leftarrow 1$

and $M_{ols} \leftarrow (XX^T + \lambda I_N)^{-1} X$ $X_{ols} \leftarrow X^T M_{ols}$ $\Delta_y \leftarrow (X_{ols} - I_M)(\mathbf{S} \odot \mathbf{y})$

while $t \leq T$ **do**

 Select $m \sim \text{Uniform}\{1, \dots, M\}$

$\delta \leftarrow -2S_m y_m (X_{ols} - I_M)_m$

$\Delta_E \leftarrow \|\Delta_y + \delta\|_2 - \|\Delta_y\|_2$

 Accept the move with probability $\exp(-\nu \Delta_E)$

if $t \bmod T_A = 0$ **then**

 Update $\nu \leftarrow \nu(1 + \epsilon)$ $\lambda \leftarrow \frac{\lambda_0 \nu_0}{\nu}$

 Update $M_{ols} \leftarrow (XX^T + \lambda I_N)^{-1} X$ $X_{ols} \leftarrow X^T M_{ols}$ $\Delta_y \leftarrow (X_{ols} - I_M)(\mathbf{S} \odot \mathbf{y})$

end if

$t \leftarrow t + 1$

end while

Return: $M_{ols}(\mathbf{S} \odot \mathbf{y})$

In Figure 3.6, we report the success rate of the SA solver for two values of the annealing rate parameter ϵ , and initial values $\nu_0 = 1$ and $\lambda_0 = 1$. As expected, the slower annealing induces a better exploration of the configuration space, leading to a higher recovery probability close to the α_{BO} threshold. Notice that the algorithmic threshold α_{SA} , here defined as the dataset size α at which the success probability is one-half, seems to approach the Bayes optimal algorithmic threshold $\alpha_{BO} = 1.13$ as larger instances and slower annealing schedules are considered.

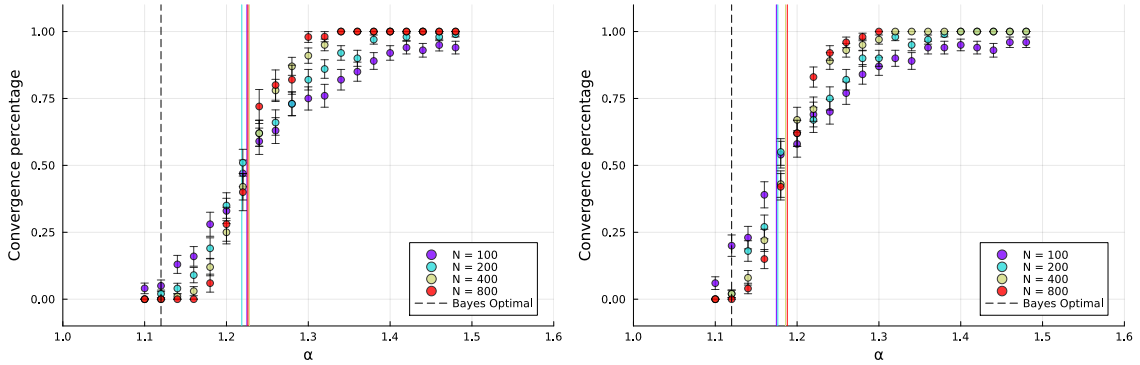


Figure 3.6: The SA-based solver is employed for solving 100 Phase Selection instances, for $T = 10^6, T_A = 100, \nu_0 = 1.0, \lambda_0 = 1.0$. After T steps the overlap m , between $\mathbf{w}^*$ and $\hat{\mathbf{w}}$, is computed as well as the MSE_y , for different values of task complexity, $\alpha \in \text{LinRange}(1.1, 1.475, 0.025)$, and different values of the system size $N \in \{100, 200, 400, 800\}$. We report the percentage of processes able to fully recover $\mathbf{w}^*$, i.e., $m \sim 1$ and $MSE_y = 0$; and a vertical line representing the α value at which we can solve half of the instances. The annealing schedules are controlled by $\epsilon = 0.1$ (left) and $\epsilon = 0.01$ (right). The error bars represent the standard deviation of the success rate across the 100 instances.

3.4.3 Approximate Message Passing based solver

Under the assumptions of our model setting, the *a posteriori* mean and variance of the estimator $\mathbf{w}$, for a given value of ν and λ , can be estimated via the Approximate Message Passing [Mezard and Montanari, 2009] approach. The exact form of the AMP equations for Phase selection, which can be derived by closing on first and second moments the associated relaxed Belief Propagation equations, is specified by the input and output channels of the problem (see Chapter 1):

$$\phi_{in}(\Sigma_0, \Sigma_1, R; \lambda) = \frac{1}{2} \log \left(\frac{\Sigma_1 + \lambda}{\Sigma_1 + \lambda - \Sigma_0} \right) + \frac{R^2}{2(\Sigma_1 + \lambda - \Sigma_0)}, \quad (3.26)$$

$$\phi_{out}(V_0, V_1, \omega; y, \nu) = \frac{1}{2} \log \left(\frac{1 + 2V_1}{1 + 2V_1 + 2\nu V_0} \right) + \log \left(e^{-\nu \frac{(\omega+y)^2}{1+2V_1+2\nu V_0}} + e^{-\nu \frac{(\omega-y)^2}{1+2V_1+2\nu V_0}} \right). \quad (3.27)$$

which are closely related to the entropic and energetic terms in the replica analysis Equation (3.20) and Equation (3.21) [Antenucci et al., 2019b]. The full algorithm is detailed in Algorithm 7. It is known that the State Evolution equations, offering a deterministic low-dimensional description of the trajectories of AMP, are in one-to-one correspondence with the saddle point iterations employed to extremize the replica symmetric free energy [Rangan, 2011].

Note that, given the two-level structure of Phase selection, the AMP equations require the estimation of two second moments, yielding a set of recursive equations reminiscent of the Approximate Survey Propagation equations, a 1RSB version of AMP [Antenucci et al., 2019b, Saglietti et al., 2020]. This similarity is directly related to the fact that, in Section 3.3, the replica trick was applied both to the external and the internal optimization levels. It is also interesting to notice that the optimization level associated with the sign selection is not immediately evident from the equations since the configuration $\mathbf{S}$ is traced out in the output channel of Equation (3.27). At large values of ν , only the largest term inside the logarithm gives a finite contribution, effectively corresponding to polarization on a specific sign choice. Moreover, because of the Gaussian nature of the inputs, in the AMP approach, we are not required to use the closed-form solution of the internal problem, avoiding the costly matrix inversion.

Here, we aim to turn AMP into a Phase selection solver by introducing our simultaneous annealing procedure on ν and λ during the message-passing iterations, forcing the measure to focus on a single configuration. This procedure should approximate state-following as long as the annealing is not too abrupt. Again, we use a hyperparameter ϵ to control the ν -annealing rate. The full heuristic is detailed in Algorithm 7. After convergence, one can use the fixed-point signal estimate to compute the estimated labels as well as the Phase Selection loss.

It's important to highlight that, if the training inputs were not i.i.d. Gaussian distributed, as is often the case in practical inference tasks, the development of correlations could undermine the reliability of the Approximate Message Passing (AMP) algorithm, potentially leading to issues with convergence or inaccurate loss estimates. As already stated in Chapter 1. In the case of a generalized covariance, the vector-AMP approach [Rangan et al., 2017] should correct for the inaccurate assumptions of our implementation, but its derivation and the associated analysis are beyond the scope of this chapter.

Algorithm 7 Approximate Message Passing-Based Solver

Input: Phase selection instance $(X, \mathbf{y})$ and the parameters $\nu_0, \lambda_0, \alpha, \epsilon, \psi$
Initialize: $\mathbf{a}^{(0)}, \mathbf{v}_0^{(0)}, \mathbf{v}_1^{(0)}, \phi_{out}, \phi_{in}, \Sigma_0^{(0)}, \Sigma_1^{(0)}, \mathbf{R}^{(0)}, \mathbf{V}_0^{(0)}, \mathbf{V}_1^{(0)}, \omega^{(0)}$ $\nu \leftarrow \nu_0, \lambda \leftarrow \lambda_0, \nu_A \leftarrow 1 + \epsilon$ and $t \leftarrow 1$
while not converged on $\mathbf{a}^{(t)}, \mathbf{v}_0^{(t)}, \mathbf{v}_1^{(t)}$ **do**
 Update $\omega_\mu^{(t)}$ and $V_{0,\mu}^{(t)}, V_{1,\mu}^{(t)}$:

$$V_{0,\mu}^{(t)} = \sum_i (x_i^\mu)^2 v_{0,i}^{(t-1)} (1 - \psi) + V_{0,\mu}^{(t-1)} \psi$$

$$V_{1,\mu}^{(t)} = \sum_i (x_i^\mu)^2 v_{1,i}^{(t-1)} (1 - \psi) + V_{1,\mu}^{(t-1)} \psi$$

$$\omega_\mu^{(t)} = \sum_i x_i^\mu w_i^{(t-1)} - \left(\sum_i (x_i^\mu)^2 v_{0,i}^{(t-1)} + \frac{1}{\nu} \sum_i (x_i^\mu)^2 v_{1,i}^{(t-1)} \right) (1 - \psi) + \omega_\mu^{(t-1)} \psi$$

Update $g_{out,\mu}^{(t)}, \Sigma_0^{(t)}, \Sigma_1^{(t)}$ and $R_i^{(t)}$:

$$g_{out,\mu}^{(t)} = \partial_{\omega_\mu^{(t)}} \phi_{out} \left(\omega_\mu^{(t)}, y_\mu, V_{0,\mu}^{(t)}, V_{1,\mu}^{(t)}, \nu \right)$$

$$\Sigma_0^{(t)} = 2\nu \sum_\mu (x_i^\mu)^2 \partial_{V_{1,\mu}^{(t)}} \phi_{out} \left(\omega_\mu^{(t)}, y_\mu, V_{0,\mu}^{(t)}, V_{1,\mu}^{(t)}, \nu \right) - \left(g_{out,\mu}^{(t-1)} \right)^2$$

$$\Sigma_1^{(t)} = \sum_\mu (x_i^\mu)^2 \partial_{\omega_\mu^{(t)}}^2 \phi_{out} \left(\omega_\mu^{(t)}, y_\mu, V_{0,\mu}^{(t)}, V_{1,\mu}^{(t)}, \nu \right) + 2\nu \partial_{V_{1,\mu}^{(t)}} \phi_{out} \left(\omega_\mu^{(t)}, y_\mu, V_{0,\mu}^{(t)}, V_{1,\mu}^{(t)}, \nu \right) - \left(g_{out,\mu}^{(t-1)} \right)^2$$

$$R_i^{(t)} = \sum_\mu x_i^\mu g_{out,\mu}^{(t)} - a_i^{(t-1)} (\Sigma_0^{(t)} - \Sigma_1^{(t)})$$

Update $a_i^{(t)}, v_{0,i}^{(t)}, v_{1,i}^{(t)}$:

$$a_i^{(t)} = \partial_{R_i^{(t)}} \phi_{in} \left(R_i^{(t)}, \Sigma_0^{(t)}, \Sigma_1^{(t)}, \lambda \right)$$

$$v_{0,i}^{(t)} = -2 \partial_{\Sigma_1^{(t)}} \phi_{in} \left(R_i^{(t)}, \Sigma_0^{(t)}, \Sigma_1^{(t)}, \lambda \right) - \left(a_i^{(t)} \right)^2$$

$$v_{1,i}^{(t)} = \partial_{R_i^{(t)}}^2 \phi_{in} \left(R_i^{(t)}, \Sigma_0^{(t)}, \Sigma_1^{(t)}, \lambda \right) - v_{0,i}^{(t)}$$

Update annealing parameters:

$$\nu \leftarrow \nu \cdot \nu_A, \quad \lambda \leftarrow \frac{\lambda_0 \nu_0}{\nu}$$

$t \leftarrow t + 1$

end while

Return: Converged marginal mean and variance $\mathbf{a}^{(t)}, \mathbf{v}_0^{(t)}$ and $\mathbf{v}_1^{(t)}$.

Figure 3.7 displays the recovery probability for two values of the annealing rate parameter $\epsilon \in \{0.1, 0.01\}$. Also in this case, the heuristic seems to approach the α_{BO} threshold in the limit of vanishing annealing rate.

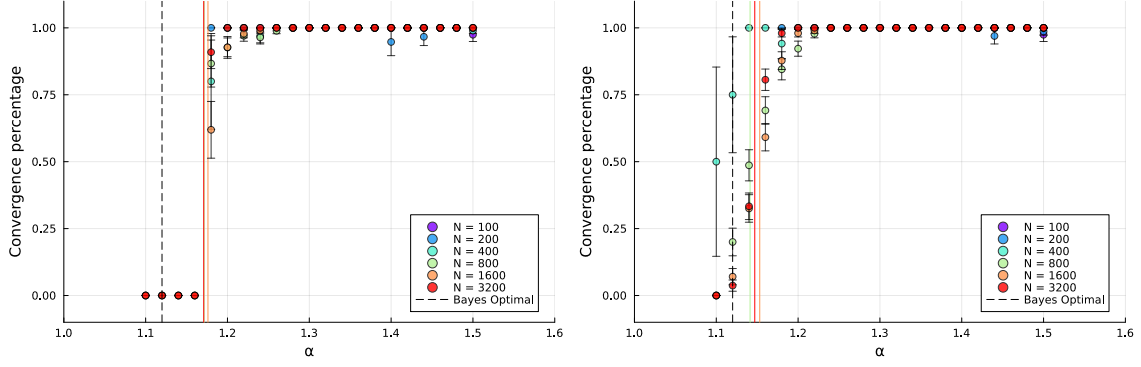


Figure 3.7: The AMP-based solver is employed for on 100 Phase Selection instances, for $T = 2 \cdot 10^5$, $\nu_0 = 1000.0$, $\lambda_0 = 1.0$. After T steps the overlap m , between $\mathbf{w}^*$ and $\hat{\mathbf{w}}$, is computed as well as the MSE_y , for different values of task complexity, $\alpha \in \text{LinRange}(1.1, 1.475, 0.025)$, and different values of the system size $N \in \{100, 200, 400, 800, 1600, 3200\}$. We report the percentage of runs able to fully recover $\mathbf{w}^*$, i.e. $m \sim 1$ and $MSE_y = 0$; and a vertical line representing the α value at which we can solve half of the instances. The annealing rate is controlled respectively by $\epsilon = 0.1$ (left) and $\epsilon = 0.01$ (right). The error bars represent the standard deviation of the success rate across the 100 instances.

3.4.4 Langevin dynamics based solver

Finally, we explore a more heuristic approach, relaxing the combinatorial problem of the sign selection into a continuous optimization problem. We replace the binary variables $\mathbf{S}$ with a continuous vector $\boldsymbol{\theta} \in \mathbb{R}^M$. The continuous optimization of $\boldsymbol{\theta}$ can be performed via Langevin dynamics over the loss:

$$\mathcal{L}(\boldsymbol{\theta}; X_{\text{ols}}, \mathbf{y}, \lambda, N) = \|\hat{\mathbf{y}} - X_{\text{ols}} \cdot \hat{\mathbf{y}}\|_2^2, \quad (3.28)$$

where $\hat{\mathbf{y}} = \boldsymbol{\theta} \odot |\mathbf{y}|$ and $X_{\text{ols}} = X^T M_{\text{ols}}$, with M_{ols} defined as in Equation (3.24). Note that, during the optimization of the continuous signs, it is useful to bound their values inside the hypercube, by rescaling the gradient and clipping the values of $\boldsymbol{\theta}$, $\theta_i \in [-1 + \delta, 1 - \delta]$, with δ small. Similarly to the SA-based solver, we consider a multiplicative annealing in the inverse temperature ν , affecting the magnitude of the Langevin Dynamics noise. The product $\lambda \cdot \nu$ is kept constant and updated once every T_A , together with M_{ols} . The pseudo-code of the Langevin dynamics-based solver is reported in Algorithm 8.

Algorithm 8 Langevin Dynamics-Based Solver**Input:** Phase selection instance $(X, \mathbf{y})$ and parameters $\epsilon, \nu_0, \lambda_0, \eta, T, T_A, \delta$ **Initialization:** Sample $\boldsymbol{\theta} \sim \text{Uniform}(-1, 1)^N$ and set $\nu \leftarrow \nu_0, \lambda \leftarrow \lambda_0, \nu_A \leftarrow 1 + \epsilon, t \leftarrow 0$ **while** $t \leq T$ **do** Sample $\mathbf{z} \sim \mathcal{N}(0, I_N)$

Compute noisy gradient:

$$g \leftarrow \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, X_{\text{ols}}, \mathbf{y}, \lambda, N) + \sqrt{\frac{2}{\nu\eta}} \mathbf{z}$$

Rescale gradient:

$$g \leftarrow g \odot (1 - \boldsymbol{\theta}^2)$$

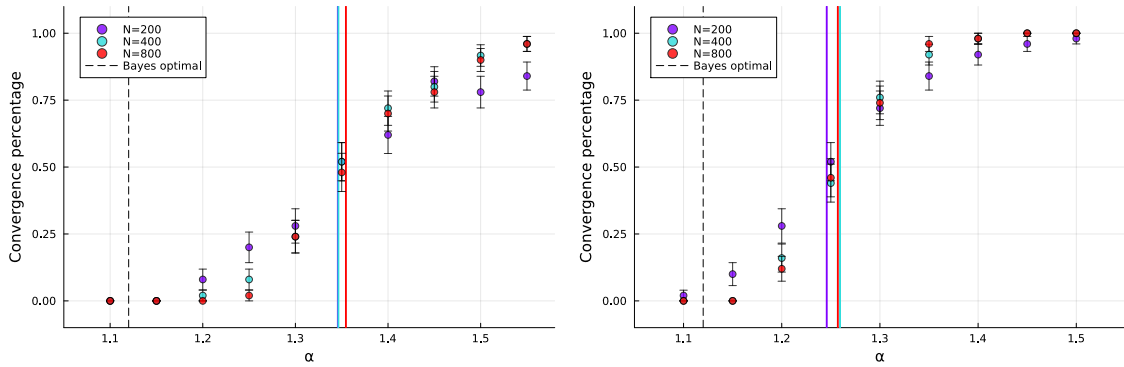
 Update $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta g$ Project $\boldsymbol{\theta} \in [-1 + \delta, 1 - \delta]^N$ **if** $t \bmod T_A = 0$ **then** Update $\nu \leftarrow \nu \cdot \nu_A \quad \lambda \leftarrow \frac{\lambda_0 \nu_0}{\nu}$ $O_{\text{ols}} \leftarrow (XX^T + \lambda I_N)^{-1} x \quad X_{\text{ols}} \leftarrow X^T O_{\text{ols}} \quad \Delta_{\boldsymbol{\theta}} \leftarrow (X_{\text{ols}} - I_M)(\boldsymbol{\theta} \odot \mathbf{y})$ **end if** $t \leftarrow t + 1$ **end while****Return:** $O_{\text{ols}}(\boldsymbol{\theta} \odot \mathbf{y})$ 

Figure 3.8: The GD-based solver is employed for solving 100 Phase Selection instances, for $T = 10^6, T_A = 100, \nu_0 = 1.0, \lambda_0 = 0.75, \delta = 10^{-5}$ and $\eta = 1.0$. After T steps the overlap m , between $\mathbf{w}^*$ and $\hat{\mathbf{w}}$, is computed as well as the MSE_y , for different values of task complexity, $\alpha \in \text{LinRange}(1.1, 1.475, 0.025)$, and different values of the system size $N \in \{200, 400, 800\}$. We report the percentage of processes able to fully recover $\mathbf{w}^*$, i.e., $m \sim 1$ and $MSE_y = 0$; the vertical lines represent the α values at which half of the runs achieve recovery. The annealing schedules are $\epsilon = 0.1$ (left) and $\epsilon = 0.01$ (right). The error bars represent the standard deviation of the success rate across the 100 instances.

Figure 3.8 shows the empirical probability of solving a phase selection instance with the Langevin dynamics approach. Once again, the effect of the annealing rate ϵ is evident: slower schedules lead to better results. By comparing Figure 3.8 with Figure 3.6 and Figure

3.7, it is clear that the retrieval performance of the Langevin dynamics-based scheme is the poorest among the considered solvers, due to the heuristic nature of the continuous relaxation strategy probably limits the performance. However, note that this approach is still able to achieve recovery with a gradient-based approach in regimes where typically gradient descent remains stuck in local minima orthogonal to the signal.

3.5 Recap of the results

In this Chapter:

- We have introduced and analyzed the *Phase Selection* formulation as a means to disentangle the combinatorial and convex continuous components of the Phase Retrieval problem. By separating the hard combinatorial subproblem (retrieving the missing measurement signs) from the convex regression task.
- We have derived a statistical physics analysis of the Phase Selection framework, in the infinite dimensional limit and proportional regime $M = \alpha N$, with $N \rightarrow \infty$, using the replica method. This analysis has been performed for both the regularized and the unregularized case. Moreover, our findings provide a refined explanation for the effectiveness of regularization in phase retrieval, an observation previously noted in the literature [Ma et al., 2018, Sagnet et al., 2020], but here linked more explicitly to the landscape and the bias toward fitting measurements with larger magnitudes.
- Using the replica analysis, the role of the regularization has been clarified: the presence of an L_2 regularization is beneficial in reconstructing the signal, however, it biases the estimated one. To avoid this bias, we have introduced an adaptive regularization scheme, which is able to accelerate convergence by biasing the optimization towards informative solutions at early stages while avoiding systematic errors at later stages.
- Driven by the theoretical findings, we have proposed three heuristic optimization methods to solve the Phase Selection problem: Simulated Annealing, Approximate Message Passing, and a continuous sign Langevin Dynamics. These methods are effective in solving the Phase Selection instances, although, they show different performances. For unstructured data AMP based solver is the fastest and the one approaching closely the BO threshold. However, SA-based or Langevin dynamics based solvers are more suitable for dealing with structured data.

Chapter 4

On the Spectral Analysis of Neural Network Losses

In Chapter 2, we explored the problem of sampling solutions to constraint satisfaction problems (CSPs). By introducing an error-counting loss function, it has been observed that for certain parameter choices (α and κ), the resulting loss landscape becomes non-convex. Similarly, in Chapter 3, we addressed the challenge of reconstructing a signal from linear measurements without phase information, where employing a squared loss in the optimization procedure induced a rugged loss landscape. This effect is generated by the absence of phase information.

In this chapter, rather than focusing on solving specific inference or machine learning tasks characterized by rough landscapes, we aim to develop a theoretical framework for studying the geometry of loss landscapes in simple neural network architectures.

Understanding the geometric structure of loss landscapes is crucial. Modern machine learning models often have several parameters that far exceed the size of the dataset, posing significant challenges for learning. In such high-dimensional settings, optimization becomes computationally demanding. Gradient Descent (GD), the standard method for model training, requires full-batch gradient evaluations and careful step-size tuning, which can be prohibitively expensive as the model size increases. These challenges have motivated the development of more efficient alternatives [Hastie et al., 2009, Bishop, 2006, Goodfellow et al., 2016, Bottou, 2010].

Stochastic Gradient Descent (SGD) has emerged as a key innovation to reduce computational costs [Bishop, 2006, Goodfellow et al., 2016]. By estimating the gradient using random minibatches, SGD accelerates convergence and improves scalability. However, this raises important questions: do GD and SGD converge to similar solutions, or do they explore different regions of the loss landscape? How does batch size influence training dynamics and generalization? Addressing these questions requires tools that go beyond first-order information (i.e., the gradient) and instead probe the geometric properties of the landscape.

A powerful approach to investigating the geometry of the loss landscape is through the loss function’s Hessian matrix, which encodes local curvature and provides valuable second-order information about the structure of the loss surface. Second-order information can be used to analyze the convergence properties of gradient-based methods. For example, it is known that if the stationary points of the loss function are isolated and the Hessian is non-degenerate, then gradient-based methods converge to minimizers [Sagun et al., 2017]. However, real-world loss landscapes in deep learning are highly non-convex and typically

characterized by a multitude of minima. Understanding how this complex structure influences optimization requires a more refined analysis. In this context, different optimizers can, in principle, target different minima, and the specific minimum found can depend on the hyperparameters values, such as the batch size in SGD [Li et al., 2019, Jastrzębski et al., 2017]. Analysis of the Hessian matrix, when the gradient of the loss is null, can also reveal how the structure of SGD solutions changes with batch size. For instance, [Sagun et al., 2017, Goyal et al., 2018] show that models trained with large batches tend to generalize worse than those trained with small batches, despite achieving similar training accuracy. This phenomenon is linked to the observation that small batches promote convergence to flatter regions, called *wide minima*, in the loss landscape [Keskar et al., 2017]. Such regions are known to correlate with better generalization, and their properties can be studied through the spectrum of the Hessian. While this argument is widely accepted, it is not universally valid. For instance, [Dinh et al., 2017] presents a deep neural network model in which sharp minima are associated with good generalization performance. This suggests that the relationship between the Hessian spectrum and generalization is complex and far from being understood.

Investigating the spectral structure of the Hessian matrix is particularly advantageous because it encapsulates the geometric structure around minimizers and can inform the design of more effective learning schemes. Various second-order optimization methods leverage this information to improve convergence rates; for example, Newton’s method [Dennis and Schnabel, 1983, Lecun et al., 2002]. While second-order methods can, in principle, drastically reduce the number of optimization steps, their computational cost is often prohibitive in large-scale settings. As a result, quasi-Newton methods such as BFGS [Nocedal and Wright, 2006] or Natural Gradient Descent [Amari, 1998] have been developed, which do not require the full Hessian but instead exploit its spectral properties by considering only a limited number of eigenvalues and eigenvectors. Although this does not provide complete information, it yields useful insights into the geometry of the loss surface and its evolution during training [Nocedal and Wright, 2006, Dennis and Schnabel, 1983, Lecun et al., 2002].

The growing interest in the geometry of loss landscapes has led to a substantial body of empirical work investigating the Hessian spectrum in deep learning models. For example, [Sagun et al., 2017] shows that the loss function Hessian spectrum (for feed-forward and fully connected networks, several losses, and trained on MNIST) at the initialization typically decomposes into two components: a bulk of eigenvalues concentrated near zero and a set of well-separated outliers. The bulk eigenvalues are associated with eigenvectors corresponding to flat directions in the loss, while outlier eigenvectors reflect directions of steep curvature. The number of outliers increases with data complexity, but the overall shape of the spectrum remains robust to changes in network and batch size.

Further insights come from Fort and Ganguli [Fort and Ganguli, 2019], who studied classification tasks and found that the number of outlier eigenvalues typically matches the number of target classes. During training, the gradient tends to align with a low-dimensional subspace spanned by these outlier eigenvectors, suggesting that optimization predominantly occurs in a narrow subspace of high curvature, while most directions in parameter space remain largely unexplored. Additionally, the magnitude of the outlier eigenvalues exhibits non-monotonic behavior (growing initially, peaking, and then decreasing) implying that gradient descent traverses high-curvature regions before settling into flatter basins with better generalization properties. An empirical investigation of the Hessian spectrum of a loss function during the training process, in simple classification tasks, is presented in [Sagun

et al., 2016].

While these findings are largely empirical, the goal of the preliminary investigation in this chapter is to provide a theoretical framework for analyzing the Hessian spectrum using tools from statistical physics. It is important to stress that the method we developed doesn't require the gradient of the loss to be zero, so we are not working on stationary weight configurations.

We employ the replica method to analytically investigate the Hessian spectrum in two specific architectures: the negative-margin perceptron and the Tree Committee Machine, for different choices of loss functions during and after training. Both models feature non-convex loss landscapes (for some choice of the margin κ and complexity α).

4.1 Hessian investigation in Neural Networks

One of the key motivations for analyzing the spectrum of the Hessian in machine learning models is its ability to inform how gradient directions can be adjusted to improve convergence. Newton-like methods exemplify this approach by leveraging second-order information [Dennis and Schnabel, 1983]. However, due to the computational expense of evaluating the full Hessian matrix associated with a loss, especially in non-convex settings, it is often more practical to focus on a small subset of the spectrum, typically the largest and smallest eigenvalues [Nocedal and Wright, 2006].

Several theoretical investigations have proposed modeling loss Hessian spectra using tools from Random Matrix Theory (RMT). Classical results, such as the Marčenko–Pastur law [Marčenko and Pastur, 1967] and Wigner's semicircle law [Wigner, 1958], have been applied to approximate the bulk distribution of eigenvalues. However, these approaches often rely on strong assumptions that deviate from realistic scenarios in Machine learning and inference.

A more refined model is the so-called *spiked covariance model* [Johnstone, 2001, Baik et al., 2005], which treats the Hessian as a sum of a full-rank noise component and a low-rank signal component. This decomposition aligns with empirical observations and accounts for the emergence of a bulk spectrum with isolated outlier eigenvalues. In this framework, the bulk arises from the noise, while the spikes reflect directions informed by data and structure.

Nonetheless, a persistent gap exists between theory and practice: the simplifying assumptions required by RMT do not always hold in real-world applications. A first attempt to perform the Hessian investigation in more general frameworks is due to Liao et al. in [Liao and Mahoney, 2021, Martin and Mahoney, 2021]. In [Liao and Mahoney, 2021], the authors consider general instances of Generalized Linear Models (GLM). Generalized linear models describe a generalized linear relationship in a dataset $\mathcal{D} = \{\mathbf{x}^\mu, y^\mu\}_{\mu \in \{1, \dots, M\}}$ between the input features $\mathbf{x}^\mu \in \mathbb{R}^N$ and their corresponding responses y^μ [McCullagh and Nelder, 1989, Nelder and Wedderburn, 1972]. The relation between input and output is described through a set of real parameters $\mathbf{w}^*$, such that for given $\mathbf{w}^* \cdot \mathbf{x}^\mu$ the response y^μ is independently drawn from the conditional density function:

$$y^\mu \sim f(y \mid \mathbf{w}^* \cdot \mathbf{x}^\mu) \quad (4.1)$$

This broad framework encompasses many well-known machine learning models, from the noiseless phase retrieval problem, $y^\mu = |\mathbf{w}^* \cdot \mathbf{x}^\mu|$ [Bonnaire et al., 2024, Mannelli et al.,

2020, Mignacco et al., 2021], to single-layer neural networks with nonlinear activations, $y^\mu = \sigma(\mathbf{w}^* \cdot \mathbf{x}^\mu)$ [Rosenblatt, 1958, Engel and Van den Broeck, 2001]. Learning a GLM consists of approximating the ground truth $\mathbf{w}^*$, with a learned set of parameters $\mathbf{w}$, estimated through a specific learning procedure.

In [Liao and Mahoney, 2021], Liao et al., for the GLM case, derive an exact characterization of the asymptotic Hessian spectrum for the square and cross-entropy losses, in the *proportional limit*, where both the input dimension N and sample size M tend to infinity with a fixed ratio $\alpha = M/N$. Their analysis assumes i.i.d. Gaussian inputs with mean $\mathbf{m}$ and covariance matrix $\mathbf{C}$ (the signal to be inferred, $\mathbf{w}^*$, generates the labels). Under these assumptions, they determine the spectral density and isolate eigenvalues associated with the signal structure. Notably, their work identifies two distinct types of spectral spikes: the first, associated with the data signal, is positively correlated with the ground-truth vector $\mathbf{w}^*$; the second, arising from the model architecture, may appear on the opposite side of the bulk, with respect to $\mathbf{w}^*$ eigenvalues, and reflect structural properties of the model. Remarkably, the leading eigenvector (associated with the largest eigenvalue) often aligns with $\mathbf{w}^*$, while an additional negative eigenvalue may capture implicit biases of the architecture.

Regarding the specific case of the Phase Retrieval problem, [Bonnaire et al., 2024] analyzes the evolution of the spectrum of the Hessian during the optimization of the loss function. The leading eigenvalue in this case is not always present, and the authors, in the infinite-dimensional limit, show how the signal-to-noise ratio influences its presence. They detect the threshold for the BBP transition [Baik et al., 2005] in the Hessian spectrum, which is a critical point where the spectral properties change qualitatively. This transition is linked to the presence concept of threshold states. A similar direction is also pursued in a previous work [Mannelli et al., 2020]. In [Asgari et al., 2025], the spectrum at the minimizer is characterized by convex losses, in the high-dimensional limit.

These findings provide a theoretical explanation for many empirical results observed in single-layer neural networks. However, generalizations to multi-layer networks remain an open problem.

We propose an alternative route based on the replica method to further investigate the Hessian spectrum, analyzing a simple two-layer architecture, the Tree Committee Machine. The proposed framework allows us to investigate the spectral properties of the Hessian in the infinite-dimensional limit, $N \rightarrow \infty$, in the so-called proportional regime, $M = \alpha N$. In this preliminary investigation, we do not constrain the Hessian computation to stationary points of the loss function, which instead is the case of the well-known Kac-Rice formalism [Auffinger et al., 2012].

4.2 Analytical investigation

This section presents the analytical framework for studying the Hessian matrix spectrum for the Tree committee machine, for a generic loss $\ell(\cdot)$ and a generic activation function $g(\cdot)$, computed using the replica method [Mézard et al., 1987].

After presenting some useful basic notions of Random Matrix Theory in Subsection 4.2.1, in Subsection 4.2.2, we introduce the tree committee machine and the definition of its Hessian [Barkai et al., 1990, 1992, Annesi et al., 2025]. It is important to underline that by fixing the activation function $g(\cdot) = \mathbb{I}(\cdot)$ and setting the number of hidden units, K , to 1, one is able to reconstruct the spectral structure for a perceptron network.

The subsequent sections provide a replica method-based approach to study the spectral properties of the Tree Committee Machine, in the infinite-dimensional limit, $N \rightarrow \infty$, and in the proportional regime $M = \alpha N$.

4.2.1 Theoretical background

Random Matrix theory provides useful tools for studying the spectrum of the Hessian matrix. Let's consider the following situation, we are given a dataset $\mathcal{D} = \{\mathbf{x}^\mu, y^\mu\}_{\mu=1,\dots,M}$ and we choose a structural form for reconstructing the output $\hat{y}^\mu = f(\mathbf{x}^\mu, \mathbf{w})$ (the structural form of the function $f(\cdot)$ depends on the architecture of the chosen model). Optimizing $\mathbf{w}$, we reduce the value of a chosen loss function $\mathcal{L}(\mathbf{w}; \mathcal{D})$.

The object of investigation of this chapter is the Hessian matrix of the loss function, which is defined as:

$$H = \left[H_{i,j}^{(\mathcal{L})} \right]_{i \in N, j \in N} \quad \text{where} \quad H_{i,j}^{(\mathcal{L})} = \frac{\partial^2 \mathcal{L}(\mathbf{w}; \mathcal{D})}{\partial w_i \partial w_j}. \quad (4.2)$$

Here we are interested in investigating its spectral properties. The spectrum of the H matrix is defined as:

$$\rho_H(z) := \frac{1}{N} \sum_{i=1}^N \delta(z - z_i) \quad (4.3)$$

which represents the distribution of its eigenvalues ($\{z_i\}_{i=1,\dots,N}$) [Potters and Bouchaud, 2020]. Since we are interested in studying the spectrum of H in the infinite-dimensional limit, and in the so-called proportional regime, $M = \alpha N$, it is infeasible to use Equation (4.3) for performing spectral analysis. Alternatively, one can use the auxiliary definition of the resolvent of a matrix:

$$R_H(z) \equiv \frac{1}{N} \sum_{i=1}^N \frac{1}{z - z_i} = \int dy \frac{\rho_H(y)}{z - y}. \quad (4.4)$$

Using this definition, it is now possible to rewrite the spectrum as [Livan et al., 2017, Potters and Bouchaud, 2020]:

$$\rho_H(z) = \lim_{\varepsilon \rightarrow 0^+} \frac{1}{\pi} \text{Im} R_H(z - i\varepsilon). \quad (4.5)$$

One now has an approach to compute the spectrum $\rho_H(z)$, starting from the definition of the resolvent. As reported in [Livan et al., 2017, Potters and Bouchaud, 2020], one can now rewrite the resolvent of H as:

$$R_H(z) = \frac{1}{N} \text{Tr} \left(\frac{1}{z\mathbb{I} - H} \right) = \frac{\partial}{\partial z} \varphi_H(z), \quad (4.6)$$

where

$$\varphi_H(z) = -\frac{2}{N} \ln Z_H(z) \quad (4.7)$$

is referred to as the "free entropy", and $Z_H(z)$ is the partition function associated with the

computation of the spectrum $\rho_H(z)$:

$$Z_H(z) = \det(H - z\mathbb{I})^{-1/2} = \int \prod_{i=1}^N \frac{du_i}{\sqrt{2\pi}} \exp\left(-\frac{1}{2} \sum_{i,j} u_i (H_{ij} - z\delta_{ij}) u_j\right). \quad (4.8)$$

Using these definitions, we have all the ingredients for investigating the spectrum of the Tree Committee Machine starting from the computation of $\varphi_H(z)$. This computation is performed using the Replica Method, [Mézard et al., 1987].

Before entering the details of the replica computation, it is important to define the cumulative distribution of the eigenvalues, and how it can be computed starting from the definition of $\varphi_H(z)$. The cumulative distribution function is defined as:

$$C_H(x) = \int_{-\infty}^x dz \rho_H(z) = \frac{1}{\pi} \text{Im} \varphi_H(z). \quad (4.9)$$

As usual in statistical physics contexts, we are interested in studying the spectral properties of H , independently of the specific realization of the disorder. For this reason, we address the investigation of the quenched free entropy:

$$\mathbb{E}_{\mathbf{w},X} \varphi_H(z) = -\frac{2}{N} \mathbb{E}_{\mathbf{w},X} \ln Z_H(z) \equiv -2\phi. \quad (4.10)$$

As previously mentioned, the spectral analysis is going to be applied to the Tree Committee Machine. In the next section, its Hessian is defined for a general loss and activation function.

4.2.2 Tree committee machine

The tree committee machine is a simple example of a two-layer architecture whose second layer has K hidden units with non-overlapping receptive fields. The weights connecting the second layer with the first are not modified during the learning procedure ($\{c_l\}_{l \in \{1, \dots, K\}}$) [Barkai et al., 1990, 1992, Annesi et al., 2025]. Only the first layer weights are optimized ($\{w_{l,i}\}_{i \in \{1, \dots, N/K\}, l \in \{1, \dots, K\}}$).

Mathematically, given a dataset $\mathcal{D} = \{\mathbf{x}^\mu, y^\mu\}_{\mu=1, \dots, M}$ (for simplicity one can assume that the labels $\{y^\mu\}_\mu$ are all equal to +1), the label estimated by the tree committee machine for $\mathbf{x}^\mu$ input, is computed as

$$\hat{y}^\mu = \text{sign} \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g \left(\sqrt{\frac{K}{N}} \sum_{i=1}^{N/K} w_{li} x_{li}^\mu \right) \right) \quad (4.11)$$

where $g(\cdot)$ is the first layer activation function. A more detailed introduction to the Tree committee machine is presented in Chapter 1.

Since for simplicity we have assumed that $y^\mu = 1$ for all $\mu \in \{1, \dots, M\}$, given a generic $\ell(\cdot)$ and a generic activation for the input layer $g(\cdot)$, one can rewrite the loss over the estimated outputs, together with a regularization term over the first layer weights, as:

$$\mathcal{L}(\mathbf{w}; \mathcal{D}) = \sum_{\mu} \ell \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g \left(\sqrt{\frac{K}{N}} \sum_{i=1}^{N/K} w_{li} x_{li}^{\mu} \right) - \kappa \right) + \frac{\lambda}{2} \|\mathbf{w}\|_2^2, \quad (4.12)$$

where $\mathbf{y} = \{y^{\mu}\}_{\mu=1,\dots,M}$, $X = \{\mathbf{x}^{\mu}\}_{\mu=1,\dots,M}$, $\mathbf{w} = \{w_{li}\}_{l=1,\dots,K; i=1,\dots,N/K}$ and λ the regularization strength. ℓ is the loss function defined over a single dataset element. Starting from this definition of the loss, one can now define the associated Hessian matrix:

$$H_{li, l'j} = \frac{1}{N} \sum_{\mu} \left(x_{l'j}^{\mu} c_{l'} g' \left(s_l^{\mu} \right) \right) \ell'' \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g \left(s_l^{\mu} \right) - \kappa \right) \left(x_{li}^{\mu} c_l g' \left(s_l^{\mu} \right) \right) \quad (4.13a)$$

$$+ \frac{\sqrt{K}}{N} \sum_{\mu} x_{l'j}^{\mu} c_{l'} \ell' \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g \left(s_l^{\mu} \right) - \kappa \right) \left(x_{li}^{\mu} c_l g'' \left(s_l^{\mu} \right) \right) + \lambda \delta_{li, l'j}, \quad (4.13b)$$

where, in the definition of the Hessian entries, li refers to the index of the first weight with respect to which the loss is differentiated, and $l'j$ refers to the second. Furthermore, to simplify the expression of the Hessian matrix, we have defined

$$s_l^{\mu} = \sqrt{\frac{K}{N}} \sum_{i=1}^{N/K} w_{li} x_{li}^{\mu}. \quad (4.14)$$

Another important ingredient for the replica investigation is the definition of the prior distribution over the first layer weights $\tilde{P}(\mathbf{w})$, which for now is not specified.

In the following, the distribution of the patterns is assumed to be normal, i.e. $x_{li}^{\mu} \sim \mathcal{N}(0, 1), \forall \mu \in \{1, \dots, M\}, i \in \{1, \dots, N/K\}$ and $l \in \{1, \dots, K\}$. From Equation (4.13), it is clear that fixing the activation function $g(\cdot) = \mathbb{I}(\cdot)$ and $K = 1$, we recover the perceptron Hessian matrix.

Once the Equation (4.12) is defined for the tree committee machine, one can now proceed with the computation of the Hessian spectrum.

4.2.3 Replica Method

As already presented in Section 4.2.1, the spectral structure of the Hessian can be computed starting from the expression of the quenched free entropy in Equation (4.10), computed in the infinite-dimensional limit:

$$\phi = \lim_{N \rightarrow \infty} \frac{1}{N} \mathbb{E}_{\mathbf{w}, X} \ln Z_H(z). \quad (4.15)$$

The expectation over the disorder is, in principle, intractable. However, one can use the well-known replica method [Mézard et al., 1987] to perform such an expectation. The replica method used for computing the logarithm of the partition function consists of

$$\ln Z_H(z) = \lim_{s \rightarrow 0} \partial_s Z_H^s(z), \quad (4.16)$$

where s is an integer. One can now rewrite the quenched free entropy as

$$\phi = \lim_{N \rightarrow \infty} \frac{1}{N} \lim_{s \rightarrow 0} \mathbb{E}_{\mathbf{w}, X} \partial_s Z_H^s(z). \quad (4.17)$$

The distribution of the tree committee machine weights follows the Gibbs distribution with inverse temperature β , where the loss function is reported in Equation (4.12). The parameter β indicates how the distribution of the weights is concentrated around the minima of the loss function. One can thus write:

$$\phi = \lim_{N \rightarrow \infty} \frac{1}{N} \mathbb{E}_X \int dP(\mathbf{w}) \lim_{s \rightarrow 0} \partial_s Z_H^s(z) \frac{e^{-\beta \sum_{\mu} \ell \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g \left(\sqrt{\frac{K}{N}} \sum_{i=1}^{N/K} w_{li} x_{li}^{\mu} \right) - \kappa \right)}}{Z}, \quad (4.18)$$

where Z is the normalization factor of the distribution of the weights, and the exponential factor $e^{-\frac{\beta \lambda}{2} \|\mathbf{w}\|_2^2}$ is absorbed in the prior $P(\mathbf{w}) = \tilde{P}(\mathbf{w}) e^{-\frac{\beta \lambda}{2} \|\mathbf{w}\|_2^2}$. From now on we define $\tilde{\lambda} = \beta \lambda$. For dealing with the normalization factor, one has to use once more a replica trick of the form:

$$\frac{1}{Z} = \lim_{n \rightarrow 0} Z^{n-1}, \quad (4.19)$$

where n is assumed to be an integer. Combining these two results, one can derive the expression of the free entropy function as:

$$\phi = \lim_{N \rightarrow \infty} \frac{1}{N} \lim_{n \rightarrow 0} \lim_{s \rightarrow 0} \partial_s \tilde{\phi} \quad (4.20)$$

$$= \lim_{N \rightarrow \infty} \frac{1}{N} \lim_{n \rightarrow 0} \lim_{s \rightarrow 0} \partial_s \mathbb{E}_X Z^{n-1} \int dP(\mathbf{w}) Z_H^s(z) e^{-\beta \sum_{\mu} \ell \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g \left(\sqrt{\frac{K}{N}} \sum_{i=1}^{N/K} w_{li} x_{li}^{\mu} \right) - \kappa \right)}. \quad (4.21)$$

The free entropy is now made of $n-1$ copies of Z , which from now on are labeled by a and b , and s copies of $Z_H(z)$, which are labeled by c and d . After some algebraic manipulations, the expression of $\tilde{\phi}$ before the analytic continuation to the origin follows this expression:

$$\tilde{\phi} = \int \prod_{a,b,l} \frac{d\hat{q}_{ab}^l dq_{ab}^l}{2\pi} \prod_{a,c,l} \frac{d\hat{t}_{ac}^l dt_{ac}^l}{2\pi} \prod_{c,d,l} \frac{d\hat{p}_{cd}^l dp_{cd}^l}{2\pi} e^{\frac{N}{K} (G_I + G_S + \alpha K G_E)} \quad (4.22)$$

$$G_I = -\frac{1}{2} \sum_{a,b,l} \hat{q}_{ab}^l q_{ab}^l - \sum_{a,c,l} \hat{t}_{ac}^l t_{ac}^l - \frac{1}{2} \sum_{c,d,l} \hat{p}_{cd}^l p_{cd}^l + \frac{z - \lambda}{2} \sum_{l,c} p_{cc}^l \quad (4.23)$$

$$G_S = \log \int \prod_{a=1}^n dP(\mathbf{w}^a) \prod_{l,i,c} \frac{du_{li}^c}{\sqrt{2\pi}} \quad (4.24)$$

$$\times e^{\frac{1}{2} \sum_{a,b,l} \hat{q}_{ab}^l \sum_i w_{il}^a w_{il}^b + \sum_{a,c,l} \hat{t}_{ac}^l \sum_i w_{il}^a u_{il}^c + \frac{1}{2} \sum_{c,d,l} \hat{p}_{cd}^l \sum_i u_{il}^c u_{il}^d} \quad (4.25)$$

$$G_E = \log \int \prod_{a,l} \frac{d\hat{\lambda}_{la} d\lambda_{la}}{2\pi} \prod_{c,l} \frac{d\hat{h}_{lc} dh_{lc}}{2\pi} e^{i \sum_{l,a} \hat{\lambda}_{la} \lambda_{la} + i \sum_{l,c} \hat{h}_{lc} h_{lc}} \quad (4.26)$$

$$\times e^{-\frac{1}{2} \sum_l \left(\sum_{a,b} \hat{\lambda}_{la} \hat{\lambda}_{lb} q_{ab}^l + 2 \sum_{a,c} \hat{\lambda}_{la} \hat{h}_{lc} t_{ac}^l + \sum_{c,d} \hat{h}_{lc} \hat{h}_{ld} p_{cd}^l \right)} \quad (4.27)$$

$$\times e^{-\frac{1}{2\sqrt{K}} \sum_l c_l h_{lc} \ell' \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g(\lambda_{l1}) - \kappa \right) g''(\lambda_{l1}) h_{lc}} \quad (4.28)$$

$$\times e^{-\beta \ell \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g(\lambda_{la}) \right)} \quad (4.29)$$

$$\times e^{-\frac{1}{2K} \sum_{ll'} c_l h_{lc} g'(\lambda_{l1}) \ell'' \left(\frac{\sigma}{\sqrt{K}} \sum_{l=1}^K c_l g(\lambda_{l1}) - \kappa \right) c_{l'} h_{l'c} g'(\lambda_{l1})} \quad (4.30)$$

where q_{ab}^l, p_{cd}^l and t_{ac}^l are the overlap parameters, defined as:

$$q_{ab}^l = \sum_i \frac{w_{il}^a w_{il}^b}{N} \quad (4.31)$$

$$p_{cd}^l = \sum_i \frac{u_{il}^c u_{il}^d}{N} \quad (4.32)$$

$$t_{ac}^l = \sum_i \frac{w_{il}^a u_{il}^c}{N} \quad (4.33)$$

As is usually done in the replica computation, we first choose an Ansatz for the structure of the overlaps, so that we can perform the analytic continuation and take the small n and s limit, followed by saddle point evaluation. The Ansatz we chose is the Replica Symmetric one (RS). Given the symmetries of the problem, it takes the form

$$q_{ab}^l = \begin{cases} Q & a = b \\ q & a \neq b \end{cases} \quad p_{cd}^l = \begin{cases} P & c = d \\ p & c \neq d \end{cases} \quad t_{ac}^l = \begin{cases} T & a = 1 \\ t & a \neq 1 \end{cases} \quad (4.34)$$

$$\hat{q}_{ab}^l = \begin{cases} -\hat{Q} & a = b \\ \hat{q} & a \neq b \end{cases} \quad \hat{p}_{cd}^l = \begin{cases} -\hat{P} & c = d \\ \hat{p} & c \neq d \end{cases} \quad \hat{t}_{ac}^l = \begin{cases} \hat{T} & a = 1 \\ \hat{t} & a \neq 1 \end{cases} \quad (4.35)$$

Using the RS ansatz and computing the limit:

$$\phi = \lim_{N \rightarrow \infty} \frac{1}{N} \lim_{n \rightarrow 0} \lim_{s \rightarrow 0} \partial_s \tilde{\phi} \quad (4.36)$$

it is possible to obtain the final expression of the free entropy:

$$\phi = \mathcal{G}_I + \mathcal{G}_S + \alpha \mathcal{G}_E \quad (4.37)$$

with the interaction and entropic terms expressed as

$$\mathcal{G}_I = -\hat{T}T + \hat{t}t + \frac{1}{2} \hat{P}P + \frac{1}{2} \hat{p}p + \frac{z - \lambda}{2} P \quad (4.38a)$$

$$\mathcal{G}_S = \frac{1}{2} \ln \frac{2\pi}{\hat{P} + \hat{p}} + \frac{1}{\hat{P} + \hat{p}} \left[\frac{\hat{p}}{2} + \frac{1}{2} \frac{\hat{Q} + 2\hat{q}}{(\hat{Q} + \hat{q})^2} (\hat{T} - \hat{t})^2 + \frac{\hat{t} (\hat{T} - \hat{t})}{\hat{Q} + \hat{q}} \right] \quad (4.38b)$$

where $Q, q, \hat{Q}, \hat{q}$ are the order parameters for the equilibrium distribution of the weights $\mathbf{w}$. Their optimal value is obtained by solving the saddle point equations of the associated free entropy $\phi^{(r)}$, whose explicit expression is reported in Section 4.2.4, directly for the $K \rightarrow \infty$ limit. The energetic term in Equation (4.37) is now reported

$$\begin{aligned} \mathcal{G}_E = & \int \prod_l Dx_l Dz_l D\lambda_l \frac{e^{-\beta \ell \left(\frac{1}{\sqrt{K}} \sum_l c_l g(\sqrt{Q-q}\lambda_l + \sqrt{q}x_l) - \kappa \right)}}{\int \prod_l D\lambda_l e^{-\beta \ell \left(\frac{1}{\sqrt{K}} \sum_l c_l g(\sqrt{Q-q}\lambda_l + \sqrt{q}x_l) - \kappa \right)}} \ln \int \prod_l Dh_l \\ & \times e^{-\frac{1}{2} \ell'' \left(\frac{1}{\sqrt{K}} \sum_l c_l g(\sqrt{Q-q}\lambda_l + \sqrt{q}x_l) - \kappa \right) \left[\frac{1}{\sqrt{K}} \sum_l c_l \left(\sqrt{P-ph_l + \frac{t}{q}} x_l + \frac{T-t}{\sqrt{Q-q}} \lambda_l + \sqrt{\gamma} z_l \right) g'(\sqrt{Q-q}\lambda_l + \sqrt{q}x_l) \right]^2} \\ & \times e^{-\frac{1}{2} \ell' \left(\frac{1}{\sqrt{K}} \sum_l c_l g(\sqrt{Q-q}\lambda_l + \sqrt{q}x_l) - \kappa \right) \frac{1}{\sqrt{K}} \sum_l c_l \left(\sqrt{P-ph_l + \frac{t}{q}} x_l + \frac{T-t}{\sqrt{Q-q}} \lambda_l + \sqrt{\gamma} z_l \right)^2 g''(\sqrt{Q-q}\lambda_l + \sqrt{q}x_l)} \end{aligned} \quad (4.39)$$

where $\gamma \equiv p - \frac{t^2}{q} - \frac{(T-t)^2}{Q-q}$.

It is important to notice that the spectrum of the Hessian corresponds to $\rho_H(z) = -iP/\pi$, where P is the value of the corresponding order parameter obtained by optimizing the saddle point equations.

Equation (4.39) is still intractable, due to the presence of the hidden layer dimension K , which in principle is arbitrary. Under the so-called *large width limit* $K \rightarrow \infty$, which consists of considering a diverging K , after taking the limit $N \rightarrow \infty$, one can find a closed form for the energetic term. In the following analysis, the large width limit is considered.

Using Equation (4.37), one can investigate the Hessian spectral properties for a tree committee machine. For small values of the inverse temperature, the free entropy is the one associated with a model whose weights are uniformly distributed. One can also compute the limit $\beta \rightarrow \infty$, from which it is possible to infer the spectrum of the Hessian whose weights are distributed according to configurations minimizing the loss. We are interested in investigating the evolution of the spectrum for different values of the inverse temperature β , starting from finite β and then focusing on $\beta \rightarrow \infty$ limit.

4.2.4 Finite temperature analysis for the large width limit

In this section, we report the result of the replica computation, under the infinite width assumption, for finite β . After reporting the expression for $\phi^{(r)}$, associated to the equilibrium distribution of the weights, we report the final expression for the Energetic term in Equation (4.39).

RS solution for the equilibrium distribution

In the previous section, we stated that the order parameters $Q, q, \hat{Q}, \hat{q}$ must be optimized by solving the saddle point equations associated with the computation of the quenched free entropy $\phi^{(r)}$ for the equilibrium distribution of $\mathbf{w}$, in the infinite-dimensional limit, under the replica symmetric (RS) ansatz and in the infinite width regime [Annesi et al., 2025,

Engel and Van den Broeck, 2001]. In this context, the equilibrium free entropy is given by:

$$\phi^{(r)} = \mathcal{G}_I^{(r)} + \mathcal{G}_S^{(r)} + \alpha \mathcal{G}_E^{(r)} \quad (4.40)$$

$$\mathcal{G}_I^{(r)} = \frac{Q\hat{Q}}{2} + \frac{q\hat{q}}{2} - \frac{\tilde{\lambda}Q}{2} \quad (4.41)$$

$$\mathcal{G}_S^{(r)} = \frac{\log(2\pi)}{2} - \frac{\log(\hat{Q} + \hat{q})}{2} + \frac{\hat{q}}{2(\hat{Q} + \hat{q})} \quad (4.42)$$

$$\mathcal{G}_E^{(r)} = \int Dz \log \left(\int Dx e^{-\beta \ell(\sqrt{\Phi_1}z + \sqrt{\Phi_0}x - \kappa)} \right) \quad (4.43)$$

where Φ_0 and Φ_1 are defined as:

$$\Phi_1 \equiv \int Dx \left(\int Dy g(\sqrt{q}x + \sqrt{Q - qy}) \right)^2 - \left(\int Dx g(\sqrt{Q}x) \right)^2 \quad (4.44)$$

$$\Phi_0 \equiv \int Dx g^2(\sqrt{Q}x) - \int Dx \left(\int Dy g(\sqrt{q}x + \sqrt{Q - qy}) \right)^2. \quad (4.45)$$

It is possible to notice that the dual parameters can be integrated out,

$$\hat{Q} = \frac{Q - 2q}{(Q - q)^2} \quad \hat{q} = \frac{q}{(Q - q)^2}. \quad (4.46a)$$

obtaining a equilibrium quenched free entropy $\phi^{(r)}$, that depends only on Q and q . The saddle point equations for the order parameters Q and q are obtained by fixing the derivatives of $\phi^{(r)}$, with respect to Q and q , to zero.

Energetic term for finite β

Once the replica computation for the equilibrium free entropy $\phi^{(r)}$ has been performed, we have all the ingredients to compute the energetic term expression, in the infinite width limit. Its expression follows:

$$\begin{aligned} \mathcal{G}_E = & -\frac{1}{2} \left\langle \ell'(O) \left(\frac{\Sigma_0}{\sqrt{\Phi_0}}x_0 + \frac{\Sigma_1}{\sqrt{\Phi_1}}x_1 \right) \right\rangle - \frac{1}{2} \left\langle \ln \left(1 + \Delta_0 \ell''(O) \right) \right\rangle \\ & + \frac{\Omega_0^2}{8\Delta_0} \left\langle \frac{\ell'(O)^2}{1 + \Delta_0 \ell''(O)} \right\rangle + \frac{\Omega_0}{2} \left\langle \frac{\ell'(O)\ell''(O) \left(\frac{\Psi_0}{\sqrt{\Phi_0}}x_0 + \frac{\Psi_1}{\sqrt{\Phi_1}}x_1 \right)}{1 + \Delta_0 \ell''(O)} \right\rangle \\ & - \frac{1}{2} \left\langle \frac{\ell''(O)}{1 + \Delta_0 \ell''(O)} \left[\left(\frac{\Psi_0}{\sqrt{\Phi_0}}x_0 + \frac{\Psi_1}{\sqrt{\Phi_1}}x_1 \right)^2 + \Gamma \right] \right\rangle + \frac{1}{8} \left(\Xi_0 - \frac{\Omega_0^2}{\Delta_0} \right) \langle \ell'(O)^2 \rangle \end{aligned} \quad (4.47)$$

where $O \equiv \sqrt{\Phi_0}x_0 + \sqrt{\Phi_1}x_1 - \kappa$, $\Gamma = \tilde{\Delta} - \frac{\Psi_0^2}{\Phi_0} - \frac{\Psi_1^2}{\Phi_1}$ and

$$\langle \bullet \rangle \equiv \int Dx_1 \frac{\int Dx_0 e^{-\beta \ell(\sqrt{\Phi_0}x_0 + \sqrt{\Phi_1}x_1 - \kappa)} \bullet}{\int Dx_0 e^{-\beta \ell(\sqrt{\Phi_0}x_0 + \sqrt{\Phi_1}x_1 - \kappa)}}. \quad (4.48)$$

The parameters appearing in Equation (4.47) are called *effective order parameters*, and are functions of the original order parameters P, p, Q, q, T, t . The expressions of the effective order parameters are reported in Appendix C.

The optimization of ϕ can now be performed using the saddle point method [Mézard et al., 1987]. This amounts to solving the following system of Equations

$$P = -2 \frac{\partial \mathcal{G}_S}{\partial \hat{P}}, \quad p = -2 \frac{\partial \mathcal{G}_S}{\partial \hat{p}}, \quad T = \frac{\partial \mathcal{G}_S}{\partial \hat{T}}, \quad t = -\frac{\partial \mathcal{G}_S}{\partial \hat{t}}, \quad (4.49a)$$

$$\hat{P} = -z + \lambda - 2 \frac{\partial \mathcal{G}_E}{\partial P}, \quad \hat{p} = -2 \frac{\partial \mathcal{G}_E}{\partial p}, \quad \hat{T} = \frac{\partial \mathcal{G}_E}{\partial T}, \quad \hat{t} = -\frac{\partial \mathcal{G}_E}{\partial t}. \quad (4.49b)$$

The optimization is usually performed iteratively, starting from an initial guess, inserting it on the right-hand side, obtaining updated values, and iterating until convergence. The details for the finite temperature replica computation are reported in Appendix C.

4.2.5 Zero Temperature analysis for the large width limit

As introduced above, choosing small values of the inverse temperature β lets us probe the spectral properties of the loss Hessian when the weight distribution is not tightly concentrated around loss-minimizing configurations. Conversely, one may wish to study the opposite regime $\beta \rightarrow \infty$. Moreover, in order for the free entropy to remain finite as $\beta \rightarrow \infty$, the order parameters themselves must be rescaled appropriately.

RS solution for the equilibrium distribution

Let us begin by performing the replica calculation for the equilibrium distribution at zero temperature. We introduce the overlap scaling

$$q = Q - \frac{\delta q}{\beta}. \quad (4.50)$$

After integrating out the conjugate (dual) variables, the scaled order parameters Q and δq satisfy the saddle-point equations that extremize the quenched free entropy $\phi^{(r)}$ under the replica-symmetric ansatz in the infinite-dimensional, infinite-width limit:

$$\phi^{(r)} = \mathcal{G}_I^{(r)} + \mathcal{G}_S^{(r)} + \alpha \mathcal{G}_E^{(r)}, \quad (4.51)$$

$$\mathcal{G}_I^{(r)} + \mathcal{G}_S^{(r)} = \frac{Q}{2\delta q} - \frac{\lambda Q}{2}, \quad (4.52)$$

$$\mathcal{G}_E^{(r)} = \int Dz \left[-\min_z \left\{ \frac{z^2}{2} + \ell(\sqrt{\delta\Phi_0} z + \sqrt{\Phi_1} x) \right\} \right]. \quad (4.53)$$

To keep the free entropy finite as $\beta \rightarrow \infty$, Φ_0 must be rescaled according to

$$\Phi_0 = \mathcal{K}_Q(Q) - \mathcal{K}_Q(q) = \frac{\mathcal{K}'_Q(Q) \delta q}{\beta} \equiv \frac{\delta\Phi_0}{\beta}, \quad (4.54)$$

while Φ_1 is defined by

$$\Phi_1 = \int Dx \left(\int Dy g(\sqrt{q}x + \sqrt{Q-q}y) \right)^2 - \left(\int Dx g(\sqrt{Q}x) \right)^2, \quad (4.55)$$

with

$$\mathcal{K}_Q(q) = \int Dx \left(\int Dy g(\sqrt{q}x + \sqrt{Q - q}y) \right)^2.$$

Free entropy expression

Now that the final expression for the equilibrium free entropy is known, the new expression of ϕ , in the zero temperature limit, can be computed. The first step consists in rescaling the order parameters so that the free entropy remains finite as $\beta \rightarrow \infty$. The scaling of q induces the following scalings on T and t and their duals:

$$t = T - \frac{\delta q \delta t}{\beta} \quad \hat{t} = \beta \frac{\delta \hat{t}}{\delta q} \quad (4.56a)$$

$$\hat{T} = \beta \frac{\delta \hat{t}}{\delta q} + \delta \hat{T} \quad \hat{T} - \hat{t} = \delta \hat{T}. \quad (4.56b)$$

The rescaled order parameters, enforce the following scaling of the effective order parameters:

$$\Psi_0 = \frac{\delta \Psi_0}{\beta} \quad \Sigma_0 = \frac{\delta \Sigma_0}{\beta}. \quad (4.57a)$$

Correspondingly, for $\beta \rightarrow \infty$, the equilibrium measure is peaked over loss minima, thus the average over the equilibrium measure becomes

$$\langle f \rangle \equiv \int Dx_1 \frac{\int Dx_0 e^{-\beta \ell(\sqrt{\Phi_0}x_0 + \sqrt{\Phi_1}x_1)} f(x_0, x_1)}{\int Dx_0 e^{-\beta \ell(\sqrt{\Phi_0}x_0 + \sqrt{\Phi_1}x_1)}} \rightarrow \int Dx_1 f(x_0^*(x_1), x_1) \quad (4.58)$$

where

$$x_0^*(x_1) \equiv \arg \max_{x_0} \left[-\frac{x_0^2}{2} - \ell \left(\sqrt{\mathcal{K}'_Q(Q)} \delta q x_0 + \sqrt{\Phi_1} x_1 \right) \right]. \quad (4.59)$$

Once all the scalings have been performed, the free entropy ϕ can be expressed as

$$\phi = \mathcal{G}_I + \mathcal{G}_S + \alpha \lim_{\beta \rightarrow \infty} \frac{\mathcal{G}_E}{\beta} \quad (4.60a)$$

$$\mathcal{G}_I = -\delta \hat{T} T - \delta \hat{t} \delta t + \frac{1}{2} \hat{P} P + \frac{1}{2} \hat{p} p + \frac{z - \lambda}{2} P \quad (4.60b)$$

$$\mathcal{G}_S = \frac{1}{2} \ln \frac{2\pi}{\hat{p} + \hat{P}} + \frac{1}{\hat{p} + \hat{P}} \left[\frac{\hat{p}}{2} + \frac{Q}{2} \delta \hat{T}^2 + \delta \hat{t} \delta \hat{T} \right] \quad (4.60c)$$

$$\begin{aligned}
\lim_{\beta \rightarrow \infty} \frac{\mathcal{G}_E}{\beta} = & -\frac{1}{2} \int Dx_1 \ell'(O_\star(x_1)) \left(\frac{\delta \Sigma_0}{\sqrt{\delta \Phi_0}} x_0^\star(x_1) + \frac{\Sigma_1}{\sqrt{\Phi_1}} x_1 \right) \\
& + \frac{1}{8} \left(\Xi_0 - \frac{\Omega_0^2}{\Delta_0} \right) \int Dx_1 \ell'(O_\star(x_1))^2 \\
& - \frac{1}{2} \int Dx_1 \ln \left(1 + \Delta_0 \ell''(O_\star(x_1)) \right) \\
& + \frac{\Omega_0^2}{8\Delta_0} \int Dx_1 \frac{\ell'(O_\star(x_1))^2}{1 + \Delta_0 \ell''(O_\star(x_1))} \\
& + \frac{\Omega_0}{2} \int Dx_1 \frac{\ell'(O_\star(x_1)) \ell''(O_\star(x_1)) \left(\frac{\delta \Psi_0}{\sqrt{\delta \Phi_0}} x_0^\star(x_1) + \frac{\Psi_1}{\sqrt{\Phi_1}} x_1 \right)}{1 + \Delta_0 \ell''(O_\star(x_1))} \\
& - \frac{1}{2} \int Dx_1 \frac{\ell''(O_\star(x_1))}{1 + \Delta_0 \ell''(O_\star(x_1))} \left[\left(\frac{\delta \Psi_0}{\sqrt{\delta \Phi_0}} x_0^\star(x_1) + \frac{\Psi_1}{\sqrt{\Phi_1}} x_1 \right)^2 + \Gamma \right]
\end{aligned} \tag{4.61}$$

where we have denoted

$$O_\star(x_1) \equiv \sqrt{\mathcal{K}'_Q(Q) \delta q} x_0^\star(x_1) + \sqrt{\Phi_1} x_1 \tag{4.62}$$

The new effective order parameters in the $\beta \rightarrow \infty$ limit are reported in Appendix C. The optimization of the free entropy ϕ in Equation (4.37), in the $\beta \rightarrow \infty$ limit, can be performed using the saddle point method, as previously done for the finite β case. The saddle point equations are similar to the ones in Equation (4.49), one only needs consider the scaled order parameters. Again the optimization of the reference system is performed ahead using the saddle point equations associated with $\phi^{(r)}$ reported in Equation (4.51).

Using this theoretical framework one can now address the evolution of the spectrum of the Hessian matrix for the tree committee machine, considering different distributions of the weights, whose concentration over loss function minima is controlled through the inverse temperature β .

4.3 Preliminary analytical results

In this section, the preliminary results for the evolution of the spectrum of the loss function Hessian are reported. We first report the evolution of the spectrum for the simple perceptron network, in Section 4.3.1, and then, in Section 4.3.2, we report the loss spectral evolution for the tree committee. The main loss functions considered in the analysis are the squared and the cross-entropy losses, already introduced in Chapter 1.

4.3.1 Evolution of Perceptron Hessian spectrum

We begin by computing the spectrum of the Hessian for a single-layer perceptron trained with the squared loss. From the general definition in Equation (4.2), one obtains

$$H_{ij} = \frac{\partial^2}{\partial w_i \partial w_j} \sum_{\mu=1}^M \left(y^\mu - \frac{\mathbf{w} \cdot \mathbf{x}^\mu}{\sqrt{N}} \right)^2 + \lambda \delta_{ij} = \sum_{\mu=1}^M \frac{x_i^\mu x_j^\mu}{N} + \lambda \delta_{ij}.$$

Hence, H is a Wishart ensemble matrix [Livan et al., 2017, Potters and Bouchaud, 2020], whose eigenvalue density follows the Marchenko–Pastur law [Marčenko and Pastur, 1967,

Franz et al., 2015], and, as it depends only on the inputs $\{\mathbf{x}^\mu\}$ and not on $\mathbf{w}$, is independent of the inverse temperature β . The prior over the weights is assumed to be spherical $P(\mathbf{w}) = \delta(\|\mathbf{w}\|_2^2 - N)$ and the regularization strength consequently is fixed to 0.

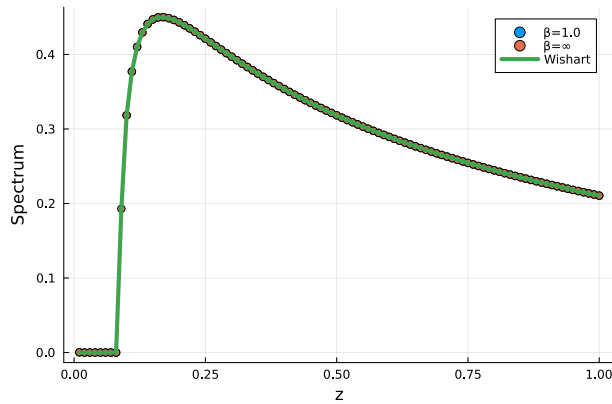


Figure 4.1: Eigenvalue spectrum of the perceptron Hessian under squared loss for $\alpha = 0.5$, $\kappa = 0$, and $\beta \in \{1, \infty\}$. The solid curve is the Marchenko–Pastur spectrum prediction for a Wishart ensemble matrix.

Figure 4.1 shows that the empirical spectrum at $\beta = 1$ and $\beta \rightarrow \infty$ coincides with the Marchenko–Pastur law. This invariance with respect to β serves as a sanity check: in the simple case of squared loss on a perceptron, our theoretical framework reproduces exactly the expected Hessian spectrum. We now turn to the analogous analysis for the tree-committee network.

4.3.2 Evolution of Tree Committee Machine Hessian spectrum

Driven by the results obtained in Section 4.3.1, we now turn our attention to the evolution of the spectrum of the Hessian matrix for the tree committee machine, as a function of the inverse temperature.

Figure 4.2 shows the evolution of the spectrum of the Hessian matrix for the tree committee machine for two choices of the loss function: the *Square* and *Cross-entropy* (with $\gamma = 1.0$) losses. The prior considered in this case is not the spherical prior, but we consider a more realistic situation: a situation in which the weights are not normalized. Their norm is controlled by an L_2 -regularization term in the loss ($\lambda = 1.0$). To keep the computation tractable, we consider the activation function to be $g(\cdot) = \text{erf}(\cdot)$. We also fix $\alpha = 0.5$ and $\kappa = 1.0$. The inverse temperature parameter β is varied in the set $\{0.0001, 1.0, 2.0, 5.0, 10.0, \infty\}$ for the square loss and in the set $\{0.001, 1.0, 2.0, 5.0, \infty\}$ for the cross-entropy loss.

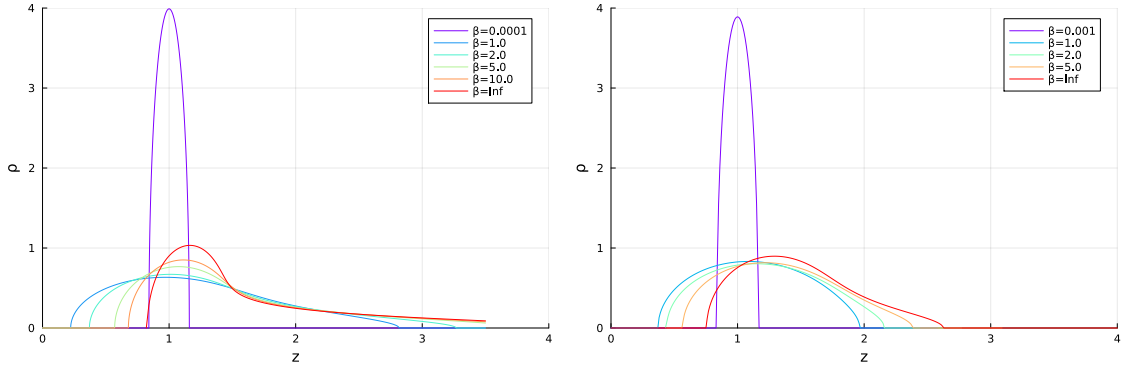


Figure 4.2: Spectrum of the Hessian for the tree committee machine, for $\alpha = 0.5$, $\lambda = 1.0$ and $\kappa = 1.0$, for different values of the inverse temperature parameter $\beta \in \{0.0001, 1.0, 2.0, 5.0, 10.0, \infty\}$ in the case of the square loss (*left*) and $\beta \in \{0.001, 1.0, 2.0, 5.0, \infty\}$, in the case of the cross entropy, with $\gamma = 1.0$ (*right*).

From Figure 4.2, one can notice that the spectrum of the Hessian matrix for the tree committee machine is affected by the value of the inverse temperature β . In particular, for small values of β ($\beta \sim 0$), the spectrum is characterized by a large number of eigenvalues close to λ . For $\beta \sim 0$, the loss function does not affect the Gibbs distribution of the weights. As β increases, the spectrum increases the quantity of eigenvalues larger than λ . The bulk of eigenvalues for the square loss is more peaked, with a longer tail on the right, while the spectrum for the cross-entropy loss is more concentrated around the bulk center. Both losses are characterized by a symmetric spectrum around $z = 1$ for $\beta \sim 0$. The symmetric shape of the spectrum for $\beta \sim 0$ is a consequence of the uniform nature of the weights distribution.

The effect of the inverse temperature can be noticed, by looking at the smallest eigenvalue of the spectrum: increasing β , first its position shifts toward the origin, then starts moving again toward $z = \lambda$. Analogously, the peak of the spectrum first reduces its height, then starts increasing again. This is a consequence of the interplay between the regularization term and the loss function that we are optimizing.

4.3.3 The role of the regularization

In this section, we briefly explore the effect of the regularization for $\beta \rightarrow \infty$. In particular, we consider the case of the square loss function and the cross-entropy one ($\gamma = 1.0$), for different values of the regularization parameter λ . The results are reported in Figure 4.3. The chosen values of $\alpha = 0.5$ and $\kappa = 1.0$ are the same as in the previous section, while the regularization parameter λ is varied in the set $\{0.1, 0.2, 0.5, 1.0, 2.0\}$.

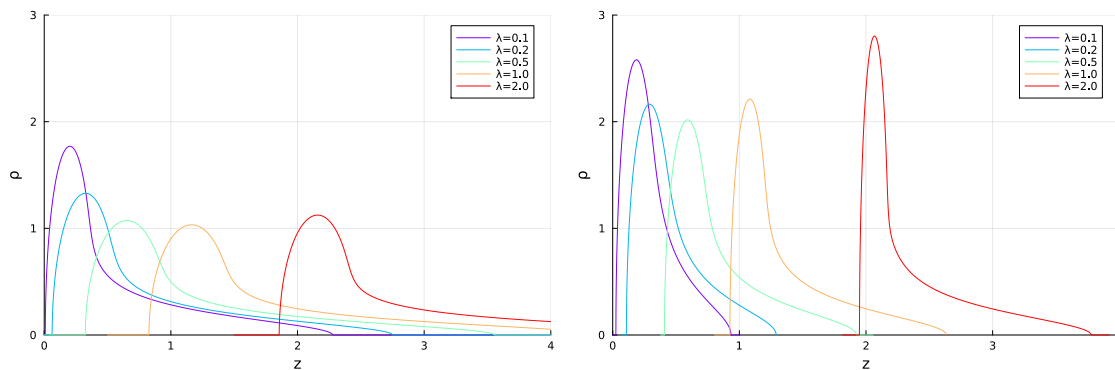


Figure 4.3: Spectrum of the Hessian for the tree committee machine, for $\alpha = 0.5$, $\beta \rightarrow \infty$ and $\kappa = 1.0$, for different values of the regularization parameter $\tilde{\lambda} \in \{0.1, 0.2, 0.5, 1.0, 2.0\}$ in the case of the square loss (*left*) and the cross entropy ($\gamma = 1.0$) (*right*).

For both choices of the loss function, one can observe that the height of the peak, by increasing λ , first decreases and then grows again. This is an intrinsic effect of the tradeoff in the optimization procedure between the regularization and the actual loss that we are optimizing. For high values of λ one should expect the spectrum to present a high peak around the value of λ . For small λ the peak height is lower, the variance of the distribution of the eigenvalues larger, and the center of the bulk should shift toward $z = 0$.

Unfortunately, using randomly generated data, it is not possible to study the effect of the regularization parameter over the outliers eigenvalues, described in [Sagun et al., 2017, Fort and Ganguli, 2019, Seung et al., 1992]. A possible extension of our analysis consists in investigating a model in the teacher-student framework, where the signal is clearly defined and can be reconstructed.

4.4 Recap of the results

In this preliminary investigation:

- We have presented the theoretical framework to compute the Hessian spectrum of simple Machine learning architectures, using the replica method and the saddle point method. Using a proper definition of the Gibbs distribution depending on the loss function, one can control the evolution of the spectrum for different values of the inverse temperature β . For low values of β the optimization of the loss is far from being completed, for $\beta \rightarrow \infty$ the Hessian spectrum is associated with weight configurations minimizing the loss function.
- We used this framework for studying the evolution of the spectrum of the weights of the tree committee machine, for two types of losses: the square and the cross-entropy losses. In this case, we considered an augmented loss with an L_2 regularization, and we investigated the effect of the regularization on the spectrum at $\beta = \infty$.

Chapter 5

Learning RBM via interaction screening optimization

In the previous chapters, we investigated the loss landscape of various inference problems in the context of supervised learning, as well as we proposed a framework for investigating the geometry of the landscape for simple FeedForward Neural Network models, whose properties can be addressed via statistical mechanics tools. In this chapter, we shift our focus to the realm of generative modeling.

In recent years, generative tasks have gained increasing prominence in the fields of Machine Learning and Artificial Intelligence. The rise of Generative AI, especially diffusion-based models [Ho et al., 2020], has enabled the sampling of complex, unknown distributions using only a limited number of observations coming from the same distribution. In other words, generative AI models can learn intricate statistical dependencies from data and generate new samples that closely resemble the original dataset. The goal of generative modeling is to extract high-level features from the dataset and generate synthetic data that are statistically indistinguishable from the real samples.

Among generative models, Energy-Based Models (EBMs) offer a practical framework inspired by concepts from statistical physics. EBMs define a probability distribution over the data via the Boltzmann distribution, using a Hamiltonian function $H(\mathbf{x}; \boldsymbol{\theta})$ that depends on a set of parameters $\boldsymbol{\theta}$:

$$p(\mathbf{x}|\boldsymbol{\theta}) = \frac{e^{-H(\mathbf{x};\boldsymbol{\theta})}}{Z_{\boldsymbol{\theta}}}, \quad (5.1)$$

where $Z_{\boldsymbol{\theta}}$ is the partition function, ensuring proper normalization, and the Hamiltonian contains the sum of many local terms. In this chapter, we are interested in approaching EBM's from an inverse problem perspective [Tarantola, 2005]: given a set of observations, the goal is to infer the parameters $\boldsymbol{\theta}$ that best explain the data. In this context, the energy function $H(\mathbf{x}; \boldsymbol{\theta})$ encodes the statistical properties of the data, and the parameters $\boldsymbol{\theta}$ are learned from a dataset $\mathcal{D} = \{\mathbf{x}^{\mu}\}_{\mu=1}^M$ of M samples. One wants the energy function to assign low energy to configurations that are likely to be observed in the data while assigning high energy to unlikely configurations. Under this perspective, the energy function can be interpreted as a measure of the "cost" associated with a given configuration $\mathbf{x}^{\mu}$, where lower energy corresponds to a higher probability.

In statistical physics, a related and extensively studied problem is the inverse Ising

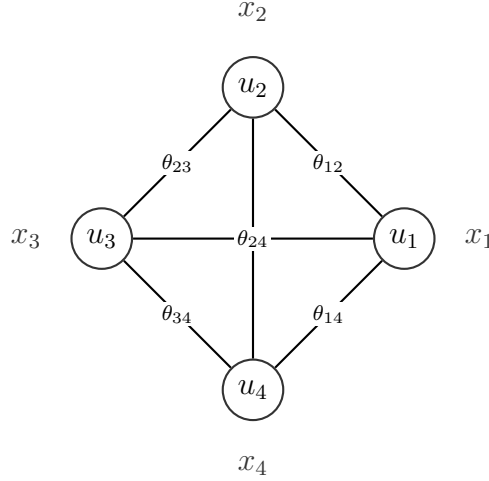


Figure 5.1: Graphical representation of a fully connected Ising model.

problem, where the goal is to infer the parameters of an Ising model (Figure 5.1) from observed configurations [Besag, 1974, Cocco and Monasson, 2011, Sessak and Monasson, 2009]. Ising model is a canonical system composed of binary variables $\mathbf{x} \in \{-1, +1\}^N$, with an energy function incorporating both local field terms and pairwise interactions [Landau and Lifshitz, 1980]:

$$H(\mathbf{x}; \mathbf{W}, \mathbf{u}) = - \sum_{i=1}^N u_i x_i - \sum_{i>j} \theta_{i,j} x_i x_j. \quad (5.2)$$

Learning the Ising model involves estimating the local fields $\mathbf{u} = \{u_i\}_{i=1,\dots,N}$ and the couplings $\Theta = \{\theta_{ij}\}_{i=1,\dots,N; j \neq i}$. Various techniques have been proposed, ranging from maximum likelihood estimation to gradient-based methods. The inverse Ising formulation has found applications in reconstructing biological and physical systems, including inferring synaptic connectivity in neural circuits [Tkačik et al., 2014] and predicting protein structures [Morcos et al., 2011]. A major advantage of this approach is its interpretability and grounding in statistical mechanics.

Despite its usefulness, the Ising model has a key limitation: it only accounts for first- and second-order interactions, i.e., single-variable and pairwise terms. This restricts its capacity to model more realistic systems where higher-order dependencies are essential.

A natural extension involves including multi-body interactions:

$$H(\mathbf{x}; \boldsymbol{\theta}) = - \sum_i \theta_i x_i - \sum_{i>j} \theta_{i,j} x_i x_j - \sum_{i>j>k} \theta_{i,j,k} x_i x_j x_k - \dots \quad (5.3)$$

However, learning such models becomes infeasible due to the exponential growth in the number of parameters required. A promising workaround is based on the assumption that real-world systems often involve sparse and localized multi-body interactions. Yet, without prior knowledge about which interactions are negligible, direct learning remains computationally expensive, scaling exponentially with the interaction order [Lokhov et al., 2018, Vuffray et al., 2016].

A compelling alternative is offered by Restricted Boltzmann Machines (RBMs) [Ackley et al., 1985, Decelle et al., 2024], a class of bipartite energy-based models with hidden units. RBMs retain the interpretability of the Ising model while effectively capturing multi-body interactions through pairwise connections with hidden variables. Notably, RBMs can be formally mapped to higher-order models without increasing the parameter space [Decelle et al., 2024, Cossu et al., 2019, Bulso and Roudi, 2021], making them powerful tools for modeling complex data.

Recently, RBMs have been used, as simple NN architectures, for performing generative modeling. Their parameters are usually optimized via Contrastive Divergence approach (CD) [Hinton, 2002], an approach biasing parameter learning. Here we propose a different learning scheme, based on pseudolikelihood and interaction screening objective, in which the CD bias effect is reduced at the cost of an increased sample complexity [Lokhov et al., 2018].

5.1 Restricted Boltzmann Machines

In this section, we revisit the structure and the properties of the Restricted Boltzmann Machine (RBM) [Ackley et al., 1985]. An RBM is an Ising-like graphical model defined on a bipartite graph. It consists of two distinct layers of variables (here assumed to be binary): the *visible layer*, denoted as $\mathbf{v} = \{v_i\}_{i=1,\dots,N_v} \in \{-1, 1\}^{N_v}$, and the *hidden layer*, denoted as $\mathbf{h} = \{h_i\}_{i=1,\dots,N_h} \in \{-1, 1\}^{N_h}$. N_v and N_h represent, respectively, the number of visible and hidden nodes. The model includes pairwise interactions only between visible and hidden units; there are no intra-layer connections within the visible or hidden layers.

Each variable is subject to an external field, often referred to as a *bias*. We denote the bias acting on visible unit v_j as a_j , and the one acting on hidden unit h_i as b_i . The energy function of an RBM is given by:

$$H(\mathbf{v}, \mathbf{h}; W, \mathbf{a}, \mathbf{b}) = - \sum_{i=1}^{N_h} \sum_{j=1}^{N_v} h_i w_{ij} v_j - \sum_{j=1}^{N_v} a_j v_j - \sum_{i=1}^{N_h} b_i h_i \quad (5.4)$$

Although the model is defined via pairwise interactions, marginalizing over the hidden variables induces effective many-body interactions among the visible units [Decelle et al., 2024]. The marginal distribution over visible configurations reads:

$$p(\mathbf{v}|W, \mathbf{a}, \mathbf{b}) = \frac{1}{Z} \sum_{\mathbf{h}} e^{-H(\mathbf{v}, \mathbf{h}; W, \mathbf{a}, \mathbf{b})} \quad (5.5)$$

where Z is the partition function ensuring proper normalization.

A graphical representation of an RBM is shown in Figure 5.2.

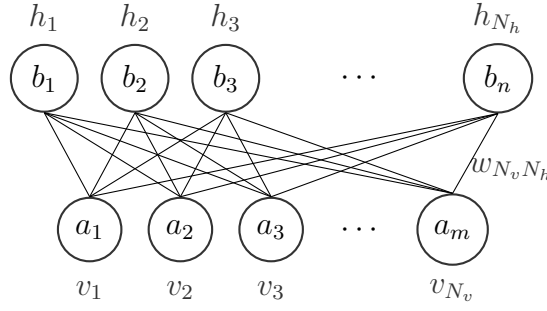


Figure 5.2: Graphical representation of a Restricted Boltzmann Machine

In contrast to deep neural networks, which often behave as black boxes and are difficult to analyze, RBMs benefit from a relatively simple structure that allows for analytical tractability. For instance, their phase diagram has been characterized in several regimes [Delle and Furtlehner, 2021], and the learning dynamics in the early stages of training are well understood. Moreover, mean-field approaches, including the derivation of TAP equations [Gabri  et al., 2015], enable the identification of free energy minima, which can be used for practical purposes such as clustering data into families.

5.1.1 Learning in Restricted Boltzmann Machines

Restricted Boltzmann Machines (RBMs) have emerged as a prominent class of generative models within the broader family of energy-based models. In contrast to classical supervised learning frameworks, RBMs operate in an unsupervised setting, where no label information is available in the dataset. The dataset consists solely of visible node configurations:

$$\mathcal{D}^V = \{\mathbf{v}^\mu\}_{\mu=1}^M,$$

where M denotes the number of data samples.

A primary challenge in training RBMs lies in the absence of observed hidden node configurations. The objective of learning is to adjust the model parameters such that the marginal distribution over visible units $p(\mathbf{v} \mid W, \mathbf{a}, \mathbf{b})$ matches the empirical distribution of the dataset. A standard approach to achieve this consists in maximizing the *log-likelihood*¹:

$$\mathcal{L}^{LL}(W, \mathbf{a}, \mathbf{b}; \mathcal{D}^V) = \frac{1}{M} \sum_{\mu=1}^M \log p(\mathbf{v}^\mu \mid W, \mathbf{a}, \mathbf{b}) \quad (5.6)$$

$$= \frac{1}{M} \sum_{\mu=1}^M \log \left(\sum_{\mathbf{h}} e^{-H(\mathbf{v}^\mu, \mathbf{h}; W, \mathbf{a}, \mathbf{b})} \right) - \log Z. \quad (5.7)$$

Optimization is performed via gradient ascent, where the gradients of the log-likelihood

¹The log-likelihood can be seen as a negative loss function for this reason we denote both the quantities with the same symbol $\mathcal{L}$, but with different apex.

are given by:

$$\frac{\partial \mathcal{L}^{LL}(W, \mathbf{a}, \mathbf{b}; \mathcal{D}^V)}{\partial w_{ij}} = \langle v_i h_j \rangle_{\mathcal{D}^V} - \langle v_i h_j \rangle_{\text{model}}, \quad (5.8)$$

$$\frac{\partial \mathcal{L}^{LL}(W, \mathbf{a}, \mathbf{b}; \mathcal{D}^V)}{\partial a_i} = \langle v_i \rangle_{\mathcal{D}^V} - \langle v_i \rangle_{\text{model}}, \quad (5.9)$$

$$\frac{\partial \mathcal{L}^{LL}(W, \mathbf{a}, \mathbf{b}; \mathcal{D}^V)}{\partial b_j} = \langle h_j \rangle_{\mathcal{D}^V} - \langle h_j \rangle_{\text{model}}, \quad (5.10)$$

with the averages defined as:

$$\langle f(\mathbf{v}, \mathbf{h}) \rangle_{\mathcal{D}^V} = \frac{1}{M} \sum_{\mu=1}^M \sum_{\mathbf{h}} f(\mathbf{v}^\mu, \mathbf{h}) p(\mathbf{h} | \mathbf{v}^\mu) \quad (\text{data-dependent average}), \quad (5.11)$$

$$\langle f(\mathbf{v}, \mathbf{h}) \rangle_{\text{model}} = \sum_{\mathbf{v}, \mathbf{h}} f(\mathbf{v}, \mathbf{h}) p(\mathbf{v}, \mathbf{h} | W, \mathbf{a}, \mathbf{b}) \quad (\text{model average}). \quad (5.12)$$

where $p(\mathbf{v}, \mathbf{h} | W, \mathbf{a}, \mathbf{b}) = \frac{e^{-H(\mathbf{v}, \mathbf{h} | W, \mathbf{a}, \mathbf{b})}}{Z}$, with Z the partition function.

A major computational bottleneck arises from the need to evaluate expectations over the model distribution, which involves the intractable partition function Z . Another source of intractability comes from the difficulty of computing the expectation values in Equation (5.12) given the model. For small models, exact computations are feasible; otherwise, sampling methods are employed. Traditionally, the sampling is performed using MCMC schemes, even if they are usually inefficient.

One such method is *RDM- k* [Decelle et al., 2021], which relies on running a fixed number of MCMC chains initialized at random and using the samples obtained after k MCMC steps. However, this method is practical only if k exceeds the mixing time of the Markov chains, which is often prohibitively large in realistic scenarios.

The most widely adopted MCMC-based scheme is *k -steps Contrastive Divergence* (CD- k), originally introduced by Hinton [Hinton, 2002, Ackley et al., 1985]. The idea is to replace the model average with a short-run Gibbs sampling starting from a data point, similarly to what is done in RDM- k , but with a "hot" initialization. Algorithm 9 outlines the CD- k learning procedure, in its vanilla, non-mean field, version. The original version of CD updates Δw_{ij} using the difference among the averages of $v_i h_j$ computed over the data distribution and the reconstructed distribution [Carreira-Perpiñán and Hinton, 2005, Hinton, 2012]. An analogous reasoning can be performed for visible and hidden biases.

For enlightening the notation in Algorithm 9, we avoid reporting the conditioning on the parameters $(W, \mathbf{a}, \mathbf{b})$ in the algorithm, but it is implicitly understood that all the probabilities are conditioned on these parameters.

Algorithm 9 k -steps Contrastive Divergence (CD- k)

Input: $\mathcal{D}^V = \{\mathbf{v}^\mu\}_{\mu=1}^M$, learning rate η , number of Gibbs steps k , total epochs $T_{\max}$

Initialize: $W, \mathbf{a}, \mathbf{b}$

for epoch = 1 to $T_{\max}$ **do**

for $\mu = 1$ to M **do**

 Sample $\mathbf{h}^{(0)} \sim p(\mathbf{h} \mid \mathbf{v}^\mu)$

$\mathbf{v}^{(0)} \leftarrow \mathbf{v}^\mu$

for $t = 1$ to k **do**

 Sample $\mathbf{v}^{(t)} \sim p(\mathbf{v} \mid \mathbf{h}^{(t-1)})$

 Sample $\mathbf{h}^{(t)} \sim p(\mathbf{h} \mid \mathbf{v}^{(t)})$

end for

 Update parameters

$$\Delta w_{ij} \leftarrow \eta \left(v_i^{(0)} h_j^{(0)} - v_i^{(k)} h_j^{(k)} \right), \quad (5.13)$$

$$\Delta a_i \leftarrow \eta \left(v_i^{(0)} - v_i^{(k)} \right), \quad (5.14)$$

$$\Delta b_j \leftarrow \eta \left(h_j^{(0)} - h_j^{(k)} \right). \quad (5.15)$$

end for

end for

Return: Updated parameters $W, \mathbf{a}, \mathbf{b}$

CD- k approximates the true log-likelihood gradient with a cheaper, truncated-MCMC version. The associated loss function is:

$$\mathcal{L}_{CD}(W, \mathbf{a}, \mathbf{b}; \mathcal{D}^V) = \mathbb{E}_{p(\mathbf{h}|\mathbf{v})p_0(\mathbf{v})}[-H(\mathbf{v}, \mathbf{h})] - \mathbb{E}_{p_k(\mathbf{h}, \mathbf{v})}[-H(\mathbf{v}, \mathbf{h})], \quad (5.16)$$

where $p_0(\mathbf{v})$ is the empirical data distribution and $p_k(\mathbf{v}, \mathbf{h})$ the distribution after k Gibbs steps, [Sutskever and Tieleman, 2010]. It has been shown in [Decelle and Furtlehner, 2021, Carreira-Perpiñán and Hinton, 2005] that CD- k actually minimizes the Kullback-Leibler divergence between the data distribution $p_0(\mathbf{v})$ and the k -step distribution $p_k(\mathbf{v})$:

$$\text{KL}(p_0 \| p_k) = \sum_{\mathbf{v}} p_0(\mathbf{v}) \log \frac{p_0(\mathbf{v})}{p_k(\mathbf{v})}.$$

Despite its efficiency, CD suffers from a fundamental limitation: it explores only the vicinity of training data in configuration space. As a consequence, it poorly estimates the probability of configurations far from the data manifold, introducing "bias" in the learning process [Decelle and Furtlehner, 2021].

A refinement of CD, called *Persistent Contrastive Divergence* (pCD), was introduced in [Tieleman, 2008]. Instead of initializing Gibbs chains at each iteration from data, pCD maintains a persistent set of chains throughout training. These chains are updated a few steps at each iteration and reused across epochs. This mitigates the problem coming from shallow mixing but still suffers when the mixing time of the MCMC chain is large.

Recent efforts aim to eliminate this mixing-time-related problem. For instance, the *Unbiased Contrastive Divergence* algorithm leverages the coupling-from-the-past technique to

produce better gradient estimates [Qiu et al., 2020]. However, such methods are computationally demanding and fall outside the scope of this chapter.

Motivated by the limitations of CD-based methods, in the next sections, we explore an alternative learning strategy based on *pseudo-likelihood* maximization for training RBMs.

5.2 Learning graphical models via interaction screening and pseudo likelihood optimization

As discussed earlier, two main challenges arise when learning graphical models using MCMC methods: the computation of the partition function, typically circumvented using the contrastive divergence method, and the statistical bias, coming from the estimation of the parameters, with CD- k , with k smaller than the mixing time of the MCMC chain [Tieleman, 2008, Qiu et al., 2020]. These two fundamental issues can be addressed through learning methods based on *pseudo-likelihood estimators* and *interaction screening objective* [Lokhov et al., 2018, Vuffray et al., 2016]. In this section, we introduce pseudo-likelihood-based and interaction-screening-based learning methods and explore their challenges in the context of graphical models. Our analysis begins with a description of how to use these methods for reconstructing Ising models parameters. Once these methods are thoroughly explained and their properties well-defined, we generalize the approach to learning Restricted Boltzmann Machines (RBMs).

5.2.1 Learning Ising Models

We are going to start by introducing the pseudo-likelihood and interaction screening based learning methods for Ising models, whose energy function is described in Equation (5.2).

Given a set of configurations, generated from an Ising model with parameters $\{\Theta^*, \mathbf{u}^*\}$, we want to infer the couplings and fields. The first approach to this inference problem consisted of optimizing the log-likelihood of the model. However, as previously stated, this problem is generally intractable due to the necessity of computing the partition function Z . The computation of Z is exponential in complexity with respect to the number of spins [Cooper, 1990], except in special cases such as tree-structured Ising models [Chow and Liu, 1968] and planar Ising models with zero magnetic fields [Johnson et al., 2010]. Despite this, one can still attempt to optimize the log-likelihood using MCMC methods, performing gradient ascent on the couplings and fields. The Monte Carlo nature of the method makes it expensive, as it requires a large number of runs to achieve a predefined accuracy. However, this approach is asymptotically exact as the number of samples tends to infinity, illustrating that "sufficient statistics" methods that rely only on estimates of first moments and pairwise correlations of spins can achieve exact reconstruction, albeit at the cost of exponential computational complexity. These methods, however, perform poorly in low-dimensional systems or in spin-glass phases, where fluctuations are crucial and cannot be neglected [Lokhov et al., 2018]. While sufficient statistics, such as the first and second moments of the data, contain all the information necessary to estimate the couplings, the computations required to extract this information are computationally expensive and prohibitive for large systems [Ackley et al., 1985].

An alternative approach to optimizing the couplings and fields in Ising models is based on the maximization of local Regularized Pseudo-Likelihood Estimators (RPLE) [Ravikumar

et al., 2010]. Each Regularized Pseudo-Likelihood estimator can be viewed as the regularized probability of the spin x_i conditioned on the remaining $N - 1$ spins [Lokhov et al., 2018]. The loss function in this context is defined as follows:

$$\mathcal{L}_i^{\text{RPLE}}(\{\theta_{ij}\}_{j \neq i}, u_i; \mathcal{D}) = -\mathbb{E}_{\mathcal{D}} \left[\ln \left(\frac{1}{1 + e^{-2x_i(u_i + \sum_{j \neq i} \theta_{ij} x_j)}} \right) \right] + \lambda \|\boldsymbol{\theta}_i\|_1 \quad (5.17)$$

where u_i is the local field of spin i , $\boldsymbol{\theta}_i$ is the vector of couplings acting on x_i (excluding the self-interaction), and λ represents the strength of the L_1 regularization, which encourages sparsity in the interaction matrix Θ . The expectation is taken over the dataset $\mathcal{D} = \{\mathbf{x}^\mu\}_{\mu=1, \dots, M}$, where $\mathbf{x}^\mu$ are the observed configurations of the Ising model. The first term in the loss function is the pseudo-likelihood, which approximates the likelihood of observing the spin x_i given its neighbors. The second term is a regularization term that penalizes large values of the couplings, promoting sparsity in the learned model.

Proving perfect graph reconstruction via RPLE relies on a rather restrictive set of conditions [Ravikumar et al., 2010]. Models that satisfy these conditions are typically high-temperature ferromagnetic models, but the procedure fails in other regimes, most notably at low temperatures [Montanari and Pereira, 2009]. [Lokhov et al., 2018] showed that RPLE is an exact method, and provided reconstruction guarantees without any extra conditions.

In [Lokhov et al., 2018] and [Vuffray et al., 2016], a different formulation for Ising parameters reconstruction is introduced, called *Regularized Interaction Screening Estimators* (RISE), or its logarithmic version, *LogRISE*. The RISE method minimizes the following local loss functions:

$$\mathcal{L}_i^{\text{RISE}}(\{\theta_{ij}\}_{j \neq i}, u_i; \mathcal{D}) = S_i(\{\theta_{ij}\}_{j \neq i}, u_i; \mathcal{D}) + \lambda \|\boldsymbol{\theta}_i\|_1 \quad (5.18)$$

where

$$S_i(\{\theta_{ij}\}_{j \neq i}, u_i; \mathcal{D}) = \mathbb{E}_{\mathcal{D}} \left[e^{-x_i(u_i + \sum_{j \neq i} \theta_{ij} x_j)} \right] \quad (5.19)$$

is the *Interaction Screening Objective (ISO)*. It represents the empirical average of the inverse factors in the Gibbs distribution, and its screening property becomes evident in the limit of a large number of samples. Changing the values of the couplings and fields in the ISO alters the effective interaction strength between a spin and its neighbors. When the model parameters match the true ones, the ISO effectively screens this interaction, rendering the spin conditionally independent of its neighbors [Lokhov et al., 2018].

In [Lokhov et al., 2018], the authors show that the L_2 -norm of the reconstruction error for the couplings is bounded with high probability as:

$$\|\hat{\boldsymbol{\theta}}_i^{\text{RISE}} - \boldsymbol{\theta}_i^*\|_2 \leq C'_d e^{3\beta d} \sqrt{\frac{\ln(N^2/\epsilon)}{M}} \quad (5.20)$$

$$\|\hat{\boldsymbol{\theta}}_i^{\text{RPLE}} - \boldsymbol{\theta}_i^*\|_2 \leq C_d e^{4\beta d} \sqrt{\frac{\ln(N^2/\epsilon)}{M}} \quad (5.21)$$

where C_d and C'_d are constants depending on the degree of the graph, d is the bounding degree of the graph, $\beta = \max_{i,j} |\theta_{i,j}^*|$ is the maximum coupling of the graph, M is the number of samples and N is the number of variables. This bound occurs with probability $1 - \epsilon/N$. In [Jayakumar et al., 2024], Jayakumar et al. propose a similar bound for the L_2 -norm of the fields in the case of the RPLe. This result is quite strong: the bias decreases as the number of training configurations increases, reducing the computational complexity of the learning process. Another significant advantage of this approach is that it does not require the computation of the partition function Z .

5.2.2 Learning Restricted Boltzmann Machines

Building on these results, we introduce a new approach based on interaction screening for learning the parameters of Restricted Boltzmann Machines (RBMs). The primary challenge in generalizing this approach to RBMs consists in the fact that the dataset $\mathcal{D}^V = \{\mathbf{v}^\mu\}_{\mu=1,\dots,M}$ contains solely visible configurations. The first step is to complete the Ising samples, by adding sampled hidden values $\{\mathbf{h}^\mu\}_{\mu=1,\dots,M}$ to $\mathcal{D}$. To address this, we propose a method inspired by Contrastive Divergence: after the system is randomly initialized, so that one has initial guesses for the fields and the couplings, we use the hidden nodes' conditional probability distribution to construct a proposal for the hidden configurations. Once the full configuration of the graphical model is available, we can reconstruct the couplings and fields using the LogRISE formulation.

The method described above can reconstruct the effective Ising model associated with full configurations $\mathcal{D} = \{\mathbf{x}^\mu\}_{\mu=1,\dots,M} = \{\mathbf{v}^\mu, \mathbf{h}^\mu\}_{\mu=1,\dots,M}$, but it requires a way to force the RISE learning scheme to reconstruct a Restricted Boltzmann Machine, rather than a fully connected Ising model. In other words, one needs to make sure that the couplings belonging to the same partition are zero (visible-visible and hidden-hidden interactions should be null). To perform this task, one can introduce a regularization term over the *visible-visible* couplings, and keep the hidden-hidden couplings fixed to zero. The only surviving interactions should be those between visible and hidden nodes, reflecting the bipartite structure of RBMs. The final loss function per site i that we aim to minimize, specifically in the case of the LogRISE formulation, is:

$$\mathcal{L}_i^{\text{LogRISE}}(\{\theta_{ij}\}_{j \neq i}, u_i; \mathcal{D}) = \log S_i(\{\theta_{ij}\}_{j \neq i}, u_i; \mathcal{D}) + \lambda \|\boldsymbol{\theta}_i^{(vv)}\|_1 \quad (5.22)$$

where $S_i(\{\theta_{ij}\}_{j \neq i}, u_i; \mathcal{D})$ is the Interaction Screening Objective, and $\boldsymbol{\theta}_i^{(vv)} = \{\theta_{i,j}\}_{j \in V}$ represents the set of couplings of visible node i , with other visible nodes (where V is the set of visible nodes, $\Theta^{(vv)} = \{\theta_i^{(vv)}\}_{i \in V}$ ²). It is important to stress that the hidden-hidden interactions are excluded from the optimization: they are not updated and are explicitly fixed to zero throughout the learning procedure.

This approach has the advantage of not relying on MCMC methods, which require truncation of the evolutionary process, and instead uses LogRISE. As a result, we can control the "bias" by adjusting the number of samples.

The pseudocode for this learning procedure for RBMs is presented in Algorithm 10:

²The model here is optimized as an Ising model where $\Theta = \begin{bmatrix} \Theta^{(vv)} & W \\ W^T & \Theta^{(hh)} \end{bmatrix}$, and $\mathbf{u} = \{\mathbf{a}, \mathbf{b}\}$.

Algorithm 10 LogRISE for RBMs

Input: $\mathcal{D}^V = \{\mathbf{v}^\mu\}_{\mu=1}^M$, η , k , T , λ_0 , $\delta\lambda = \lambda_0/T$, N_v , N_h ($N = N_v + N_h$)

Initialization: $\Theta^{(0)} = \begin{bmatrix} \Theta^{(vv)(0)} & W^{(0)} \\ W^{(0)T} & \Theta^{(hh)(0)} \end{bmatrix}$, $\mathbf{u}^{(0)} = \{a^{(0)}, b^{(0)}\}$

for $t = 1$ **to** T **do**

 Fix $\lambda = \lambda_0 - t\delta\lambda$

 Compute $p(\mathbf{h} \mid \mathbf{v}, \mathbf{u}^{(t-1)}, \Theta^{(vh)(t-1)})$

 Sample $\mathbf{h}^\mu \sim p(\mathbf{h} \mid \mathbf{v}^\mu, \mathbf{u}^{(t-1)}, \Theta^{(vh)(t-1)})$ for $\mu \in \{1, \dots, M\}$

 Given $\mathcal{D} = \{\mathbf{v}^\mu, \mathbf{h}^\mu\}_{\mu=1, \dots, M}$, minimize

$$\mathcal{L}(\Theta^{(t-1)}, \mathbf{u}^{(t-1)}; \mathcal{D}) = \frac{1}{N} \sum_{i=1}^N \mathcal{L}_i^{\text{LogRISE}}(\{\theta_{ij}^{(t-1)}\}_{j \neq i}, u_i^{(t-1)}; \mathcal{D}) \quad (5.23)$$

 with respect to the couplings $\Theta^{(t)}$ and fields $\mathbf{u}^{(t)}$, using a GD based method

end for

Return: $\Theta^{(T)}, \mathbf{u}^{(T)}$

The optimization of the LogRISE loss function is performed using three different gradient-based algorithms: *Proximal Gradient Descent* [Daubechies et al., 2004], *Entropic Gradient Descent* [Pittorino et al., 2021, Vuffray et al., 2022], and a second-order optimization method (IPOPT) via the *JuMP* library [Dunning et al., 2017]. It is important to note that, due to the nature of the RBM reconstruction problem from visible configurations, each optimization algorithm reconstructs a different RBM. A natural question arises: how can one check whether the reconstructed model accurately represents the data?

This question translates to defining appropriate metrics for evaluating the performance of the learning scheme. The only accessible element for performance assessment is the marginal distribution of the visible nodes. We now present the main metric for evaluation: *Total Variation distance (TV)* among the empirical marginal distribution of visible nodes ($p_{\mathcal{D}^V}$) and reconstructed visible nodes marginal distribution (p_{LogRISE}):

$$d_{\text{TV}}(P, Q) = \sup_{A \subseteq \mathcal{X}} |p_{\mathcal{D}^V}(A) - p_{\text{LogRISE}}(A)| \quad (5.24)$$

$$= \frac{1}{2} \sum_{\mathbf{v}} |p_{\mathcal{D}^V}(\mathbf{v}) - p_{\text{LogRISE}}(\mathbf{v})| \quad (5.25)$$

where the sum is performed over the set of all possible configurations $\mathbf{v}$.

5.3 Numerical Results

This section presents preliminary results on reconstructing RBM models using LogRISE-based learning in a controlled, artificial setting.

For each problem instance, data are generated from an RBM having zero visible and hidden fields, and whose couplings are i.i.d. sampled from a Gaussian distribution with zero mean and variance $1/9$. We have $N_v = 5$ visible nodes and $N_h = 6$ hidden nodes³. The dataset consists of $M = 10^4$ samples, each representing a configuration of the visible nodes.

³All the analysis of this section is performed with these choices N_v and N_h

5.3.1 Total variation distance evolution

Figure 5.3 (left panel) shows the Total Variation (TV) distance between the true marginal distribution of the visible nodes and the estimated one for different learning schemes (based on PGD, EGD, and JuMP). The regularization parameter evolves according to $\lambda(t) = \lambda_0 - \delta\lambda t$, where $\delta\lambda = \lambda_0/T$, $t \in [1, \dots, T]$, and $T = 100$ is the total number of learning iterations, with $\lambda_0 = 10.0$. The number of gradient descent (GD) steps is fixed at 100. For each learning scheme, we report the evolution of the TV distance over time, averaged over 10 different problem instances (solid line), with the shaded area indicating one standard deviation distance from the average.

The right panel of Figure 5.3 shows the heat map of the couplings of the RBM reconstructed at the end of the learning procedure of a single instance using the EGD scheme. At convergence, the visible-visible couplings are nearly zero, while the visible-hidden couplings remain nonzero. The color scale is normalized to the maximum coupling value.

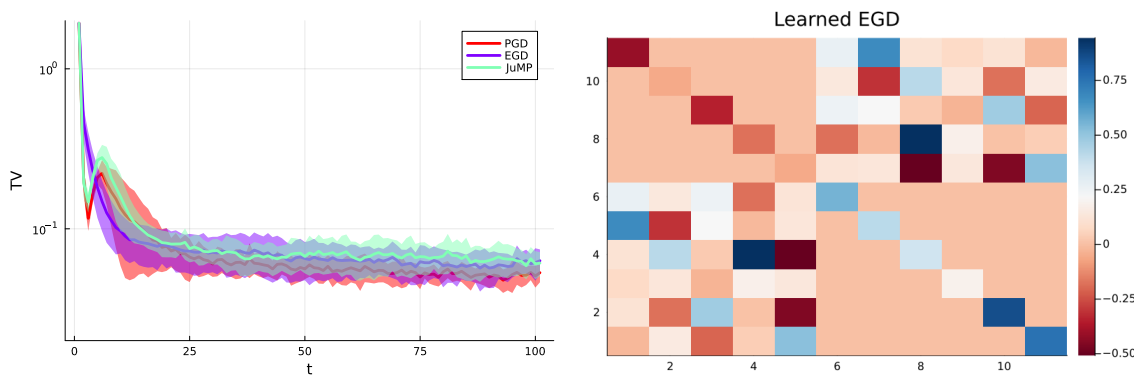


Figure 5.3: *Left panel:* Total Variation distance between the marginal distribution of the visible nodes in the data and the estimated one, for three different instances of a 5-visible, 6-hidden node RBM. The dataset consists of $M = 10^4$ configurations. Hyperparameters are set to $\lambda_0 = 10.0$ and $T = 100$. The regularization evolves as $\lambda(t) = \lambda_0 - \delta\lambda t$, with $\delta\lambda = \lambda_0/T$. The number of GD steps is fixed at 100. The solid line shows the mean over 10 runs, and the shaded area represents one standard deviation. *Right panel:* Heat map of the couplings of the RBM reconstructed at the end of learning a random instance using the EGD scheme. The color scale is normalized to the maximum coupling value.

As shown in Figure 5.3, the proposed scheme in Section 5.2.2 successfully learns an RBM, as evidenced by the decreasing TV distance over time. For this small model and dataset size, all three learning schemes achieve similar performance in terms of TV distance. However, in terms of computational time, PGD and EGD are about five times faster than JuMP optimization with IPOPT. This is expected, as IPOPT is a second-order optimization method that requires computing the full loss Hessian, which is computationally expensive. Although it is not optimal for RBM learning, it is included in the initial analysis due to the precision of the solutions it provides.

5.3.2 Dependence on Dataset Size

We expect the Total Variation distance to decrease as the number of data points M increases, a trend also observed in [Lokhov et al., 2018] for Ising model reconstruction.

In the left panel of Figure 5.4, we report the final mean TV distance for 10 different problem instances as a function of M , with error bars indicating one standard deviation distance from the mean. The number of GD steps is fixed at 100 to ensure sufficient optimization at each t . Hyperparameters are set to $\lambda_0 = 10.0$ and $T = 100$, and regularization annealing follows $\lambda(t) = \lambda_0 - \delta\lambda t$, with $\delta\lambda = \lambda_0/T$.

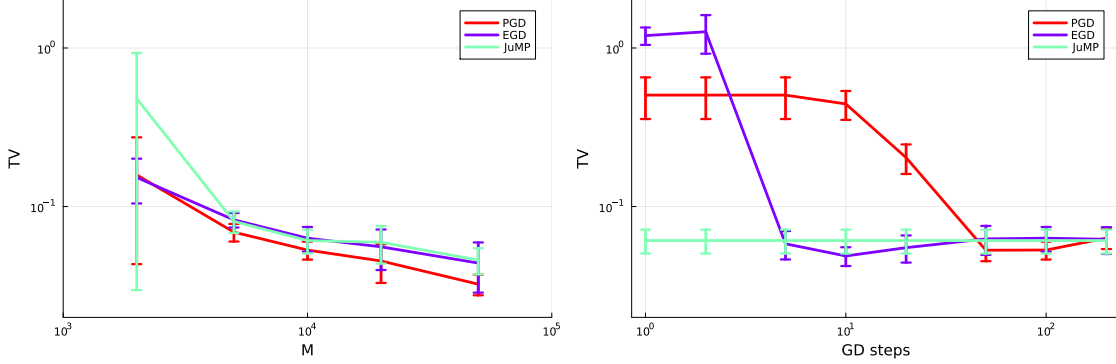


Figure 5.4: *Left panel:* Final value of the Total Variation distance as a function of dataset size M . Hyperparameters are set to $\lambda_0 = 10.0$ and $T = 100$, with 100 GD steps. Regularization annealing is $\lambda(t) = \lambda_0 - \delta\lambda t$. The line shows the mean over 10 runs, and error bars indicate one standard deviation. *Right panel:* Final value of the TV distance as a function of the number of GD steps in the optimization of the LogRISE loss. Here, $M = 10^4$ and GD steps vary.

The left panel of Figure 5.4 demonstrates that learning performance improves with increasing dataset size, largely independent of the optimization algorithm used. The three optimization schemes yield comparable results.

5.3.3 Dependence on the Number of GD Steps

In the previous section, we showed that increasing M improves learning performance, as expected. It is also important to assess how the number of optimization steps affects performance. One may ask whether full optimization of the pseudo-likelihood loss is necessary for convergence, or if fewer steps suffice. This analysis is primarily relevant to GD-based algorithms, but results for the IPOPT scheme are also reported for comparison. In IPOPT, the number of GD steps is not explicitly defined.

To investigate this, we report the evolution of the TV distance as a function of the number of GD steps in the right panel of Figure 5.4, for $\lambda(t) = \lambda_0 - \delta\lambda t$, $T = 100$, $M = 10^4$ and $\lambda_0 = 10.0$. The lines represent the mean TV distance over 10 problem instances, with error bars indicating one standard deviation.

Comparing PGD and EGD, it is clear that EGD reconstructs the RBM parameters with fewer GD steps, while PGD requires more iterations to reach similar performance. This is consistent with previous findings [Pittorino et al., 2021, Vuffray et al., 2022], where EGD is shown to converge faster than other GD-based methods. For a sufficiently large number of GD steps, both methods achieve similar performance, comparable to IPOPT. All methods reach a plateau in TV distance, indicating that the reconstruction error is ultimately limited by the dataset size M , rather than the number of GD steps, as also observed in [Lokhov et al., 2018].

5.3.4 Dependence on Regularization strength

It is also relevant to investigate whether the regularization strength affects learning performance. To this end, we fix the number of GD steps to 100 and consider different initial values of λ_0 in $[0.1, 1, 10, 100]$. Regularization evolves as $\lambda(t) = \lambda_0 - \delta\lambda t$, with $T = 100$ and $M = 10^4$. Results are shown in Figure 5.5. For all algorithms, the initial value of λ_0 does not significantly affect learning performance, as the TV distance remains nearly constant for different λ_0 (We report the average value of the TV over 10 instances for each λ_0 together with their standard deviation). The central and right panels of Figure 5.5 show the couplings of the RBM reconstructed at the end of learning using IPOPT for $\lambda_0 = 0.1$ and $\lambda_0 = 1$, respectively. For $\lambda_0 = 0.1$, the visible-visible couplings are nonzero, highlighting the impact of regularization. For $\lambda_0 = 1$, these couplings are nearly zero. This suggests that the annealing schedule and initial value of λ_0 do not significantly affect performance, provided the initial value is sufficiently large.

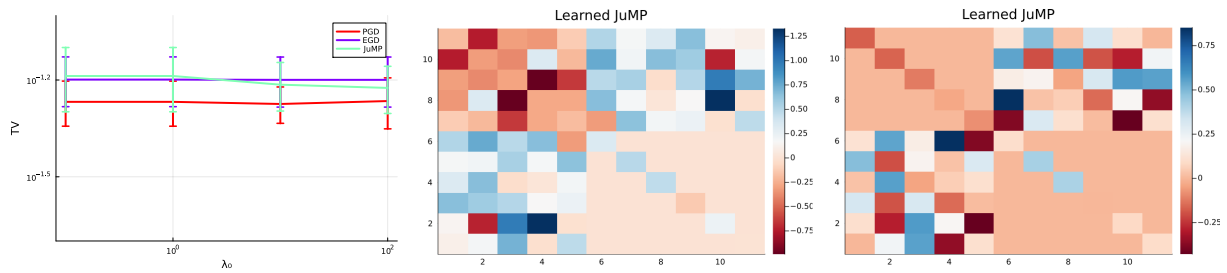


Figure 5.5: *Left panel:* Total Variation distance as a function of the initial value of λ_0 . The line shows the mean over 10 runs, and error bars indicate one standard deviation. $T = 100$ and $M = 10^4$. Regularization evolves as $\lambda(t) = \lambda_0 - \delta\lambda t$. The number of GD steps is fixed at 100. *Central panel:* Heat map of the couplings of the RBM reconstructed for a single instance of the problem at the end of learning using JuMP for $\lambda_0 = 0.1$. *Right panel:* Heat map of the couplings for $\lambda_0 = 1$. The color scale is normalized to the maximum coupling value.

5.4 Recap of the results

In this chapter, we presented some preliminary results for learning parameters of Restricted Boltzmann Machines (RBMs) using the LogRISE formulation:

- We presented a novel approach for learning RBMs based on Pseudo Likelihood maximization or ISO minimization, which doesn't share the same limitation as MCMC-based methods: requiring the length of the MCMC chain to be larger than its mixing time. The optimization of the new loss functions is performed using three different algorithms: Proximal Gradient Descent (PGD), Entropic Gradient Descent (EGD), and a second-order optimization method (IPOPT) via the JuMP library.
- We showed that the proposed scheme can learn the parameters of an RBM, as shown by the Total Variation distance evolution. The performance of interaction screening-based learning has been investigated as a function of the dataset size, the number of gradient descent steps, and the regularization strength. As expected, larger datasets lead to improved reconstruction accuracy, consistent with previous findings in the Ising model literature. Interestingly, we found that EGD maintains good performance even with a small number of gradient steps per iteration, showing its rapid convergence properties. The initial value of the regularization parameter λ_0 does not significantly affect the performance of the learning scheme, as long as it is sufficiently large, to guarantee a correct reconstruction of an RBM.

Chapter 6

Conclusions and Future Directions

In this thesis we employed tools from statistical physics, namely the Replica Method, Approximate Message Passing, and Random Matrix Theory, to analyse a range of inference and learning problems in machine learning. These problems share two main difficulties: highly non-convex loss landscapes and a large number of parameters, which statistical physics helps to solve.

Chapter 2 addresses fair sampling solutions of non-convex optimisation tasks. We introduced the *Stochastic Localization* scheme and applied it to the negative and binary perceptrons. Using the replica method, one can derive SL failing modes in perceptron problems: whereas SL samples uniformly from the flat measure of the negative perceptron (for a wide range of α depending on the margin), it fails sampling binary solutions. We therefore designed a *tunable* measure, enabling efficient sampling of binary solutions. Two efficient samplers are then constructed: the ASL scheme, where AMP supplies the first moment of the tilted distribution, and the τ -annealing MCMC scheme relaxing the i.i.d. requirements of AMP. Both provide samples from the tuned binary perceptron distribution. However, they have some limitations: ASL is limited to i.i.d. data and τ -annealing may suffer from slow mixing. Moreover, the impossibility to sample the binary-perceptron solution from the flat measure comes from the clustered nature of its solution space. Understanding why the tunable binary perceptron distribution enables efficient sampling remains an open challenge.

Chapter 3 tackles the real-valued Phase Retrieval problem, where one must reconstruct a signal from noiseless, phaseless measurements. We proposed the *Phase Selection* framework, which separates the Phase Retrieval task into a sign inference and signal reconstruction subtasks. An analysis of Phase Selection is performed using the replica formalism, under the Replica Symmetric ansatz. Guided by the theoretical analysis, we built three solvers for Phase Selection instances, based on Simulated Annealing, Approximate Message Passing, and Langevin dynamics on a continuous-sign relaxation. They indeed succeed in solving Phase Selection instances up to α comparable with the Bayes optimal prediction. Our work focuses on analyzing the Phase Selection framework in the absence of measurement noise. Future work will add measurement noise and will consider more realistic datasets.

Chapter 4 studies neural-network loss geometry through its Hessian spectrum. For a two-layer *tree committee machine* we derived, under an RS ansatz, the quenched spectrum of the square and cross entropy loss, assuming the distribution of first layer weights to follow the Gibbs probability measure with inverse temperature β . The case $\beta \sim 0$ reproduces a uniform weight prior, while $\beta \rightarrow \infty$ corresponds to a distribution giving mass to configurations minimizing of the loss. This theoretical framework can help to explain several experimental

observations. However, as a limitation, our approach considered only random data, without structure. The next step of our geometrical investigation includes extending the analysis from pure storage tasks to more structured data, potentially explaining several experimental regularities reported in the literature.

Chapter 5 revisits learning in Restricted Boltzmann Machines. Standard algorithms, like Contrastive Divergence and Persistent CD, can struggle because of slow MCMC mixing times. We therefore adopted a learning approach based on the optimization of regularized pseudo-likelihood and regularized interaction screening objective. We first test their performances in the inverse-Ising setting and then we use them for learning RBMs. Experiments on synthetic data from small RBMs showed the efficacy of these learning methods. We quantified the dependence on the regularisation strength, the sample size, and the number of optimisation steps. Future work will benchmark this new learning methods against state of the art methods on real-world data, such as images.

This thesis demonstrates how statistical-physics techniques can illuminate the complexity of modern machine-learning models. Understanding the effect of the tuned distribution, pushing Phase Selection into noisy regimes, refining Hessian analyses for realistic inputs, and benchmarking interaction screening learning to large RBMs are only the first milestones. Much is left to be discovered, and statistical physics can help us.

Appendix A

Stochastic Localization Appendix

A.1 Replica Computation for ASL on the Perceptron

A.1.1 General framework

In this Section, we outline the key steps of the replica calculation used to derive the time-dependent free-entropy ϕ_t of the Stochastic Localization process, as outlined in Section 2.3. The formalism is applied to the general perceptron model given in Equation (2.14), and then specialized on the spherical and binary case in Appendix A.3 and Appendix A.4 respectively.

We assume $M = \alpha N$ Gaussian-distributed examples, $X = \{\mathbf{x}^\mu\}_{\mu=1}^M$, $\mathbf{x}^\mu \sim \mathcal{N}(\mathbf{0}, I_N)$. The weights have a factorized prior $P(\mathbf{w}) = \prod_i P(w_i)$, where we slightly abused the notation. The case of the non-factorizable global spherical prior is handled naturally by setting $P(w) = 1$ but then constraining the diagonal overlap instead of optimizing it, as explained later.

We want to compute the free entropy at time t of the model in Equation (2.14) averaged over the realization of the field vector in the SL process (Equation (2.6)):

$$\phi_t = \lim_{N \rightarrow +\infty} \frac{1}{N} \mathbb{E} \log Z_{\mathbf{h}_t, t}. \quad (\text{A.1})$$

This computation is performed using the replica trick twice (the second one is used to express the expectation over the reference configurations):

$$\mathbb{E} \log Z = \lim_{n \rightarrow 0} \partial_n \mathbb{E} Z^n \quad \text{and} \quad Z^{-1} = \lim_{s \rightarrow 0} Z^{s-1}.$$

We can thus write

$$\phi_t = \lim_{N \rightarrow +\infty} \frac{1}{N} \lim_{s \rightarrow 0} \lim_{n \rightarrow 0} \partial_n \mathbb{E}_{\psi, \mathbf{g}} \int \prod_{\alpha=1}^s \psi(d\mathbf{w}_\alpha^*) \prod_{a=1}^n \psi(d\mathbf{w}_a) e^{\langle \mathbf{h}_t, \mathbf{w}_a \rangle - \frac{t}{2} \|\mathbf{w}_a\|^2}, \quad (\text{A.2})$$

where $\mathbf{h}_t = t\mathbf{w}_1^* + \sqrt{t}\mathbf{g}$, with $\mathbf{g} \sim \mathcal{N}(0, I_N)$. For convenience, we define the replicated partition function

$$Z_t^{n,s} = \int \prod_{\alpha=1}^s \psi(d\mathbf{w}_\alpha^*) \prod_{a=1}^n \psi(d\mathbf{w}_a) e^{\langle \mathbf{h}_t, \mathbf{w}_a \rangle - \frac{t}{2} \|\mathbf{w}_a\|^2}, \quad (\text{A.3})$$

Using the definition of the SL fields, the average replicated partition function reads

$$\mathbb{E}Z_t^{n,s} = \mathbb{E}_{X,\mathbf{g}} \int \prod_{\alpha=1}^s P(d\mathbf{w}_\alpha^*) \prod_{a=1}^n P(d\mathbf{w}_a) \prod_{\mu\alpha} \tilde{\Theta} \left(\sum_i \frac{w_{\alpha i}^* x_i^\mu}{\sqrt{N}} \right) \prod_{\mu a} \tilde{\Theta} \left(\sum_i \frac{w_{ai} x_i^\mu}{\sqrt{N}} \right) \times \quad (\text{A.4})$$

$$\times e^{t \sum_{ai} w_{1i}^* w_{ai} + \sqrt{t} \sum_{ai} g_i w_{ai} - \frac{1}{2} t \sum_{ai} w_{ai}^2}. \quad (\text{A.5})$$

where we used the argument that the realization of ψ depends from the realization of the dataset X , and $\tilde{\Theta}$ is defined as

$$\tilde{\Theta}(s) = \Theta(s - \kappa) e^{-\frac{1}{T} U(s - \kappa)}. \quad (\text{A.6})$$

Here $\Theta(s)$ is the Heaviside theta function. Introducing $\lambda_\alpha^\mu = \frac{\mathbf{w}_\alpha^* \cdot \mathbf{x}^\mu}{\sqrt{N}}$ and $u_a^\mu = \frac{\mathbf{w}_a \cdot \mathbf{x}^\mu}{\sqrt{N}}$, and their conjugate Lagrange multipliers, we obtain

$$\begin{aligned} \mathbb{E}Z_t^{n,s} = \mathbb{E}_{X,\mathbf{g}} \int \prod_{\alpha=1}^s P(d\mathbf{w}_\alpha^*) \prod_{a=1}^n P(d\mathbf{w}_a) \prod_{\mu\alpha} \frac{d\lambda_\alpha^\mu d\hat{\lambda}_\alpha^\mu}{2\pi} \prod_{\mu a} \frac{du_a^\mu d\hat{u}_a^\mu}{2\pi} \prod_{\mu\alpha} \tilde{\Theta}(\lambda_\alpha^\mu) \prod_{\mu a} \tilde{\Theta}(u_a^\mu) \quad (\text{A.7}) \\ \times e^{-i \sum_{\mu\alpha} \hat{\lambda}_\alpha^\mu \lambda_\alpha^\mu - i \sum_{\mu a} \hat{u}_a^\mu u_a^\mu + i \sum_{\mu\alpha i} \hat{\lambda}_\alpha^\mu \frac{w_{\alpha i}^* x_i^\mu}{\sqrt{N}} + i \sum_{\mu a i} \hat{u}_a^\mu \frac{w_{ai} x_i^\mu}{\sqrt{N}} + t \sum_{ai} w_{1i}^* w_{ai} + \sqrt{t} \sum_{ai} g_i w_{ai} - \frac{1}{2} t \sum_{ai} w_{ai}^2}. \end{aligned} \quad (\text{A.8})$$

It is now possible to compute the average over the dataset X and over the Gaussian noise $\mathbf{g}$, obtaining

$$\mathbb{E}Z_t^{n,s} = \int \prod_{\alpha=1}^s P(d\mathbf{w}_\alpha^*) \prod_{a=1}^n P(d\mathbf{w}_a) \prod_{\mu\alpha} \frac{d\lambda_\alpha^\mu d\hat{\lambda}_\alpha^\mu}{2\pi} \prod_{\mu a} \frac{du_a^\mu d\hat{u}_a^\mu}{2\pi} \prod_{\mu\alpha} \tilde{\Theta}(\lambda_\alpha^\mu) \prod_{\mu a} \tilde{\Theta}(u_a^\mu) \quad (\text{A.9})$$

$$\times e^{-i \sum_{\mu\alpha} \hat{\lambda}_\alpha^\mu \lambda_\alpha^\mu - i \sum_{\mu a} \hat{u}_a^\mu u_a^\mu + t \sum_{ai} w_{1i}^* w_{ai} + \frac{1}{2} t \sum_i \sum_{ab} w_{ai} w_{bi} - \frac{1}{2} t \sum_{ai} w_{ai}^2} \quad (\text{A.10})$$

$$\times e^{-\frac{1}{2N} \sum_{\mu i} (\sum_{\alpha\beta} \hat{\lambda}_\alpha^\mu \hat{\lambda}_\beta^\mu w_{\alpha i}^* w_{\beta i} + \sum_{ab} \hat{u}_a^\mu \hat{u}_b^\mu w_{ai} w_{bi} + 2 \sum_{\alpha a} \hat{\lambda}_\alpha^\mu \hat{u}_a^\mu w_{\alpha i}^* w_{ai})}. \quad (\text{A.11})$$

We now introduce the overlap parameters

$$q_{ab} = \frac{\mathbf{w}_a \cdot \mathbf{w}_b}{N}, \quad r_{\alpha\beta} = \frac{\mathbf{w}_\alpha^* \cdot \mathbf{w}_\beta^*}{N}, \quad p_{\alpha a} = \frac{\mathbf{w}_\alpha^* \cdot \mathbf{w}_a}{N}, \quad (\text{A.12})$$

where q_{ab} is the overlap between replicas of the tilted distribution, $r_{\alpha\beta}$ is an overlap in the reference system, and $p_{\alpha a}$ is a cross-overlap between reference and tilted system. We also introduce the conjugate parameters $\hat{q}_{ab}, \hat{r}_{\alpha\beta}, \hat{p}_{\alpha a}$ to enforce their definitions.

The overlaps q_{aa} and $r_{\alpha\alpha}$ will be fixed to 1, since $\|\mathbf{w}_b\|^2 = N$ in both the spherical and binary case. For the spherical case, the diagonal multipliers $\hat{q}_{aa}$ will have to be optimized, while their value will be irrelevant in the binary case.

With the introduction of the overlaps, and using the factorization of $P(\mathbf{w})$, we can write

the average replicated partition function as

$$\mathbb{E}Z_t^{n,s} = \int \prod_{\alpha=1}^s \prod_i P(dw_{\alpha i}^*) \prod_{a=1}^n P(w_{ai}) \prod_{\alpha \leq \beta} dr_{\alpha\beta} d\hat{r}_{\alpha\beta} \prod_{a \leq b} dq_{ab} d\hat{q}_{ab} \prod_{\alpha a} dp_{\alpha a} d\hat{p}_{\alpha a} \quad (\text{A.13})$$

$$\times e^{-N \frac{1}{2} \sum_{\alpha\beta} \hat{r}_{\alpha\beta} r_{\alpha\beta} - \frac{1}{2} N \sum_{ab} \hat{q}_{ab} q_{ab} - N \sum_{\alpha a} \hat{p}_{\alpha a} p_{\alpha a}} \quad (\text{A.14})$$

$$\times e^{+\frac{1}{2} \sum_{\alpha\beta} \hat{r}_{\alpha\beta} \sum_i w_{\alpha i}^* w_{\beta i}^* + \frac{1}{2} \sum_{ab} \hat{q}_{ab} \sum_i w_{ai} w_{bi} + \sum_{\alpha a} \hat{p}_{\alpha a} \sum_i w_{\alpha i}^* w_{ai}} \quad (\text{A.15})$$

$$\times \int \prod_{\mu\alpha} \frac{d\lambda_{\alpha}^{\mu} d\hat{\lambda}_{\alpha}^{\mu}}{2\pi} \prod_{\mu a} \frac{du_a^{\mu} d\hat{u}_a^{\mu}}{2\pi} \prod_{\mu\alpha} \tilde{\Theta}(\lambda_{\alpha}^{\mu}) \prod_{\mu a} \tilde{\Theta}(u_a^{\mu}) \quad (\text{A.16})$$

$$\times e^{-i \sum_{\mu\alpha} \hat{\lambda}_{\alpha}^{\mu} \lambda_{\alpha}^{\mu} - i \sum_{\mu a} \hat{u}_a^{\mu} u_a^{\mu} + t N \sum_a p_{1a} + \frac{1}{2} t N \sum_{ab} q_{ab} - \frac{t}{2} N \sum_a q_{aa}} \quad (\text{A.17})$$

$$\times e^{-\frac{1}{2} \sum_{\mu} (\sum_{\alpha\beta} \hat{\lambda}_{\alpha}^{\mu} \hat{\lambda}_{\beta}^{\mu} r_{\alpha\beta} + \sum_{ab} \hat{u}_a^{\mu} \hat{u}_b^{\mu} q_{ab} + 2 \sum_{\alpha a} \hat{\lambda}_{\alpha}^{\mu} \hat{u}_a^{\mu} p_{\alpha a})}. \quad (\text{A.18})$$

Site-dependent and pattern-dependent terms are now decoupled and factorized. A few more straightforward steps lead to the following expression, amenable to saddle point evaluation:

$$\mathbb{E}Z_t^{n,s} = \int \prod_{\alpha \leq \beta} dr_{\alpha\beta} d\hat{r}_{\alpha\beta} \prod_{a \leq b} dq_{ab} d\hat{q}_{ab} \prod_{\alpha a} dp_{\alpha a} d\hat{p}_{\alpha a} e^{N\Phi_t}, \quad (\text{A.19})$$

where

$$\Phi_t = G_I + G_S + \alpha G_E \quad (\text{A.20})$$

$$G_I = -\frac{1}{2} \sum_{\alpha\beta} \hat{r}_{\alpha\beta} r_{\alpha\beta} - \frac{1}{2} \sum_{ab} \hat{q}_{ab} q_{ab} - \sum_{\alpha a} \hat{p}_{\alpha a} p_{\alpha a} + t \sum_a p_{1a} + \frac{1}{2} t \sum_{ab} q_{ab} - \frac{t}{2} \sum_a q_{aa}, \quad (\text{A.21})$$

$$G_S = \log \int \prod_{\alpha=1}^s P(dw_{\alpha}^*) \prod_{a=1}^n P(dw_a) e^{+\frac{1}{2} \sum_{\alpha\beta} \hat{r}_{\alpha\beta} w_{\alpha}^* w_{\beta}^* + \frac{1}{2} \sum_{ab} \hat{q}_{ab} w_a w_b + \sum_{\alpha a} \hat{p}_{\alpha a} w_{\alpha}^* w_a}, \quad (\text{A.22})$$

$$G_E = \log \int \prod_{\alpha} \frac{d\lambda_{\alpha} d\hat{\lambda}_{\alpha}}{2\pi} \prod_a \frac{du_a d\hat{u}_a}{2\pi} \prod_{\alpha} \tilde{\Theta}(\lambda_{\alpha}) \prod_a \tilde{\Theta}(u_a) \quad (\text{A.23})$$

$$\times e^{-i \sum_{\alpha} \hat{\lambda}_{\alpha} \lambda_{\alpha} - i \sum_a \hat{u}_a u_a - \frac{1}{2} \sum_{\alpha\beta} \hat{\lambda}_{\alpha} \hat{\lambda}_{\beta} r_{\alpha\beta} - \frac{1}{2} \sum_{ab} \hat{u}_a \hat{u}_b q_{ab} - \sum_{\alpha a} \hat{\lambda}_{\alpha} \hat{u}_a p_{\alpha a}}.$$

As is usually done in the replica method, we first choose an Ansatz for the structure of the overlaps, so that we can perform analytic continuation and take the small n and s limit, followed by saddle point evaluation.

A.1.2 Replica Symmetric Ansatz

The Ansatz (RS) we choose is the Replica Symmetric one. Given the symmetries of the problem, it takes the form

$$q_{ab} = \begin{cases} q_d = 1 & \text{if } a = b \\ q & \text{if } a \neq b \end{cases}, \quad r_{\alpha\beta} = \begin{cases} r_d = 1 & \alpha = \beta \\ r & \alpha \neq \beta \end{cases}, \quad p_{\alpha a} = \begin{cases} p^* & \alpha = 1 \\ p & \alpha \neq 1. \end{cases} \quad (\text{A.24})$$

The Nishimori conditions discussed in the next Section guarantee that if the RS ansatz is correct for the reference system, e.g. it is by definition for the spherical perceptron in the

RS regime, then the RS ansatz above gives a correct prediction for ϕ_t . More generally, the correct Ansatz for the tilted system has the same number of steps of symmetry breaking as the correct one for the reference system.

A.1.3 Nishimori conditions

Thanks to the Bayesian nature of the problem, the Nishimori conditions apply [Iba, 1999, Nishimori, 1980]. They are a set of identities that we use to further simplify the overlap structure. Here we state these identities in our context and review their derivation.

Let's fix the time t , and also fix the realization of the disorder in the distribution p (i.e. the realization of the examples X). Assume $\mathbf{w}^* \sim p$, and call $\mathbf{h} = t\mathbf{w}^* + \sqrt{t}\mathbf{g}$, with $\mathbf{g}$ standard Gaussian, the observed field (as in Equation (2.6)). The corresponding posterior over $\mathbf{w}^*$ is $p_{h,t}$ given in Equation (2.4). For convenience we drop the time index and write $\mathbf{h} = \mathbf{h}_t$ and $p(\mathbf{w}|\mathbf{h}) = p_{h,t}(\mathbf{w})$.

Given K i.i.d. samples from the posterior, $\mathbf{w}_k \sim p(\mathbf{w}|\mathbf{h})$, given a continuous bounded test function $f : \mathbb{R}^{K \times N} \rightarrow \mathbb{R}$, the following identity (Nishimori identity) holds:

$$\mathbb{E}_{\mathbf{w}^*} \mathbb{E}_{\mathbf{h}|\mathbf{w}^*} \mathbb{E}_{\{\mathbf{w}_k\}_{k=1}^K|\mathbf{h}} f(\mathbf{w}^*, \mathbf{w}_2, \dots, \mathbf{w}_K) = \mathbb{E}_{\mathbf{w}^*} \mathbb{E}_{\mathbf{h}|\mathbf{w}^*} \mathbb{E}_{\{\mathbf{w}_k\}_{k=1}^K|\mathbf{h}} f(\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_K) \quad (\text{A.25})$$

In order to show this, fixing $K = 2$ for simplicity, we rewrite the *l.h.s.* of the previous equation as

$$\mathbb{E} f(\mathbf{w}^*, \mathbf{w}_2) = \int d\mathbf{w}^* d\mathbf{h} d\mathbf{w}_2 P(\mathbf{w}^*) P(\mathbf{h}|\mathbf{w}^*) P(\mathbf{w}_2|\mathbf{h}) f(\mathbf{w}^*, \mathbf{w}_2) \quad (\text{A.26})$$

$$= \int d\mathbf{w}^* d\mathbf{h} d\mathbf{w}_2 P(\mathbf{h}) P(\mathbf{w}^*|\mathbf{h}) P(\mathbf{w}_2|\mathbf{h}) f(\mathbf{w}^*, \mathbf{w}_2), \quad (\text{A.27})$$

where we used Bayes' rule. The *r.h.s.* instead can be rewritten as

$$\mathbb{E} f(\mathbf{w}_1, \mathbf{w}_2) = \int d\mathbf{w}^* d\mathbf{h} d\mathbf{w}_1 d\mathbf{w}_2 P(\mathbf{w}^*) P(\mathbf{h}|\mathbf{w}^*) P(\mathbf{w}_1|\mathbf{h}) P(\mathbf{w}_2|\mathbf{h}) f(\mathbf{w}_1, \mathbf{w}_2) \quad (\text{A.28})$$

$$= \int d\mathbf{h} d\mathbf{w}_1 d\mathbf{w}_2 P(\mathbf{h}) P(\mathbf{w}_1|\mathbf{h}) P(\mathbf{w}_2|\mathbf{h}) f(\mathbf{w}_1, \mathbf{w}_2). \quad (\text{A.29})$$

The *l.h.s.* and the *r.h.s.* of Equation (A.25) are now clearly shown to be the same, once we rename $\mathbf{w}^*$ to $\mathbf{w}_1$. We can use Equation (A.25) to establish identities among the overlaps appearing in the replicated free entropy Φ_t of Equation (A.20). In fact, within the RS Ansatz, assuming large N and at the saddle point value, one can write the overlaps as:

$$p^* = \frac{\mathbf{w}_1^* \cdot \mathbf{w}_b}{N} \quad \forall b \in [n]; \quad q = \frac{\mathbf{w}_a \cdot \mathbf{w}_b}{N} \quad \forall a, b \in [n], a \neq b. \quad (\text{A.30})$$

Since in the replicated partition function $\mathbf{w}_a$ and $\mathbf{w}_b$ are independent samples of the posterior conditional on $\mathbf{w}_1^*$, the Nishimori identities imply

$$p^* = q. \quad (\text{A.31})$$

Now, consider the overlaps

$$p = \frac{\mathbf{w}_\alpha^* \cdot \mathbf{w}_a}{N} \quad \forall a \in [n], \forall \alpha \in [s], \alpha \neq 1; \quad r = \frac{\mathbf{w}_\alpha^* \cdot \mathbf{w}_\beta^*}{N} \quad \forall \alpha, \beta \in [s], \alpha \neq \beta. \quad (\text{A.32})$$

Since $\mathbf{w}_a$ is decoupled from $\mathbf{w}_\alpha^*$ when $\alpha \neq 0$, its distribution is the same as a reference configuration when $\mathbf{w}_1^*$ is traced out, therefore we have

$$p = r. \quad (\text{A.33})$$

Similar reasoning can be applied to the conjugate order parameters, so that we obtain:

$$\hat{p}^* = \hat{q}, \quad (\text{A.34})$$

$$\hat{p} = \hat{r}. \quad (\text{A.35})$$

These conditions simplify considerably the expressions in the computation of the free entropy. The Nishimori conditions can be also used within Replica Symmetry Breaking scenarios for the reference distribution p . They guarantee that no further RSB steps are needed in the analysis of the tilted free entropy ϕ_t .

A.1.4 Replica Symmetric Free Entropy

Entropic term

Let's first look at the entropic term G_S within the RS ansatz:

$$G_S = \log \int \prod_{\alpha=1}^s P(dw_\alpha^*) \prod_{a=1}^n P(dw_a) e^{+\frac{1}{2}(\hat{r}_d - \hat{r}) \sum_\alpha w_\alpha^{*2} + \frac{1}{2}(\hat{r} - \hat{p})(\sum_\alpha w_\alpha^*)^2} \quad (\text{A.36})$$

$$\times e^{+\frac{1}{2}(\hat{q}_d - \hat{q}) \sum_a w_a^2 + \frac{1}{2}(\hat{q} - \hat{p})(\sum_a w_a)^2 + (\hat{p}^* - \hat{p})w_1^* \sum_a w_a + \frac{\hat{p}}{2}(\sum_\alpha w_\alpha^* + \sum_a w_a)^2}. \quad (\text{A.37})$$

Using the Nishimori conditions, this simplifies to:

$$G_S = \log \int \prod_{\alpha=1}^s P(dw_\alpha^*) \prod_{a=1}^n P(dw_a) e^{\frac{1}{2}(\hat{r}_d - \hat{r}) \sum_\alpha w_\alpha^{*2}} \quad (\text{A.38})$$

$$\times e^{+\frac{1}{2}(\hat{r}_d - \hat{q}) \sum_a w_a^2 + \frac{1}{2}(\hat{q} - \hat{r})(\sum_a w_a)^2 + (\hat{q} - \hat{r})w_1^* \sum_a w_a + \frac{\hat{r}}{2}(\sum_\alpha w_\alpha^* + \sum_a w_a)^2}. \quad (\text{A.39})$$

Using the Hubbard-Stratonovich transformation twice we can get rid of the quadratic summations like $(\sum_a w_a)^2$ in the exponentials, at the cost of introducing two extra integration variables:

$$G_S = \log \int \prod_{\alpha=1}^s P(dw_\alpha^*) \prod_{a=1}^n P(dw_a) \int Dz D\gamma e^{\frac{1}{2}(\hat{r}_d - \hat{r}) \sum_\alpha w_\alpha^{*2}} \quad (\text{A.40})$$

$$\times e^{\frac{1}{2}(\hat{r}_d - \hat{q}) \sum_a w_a^2 + z\sqrt{\hat{q} - \hat{r}} \sum_a w_a + (\hat{q} - \hat{r})w_1^* \sum_a w_a + \gamma\sqrt{\hat{r}}(\sum_\alpha w_\alpha^* + \sum_a w_a)}, \quad (\text{A.41})$$

where we used $Dz = dz \frac{e^{-z^2/2}}{\sqrt{2\pi}}$ as a shorthand to denote a gaussian integral. This can now be factorized:

$$G_S = \log \int P(dw_1^*) \int Dz D\gamma e^{\frac{1}{2}(\hat{r}_d - \hat{r})w_1^{*2} + \gamma\sqrt{\hat{p}}w_1^*} [\mathcal{Z}(w_1^*, z, \gamma)]^n [\mathcal{Z}^*(\eta, \gamma)]^{s-1}, \quad (\text{A.42})$$

where

$$\mathcal{Z}(w_1^*, z, \gamma) = \int P(dw) e^{\frac{1}{2}(\hat{r}_d - \hat{q})w^2 + (z\sqrt{\hat{q}-\hat{r}} + \gamma\sqrt{\hat{r}} + (\hat{q}-\hat{r})w_1^*)w} \quad (\text{A.43})$$

$$\mathcal{Z}^*(z, \gamma) = \int P(dw^*) e^{\frac{1}{2}(\hat{r}_d - \hat{r})w^{*2} + \gamma\sqrt{\hat{r}}w^*} \quad (\text{A.44})$$

$$(\text{A.45})$$

The last step of the replica computation consists in performing the limit $n \rightarrow 0$ and $s \rightarrow 0$, which enables to obtain the final expression of ϕ_t :

$$\mathcal{G}_S(\hat{q}) = \lim_{s \rightarrow 0} \lim_{n \rightarrow 0} \partial_n G_S \quad (\text{A.46})$$

The final expression of the entropic term depends on the prior of the weights. Its functional form is reported in section A.3 and A.4, respectively for the spherical and binary perceptron cases.

Energetic term

Let's now focus our attention on the energetic term

$$G_E = \log \int \prod_{\alpha} \frac{d\lambda_{\alpha} d\hat{\lambda}_{\alpha}}{2\pi} \prod_a \frac{du_a d\hat{u}_a}{2\pi} \prod_{\alpha} \tilde{\Theta}(\lambda_{\alpha}) \prod_a \tilde{\Theta}(u_a) \quad (\text{A.47})$$

$$\times e^{-i \sum_{\alpha} \hat{\lambda}_{\alpha} \lambda_{\alpha} - i \sum_a \hat{u}_a u_a - \frac{1}{2} \sum_{\alpha\beta} \hat{\lambda}_{\alpha} \hat{\lambda}_{\beta} r_{\alpha\beta} - \frac{1}{2} \sum_{ab} \hat{u}_a \hat{u}_b q_{ab} - \sum_{\alpha a} \hat{\lambda}_{\alpha} \hat{u}_a p_{\alpha a}}. \quad (\text{A.48})$$

After using the RS ansatz over the overlap parameters, the Nishimori conditions, and some manipulations, the energetic term simplifies

$$G_E = \log \int \prod_{\alpha} \frac{d\lambda_{\alpha} d\hat{\lambda}_{\alpha}}{2\pi} \prod_a \frac{du_a d\hat{u}_a}{2\pi} \prod_{\alpha} \tilde{\Theta}(\lambda_{\alpha}) \prod_a \tilde{\Theta}(u_a) \quad (\text{A.49})$$

$$\times e^{-i \sum_{\alpha} \hat{\lambda}_{\alpha} \lambda_{\alpha} - i \sum_a \hat{u}_a u_a - \frac{1}{2}(1-r) \sum_{\alpha} \hat{\lambda}_{\alpha}^2 + \frac{q-r}{2} \hat{\lambda}_1^2} \quad (\text{A.50})$$

$$\times e^{-\frac{1}{2}(1-q) \sum_{\alpha} \hat{u}_{\alpha}^2 - \frac{r}{2} (\sum_{\alpha} \hat{\lambda}_{\alpha} + \sum_a \hat{u}_a)^2 - \frac{q-r}{2} (\hat{\lambda}_1 + \sum_a \hat{u}_a)^2}. \quad (\text{A.51})$$

We use again two Hubbard-Stratonovich substitutions to make the integrand factorized in the a and α indices, at the cost of introducing two more gaussian integrals:

$$G_E = \log \int Dz D\gamma \prod_{\alpha} \frac{d\lambda_{\alpha} d\hat{\lambda}_{\alpha}}{2\pi} \prod_a \frac{du_a d\hat{u}_a}{2\pi} \prod_{\alpha} \tilde{\Theta}(\lambda_{\alpha}) \prod_a \tilde{\Theta}(u_a) \quad (\text{A.52})$$

$$\times e^{-i \sum_{\alpha} \hat{\lambda}_{\alpha} (\lambda_{\alpha} + \gamma\sqrt{r}) - i \sum_a \hat{u}_a (u_a + \gamma\sqrt{r} + z\sqrt{q-r}) - \frac{1}{2}(1-r) \sum_{\alpha} \hat{\lambda}_{\alpha}^2 + \frac{q-r}{2} \hat{\lambda}_1^2} \quad (\text{A.53})$$

$$\times e^{-\frac{1}{2}(1-q) \sum_{\alpha} \hat{u}_{\alpha}^2 - iz\sqrt{q-r} \hat{\lambda}_1}. \quad (\text{A.54})$$

Finally, collecting the factors and taking the limits for $n \rightarrow 0$ and $s \rightarrow 0$, we arrive at

$$\mathcal{G}_E(q) = \lim_{s \rightarrow 0} \lim_{n \rightarrow 0} \partial_n G_E \quad (\text{A.55})$$

$$= \int Dz D\gamma \frac{\tilde{H}_{1-q}(\gamma\sqrt{r} + z\sqrt{q-r})}{\tilde{H}_{1-r}(\gamma\sqrt{r})} \log \tilde{H}_{1-q}(\gamma\sqrt{r} + z\sqrt{q-r}). \quad (\text{A.56})$$

where

$$\tilde{H}_a(b) = \int Dz \Theta(-b - \kappa + \sqrt{a}z) e^{-\frac{1}{T}U(-b-\kappa+\sqrt{a}z)} \quad (\text{A.57})$$

In the uniform measure case $U(s) = 0$, $\tilde{H}$ simplifies to $\tilde{H}_a(b) = H\left(-\frac{b+\kappa}{\sqrt{a}}\right)$, with $H(x) = \int_x^{+\infty} Dz = \frac{1}{2}\text{erfc}\left(\frac{x}{\sqrt{2}}\right)$, as defined in the main text.

Interaction term

Finally, we compute the interaction term

$$G_I = -\frac{1}{2} \sum_{\alpha\beta} \hat{r}_{\alpha\beta} r_{\alpha\beta} - \frac{1}{2} \sum_{ab} \hat{q}_{ab} q_{ab} - \sum_{\alpha a} \hat{p}_{\alpha a} p_{\alpha a} + t \sum_a p_{1a} + \frac{1}{2} t \sum_{ab} q_{ab} - \frac{t}{2} \sum_a q_{aa}, \quad (\text{A.58})$$

which, after using the RS ansatz, the Nishimori conditions, and taking the limits $n \rightarrow 0$ and $s \rightarrow 0$, reduces to:

$$\mathcal{G}_I = \lim_{s \rightarrow 0} \lim_{n \rightarrow 0} \partial_n G_I = -\frac{1}{2} \hat{r}_d - \frac{1}{2} \hat{q} q + \hat{r} r + \frac{1}{2} t q \quad (\text{A.59})$$

Saddle Point Equations

The optimization of ϕ_t can now be performed using the saddle point method [Mézard et al., 1987]. This amounts at solving the following system of equations

$$\hat{q} = t + 2\alpha \frac{\partial}{\partial q} \mathcal{G}_E(q), \quad (\text{A.60})$$

$$q = 2 \frac{\partial}{\partial \hat{q}} \mathcal{G}_S(\hat{q}). \quad (\text{A.61})$$

The optimization is usually performed iteratively, starting from an initial guess, inserting it in the right hand side, obtaining updated values, and iterating until convergence.

It is important to point out that the overlap parameters for the reference network are not obtained from the free entropy expression derived above; rather, they need to be determined independently from an analogous (but simpler) free entropy expression that only involves the reference replicas. The procedure is schematically shown in the next section, and the full computation is reported in [Engel and Van den Broeck, 2001] and in Chapter 1.

Reference system

The free entropy associated to the reference system is simply that of a standard perceptron problem, which is well known and can be found in [Engel and Van den Broeck, 2001], and in Chapter 1. The equations of the overlap parameters depend only on $(r, \hat{r}, \hat{r}_d)$. This results from the Bayesian structure of the problem: the reference probability measure is independent of the tilted one; for this reason, the reference free entropy is independent of $q, \hat{q}$. One can recover the expression of reference system free entropy $\phi^{(r)}(r, \hat{r}, \hat{r}_d)$ fixing

$n = 0$ and taking the limit $s \rightarrow 0$ of Φ_t , in Equation (A.20). The RS expression reads:

$$\phi^{(r)}(r, \hat{r}, \hat{r}_d) = \mathcal{G}_I^{(r)}(r, \hat{r}, \hat{r}_d) + \mathcal{G}_S^{(r)}(\hat{r}, \hat{r}_d) + \alpha \mathcal{G}_E^{(r)}(r), \quad (\text{A.62})$$

$$\mathcal{G}_I^{(r)}(r, \hat{r}, \hat{r}_d) = \frac{1}{2}(r\hat{r} - \hat{r}_d), \quad (\text{A.63})$$

$$\mathcal{G}_S^{(r)}(\hat{r}, \hat{r}_d) = \log \left(\int P(dw^*) e^{\frac{1}{2}(\hat{r}_d - \hat{r})w^{*2} + \gamma\sqrt{\hat{r}}w^*} \right), \quad (\text{A.64})$$

$$\mathcal{G}_E^{(r)}(r) = \int Dz \log \tilde{H}_{1-r}(\sqrt{r}z). \quad (\text{A.65})$$

The corresponding saddle point equations, that can be used to determine r , $\hat{r}$ and $\hat{r}_d$ as a function of α and κ , are:

$$r = 1 - 2 \frac{\partial}{\partial \hat{r}} \mathcal{G}_S^{(r)}(\hat{r}, \hat{r}_d), \quad (\text{A.66})$$

$$1 = 1 - 2 \frac{\partial}{\partial \hat{r}_d} \mathcal{G}_S^{(r)}(\hat{r}, \hat{r}_d), \quad (\text{A.67})$$

$$\hat{r} = -2\alpha \frac{\partial}{\partial r} \mathcal{G}_E^{(r)}(r), \quad (\text{A.68})$$

Our treatment differs from the standard one present in the literature only by the addition of the potential $U(s)$, which, only affects the energetic term and is fully absorbed in the definition of the $\tilde{H}$ function.

A.2 Computation of the stability distribution

In this Appendix, we perform the computation of the stability distribution in the perceptron problem with an arbitrary potential $U(s)$. This is a simple generalization to include the potential of a standard calculation that can be performed with the replica trick [Engel and Van den Broeck, 2001].

Given an N -dimensional weight vector $\mathbf{w}$, representing a solution of the perceptron problem with a margin κ , and a pattern $\mathbf{x}^\mu$, the stability is defined as $s^\mu = \frac{\mathbf{w} \cdot \mathbf{x}^\mu}{\sqrt{N}}$. Our goal is to compute, for a given target stability s , the average of $\delta(s - s^\mu)$ over all patterns and all solutions sampled from the target distribution p (Equation (2.1)), and subsequently average over all realizations of the patterns:

$$P(s) = \mathbb{E}_X \mathbb{E}_{\mathbf{w} \sim p} \frac{1}{M} \sum_{\mu=1}^M \delta(s - s^\mu) \quad (\text{A.69})$$

Since all patterns are statistically equivalent, we can just compute the average stability of the first pattern:

$$P(s) = \mathbb{E}_X \mathbb{E}_{\mathbf{w} \sim p} \delta(s - s^1). \quad (\text{A.70})$$

We now expand the formula using the definition of $p(\mathbf{w})$ in (2.14):

$$P(s) = \mathbb{E}_X \frac{\int P(d\mathbf{w}) \prod_{\mu=1}^M \tilde{\Theta}(s^\mu) \delta(s - s^1)}{\int P(d\mathbf{w}) \tilde{\Theta}(s^\mu)}, \quad (\text{A.71})$$

where we also used the definition of $\tilde{\Theta}$ in Equation (A.6). To take the expectation, we introduce n replicas, with n integer at first, and to be sent to zero through analytical continuation, so that we can rewrite the denominator as $Z^{-1} = \lim_{n \rightarrow 0} Z^{n-1}$. We give a special role to replica 1, and obtain the formula

$$P(s) = \lim_{n \rightarrow 0} \mathbb{E}_X \int \prod_a P(d\mathbf{w}_{a=1}^n) \prod_{\mu,a} \left[\tilde{\Theta} \left(\frac{\mathbf{w}_a \cdot \mathbf{x}^\mu}{\sqrt{N}} \right) \right] \delta \left(s - \frac{\mathbf{w}_1 \cdot \mathbf{x}^1}{\sqrt{N}} \right). \quad (\text{A.72})$$

The computation can then be carried out as usual, leading to the free entropy of the reference system in Section A.1.4, followed by saddle point evaluation. In fact, the observable $\delta \left(s - \frac{\mathbf{w}_1 \cdot \mathbf{x}^1}{\sqrt{N}} \right)$ being subdominant does not influence the saddle point. Given the stationary values of the overlaps, one can then obtain the RS expression for the stability distribution, given by

$$P(s) = \tilde{\Theta}(s) \frac{1}{\sqrt{2\pi(1-r)}} \int Dt \frac{e^{-\frac{(s-\sqrt{r}t)^2}{2(1-r)}}}{\tilde{H}_{1-r}(\sqrt{r}t)}, \quad (\text{A.73})$$

where $\tilde{H}$ has been defined in Equation (A.57).

A.3 Spherical Perceptron

In this section, we report the replica computation results for the spherical perceptron case, in which the prior is $P(\mathbf{w}) = \delta(\|\mathbf{w}\|^2 - N)$.

A.3.1 Free Entropy

The final expression for the free entropy can be obtained by specializing the entropic term, in Equation (A.22) to the spherical prior, which together with Equation (A.21) and Equation (A.23) gives:

$$\phi_t(q, \hat{q}) = \mathcal{G}_I(q, \hat{q}) + \mathcal{G}_S(\hat{q}) + \alpha \mathcal{G}_E(q) \quad (\text{A.74})$$

$$\mathcal{G}_I = -\frac{1}{2}\hat{r}_d - \frac{1}{2}\hat{q}q + \hat{r}r + \frac{1}{2}tq \quad (\text{A.75})$$

$$\mathcal{G}_S(\hat{q}) = -\frac{1}{2}\log(\hat{q} - \hat{r}_d) + \frac{1}{2(\hat{q} - \hat{r}_d)} \left(\hat{q} + 2\hat{r} \frac{(\hat{q} - \hat{r})}{(\hat{r} - \hat{r}_d)} + \frac{(\hat{q} - \hat{r})^2}{(\hat{r} - \hat{r}_d)} \left(\frac{\hat{r}}{\hat{r} - \hat{r}_d} + 1 \right) \right) \quad (\text{A.76})$$

$$\mathcal{G}_E(q) = \int Dz D\gamma \frac{\tilde{H}_{1-q}(\gamma\sqrt{r} + z\sqrt{q-r})}{\tilde{H}_{1-r}(\gamma\sqrt{r})} \log \tilde{H}_{1-q}(\gamma\sqrt{r} + z\sqrt{q-r}) \quad (\text{A.77})$$

A.3.2 Reference system

Once the prior for the perceptron model is specified, the free energy equations for the reference system become the following:

$$\phi^{(r)}(r, \hat{r}, \hat{r}_d) = \mathcal{G}_I^{(r)}(r, \hat{r}, \hat{r}_d) + \mathcal{G}_S^{(r)}(\hat{r}, \hat{r}_d) + \alpha \mathcal{G}_E^{(r)}(r), \quad (\text{A.78})$$

$$\mathcal{G}_I^{(r)}(r, \hat{r}, \hat{r}_d) = \frac{1}{2}(r\hat{r} - \hat{r}_d), \quad (\text{A.79})$$

$$\mathcal{G}_S^{(r)}(\hat{r}, \hat{r}_d) = \frac{1}{2} \left(\log(2\pi) - \log(\hat{r} - \hat{r}_d) + \frac{\hat{r}}{(\hat{r} - \hat{r}_d)} \right), \quad (\text{A.80})$$

$$\mathcal{G}_E^{(r)}(r) = \int Dz \log \tilde{H}_{1-r}(\sqrt{r}z). \quad (\text{A.81})$$

They are obtained by specializing the entropic term for the reference system in Equation (A.64) to the case of spherical perceptron prior, together with Equation (A.63) and Equation (A.65).

A.4 Binary Perceptron

In this section, we consider free entropy expression for the binary perceptron case, whose (improper) prior is $P(w_i) = \delta(w_i - 1) + \delta(w_i + 1)$.

A.4.1 Free Entropy

We obtain the free energy by specializing the expression for the entropic term in Equation (A.22) to the binary prior, together with Equation (A.21) and Equation (A.23):

$$\phi_t(q, \hat{q}) = \mathcal{G}_I(q, \hat{q}) + \mathcal{G}_S(\hat{q}) + \alpha \mathcal{G}_E(q), \quad (\text{A.82})$$

$$\mathcal{G}_S(\hat{q}) = \sum_{w^*=\pm 1} \int Dz D\gamma \frac{e^{\gamma\sqrt{\hat{r}}w^*}}{2 \cosh(\gamma\sqrt{\hat{r}})} \log \left(2e^{\frac{1}{2}(\hat{r}_d - \hat{q})} \cosh \left(z\sqrt{\hat{q} - \hat{r}} + \gamma\sqrt{\hat{r}} + (\hat{q} - \hat{r})w^* \right) \right), \quad (\text{A.83})$$

$$\mathcal{G}_E(q) = \int Dz D\gamma \frac{\tilde{H}_{1-q}(\gamma\sqrt{r} + z\sqrt{q-r})}{\tilde{H}_{1-r}(\gamma\sqrt{r})} \log \tilde{H}_{1-q}(\gamma\sqrt{r} + z\sqrt{q-r}), \quad (\text{A.84})$$

$$\mathcal{G}_I = -\frac{1}{2}\hat{r}_d - \frac{1}{2}\hat{q}q + \hat{r}r + \frac{1}{2}tq. \quad (\text{A.85})$$

A.4.2 Reference system

The free entropy equations for the reference system are the following:

$$\phi^{(r)}(r, \hat{r}, \hat{r}_d) = \mathcal{G}_I^{(r)}(r, \hat{r}, \hat{r}_d) + \mathcal{G}_S^{(r)}(\hat{r}, \hat{r}_d) + \alpha \mathcal{G}_E^{(r)}(r), \quad (\text{A.86})$$

$$\mathcal{G}_I^{(r)}(r, \hat{r}, \hat{r}_d) = \frac{1}{2}(r\hat{r} - \hat{r}_d), \quad (\text{A.87})$$

$$\mathcal{G}_S^{(r)}(\hat{r}, \hat{r}_d) = -\frac{1}{2}(\hat{r} - \hat{r}_d) + \int Dz \log \left(2 \cosh(z\sqrt{r}) \right), \quad (\text{A.88})$$

$$\mathcal{G}_E^{(r)}(r) = \int Dz \log \tilde{H}_{1-r}(\sqrt{r}z). \quad (\text{A.89})$$

where we have specialized the entropic term in (A.64), together with the interaction term and the energetic one in (A.63) and (A.65). As a side note, we may observe that in this case the $\hat{r}_d$ order parameter simplifies away with the expression of the interaction term $\mathcal{G}_I$, in both the reference and the tilted systems. This is expected since in the binary case the norm is fixed automatically. We keep this term in the expressions only for uniformity of notation.

A.5 Limiting behavior of the free entropy derivative

In this section, we report the analysis of $\partial_q \phi_t(q)$ as $q \rightarrow 1$. We set $q = 1 - \epsilon$ and expand for small ϵ , leading to the results reported in Section 2.4.4.

The starting expression is:

$$\phi_t(q, \hat{q}) = -\frac{1}{2}(\hat{r}_d + \hat{q}q) + \hat{r}r + \frac{1}{2}tq + \mathcal{G}_S(\hat{q}) + \alpha \mathcal{G}_E(q) \quad (\text{A.90})$$

and its derivative is:

$$\frac{d\phi_t}{dq} = \frac{t - \hat{q}}{2} + \alpha \frac{d\mathcal{G}_E(q)}{dq} \quad (\text{A.91})$$

We consider the two terms separately.

A.5.1 Interaction term

We start from the first (interaction) term. The dependency of $\hat{q}$ on ϵ can be determined from the saddle point equations. The spherical and binary perceptron models need to be considered separately, since the relevant equation involves $\mathcal{G}_S$, which differs between the two.

Spherical Perceptron

In the spherical case, the saddle point equation for q reads:

$$q = 2 \frac{\partial \mathcal{G}_S(\hat{q})}{\partial \hat{q}} \quad (\text{A.92})$$

$$= -\frac{\partial}{\partial \hat{q}} \left(\frac{1}{2} \log(\hat{q} - \hat{r}_d) - \frac{1}{2(\hat{q} - \hat{r}_d)} \left(\hat{q} + 2\hat{r} \frac{(\hat{q} - \hat{r})}{(\hat{r} - \hat{r}_d)} + \frac{(\hat{q} - \hat{r})^2}{(\hat{r} - \hat{r}_d)} \left(\frac{\hat{r}}{\hat{r} - \hat{r}_d} + 1 \right) \right) \right), \quad (\text{A.93})$$

$$= -\frac{1}{\hat{q} - \hat{r}_d} + \frac{2\hat{r} - \hat{r}_d}{(\hat{r} - \hat{r}_d)^2}. \quad (\text{A.94})$$

We can now use the saddle point for the reference system, see Section A.1.4, to rewrite the expression of q , concluding that $\hat{q}$ diverges as

$$\hat{q} = \frac{1}{\epsilon} + \hat{r}_d. \quad (\text{A.95})$$

This shows that the part of the derivative due to the interaction term diverges as $O(\epsilon^{-1})$. The sign of the derivative is negative, meaning that the free entropy has a local minimum at $q = 1$ unless this is overcome by the energetic contribution (which is not the case with $U(s) = 0$, as we show below).

Binary perceptron

An analogous computation for the binary perceptron case, already presented in Huang and Kabashima [2014], results in:

$$\frac{1+q}{2} = \frac{\partial}{\partial q} \sum_{w^*=\pm 1} \int Dz D\gamma \frac{e^{\gamma\sqrt{\hat{r}}w^*}}{2 \cosh(\gamma\sqrt{\hat{r}})} \log \left(2 \cosh \left(z\sqrt{\hat{q} - \hat{r}} + \gamma\sqrt{\hat{r}} + (\hat{q} - \hat{r}) w^* \right) \right) \quad (\text{A.96})$$

$$= \sum_{w^*=\pm 1} \int Dz D\gamma \frac{e^{\gamma\sqrt{\hat{r}}w^*}}{2 \cosh(\gamma\sqrt{\hat{r}})} \tanh \left(z\sqrt{\hat{q} - \hat{r}} + \gamma\sqrt{\hat{r}} + (\hat{q} - \hat{r}) w^* \right) \left(\frac{z}{\sqrt{\hat{q} - \hat{r}}} + w^* \right) \quad (\text{A.97})$$

As for the spherical case, $\hat{q} \rightarrow \infty$ as $q \rightarrow 1$. We can then rewrite the previous expression as

$$1 - \frac{1}{2}\epsilon = 1 - e^{-2(\hat{q}-\hat{r})} \int Dz D\gamma \frac{e^{\gamma\sqrt{\hat{r}}} e^{-2(z\sqrt{\hat{q}-\hat{r}}+\gamma\sqrt{\hat{r}})} - e^{-\gamma\sqrt{\hat{r}}} e^{2(z\sqrt{\hat{q}-\hat{r}}+\gamma\sqrt{\hat{r}})}}{\cosh(\gamma\sqrt{\hat{r}})} \quad (\text{A.98})$$

and thus

$$\hat{q} = -\frac{1}{2} \log \left(\frac{\epsilon}{2} \right) + C \quad (\text{A.99})$$

$$C = \hat{r} + \frac{1}{2} \log \left(\int Dz D\gamma \frac{e^{\gamma\sqrt{\hat{r}}} e^{-2(z\sqrt{\hat{q}-\hat{r}}+\gamma\sqrt{\hat{r}})} + e^{-\gamma\sqrt{\hat{r}}} e^{2(z\sqrt{\hat{q}-\hat{r}}+\gamma\sqrt{\hat{r}})}}{\cosh(\gamma\sqrt{\hat{r}})} \right). \quad (\text{A.100})$$

This shows that, as $q \rightarrow 1$, the part of the derivative that comes from the interaction term also diverges in the binary case, but this time logarithmically, as $O(\log \epsilon)$. The sign of the derivative is again negative, meaning that the free entropy has a local minimum at $q = 1$

unless the effect is overcome by the energetic contribution. As we shall show, the energetic contribution does indeed overcome this effect and produce a local maximum at $q = 1$ unless the potential $U(s)$ diverges at the origin.

A.5.2 Energetic term

Next, we compute the derivative of the energetic term. This does not depend on the model. We expand $\mathcal{G}_E(1 - \epsilon)$ for small ϵ and then study $(\mathcal{G}_E(1 - \epsilon) - \mathcal{G}_E(1)) / \epsilon^1$, keeping in mind that:

$$\frac{d\mathcal{G}_E}{dq} = -\frac{d\mathcal{G}_E}{d\epsilon} \quad (\text{A.101})$$

We perform the expansion, trying to keep the setting general with respect to the potential U . Indeed, our only starting assumption is that U is twice differentiable over $(0, \infty)$.

The energetic term, after some manipulations, and using $q = 1 - \epsilon$, can be written as:

$$\mathcal{G}_E = \int D\gamma Dz \frac{\mathcal{N}(z, \gamma)}{\mathcal{D}(\gamma)} \quad (\text{A.102})$$

$$\mathcal{D}(\gamma) = \int_{\frac{\gamma\sqrt{r}}{\sqrt{1-r}}}^{\infty} D\lambda e^{-\beta U(\lambda\sqrt{1-r}-\gamma\sqrt{r})} \quad (\text{A.103})$$

$$\mathcal{N}(z, \gamma) = \int_{\frac{\gamma\sqrt{r}}{\sqrt{1-r}}}^{\infty} D\lambda e^{-\beta U(\lambda\sqrt{1-r}-\gamma\sqrt{r})} \mathcal{A}(z, \gamma, \lambda) \quad (\text{A.104})$$

$$\mathcal{A}(z, \gamma, \lambda) = \log \int_{-\frac{a(z, \gamma, \lambda, u)}{\sqrt{\epsilon}}}^{\infty} Du e^{-\beta U(u\sqrt{\epsilon}+a(z, \gamma, \lambda, u))} \quad (\text{A.105})$$

$$a(z, \gamma, \lambda, u) = -\sqrt{r}\gamma - z\sqrt{1-r} - \epsilon\sqrt{\frac{\epsilon}{1-r}} + \sqrt{1-r}\lambda - \frac{\epsilon\lambda}{\sqrt{1-r}} \quad (\text{A.106})$$

We first perform two change of variables, in both $\mathcal{N}$ and $\mathcal{D}$, from λ to $\rho = -\gamma\sqrt{r} + \lambda\sqrt{1-r}$:

$$\mathcal{D}(\gamma) = \frac{1}{\sqrt{1-r}} \int_0^{\infty} d\rho G\left(\frac{\rho + \gamma\sqrt{r}}{\sqrt{1-r}}\right) e^{-\beta U(\rho)} \quad (\text{A.107})$$

$$\mathcal{N}(z, \gamma) = \frac{1}{\sqrt{1-r}} \int_0^{\infty} d\rho G\left(\frac{\rho + \gamma\sqrt{r}}{\sqrt{1-r}}\right) e^{-\beta U(\rho)} \tilde{\mathcal{A}}(z, \gamma, \rho) \quad (\text{A.108})$$

$$\tilde{\mathcal{A}}(z, \gamma, \rho) = \log \int_{-\frac{a'(z, \gamma, \rho, u)}{\sqrt{\epsilon}}}^{\infty} Du e^{-\beta U(u\sqrt{\epsilon}+\tilde{a}(z, \gamma, \rho, u))} \quad (\text{A.109})$$

$$\tilde{a}(z, \gamma, \rho, u) = \rho - z\sqrt{\epsilon}\sqrt{1-\frac{\epsilon}{1-r}} - \epsilon\frac{\rho + \gamma\sqrt{r}}{\sqrt{1-r}} \quad (\text{A.110})$$

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

Next, we expand the potential U inside the expression of $\tilde{\mathcal{A}}$ for small ϵ up to the first

¹To the sake of simplicity, from now on, we do not explicit the dependence of $\mathcal{G}_E$.

order:

$$U(u\sqrt{\epsilon} + \tilde{a}(z, \gamma, \lambda, u)) \approx U(\rho) + \sqrt{\epsilon}U'(\rho)(u - z) \quad (\text{A.111})$$

$$- \epsilon \frac{U'(\rho)}{1-r} (\sqrt{r}\gamma + \rho) + \frac{1}{2}\epsilon(u - z)^2 U''(\rho) \quad (\text{A.112})$$

This allows us to compute the integral $\tilde{\mathcal{A}}$, which converges as long as $1 + U''(\rho)\beta\epsilon \geq 0$ (which is obviously always true if U is convex). After some explicit integration and further expansions, we arrive at:

$$\tilde{\mathcal{A}}(z, \gamma, \rho) \approx \mathcal{A}_0 + \sqrt{\epsilon}\mathcal{A}_{1/2} + \epsilon\mathcal{A}_1 + \log H\left(z - \frac{\rho}{\sqrt{\epsilon}}\right) \quad (\text{A.113})$$

$$\mathcal{A}_0 = -\beta U(\rho) \quad (\text{A.114})$$

$$\mathcal{A}_{1/2} = \beta z U'(\rho) \quad (\text{A.115})$$

$$\mathcal{A}_1 = \beta \left(-\frac{U''(\rho)}{2} (1 + z^2) + \frac{\beta (U'(\rho))^2}{2} + \frac{U'(\rho)(\rho + \gamma\sqrt{r})}{1-r} \right) \quad (\text{A.116})$$

At this point, we can observe that the term $\mathcal{A}_{1/2}$ does not contribute because it gets canceled by the integral over z . Indeed, with further manipulations and changes of variables we arrive at:

$$\mathcal{G}_E = \int D\gamma \frac{\mathcal{N}'(\gamma)}{\mathcal{D}'(\gamma)} \quad (\text{A.117})$$

$$\tilde{\mathcal{D}}(\gamma) = \int_0^\infty d\rho G\left(\frac{\rho + \gamma\sqrt{r}}{\sqrt{1-r}}\right) e^{-\beta U(\rho)} \quad (\text{A.118})$$

$$\tilde{\mathcal{N}}(\gamma) = \int_0^\infty d\rho G\left(\frac{\rho + \gamma\sqrt{r}}{\sqrt{1-r}}\right) e^{-\beta U(\rho)} [\mathcal{A}_0 + \epsilon\tilde{\mathcal{A}}_1] + \quad (\text{A.119})$$

$$+ \sqrt{\epsilon} \int_0^\infty d\tau G\left(\frac{\tau\epsilon + \gamma\sqrt{r}}{\sqrt{1-r}}\right) e^{-\beta U(\tau\epsilon)} \int Dz \log H(z - \tau)$$

$$\mathcal{A}_0 = -\beta U(\rho) \quad (\text{A.120})$$

$$\tilde{\mathcal{A}}_1 = \beta \left(-U''(\rho) + \frac{\beta (U'(\rho))^2}{2} + \frac{U'(\rho)(\rho + \gamma\sqrt{r})}{1-r} \right) \quad (\text{A.121})$$

We can observe that $\mathcal{G}_E$ receives three contributions from $\tilde{\mathcal{N}}$. The ones from $\mathcal{A}_0$ and $\tilde{\mathcal{A}}_1$ are straightforward: they represent a zero-order and a first-order contributions. In particular the first one gives us the limiting value $\mathcal{G}_E^0 = \lim_{\epsilon \rightarrow 0} \mathcal{G}_E$. The first-order term then gives a finite contribution to the derivative.

Thus, the only way to avoid the local maximum at $q = 1$ in the binary perceptron case is to choose a potential $U(s)$ that diverges for $s \rightarrow 0$. Then the scaling of the last term with ϵ can be manipulated. For the specific choice $U(s) = -\log(s)$ we can easily see that we obtain an additional factor $\epsilon^{\beta/2}$, which makes the term $O\left(\epsilon^{\frac{\beta+1}{2}}\right)$ and the derivative $O\left(\epsilon^{\frac{\beta-1}{2}}\right)$. For $\beta > 1$, this term therefore becomes sub-dominant and it does not contribute to the derivative: we revert to the situation where the dominant contribution no longer

comes from the energetic term, but from the interaction term, and it has the right sign to ensure that $q = 1$ will correspond to a local minimum rather than a maximum.

Appendix B

Phase Selection Appendix

B.1 Replica computation

In this section, we perform the replica computation for the analytical investigation of the Phase Selection problem, introduced in Chapter 3. To compute the quenched free-entropy for Phase selection, the starting point is the definition of the partition function Z , associated with the probability distribution over the sign configurations. Introducing an inverse temperature ν and a chemical potential ζ , we can write the partition function as:

$$Z(\nu, \zeta; X, \mathbf{w}^*) = \sum_{\{\mathbf{S}\}} e^{\zeta \sum_{\mu=1}^M S^\mu - \nu \mathcal{L}(\mathbf{S}; X, \mathbf{w}^*)}, \quad (\text{B.1})$$

where the energy of the internal system, $\mathcal{L}(\mathbf{S}; X, \mathbf{w}^*)$, is recovered as the maximum a posteriori, i.e., in the zero temperature limit of the posterior, for the estimated signal

$$\mathcal{L}(\mathbf{S}; X, \mathbf{w}^*) = \lim_{\rho \rightarrow \infty} -\frac{1}{\rho} \log \int \prod_{i=1}^N dw_i P(w_i) \frac{e^{-\frac{\rho}{2} \left(\sum_{\mu=1}^M \left(S^\mu \frac{\mathbf{x}^\mu \cdot \mathbf{w}^*}{\sqrt{N}} - \frac{\mathbf{x}^\mu \cdot \mathbf{w}}{\sqrt{N}} \right)^2 + \frac{\lambda}{2} \|\mathbf{w}\|_2^2 \right)}}{\sqrt{2\pi/\rho}}, \quad (\text{B.2})$$

and where $\mathbf{x}^\mu$ represents a row of the measurement matrix. Note that, for convenience of notation, compared to the definition Equation (3.5) for the MSE_y , we have considered the gauge transformation:

$$\mathbf{S} \odot \left| \frac{X\mathbf{w}^*}{\sqrt{N}} \right| \rightarrow \mathbf{S} \odot \frac{X\mathbf{w}^*}{\sqrt{N}}, \quad (\text{B.3})$$

which corresponds to reabsorbing in $\mathbf{y}$ the correct signs for the measurements.

In this type of analysis, the goal is to average over the specific realizations of the dataset, referred to as quenched disorder, as well as over the particular choice of the target signal, and to compute the typical value of the free entropy:

$$\phi(\nu, \zeta) = \lim_{N \rightarrow \infty} \frac{1}{N} \mathbb{E}_{X, \mathbf{w}^*} \ln Z(\nu, \zeta; X, \mathbf{w}^*). \quad (\text{B.4})$$

Following [Obuchi et al., 2018], we employ two separate instances of the replica trick to estimate this average. First, one can remove the intractable expectation of the logarithm

by using:

$$\mathbb{E}_{X, \mathbf{w}^*} \ln Z = \mathbb{E}_{X, \mathbf{w}^*} \frac{1}{s} \lim_{s \rightarrow 0} Z^s, \quad (\text{B.5})$$

Since the disorder average can be easily evaluated for integer values of s , and the result can be analytically continued to $s \rightarrow 0$. As for the second replica trick, we assume the ratio between the two inverse temperatures in the problem, external, ν , and internal, ρ , to be integer, $\nu/\rho = \beta \in \mathbb{N}$ and then extrapolate the limit $\beta \rightarrow 0$. Using these replica tricks, one is now able to compute the quenched average of the free entropy as:

$$\phi(\nu, \zeta) = \lim_{N \rightarrow \infty} \lim_{s, \beta \rightarrow 0} \frac{1}{Ns} \log \mathbb{E}_{X, \mathbf{w}^*} \left\{ \sum_{\{S\}} e^{\zeta \sum_{\mu} S_{\mu}} \right\} \quad (\text{B.6})$$

$$\left[\int d\mathbf{w} P(\mathbf{w}) e^{-\frac{\nu}{2\beta} \left[\sum_{\mu} \left(S_{\mu} \left(\frac{\mathbf{x}^{\mu} \cdot \mathbf{w}^*}{\sqrt{N}} \right) - \frac{\mathbf{x}^{\mu} \cdot \mathbf{w}}{\sqrt{N}} \right)^2 + \frac{\lambda}{2} \|\mathbf{w}\|_2^2 \right]} \right]^{\beta s}, \quad (\text{B.7})$$

where we assumed that $\beta = \nu/\rho$ is a positive integer. This assumption is fundamental for treating ν/ρ and performing a second replica computation. Now we replicate the partition function, enabling the expectation over the disorder:

$$Z^s = \sum_{\{S^a\}} \prod_a e^{\zeta \sum_{\mu} S_{\mu}^a} \mathbb{E}_{\mathbf{w}^*, X} \left[\int \prod_{a=1}^s \prod_{c=1}^{\beta} dw_{ac} P(w_{ac}) \prod_{\mu=1}^{\alpha} \frac{e^{-\frac{\nu}{2\beta} \sum_{ac} \left[\left(S_{\mu}^a \left(\frac{\mathbf{x}^{\mu} \cdot \mathbf{w}^*}{\sqrt{N}} \right) - \frac{\mathbf{x}^{\mu} \cdot \mathbf{w}_{ac}}{\sqrt{N}} \right)^2 + \frac{\lambda}{2} \|\mathbf{w}_{ac}\|_2^2 \right]}}{\sqrt{2\pi\beta/\nu}} \right]. \quad (\text{B.8})$$

We can now introduce the change of variables $u_0^{\mu} = \frac{\mathbf{x}^{\mu} \cdot \mathbf{w}^*}{\sqrt{N}}$ and $u_{ac}^{\mu} = \frac{\mathbf{x}^{\mu} \cdot \mathbf{w}_{ac}}{\sqrt{N}}$, which allows us to rewrite the replicated partition function as:

$$Z^s = \sum_{\{S^a\}} \prod_a e^{\zeta \sum_{\mu} S_{\mu}^a} \mathbb{E}_{\mathbf{w}^*, X} \left[\int \prod_{\mu} \frac{du_0^{\mu} d\hat{u}_0^{\mu}}{2\pi} e^{i \sum_{\mu} \hat{u}_0^{\mu} \left(u_0^{\mu} - \frac{\mathbf{x}^{\mu} \cdot \mathbf{w}^*}{\sqrt{N}} \right)} \right] \quad (\text{B.9})$$

$$\times \prod_{\mu, ac} \frac{du_{ac}^{\mu} d\hat{u}_{ac}^{\mu}}{2\pi} \prod_{ac} e^{i \sum_{\mu} \hat{u}_{ac}^{\mu} \left(u_{ac}^{\mu} - \frac{\mathbf{x}^{\mu} \cdot \mathbf{w}_{ac}}{\sqrt{N}} \right)} \quad (\text{B.10})$$

$$\times \int \prod_{a=1}^s \prod_{c=1}^{\beta} dw_{ac} P(w_{ac}) \prod_{ac} e^{-\frac{\lambda}{2} \frac{\nu}{\beta} \|\mathbf{w}_{ac}\|_2^2} \prod_{\mu} \frac{e^{-\frac{\nu}{2\beta} \sum_{ac} (S_{\mu}^a u_0^{\mu} - u_{ac}^{\mu})^2}}{\sqrt{2\pi\beta/\nu}} \right]. \quad (\text{B.11})$$

One can now define the overlap order parameters of the model [Auffinger et al., 2013], which represent a set of sufficient descriptors of the system in high dimensions:

$$\tilde{q} = \frac{\mathbf{w}^* \cdot \mathbf{w}^*}{N}, \quad q_{ac, bd} = \frac{\mathbf{w}_{ac} \cdot \mathbf{w}_{bd}}{N}, \quad m_{ac} = \frac{\mathbf{w}^* \cdot \mathbf{w}_{ac}}{N}, \quad (\text{B.12})$$

where $\tilde{q}$ represents the norm of the hidden signal, $q_{ac,bd}$ the overlap between two signal estimates sampled from the above-defined measure, and m_{ac} is the magnetization, i.e., the overlap between hidden and estimated signal.

By plugging their definitions in the replicated partition function, and defining $D = \{a, b, c, d | (a = b) \wedge (c \geq d)\} \vee \{a, b, c, d | (a > b) \wedge c, d\}$, it is possible to obtain:

$$Z^s = \sum_{\{S^a\}} \prod_a e^{\zeta \sum_\mu S_\mu^a} \int d\mathbf{w}^* P(\mathbf{w}^*) \int d\tilde{q} \delta(N\tilde{q} - \mathbf{w}^* \cdot \mathbf{w}^*) \quad (\text{B.13})$$

$$\times \int \prod_{ac,bd} dq_{ab,cd} \delta(Nq_{ac,bd} - \mathbf{w}_{ab} \cdot \mathbf{w}_{bd}) \prod_{ac} dm_{ac} \delta(Nm_{ac} - \mathbf{w}^* \cdot \mathbf{w}_{ab}) \quad (\text{B.14})$$

$$\times \int \prod_\mu \frac{du_0^\mu d\hat{u}_0^\mu}{2\pi} e^{i \sum_\mu \hat{u}_0^\mu u_0^\mu} \prod_{\mu,ac} \frac{du_{ac}^\mu d\hat{u}_{ac}^\mu}{2\pi} \prod_{ac} e^{i \sum_\mu \hat{u}_{ac}^\mu u_{ac}^\mu} \times \quad (\text{B.15})$$

$$\times \prod_\mu e^{-\frac{1}{2} \left((\hat{u}_0^\mu)^2 \tilde{q} + \sum_{ac,bd} \hat{u}_{ac}^\mu \hat{u}_{bd}^\mu q_{ac,bd} + 2\hat{u}_0^\mu \sum_{ac} \hat{u}_{ac}^\mu m_{ac} \right)} \quad (\text{B.16})$$

$$\times \int \prod_{a=1}^s \prod_{c=1}^\beta d\mathbf{w}_{ab} P(\mathbf{w}_{ab}) e^{-\frac{\lambda}{2} \frac{\nu}{\beta} \|\mathbf{w}_{ab}\|_2^2} \prod_\mu \frac{e^{-\frac{\nu}{2\beta} \sum_{ac} (S_\mu^a u_0^\mu - u_{ac}^\mu)^2}}{\sqrt{2\pi\beta/\nu}} \quad (\text{B.17})$$

$$= \int \prod_{\{a,b,c,d\} \in D} \frac{dq_{ac,bd} d\hat{q}_{ac,bd}}{2\pi} \int \prod_{ac} \frac{dm_{ac} d\hat{m}_{ac}}{2\pi} e^{N\Phi(q,m)}, \quad (\text{B.18})$$

where:

$$\Phi(q, R) = G_I + \log G_S + \alpha \log G_E \quad (\text{B.19})$$

$$G_I = i\hat{q}\tilde{q} + i \sum_{\{a,b,c,d\} \in D} \hat{q}_{ac,bd} q_{ac,bd} + i \sum_{ac} \hat{m}_{ac} m_{ac}, \quad (\text{B.20})$$

$$G_S = \int d\mathbf{w}^* P(\mathbf{w}^*) \int \prod_{ac} d\mathbf{w}_{ab} e^{-\frac{\lambda}{2} \frac{\nu}{\beta} (\mathbf{w}_{ab})^2} P(\mathbf{w}_{ab}) \quad (\text{B.21})$$

$$\times e^{-i\hat{q}(\mathbf{w}^*)^2 - i \sum_{\{a,b,c,d\} \in D} \hat{q}_{ac,bd} \mathbf{w}_{ab} \cdot \mathbf{w}_{bd} - i \sum_{ac} \hat{m}_{ac} \mathbf{w}^* \cdot \mathbf{w}_{ab}}, \quad (\text{B.22})$$

$$G_E = \sum_{S^a=\{1,0\}} e^{\zeta \sum_a S^a} \int \frac{du_0 d\hat{u}_0}{2\pi} \prod_{ac} \frac{du_{ac} d\hat{u}_{ac}}{2\pi} \quad (\text{B.23})$$

$$\times e^{i\hat{u}_0 u_0 + \sum_{ac} i\hat{u}_{ac} u_{ac} - \frac{1}{2} \left((\hat{u}_0)^2 \tilde{q} + \sum_{ac,bd} \hat{u}_{ac} \hat{u}_{bd} q_{ac,bd} + 2\hat{u}_0 \sum_{ac} \hat{u}_{ac} m_{ac} \right)} \quad (\text{B.24})$$

$$\times \frac{e^{-\frac{\nu}{2\beta} \sum_{ac} (S^a u_0 - u_{ac})^2}}{\sqrt{2\pi\beta/\nu}}. \quad (\text{B.25})$$

B.1.1 Replica Symmetric Ansatz

In the $N \rightarrow \infty$ limit, the free entropy value can be evaluated using a saddle-point approximation over the order parameters. This requires imposing stationarity (extremum) conditions for the overlaps that appear in the free entropy expression. To make the problem tractable, a parametric ansatz for the order parameters is adopted; here, we employ the replica symmetric (RS) ansatz, which assumes symmetry under permutation of the replica indices and significantly reduces the complexity of the extremization. The overlap $q^{ac,bd}$

becomes

$$q^{ac,bd} = \begin{cases} q_2 & \text{if } a = b, c = d \\ q_1 & \text{if } a = b, c > d \\ q_0 & \text{if } a > b, \forall c, d \end{cases}, \quad (\text{B.26})$$

and a replica-index-independent magnetization with the signal m . We now consider separately the different term appearing in the free entropy expression.

Interaction term

Let's start our analysis from the Interaction term. We can first insert the RS ansatz in the expression for the quenched free entropy, and then perform some mathematical manipulations. The interaction term reads as follows:

$$G_I = -\hat{q}\tilde{q} - \sum_{\{a,b,c,d\} \in D} \hat{q}_{ac,bd} q_{ac,bd} - \sum_{ac} \hat{m}_{ac} m_{ac}, \quad (\text{B.27})$$

$$= -\hat{q}\tilde{q} - \frac{1}{2} \sum_{a=b,c=d} \hat{q}_2 q_2 - \sum_{a=b,c>d} \hat{q}_1 q_1 - \sum_{a>b,cd} \hat{q}_0 q_0 - \sum_{ac} \hat{m} m, \quad (\text{B.28})$$

$$= -\hat{q}\tilde{q} + s \left(-\frac{1}{2} \beta \hat{q}_2 q_2 - \frac{\beta(\beta-1)}{2} \hat{q}_1 q_1 + \frac{1}{2} \beta^2 \hat{q}_0 q_0 - \beta \hat{m} m \right), \quad (\text{B.29})$$

in the $\rho \rightarrow \infty$ limit we need to rescale the order parameters as follows:

$$q_2 = q_1 + \frac{\delta q}{\rho} \quad \hat{q}_1 - \hat{q}_2 = \frac{\rho}{\nu} \delta \hat{q} \quad (\text{B.30})$$

$$\hat{q}_{1/0} = \left(\frac{\rho}{\nu} \right)^2 \hat{q}_{1/0} \quad \hat{m} = \frac{\rho}{\nu} \hat{m}. \quad (\text{B.31})$$

Plugging the new order parameters into the expression for the interaction term, one gets:

$$G_I = -\hat{q}\tilde{q} + s \left(-\frac{1}{2} \beta \left(\frac{\rho^2}{\nu^2} \hat{q}_1 - \frac{\rho}{\nu} \delta \hat{q} \right) \left(q_1 + \frac{\delta q}{\rho} \right) - \frac{\rho^2}{\nu^2} \frac{\beta(\beta-1)}{2} \hat{q}_1 q_1 + \frac{\rho^2}{\nu^2} \frac{1}{2} \beta^2 \hat{q}_0 q_0 - \frac{\rho}{\nu} \beta \hat{m} m \right). \quad (\text{B.32})$$

The values of $\hat{q}$ and $\tilde{q}$ can be determined independently of the other order parameters in the small s limit. In this setting we have $\tilde{q} = \int dw^* P(w^*) (w^*)^2$ and $\hat{q} = 0$. This is also obtained by requiring $\phi(s=0) = 0$. In the case of a Gaussian prior for the true signal, we have $\tilde{q} = \int Dw^* (w^*)^2 = 1$. Since the saddle point value for $\hat{q} = 0$, by taking the limit for $s \rightarrow 0$ and $\beta \rightarrow 0$, one obtains:

$$\mathcal{G}_I = \lim_{s \rightarrow 0} \lim_{\beta \rightarrow 0} \frac{G_I}{s} = \frac{1}{2} \left(\delta \hat{q} q_1 - \frac{\hat{q}_1 \delta q}{\nu} - \hat{q}_1 q_1 + \hat{q}_0 q_0 - 2 \hat{m} m \right) \quad (\text{B.33})$$

Entropic term

We can now consider the entropic term, which, after inserting the RS ansatz and performing some mathematical manipulation, can be rewritten as:

$$G_S = \int dw^* P(w^*) e^{\hat{q}(w^*)^2} \int \prod_{ac} dw_{ac} e^{-\frac{\lambda}{2} \frac{\nu}{\beta} (w_{ac})^2} P(w_{ac}) \quad (\text{B.34})$$

$$\times e^{\sum_{\{a,b,c,d\} \in D} \hat{q}_{ac,bd} w_{ac} w_{bd} + \sum_{ac} \hat{m}_{ac} w^* w_{ac,i}} \quad (\text{B.35})$$

$$= \int dw^* P(w^*) e^{\hat{q}(w^*)^2} \int \prod_{ac} dw_{ac} P(w_{ac}) \quad (\text{B.36})$$

$$\times e^{\frac{1}{2} \left(\hat{q}_2 - \hat{q}_1 - \frac{\nu\lambda}{\beta} \right) \sum_a \sum_c (w_{ac})^2 + \frac{1}{2} (\hat{q}_1 - \hat{q}_0) \sum_a \left(\sum_c w_{ac} \right)^2 + \frac{1}{2} \hat{q}_0 \left(\sum_a \sum_c w_{ac} \right)^2 + \hat{m} \sum_{ac} w_{ac}} \quad (\text{B.37})$$

$$= \int dw^* P(w^*) e^{\hat{q}(w^*)^2} \int Dz_0 \quad (\text{B.38})$$

$$\times \left[\int Dz_1 \left(\int dx P(x) e^{\frac{1}{2} \left(\hat{q}_2 - \hat{q}_1 - \frac{\nu\lambda}{\beta} \right) x^2 + \left(\sqrt{\hat{q}_1 - \hat{q}_0} z_1 + \sqrt{\hat{q}_0} z_0 + \hat{m} w^* \right) x} \right)^{\beta} \right]^s. \quad (\text{B.39})$$

We have introduced the integration over the auxiliary variables z_0 and z_1 , using the Hubbard Stratonovich transformation. This transformation is used for linearizing the exponential term in the interaction and entropic terms (the transformation is presented in a detailed way in Chapter 1). By assuming Gaussian prior for the signal $\mathbf{w}^*$ and flat prior for the weights $\mathbf{w}_{ac}$, we can rewrite the entropic term, in the limit $\beta \rightarrow 0$ and $s \rightarrow 0$, as:

$$\mathcal{G}_S = \lim_{s \rightarrow 0} \lim_{\beta \rightarrow 0} \frac{1}{s} \log G_S \quad (\text{B.40})$$

$$= \lim_{s \rightarrow 0} \lim_{\beta \rightarrow 0} \frac{1}{s} \log \int Dw^* \int Dz_0 \quad (\text{B.41})$$

$$\times \left[\int Dz_1 \left(\int \frac{dx}{\sqrt{2\pi}} e^{-\frac{\rho}{\nu} \left(\frac{1}{2} (\delta\hat{q} + \nu\lambda) x^2 + \left(\sqrt{\hat{q}_1 - \hat{q}_0} z_1 + \sqrt{\hat{q}_0} z_0 + \hat{m} w^* \right) x \right)} \right)^{\beta} \right]^s \quad (\text{B.42})$$

$$= \int Dw^* \int Dz_0 \log \left[\frac{e^{\frac{(\sqrt{\hat{q}_0} z_0 + \hat{m} w^*)^2}{2((\delta\hat{q} + \nu\lambda) - (\hat{q}_1 - \hat{q}_0))}}}{\sqrt{1 - \frac{\hat{q}_1 - \hat{q}_0}{(\delta\hat{q} + \nu\lambda)}}} \right] \quad (\text{B.43})$$

$$= \frac{1}{2} \log \left(\frac{\delta\hat{q} + \nu\lambda}{(\delta\hat{q} + \nu\lambda) - (\hat{q}_1 - \hat{q}_0)} \right) + \frac{\hat{q}_0 + \hat{m}^2}{2((\delta\hat{q} + \nu\lambda) - (\hat{q}_1 - \hat{q}_0))} \quad (\text{B.44})$$

Energetic term

Finally we can consider the energetic term. After inserting the RS ansatz and performing some mathematical manipulations, we can rewrite the energetic term as:

$$G_E = \sum_{S^a=\pm 1} e^{\zeta \sum_a S^a} \int \frac{du_0}{\sqrt{2\pi}} \prod_{ac} \frac{du_{ac} d\hat{u}_{ac}}{2\pi} \left(\int \frac{d\hat{u}_0}{\sqrt{2\pi}} e^{-\frac{1}{2}(\hat{u}_0)^2 \bar{q} + i\hat{u}_0(u_0 + i \sum_{ac} \hat{u}_{ac} m)} \right) \quad (\text{B.45})$$

$$\times e^{\sum_{ac} i\hat{u}_{ac} u_{ac} - \frac{1}{2} \sum_{ac,bd} \hat{u}_{ac} \hat{u}_{bd} q_{ac,bd}} \prod_a \frac{1}{\sqrt{2\pi\Delta_a}} e^{-\sum_{ac} \frac{(S^a u_0 - u_{ac})^2}{2\Delta_a}} \quad (\text{B.46})$$

$$= \int Dz_0 \int \frac{du_0}{\sqrt{2\pi}} e^{-\frac{u_0^2}{2\bar{q}}} \left[\sum_{S=\pm 1} e^{\zeta S} \int Dz_1 \right. \quad (\text{B.47})$$

$$\times \left(\int \frac{du d\hat{u}}{2\pi} e^{-\frac{1}{2}(q_2 - q_1)\hat{u}^2 + i\hat{u} \left(u - m \frac{u_0}{\bar{q}} + \sqrt{q_1 - q_0} z_1 + \sqrt{q_0 - \frac{m^2}{\bar{q}}} z_0 \right)} \frac{1}{\sqrt{2\pi\Delta}} e^{-\frac{(Su_0 - u)^2}{2\Delta}} \right)^{\beta} \Bigg]^s \quad (\text{B.48})$$

$$= \int Dz_0 \int Du_0 \quad (\text{B.49})$$

$$\times \left[e^{-\zeta} \int Dz_1 \left(\int Du \frac{1}{\sqrt{2\pi\Delta}} e^{-\frac{\left(-\left(\sqrt{\bar{q} - \frac{m^2}{q_0}} u_0 + \frac{m}{\sqrt{q_0}} z_0 \right) - \left(u \sqrt{q_2 - q_1} + \sqrt{q_1 - q_0} z_1 + \sqrt{q_0} z_0 \right) \right)^2}{2\Delta}} \right)^{\beta} \right]^s \quad (\text{B.50})$$

$$+ e^{\zeta} \int Dz_1 \left(\int Du \frac{1}{\sqrt{2\pi\Delta}} e^{-\frac{\left(\left(\sqrt{\bar{q} - \frac{m^2}{q_0}} u_0 + \frac{m}{\sqrt{q_0}} z_0 \right) - \left(u \sqrt{q_2 - q_1} + \sqrt{q_1 - q_0} z_1 + \sqrt{q_0} z_0 \right) \right)^2}{2\Delta}} \right)^{\beta} \Bigg]^s. \quad (\text{B.51})$$

After introducing the auxiliary variables z_0 and z_1 , via the Hubbard Stratonvic transformation, performing the finite temperature limit $\rho \rightarrow \infty$ (it is clear that the definition of ρ is contained in $\Delta = \beta/\nu$), and the small s and β limits, we can rewrite the energetic term as:

$$\mathcal{G}_E = \lim_{s \rightarrow 0} \lim_{\beta \rightarrow 0} \frac{\log G_E}{s} \quad (\text{B.52})$$

$$= \int Dz_0 \int Du_0 \log \quad (\text{B.53})$$

$$\times \left[e^{-\zeta} \int Dz_1 e^{-\nu \left[\min_u \frac{u^2}{2} + \left(u\sqrt{\delta q} + \sqrt{\tilde{q}}u_0 + \sqrt{q_0 - m^2}z_0 + mu_0 + \sqrt{q_1 - q_0}z_1 \right)^2 \right]} \right] \quad (\text{B.54})$$

$$+ e^{\zeta} \int Dz_1 e^{-\nu \left[\min_u \frac{u^2}{2} + \left(u\sqrt{\delta q} - \sqrt{\tilde{q}}u_0 + \sqrt{q_0 - m^2}z_0 + mu_0 + \sqrt{q_1 - q_0}z_1 \right)^2 \right]} \right] \quad (\text{B.55})$$

$$= \int Dz_0 \int Du_0 \log \quad (\text{B.56})$$

$$\times \left[\frac{\sqrt{1 + 2\delta q}}{\sqrt{1 + 2\delta q + 2\nu(q_1 - q_0)}} e^{-\nu \frac{(m^2 + \tilde{q})u_0^2 + (q_0 - m^2)z_0^2 + 2m\sqrt{q_0 - m^2}u_0z_0}{(1 + 2\delta q) + 2\nu(q_1 - q_0)}} \right] \quad (\text{B.57})$$

$$\times \left(e^{-\zeta - 2\nu \frac{\sqrt{\tilde{q}}u_0(mu_0 + \sqrt{q_0 - m^2}z_0)}{(1 + 2\delta q) + 2\nu(q_1 - q_0)}} + e^{\zeta + 2\nu \frac{\sqrt{\tilde{q}}u_0(mu_0 + \sqrt{q_0 - m^2}z_0)}{(1 + 2\delta q) + 2\nu(q_1 - q_0)}} \right). \quad (\text{B.58})$$

And finally:

$$\mathcal{G}_E = \frac{1}{2} \log \left(\frac{1 + 2\delta q}{1 + 2\delta q + 2\mu(q_1 - q_0)} \right) - \nu \frac{\tilde{q} + q_0}{(1 + 2\delta q) + 2\nu(q_1 - q_0)} \quad (\text{B.59})$$

$$+ \int Dz_0 \int Du_0 \log \left[2 \cosh \left(\zeta + 2\nu \frac{\sqrt{\tilde{q}}(mu_0 + \sqrt{q_0 - m^2}z_0)u_0}{(1 + 2\delta q) + 2\nu(q_1 - q_0)} \right) \right] \quad (\text{B.60})$$

Saddle point equations

Once the final expression of the free entropy is determined, one can compute its typical value by solving the following Saddle point equations:

$$\frac{\partial \mathcal{G}_I}{\partial \delta q} + \alpha \frac{\partial \mathcal{G}_E}{\partial \delta q} = 0 \quad \frac{\partial \mathcal{G}_I}{\partial \delta \hat{q}} + \frac{\partial \mathcal{G}_S}{\partial \delta \hat{q}} = 0 \quad (\text{B.61})$$

$$\frac{\partial \mathcal{G}_I}{\partial q_1} + \alpha \frac{\partial \mathcal{G}_E}{\partial q_1} = 0 \quad \frac{\partial \mathcal{G}_I}{\partial \hat{q}_1} + \frac{\partial \mathcal{G}_S}{\partial \hat{q}_1} = 0 \quad (\text{B.62})$$

$$\frac{\partial \mathcal{G}_I}{\partial q_0} + \alpha \frac{\partial \mathcal{G}_E}{\partial q_0} = 0 \quad \frac{\partial \mathcal{G}_I}{\partial \hat{q}_0} + \frac{\partial \mathcal{G}_S}{\partial \hat{q}_0} = 0 \quad (\text{B.63})$$

$$\frac{\partial \mathcal{G}_I}{\partial m} + \alpha \frac{\partial \mathcal{G}_E}{\partial m} = 0 \quad \frac{\partial \mathcal{G}_I}{\partial \hat{m}} + \frac{\partial \mathcal{G}_S}{\partial \hat{m}} = 0 \quad (\text{B.64})$$

$$(\text{B.65})$$

B.2 Stabilities analysis

In this section, we provide the detailed computation of the distribution of stabilities presented in Section 3.3.3. Stabilities represent the measurement amplitudes given by a signal estimate $\mathbf{w}$:

$$s^\mu = \frac{\mathbf{w} \cdot \mathbf{x}^\mu}{\sqrt{N}}.$$

The distribution of stabilities is defined as the expected value of the Dirac delta $\delta(\cdot)$ of the difference between the magnitude of the measurement s and the stability. Since all patterns are statistically equivalent, we can just compute the average stability of the first pattern:

$$s^1 = \frac{\mathbf{w} \cdot \mathbf{x}^1}{\sqrt{N}},$$

thus obtaining the distribution of stabilities as:

$$P(s) = \left\langle \delta \left(s - \frac{\mathbf{w} \cdot \mathbf{x}^1}{\sqrt{N}} \right) \right\rangle = \mathbb{E}_{X, \mathbf{w}^*} \left[\frac{\sum_{\{S\}} e^{\zeta \sum_{\mu=1}^M S_\mu - \nu \mathcal{L}(\mathbf{S} | X, \mathbf{w}^*)} \delta \left(s - \frac{\mathbf{w} \cdot \mathbf{x}^1}{\sqrt{N}} \right)}{\sum_{\{S\}} e^{\zeta \sum_{\mu=1}^M S_\mu - \nu \mathcal{L}(\mathbf{S} | X, \mathbf{w}^*)}} \right],$$

The presence of the patterns $\mathbf{x}^\mu$ at the denominator makes this computation hard; however, we can introduce two sets of replicas $a \in \{1, \dots, n\}$ and $c \in \{1, \dots, \beta\}$, with $\beta = \nu/\rho$. By replicating the procedure reported in Section B.1, we can rewrite the distribution of stabilities as:

$$P(s) = \lim_{n \rightarrow 0} \mathbb{E}_{X, \mathbf{w}^*} \sum_{\{S^a\}} \prod_a e^{\zeta \sum_\nu S_\mu^a} \left[\int \prod_{a=1}^s \prod_{c=1}^\beta d\mathbf{w}_{ac} P(\mathbf{w}_{ac}) \right. \\ \left. \times \prod_{\mu=1}^{\alpha N} \exp \left(-\frac{\nu}{\beta} \sum_{ac} \left(S_\mu^a \frac{\mathbf{x}^\mu \cdot \mathbf{w}^*}{\sqrt{N}} - \frac{\mathbf{x}^\mu \cdot \mathbf{w}_{ac}}{\sqrt{N}} \right)^2 - \sum_{ac} \frac{\lambda \nu}{2\beta} \|\mathbf{w}_{ac}\|_2^2 \right) \delta \left(s - \frac{\mathbf{x}^1 \cdot \mathbf{w}_{1,1}}{\sqrt{N}} \right) \right],$$

By construction, it is clear that the replica $a = 1$ and $c = 1$, and the pattern index $\mu = 1$ are special and must be considered separately in the computation. For this reason, we need to separate their contribution from the others:

$$\begin{aligned}
P(s) = & \lim_{n \rightarrow 0} \mathbb{E}_{X, \mathbf{w}^*} \int \prod_{a=1}^s \prod_{c=1}^{\beta} d\mathbf{w}_{ac} P(\mathbf{w}_{ac}) \delta \left(s - \frac{\mathbf{x}^1 \cdot \mathbf{w}_{1,1}}{\sqrt{N}} \right) \times \\
& \times \sum_{S^1} e^{\zeta S^1} \exp \left(-\frac{\nu}{\beta} \left(S^1 \frac{\mathbf{x}^1 \cdot \mathbf{w}^*}{\sqrt{N}} - s \right)^2 - \frac{\lambda \nu}{2\beta} \|\mathbf{w}_{1,1}\|_2^2 \right) \times \\
& \times \prod_{c>1} \exp \left(-\frac{\nu}{\beta} \sum_{c>1} \left(S^1 \frac{\mathbf{x}^1 \cdot \mathbf{w}^*}{\sqrt{N}} - \frac{\mathbf{x}^1 \cdot \mathbf{w}_{1c}}{\sqrt{N}} \right)^2 - \sum_{c>1} \frac{\lambda \nu}{2\beta} \|\mathbf{w}_{1c}\|_2^2 \right) \times \\
& \times \sum_{\{S^a\}} \prod_{a>1, c} e^{\zeta S^a} \exp \left(-\frac{\nu}{\beta} \sum_{a>1, c} \left(S^a \frac{\mathbf{x}^1 \cdot \mathbf{w}^*}{\sqrt{N}} - \frac{\mathbf{x}^1 \cdot \mathbf{w}_{ac}}{\sqrt{N}} \right)^2 - \sum_{a>1, c} \frac{\lambda \nu}{2\beta} \|\mathbf{w}_{ac}\|_2^2 \right) \times \\
& \times \sum_{\{S^a\}} \prod_{a, c} e^{\zeta \sum_{\nu} S_{\mu}^a} \prod_{\mu=2}^{\alpha N} \exp \left(-\frac{\nu}{\beta} \sum_{ac} \left(S_{\mu}^a \frac{\mathbf{x}^{\mu} \cdot \mathbf{w}^*}{\sqrt{N}} - \frac{\mathbf{x}^{\mu} \cdot \mathbf{w}_{ac}}{\sqrt{N}} \right)^2 - \sum_{ac} \frac{\lambda \nu}{2\beta} \|\mathbf{w}_{ac}\|_2^2 \right).
\end{aligned}$$

where we choose a flat prior for $\mathbf{w}_{ac}$, a normal prior for the hidden signal, and we assume the patterns to be distributed normally. Using the Replica Symmetric ansatz reported in Section B.1 and averaging over the patterns, one can rewrite the expression for the distribution of stabilities as [Engel and Van den Broeck \[2001\]](#):

$$\begin{aligned}
P(s) = & n \int Dz_0 \int Du_0 \frac{\sum_{s=\{\pm 1\}} e^{-s\zeta} \int Dz_1 \left(\int \frac{du}{\sqrt{2\pi}} e^{-\rho \left(\frac{u^2}{2} + \omega(\delta q, s, u, u_0, q_0, q_1, m, z_0, z_1)^2 \right)} \right)^{\nu/\rho-1}}{\sum_s e^{-s\zeta} \int Dz_1 \left(\int \frac{du}{\sqrt{2\pi}} e^{-\rho \left(\frac{u^2}{2} + \omega(\delta q, s, u, u_0, q_0, q_1, m, z_0, z_1)^2 \right)} \right)^{\nu/\rho}} \times \\
& \times \int \frac{du}{\sqrt{2\pi}} e^{-\rho \left(\frac{u^2}{2} + \omega(\delta q, s, u, u_0, q_0, q_1, m, z_0, z_1)^2 \right)} \times \\
& \times \delta \left(s - \left(u\sqrt{\delta q} + z_1\sqrt{q_1 - q_0} + \sqrt{q_0 - m^2 z_0 + m u_0} \right) \right)
\end{aligned}$$

where

$$\omega(\delta q, \tilde{q}, s, u, u_0, q_0, q_1, m, z_0, z_1) = u\sqrt{\delta q} + s u_0 \sqrt{\tilde{q}} + \sqrt{q_0 - m^2 z_0 + m u_0} + \sqrt{q_1 - q_0} z_1$$

Then, by taking the limit for vanishing number of replicas $n \rightarrow 0$ $\beta = \nu/\rho \rightarrow 0$, and manipulating, it is possible to write the expression for the distribution of stabilities as

$$P(s) = \int Dz_0 \int Du_0 \tag{B.66}$$

$$\begin{aligned}
& \times \frac{\frac{(1+2\delta q)}{\sqrt{2\pi(q_1-q_0)}} \sum_{s \in \{-1, 1\}} \frac{e^{-\frac{z_+^*(s, \delta q, q_1, q_0, m, s)^2}{2}}}{\sqrt{2\pi(q_1-q_0)}} e^{-s\zeta} e^{-\nu(su_0+s)^2(1+2\delta q)}}{\sqrt{\frac{1+2\delta q}{(1+2\delta q)+2\nu(q_1-q_0)}} e^{-\nu \frac{\left(\sqrt{q_0-m^2 z_0+m u_0} \right)^2 + u_0^2}{(1+2\delta q)+2\nu(q_1-q_0)}} 2 \cosh \left(\zeta + 2\nu \frac{\left(\sqrt{q_0-m^2 z_0+m u_0} \right) u_0}{(1+2\delta q)+2\nu(q_1-q_0)} \right)}, \tag{B.67}
\end{aligned}$$

with

$$z_1^*(s) = \frac{2\delta q s \sqrt{\tilde{q}} u_0 - \sqrt{q_0 - m^2} z_0 - m u_0 + (1 + 2\delta q) s}{\sqrt{q_1 - q_0}}.$$

The distribution of stabilities, associated with the correct sign choice, is obtained by choosing $s = 1$ in Equation (B.67) expression.

B.3 Replica Symmetry breaking

As discussed in Section 3.3.2, the Replica Symmetric (RS) ansatz, which assumes uniform behavior across all replicas, may fail to encapsulate the system's complexity fully. To evaluate whether a more refined ansatz incorporating Replica Symmetry Breaking (RSB) is necessary, one can analyze the nature of the RS solutions, which may correspond to either local or global optimizers of the free entropy ϕ [Antenucci et al., 2019a, Franz et al., 2017]. Specifically, if the RS solutions are local optimizers of the free entropy, large enough perturbations to the system lead to modifications in the solutions of the saddle point equations. Conversely, no perturbation alters the RS solution if the RS solutions represent a global optimum. While local stability guarantees that the RS phase is robust against small perturbations, it does not imply that the RS phase constitutes the most stable (i.e., globally optimal) configuration.

In the following section, we do not perform the analysis of the stability of the RS solution (which is out of the scope of this chapter). However, we provide the necessary tools to perform such analysis: we report the expression of the free entropy by breaking the symmetry for the replica index associated with the external (sign reconstruction) problem. We expect that, once Phase retrieval is divided into two sub-optimization problems, the internal problem is convex and thus is correctly described by the RS solution, while the external one is not.

1-step Replica Symmetry Breaking

To start the analysis of the RS solution's nature, it is natural first to question whether it constitutes a global optimizer of the free entropy ϕ . A rigorous assessment of its global stability requires replacing the Replica Symmetric (RS) ansatz in Equation (3.16) with a 1-Replica Symmetry Breaking (1RSB) ansatz. While the RS ansatz assumes identical interactions among all replicas, the 1RSB ansatz introduces a hierarchical structure, grouping replicas into N/τ non-overlapping clusters, each containing τ elements (τ is called *Parisi parameter*) [Parisi, 1979, 1983]. Interactions between replicas within the same group differ from those between replicas in different groups. The new form of the overlaps $q_{ac,bd}$ and m_{ac} follows:

$$q_{ac,bd} = \begin{cases} q_2 & a = b, c = d \\ q_1 & a = b, c > d \\ q_0^d & a > b, B_a = B_b \\ q_0^0 & a > b, B_a \neq B_b \end{cases}; \quad (\text{B.68})$$

$$m_{ac} = m.$$

In Equation (B.68), $\{B_a\}_{a \in \{1, \dots, N/\tau\}}$ is the set of different groups of replicas.

The new structure of interacting replicas can be used to compute the 1RSB free entropy expression. The procedure is the same as described in Section 3.3.1. In the zero temperature limit, $\rho \rightarrow \infty$, we impose the standard rescaling of the order parameters:

$$\begin{aligned} q_2 &= q_1 + \frac{\delta q}{\rho}; & \hat{q}_1 - \hat{q}_2 &= \frac{\rho}{\nu} \delta \hat{q}; \\ \hat{q}_{1/0} &= \left(\frac{\rho}{\nu}\right)^2 \hat{q}_{1/0}; & \hat{m} &= \frac{\rho}{\nu} \hat{m}. \end{aligned} \quad (\text{B.69})$$

By plugging the rescaled order parameters in Equation (3.14) we can compute the final expression for the 1RSB versions of the Entropic, Energetic, and Interaction terms:

$$\begin{aligned} \mathcal{G}_I &= \lim_{s \rightarrow 0} \frac{G_I}{s} = \frac{1}{2} \left(\delta \hat{q} q_1 - \frac{\hat{q}_1 \delta q}{\nu} - \hat{q}_1 q_1 - (\tau - 1) \hat{q}_0^d q_0^d + \tau \hat{q}_0^0 q_0^0 - 2 \hat{m} m \right); \\ \mathcal{G}_S &= \lim_{s \rightarrow 0} \frac{\ln G_S}{s} = -\frac{1}{2} \log \left(1 - \frac{\hat{q}_1 - \hat{q}_0^d}{\delta \hat{q}} \right) - \frac{1}{2\tau} \log \left(1 - \frac{\tau(\hat{q}_0^d - \hat{q}_0^0)}{(\delta \hat{q} - (\hat{q}_1 - \hat{q}_0^d))} \right) \\ &\quad + \frac{(\hat{q}_0^0 + \hat{m}^2)}{2 \left((\delta \hat{q} - (\hat{q}_1 - \hat{q}_0^d)) - \tau(\hat{q}_0^d - \hat{q}_0^0) \right)}; \\ \mathcal{G}_E &= \lim_{s \rightarrow 0} \frac{\ln G_E}{s} = \frac{1}{2} \log \left(\frac{1}{1 + \frac{2\nu(q_1 - q_0^d)}{1 + 2\delta q}} \right) \\ &\quad + \frac{1}{\tau} \int D z_0 \int D u_0 \log \int D z \left[\frac{\cosh \left(\zeta + 2\nu \frac{\varrho(z_0, z, u_0, q_0^0, m, q_0^d)}{\varpi(\delta q, q_1, q_0^d)/\sqrt{\bar{q}} u_0} \right)}{e^{\frac{\nu \bar{q} u_0^2 + \varrho(z_0, z, u_0, q_0^0, m, q_0^d)^2}{\varpi(\delta q, q_1, q_0^d)}}} \right]^\tau, \end{aligned} \quad (\text{B.70})$$

where

$$\begin{aligned} \varpi(\delta q, q_1, q_0^d) &= (1 + 2\delta q) + 2\nu (q_1 - q_0^d), \\ \varrho(z_0, z, u_0, q_0^0, m, q_0^d) &= z_0 \sqrt{q_0^0 - m^2} + z \sqrt{q_0^d - q_0} + m u_0. \end{aligned}$$

Equation (B.70) can now be used to investigate the RS solutions' global stability. They are globally stable if the RS saddle point is a stable fixed point of the 1RSB free entropy with unitary Parisi parameter. This finally indicates that RS solutions are not globally stable [Barbier et al., 2024].

Appendix C

Hessian appendix

In the main text, the fundamental steps of the replica computation have been reported. The scope of this appendix is to give more insights into how the final expression for the free entropy ϕ is obtained.

C.1 General Replica computation

Let's start from the definition of ϕ , reported in Equation (4.21):

$$\phi = \lim_{N \rightarrow \infty} \frac{1}{N} \lim_{n \rightarrow 0} \lim_{s \rightarrow 0} \partial_s \tilde{\phi} \quad (\text{C.1})$$

$$= \lim_{N \rightarrow \infty} \frac{1}{N} \mathbb{E}_X \lim_{n \rightarrow 0} Z^{n-1} \int dP(\mathbf{w}) \lim_{s \rightarrow 0} \partial_s Z_H^s(z) e^{-\beta \sum_{\mu} \ell \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g \left(\sqrt{\frac{K}{N}} \sum_{i=1}^{N/K} w_{li}^a x_{li}^{\mu} \right) - \kappa \right)}. \quad (\text{C.2})$$

where Z is the partition function, the normalization constant of the Gibbs distribution of the weights, $Z_H(z)$ is given by Equation (4.8) and $P(\mathbf{w})$ is the prior over the weights of the first layer of the tree committee machine. One can now explicit the expression for Z and Z_H ,

$$\phi = \lim_{N \rightarrow \infty} \frac{1}{N} \lim_{n \rightarrow 0} \lim_{s \rightarrow 0} \mathbb{E}_X \partial_s \int \prod_{a=1}^n dP(\mathbf{w}^a) e^{-\beta \sum_{\mu} \ell \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g \left(\sqrt{\frac{K}{N}} \sum_{i=1}^{N/K} w_{li}^a x_{li}^{\mu} \right) - \kappa \right)} \quad (\text{C.3})$$

$$\times \int \prod_{l,i,c} \frac{dw_{li}^c}{\sqrt{2\pi}} e^{+\frac{z-\lambda}{2} \sum_{l,i,c} (w_{li}^c)^2} \exp \left\{ -\frac{1}{2N} \sum_{ij,ll'} \sum_{\mu=1}^M c_l u_{li}^c x_{li}^{\mu} g' \left(\sqrt{\frac{K}{N}} \sum_{i=1}^{N/K} w_{li}^1 x_{li}^{\mu} \right) \right. \quad (\text{C.4})$$

$$\times \left. \ell'' \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g \left(\sqrt{\frac{K}{N}} \sum_{i=1}^{N/K} w_{li}^1 x_{li}^{\mu} \right) - \kappa \right) c_{l'} x_{l'j}^{\mu} u_{l'j}^c g' \left(\sqrt{\frac{K}{N}} \sum_{i=1}^{N/K} w_{l'i}^1 x_{l'i}^{\mu} \right) \right\} \quad (\text{C.5})$$

$$\times e^{-\frac{\sqrt{K}}{2N} \sum_{ij} \sum_{l=1}^K c_l \sum_{\mu=1}^M u_{li}^c x_{li}^{\mu} \ell' \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g \left(\sqrt{\frac{K}{N}} \sum_{i=1}^{N/K} w_{li}^1 x_{li}^{\mu} \right) - \kappa \right) u_{l'j}^c x_{l'j}^{\mu} g'' \left(\sqrt{\frac{K}{N}} \sum_{i=1}^{N/K} w_{l'i}^1 x_{l'i}^{\mu} \right)} \quad (\text{C.6})$$

To maintain the simplest possible notation, we first consider the computation of $\tilde{\phi}$. Once its expression is fully simplified, we compute its derivative for s and the limits for $s, n \rightarrow 0$.

We then perform the saddle point method [Mézard et al., 1987]. The first step of the replica computation consists of performing the change of variable:

$$\lambda_{la}^\mu = \sqrt{\frac{K}{N}} \sum_{i=1}^{N/K} w_{li}^a x_{li}^\mu \quad \text{and} \quad h_{lc}^\mu = \sqrt{\frac{K}{N}} \sum_j u_{lj}^c x_{lj}^\mu. \quad (\text{C.7})$$

By inserting their definition in $\tilde{\phi}$, by means of Dirac delta definition, one obtain:

$$\tilde{\phi} = \mathbb{E}_X \int \prod_{a=1}^n dP(\mathbf{w}^a) \prod_{l,i,c} \frac{du_{li}^c}{\sqrt{2\pi}} e^{\frac{z-\lambda}{2} \sum_{li,c} (u_{li}^c)^2} \prod_{a,\mu,l} \frac{d\hat{\lambda}_{la}^\mu d\lambda_{la}^\mu}{2\pi} \prod_{c,\mu,l} \frac{d\hat{h}_{lc}^\mu dh_{lc}^\mu}{2\pi} \quad (\text{C.8})$$

$$\times e^{i \sum_{l,a,\mu} \hat{\lambda}_{la}^\mu \lambda_{la}^\mu + i \sum_{l,c,\mu} \hat{h}_{lc}^\mu h_{lc}^\mu} \quad (\text{C.9})$$

$$\times e^{-i \sqrt{\frac{K}{N}} \sum_{\mu=1}^M \sum_{li} x_{li}^\mu (\sum_a \hat{\lambda}_{la}^\mu w_{li}^a + \sum_c \hat{h}_{lc}^\mu u_{li}^c)} \quad (\text{C.10})$$

$$\times e^{-\frac{1}{2\sqrt{K}} \sum_l c_l \sum_{\mu=1}^M h_{lc}^\mu \ell' \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g(\lambda_{l1}^\mu) - \kappa \right) g''(\lambda_{l1}^\mu) h_{lc}^\mu} \quad (\text{C.11})$$

$$\times e^{-\beta \sum_\mu \ell \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g(\lambda_{la}^\mu) - \kappa \right)} \quad (\text{C.12})$$

$$\times e^{-\frac{1}{2K} \sum_{ll'} \sum_{\mu=1}^M c_l h_{lc}^\mu g'(\lambda_{l1}^\mu) \ell'' \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g(\lambda_{l1}^\mu) - \kappa \right) c_{l'} h_{l'c}^\mu g'(\lambda_{l1}^\mu)} \quad (\text{C.13})$$

The exponential function is now linear in the patterns; thus, we can now perform the expectation over X from which one obtains:

$$= \int \prod_{a=1}^n dP(\mathbf{w}^a) \prod_{l,i,c} \frac{du_{li}^c}{\sqrt{2\pi}} e^{\frac{z-\lambda}{2} \sum_{li,c} (u_{li}^c)^2} \prod_{a,\mu} \frac{d\hat{\lambda}_{la}^\mu d\lambda_{la}^\mu}{2\pi} \prod_{c,\mu} \frac{d\hat{h}_{lc}^\mu dh_{lc}^\mu}{2\pi} \quad (\text{C.14})$$

$$\times e^{i \sum_{l,a,\mu} \hat{\lambda}_{la}^\mu \lambda_{la}^\mu + i \sum_{l,c,\mu} \hat{h}_{lc}^\mu h_{lc}^\mu} \quad (\text{C.15})$$

$$\times e^{-\frac{K}{2N} \sum_{\mu=1}^M \sum_{li} (\sum_a \hat{\lambda}_{la}^\mu w_{li}^a \sum_b \hat{\lambda}_{lb}^\mu w_{li}^b + 2 \sum_a \hat{\lambda}_{la}^\mu w_{li}^a \sum_c \hat{h}_{lc}^\mu u_{li}^c + \sum_c \hat{h}_{lc}^\mu u_{li}^c \sum_d \hat{h}_{ld}^\mu u_{li}^d)} \quad (\text{C.16})$$

$$\times e^{-\beta \sum_\mu \ell \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g(\lambda_{la}^\mu) - \kappa \right)} \quad (\text{C.17})$$

$$\times e^{-\frac{1}{2\sqrt{K}} \sum_l c_l \sum_{\mu=1}^M h_{lc}^\mu \ell' \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g(\lambda_{l1}^\mu) - \kappa \right) g''(\lambda_{l1}^\mu) h_{lc}^\mu} \quad (\text{C.18})$$

$$\times e^{-\frac{1}{2K} \sum_{ll'} \sum_{\mu=1}^M c_l h_{lc}^\mu g'(\lambda_{l1}^\mu) \ell'' \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g(\lambda_{l1}^\mu) - \kappa \right) c_{l'} h_{l'c}^\mu g'(\lambda_{l1}^\mu)}. \quad (\text{C.19})$$

For simplifying the notation, it is convenient to define the overlap parameters [Engel and Van den Broeck, 2001]

$$q_{ab}^l = \sum_i \frac{w_{il}^a w_{il}^b}{N} \quad p_{cd}^l = \sum_i \frac{u_{il}^c u_{il}^d}{N} \quad t_{ac}^l = \sum_i \frac{w_{il}^a u_{il}^c}{N}. \quad (\text{C.20})$$

By plugging the definition of the overlaps in the expression of $\tilde{\phi}$, using again the auxiliary Dirac delta function, one can rewrite $\tilde{\phi}$ as

$$\tilde{\phi} = \int \prod_{a=1}^n dP(\mathbf{w}^a) \prod_{l,i,c} \frac{du_{li}^c}{\sqrt{2\pi}} e^{\frac{z-\lambda}{2} \frac{N}{K} \sum_{lc} p_{cc}^l} \prod_{a,\mu} \frac{d\hat{\lambda}_{la}^\mu d\lambda_{la}^\mu}{2\pi} \prod_{c,\mu} \frac{d\hat{h}_{lc}^\mu dh_{lc}^\mu}{2\pi} \quad (\text{C.21})$$

$$\times e^{i \sum_{l,a,\mu} \hat{\lambda}_{la}^\mu \lambda_{la}^\mu + i \sum_{l,c,\mu} \hat{h}_{lc}^\mu h_{lc}^\mu} \quad (\text{C.22})$$

$$\times \int \prod_{a,b,l} \frac{\hat{q}_{ab}^l q_{ab}^l}{2\pi} \prod_{a,c,l} \frac{\hat{t}_{ac}^l t_{ac}^l}{2\pi} \prod_{c,d,l} \frac{\hat{p}_{cd}^l p_{cd}^l}{2\pi} \quad (\text{C.23})$$

$$\times e^{-\frac{N}{2K} \sum_{a,b,l} \hat{q}_{ab}^l q_{ab}^l - \frac{N}{K} \sum_{a,c,l} \hat{t}_{ac}^l t_{ac}^l - \frac{N}{2K} \sum_{c,d,l} \hat{p}_{cd}^l p_{cd}^l} \quad (\text{C.24})$$

$$\times e^{\frac{1}{2} \sum_{a,b,l} \hat{q}_{ab}^l \sum_i w_{il}^a w_{il}^b + \sum_{a,c,l} \hat{t}_{ac}^l \sum_i w_{il}^a u_{il}^c + \frac{1}{2} \sum_{c,d,l} \hat{p}_{cd}^l \sum_i u_{il}^c u_{il}^d} \quad (\text{C.25})$$

$$\times e^{-\frac{K}{2N} \sum_{\mu=1}^M \sum_l \left(\sum_{a,b} \hat{\lambda}_{la}^\mu \hat{\lambda}_{lb}^\mu q_{ab}^l + 2 \sum_{a,c} \hat{\lambda}_{la}^\mu \hat{h}_{lc}^\mu t_{ac}^l + \sum_{c,d} \hat{h}_{lc}^\mu \hat{h}_{ld}^\mu p_{cd}^l \right)} \quad (\text{C.26})$$

$$\times e^{-\beta \sum_\mu \ell \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g(\lambda_{l1}^\mu) - \kappa \right)} \quad (\text{C.27})$$

$$\times e^{-\frac{1}{2K} \sum_{ll'} \sum_{\mu=1}^M c_l h_{lc}^\mu g'(\lambda_{l1}^\mu) \ell'' \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g(\lambda_{l1}^\mu) - \kappa \right) c_{l'} h_{l'c}^\mu g'(\lambda_{l1}^\mu)} \quad (\text{C.28})$$

$$\times e^{-\frac{1}{2\sqrt{K}} \sum_l c_l \sum_{\mu=1}^M h_{lc}^\mu \ell' \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g(\lambda_{l1}^\mu) - \kappa \right) g''(\lambda_{l1}^\mu) h_{lc}^\mu}. \quad (\text{C.29})$$

After some algebraic manipulations, one can obtain the final expression for $\tilde{\phi}$, which is the starting point for the computation of the free entropy ϕ :

$$\tilde{\phi} = \int \prod_{a,b,l} \frac{d\hat{q}_{ab}^l dq_{ab}^l}{2\pi} \prod_{a,c,l} \frac{d\hat{t}_{ac}^l dt_{ac}^l}{2\pi} \prod_{c,d,l} \frac{d\hat{p}_{cd}^l dP_{cd}^l}{2\pi} e^{\frac{N}{K} (G_I + G_S + \alpha K G_E)} \quad (\text{C.30})$$

$$G_I = -\frac{1}{2} \sum_{a,b,l} \hat{q}_{ab}^l q_{ab}^l - \sum_{a,c,l} \hat{t}_{ac}^l t_{ac}^l - \frac{1}{2} \sum_{c,d,l} \hat{p}_{cd}^l p_{cd}^l + \frac{z-\lambda}{2} \sum_{l,c} p_{cc}^l \quad (\text{C.31})$$

$$G_S = \log \int \prod_{a=1}^n dP(\mathbf{w}^a) \prod_{l,i,c} \frac{du_{li}^c}{\sqrt{2\pi}} \quad (\text{C.32})$$

$$\times e^{\frac{1}{2} \sum_{a,b,l} \hat{q}_{ab}^l \sum_i w_{il}^a w_{il}^b + \sum_{a,c,l} \hat{t}_{ac}^l \sum_i w_{il}^a u_{il}^c + \frac{1}{2} \sum_{c,d,l} \hat{p}_{cd}^l \sum_i u_{il}^c u_{il}^d} \quad (\text{C.33})$$

$$G_E = \log \int \prod_{a,l} \frac{d\hat{\lambda}_{la} d\lambda_{la}}{2\pi} \prod_{c,l} \frac{d\hat{h}_{lc} dh_{lc}}{2\pi} e^{i \sum_{l,a} \hat{\lambda}_{la} \lambda_{la} + i \sum_{l,c} \hat{h}_{lc} h_{lc}} \quad (\text{C.34})$$

$$\times e^{-\frac{1}{2} \sum_l \left(\sum_{a,b} \hat{\lambda}_{la} \hat{\lambda}_{lb} q_{ab}^l + 2 \sum_{a,c} \hat{\lambda}_{la} \hat{h}_{lc} t_{ac}^l + \sum_{c,d} \hat{h}_{lc} \hat{h}_{ld} p_{cd}^l \right)} \quad (\text{C.35})$$

$$\times e^{-\frac{1}{2\sqrt{K}} \sum_l c_l h_{lc} \ell' \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g(\lambda_{l1}) - \kappa \right) g''(\lambda_{l1}) h_{lc}} \quad (\text{C.36})$$

$$\times e^{-\beta \ell \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g(\lambda_{l1}) - \kappa \right)} \quad (\text{C.37})$$

$$\times e^{-\frac{1}{2K} \sum_{ll'} c_l h_{lc} g'(\lambda_{l1}) \ell'' \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g(\lambda_{l1}) - \kappa \right) c_{l'} h_{l'c} g'(\lambda_{l1})} \quad (\text{C.38})$$

As usual in the context of statistical physics, in order to compute the free entropy, we need to chose an appropriate ansatz for the overlaps. We consider the Replica symmetric ansatz

(RS), which is the simplest one, and it is given by the following expressions:

$$\begin{aligned} q_{ab}^l &= \begin{cases} Q & a = b \\ q & a \neq b \end{cases} & \hat{q}_{ab}^l &= \begin{cases} -\hat{Q} & a = b \\ \hat{q} & a \neq b \end{cases} \\ p_{cd}^l &= \begin{cases} P & c = d \\ p & c \neq d \end{cases} & \hat{p}_{cd}^l &= \begin{cases} -\hat{P} & c = d \\ \hat{p} & c \neq d \end{cases} \\ t_{ac}^l &= \begin{cases} T & a = 1 \\ t & a \neq 1 \end{cases} & \hat{t}_{ac}^l &= \begin{cases} \hat{T} & a = 1 \\ \hat{t} & a \neq 1 \end{cases} \end{aligned}$$

We can now rewrite the three different terms in the expression of $\tilde{\phi}$, namely the interaction term G_I , the entropic term G_S , and the energetic term G_E , in terms of the overlaps defined above. We consider them one by one, starting from the interaction term G_I .

Interaction term

Using the RS ansatz, the interaction term G_I can be rewritten as follows:

$$G_I = -\frac{1}{2} \sum_{a,b,l} \hat{q}_{ab}^l q_{ab}^l - \sum_{a,c,l} \hat{t}_{ac}^l t_{ac}^l - \frac{1}{2} \sum_{c,d,l} \hat{p}_{cd}^l p_{cd}^l + \frac{z-\lambda}{2} \sum_{l,c} p_{cc}^l \quad (\text{C.39})$$

$$= \sum_{l=1}^K \left(\frac{1}{2} n \hat{Q} Q - \frac{1}{2} n(n-1) \hat{q} q - s \hat{T} T - s(n-1) \hat{t} t + \frac{1}{2} s \hat{P} P - \frac{1}{2} s(s-1) \hat{p} p + \frac{z-\lambda}{2} s P \right). \quad (\text{C.40})$$

We can now perform the derivative ∂_s and the limit for $n, s \rightarrow 0$, obtaining the final expression of the interaction term in the free entropy ϕ :

$$\mathcal{G}_I = \lim_{n \rightarrow 0} \lim_{s \rightarrow 0} \partial_s G_I \quad (\text{C.41})$$

$$= \sum_{l=1}^K \left(-\hat{T} T + \hat{t} t + \frac{1}{2} \hat{P} P + \frac{1}{2} \hat{p} p + \frac{z-\lambda}{2} P \right) \quad (\text{C.42})$$

$$= K \left(-\hat{T} T + \hat{t} t + \frac{1}{2} \hat{P} P + \frac{1}{2} \hat{p} p + \frac{z-\lambda}{2} P \right) \quad (\text{C.43})$$

Entropic term

The expression of the entropic term G_S , under the replica symmetric ansatz, can be rewritten as follows:

$$G_S = \log \int \prod_{a=1}^n dP(\mathbf{w}^a) \prod_{l,c} \frac{du_l^c}{\sqrt{2\pi}} e^{\frac{1}{2} \sum_{a,b,l} \hat{q}_{ab} w_l^a w_l^b + \sum_{a,c,l} \hat{t}_{ac} w_l^a u_l^c + \frac{1}{2} \sum_{c,d,l} \hat{p}_{cd} u_l^c u_l^d} \quad (\text{C.44})$$

$$= K \ln \int \prod_{a=1}^n dP(\mathbf{w}^a) \prod_c \frac{du_c}{\sqrt{2\pi}} e^{-\frac{1}{2} (\hat{P} + \hat{p}) \sum_c (u^c)^2 + \frac{1}{2} \hat{p} (\sum_c u^c)^2} \quad (\text{C.45})$$

$$\times e^{-\frac{1}{2} (\hat{Q} + \hat{q}) \sum_a (w^a)^2 + \frac{1}{2} \hat{q} (\sum_a w^a)^2 + (\hat{T} - \hat{t}) w^1 \sum_c u^c + \hat{t} \left[(\sum_a w^a + \sum_c u^c)^2 - (\sum_a w^a)^2 - \sum_c (u^c)^2 \right]} \quad (\text{C.46})$$

Then by linearizing the quadratic terms in the sum of the weights, $(\sum_a w^a)^2$, through

the Hubbard-Stratonovich approach, deriving for s and performing the limits $n, s \rightarrow \infty$, one obtains:

$$\mathcal{G}_S = \lim_{n \rightarrow 0} \lim_{s \rightarrow 0} \partial_s G_S$$

with

$$\frac{N}{K} \mathcal{G}_S = N \left(-\frac{1}{2} \ln (\hat{P} + \hat{p}) + \frac{1}{(\hat{P} + \hat{p})} \left[\frac{\hat{p}}{2} + \frac{(\hat{T} - \hat{t})^2 (\hat{q} + \frac{\hat{Q}}{2})}{(\hat{Q} + \hat{q})^2} + \hat{t} \frac{(\hat{T} - \hat{t})}{(\hat{Q} + \hat{q})} \right] \right) \quad (\text{C.47})$$

under the assumption that the prior over the weights is given by the exponential factor $e^{-\tilde{\lambda} \frac{x^2}{2}}$.

Energetic term

Regarding the energetic term, after the introduction of the Replica Symmetric ansatz and linearizing the quadratic term using the Hubbard-Stratonovich transformation, one obtains:

$$G_E = \log \int \prod_{a,l} \frac{d\hat{\lambda}_{la} d\lambda_{la}}{2\pi} \prod_{c,l} \frac{d\hat{h}_{lc} dh_{lc}}{2\pi} e^{i \sum_{l,a} \hat{\lambda}_{la} \lambda_{la} + i \sum_{l,c} \hat{h}_{lc} h_{lc}} \quad (\text{C.48})$$

$$\times e^{-\frac{1}{2} \sum_l \left(\sum_{a,b} \hat{\lambda}_{la} \hat{\lambda}_{lb} q_{ab}^l + 2 \sum_{a,c} \hat{\lambda}_{la} \hat{h}_{lc} t_{ac}^l + \sum_{c,d} \hat{h}_{lc} \hat{h}_{ld} p_{cd}^l \right)} \quad (\text{C.49})$$

$$\times e^{-\frac{1}{2\sqrt{K}} \sum_l c_l h_{lc} \ell' \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g(\lambda_{l1}) - \kappa \right) g''(\lambda_{l1}) h_{lc}} \quad (\text{C.50})$$

$$\times e^{-\beta \ell \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g(\lambda_{la}) - \kappa \right)} \quad (\text{C.51})$$

$$\times e^{-\frac{1}{2K} \sum_{l,l'} c_l h_{lc} g'(\lambda_{l1}) \ell'' \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g(\lambda_{l1}) - \kappa \right) c_{l'} h_{l'c} g'(\lambda_{l1})} \quad (\text{C.52})$$

$$= s \int \prod_l D x_l \prod_l D z_l \prod_l D \lambda_l \frac{e^{-\beta \ell \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g(\sqrt{Q-q} \lambda_l - x_l \sqrt{q}) - \kappa \right)}}{\mathcal{N}} \quad (\text{C.53})$$

$$\times \log \int \prod_l \frac{dh_l}{\sqrt{2\pi}} e^{-\sum_l \frac{\left(\sqrt{P-p} h_l + \frac{t}{\sqrt{q}} x_l - \frac{\lambda^l (T-t)}{\sqrt{Q-q}} + z_l \sqrt{\gamma} \right)^2}{2(P-p)}} \quad (\text{C.54})$$

$$\times e^{-\frac{1}{2\sqrt{K}} \ell' \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g(\sqrt{Q-q} \lambda_l - x_l \sqrt{q}) - \kappa \right) \sum_l c_l g''(\sqrt{Q-q} \lambda_l - x_l \sqrt{q}) (P-p) (h_l)^2} \quad (\text{C.55})$$

$$\times e^{-\frac{1}{2K} \ell'' \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g(\sqrt{Q-q} \lambda_l - x_l \sqrt{q}) - \kappa \right) \left(\sum_l c_l \sqrt{P-p} h_l g'(\sqrt{Q-q} \lambda_l - x_l \sqrt{q}) \right)^2}. \quad (\text{C.56})$$

Where we have defined

$$\mathcal{N} = \int \prod_l D \lambda_l e^{-\beta \ell \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g(\sqrt{Q-q} \lambda_l - x_l \sqrt{q}) - \kappa \right)} \quad (\text{C.57})$$

$$\gamma = p - \frac{t^2}{q} - \frac{(T-t)^2}{Q-q}. \quad (\text{C.58})$$

We can now compute the limit for $n, s \rightarrow \infty$ for the derivative in s of the energetic term, $\partial_s G_E$, which reads

$$\mathcal{G}_E = \lim_{n \rightarrow 0} \lim_{s \rightarrow 0} \partial_s G_E \quad (\text{C.59})$$

$$= \int \prod_l Dx_l \prod_l Dz_l \prod_l D\lambda_l \frac{e^{-\beta \ell \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g(\sqrt{Q-q}\lambda_l - x_l \sqrt{q}) - \kappa \right)}}{\mathcal{N}_l} \times \log \int \prod_l Dh_l \quad (\text{C.60})$$

$$\times e^{-\frac{1}{2K} \ell'' \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g(\sqrt{Q-q}\lambda_l - x_l \sqrt{q}) - \kappa \right) \left(\sum_l c_l \left(\sqrt{P-p}h_l - \frac{t}{\sqrt{q}}x_l + \frac{\lambda^l(T-t)}{\sqrt{Q-q}} - z_l \sqrt{\gamma} \right) g'(\sqrt{Q-q}\lambda_l - x_l \sqrt{q}) \right)^2} \quad (\text{C.61})$$

$$\times e^{-\frac{1}{2\sqrt{K}} \ell' \left(\frac{1}{\sqrt{K}} \sum_{l=1}^K c_l g(\sqrt{Q-q}\lambda_l - x_l \sqrt{q}) - \kappa \right) \sum_l c_l g''(\sqrt{Q-q}\lambda_l - x_l \sqrt{q}) \left(\sqrt{P-p}h_l - \frac{t}{\sqrt{q}}x_l + \frac{\lambda^l(T-t)}{\sqrt{Q-q}} - z_l \sqrt{\gamma} \right)^2} \quad (\text{C.62})$$

Using the results from the previous calculations, the final expression for the free entropy ϕ , under the Replica symmetric Ansatz is

$$\phi = \mathcal{G}_I + \mathcal{G}_S + \alpha \mathcal{G}_E, \quad (\text{C.63})$$

where the interaction, entropic are

$$\mathcal{G}_I = -\hat{T}T + \hat{t}t + \frac{1}{2}\hat{P}P + \frac{1}{2}\hat{p}p + \frac{z - \lambda}{2}P \quad (\text{C.64a})$$

$$\begin{aligned} \mathcal{G}_S &= \frac{1}{2} \ln \frac{2\pi}{\hat{P} + \hat{p}} + \frac{1}{\hat{P} + \hat{p}} \left[\frac{\hat{p}}{2} + \frac{(\hat{T} - \hat{t})^2 \left(\hat{q} + \frac{\hat{Q}}{2} \right)}{(\hat{Q} + \hat{q})^2} + \frac{\hat{t}(\hat{T} - \hat{t})}{\hat{Q} + \hat{q}} \right] \\ &= \frac{1}{2} \ln \frac{2\pi}{\hat{P} + \hat{p}} + \frac{1}{\hat{P} + \hat{p}} \left[\frac{\hat{p}}{2} + \frac{Q}{2} (\hat{T} - \hat{t})^2 + \hat{t}(\hat{T} - \hat{t})(Q - q) \right] \end{aligned} \quad (\text{C.64b})$$

$$\mathcal{G}_E = \int \prod_l Dx_l Dz_l D\lambda_l \frac{e^{-\beta \ell \left(\frac{1}{\sqrt{K}} \sum_l c_l g(\sqrt{Q-q}\lambda_l + \sqrt{q}x_l) - \kappa \right)}}{\int \prod_l D\lambda_l e^{-\beta \ell \left(\frac{1}{\sqrt{K}} \sum_l c_l g(\sqrt{Q-q}\lambda_l + \sqrt{q}x_l) - \kappa \right)}} \times \ln \int \prod_l Dh_l \quad (\text{C.64c})$$

$$\times e^{-\frac{1}{2} \ell'' \left(\frac{1}{\sqrt{K}} \sum_l c_l g(\sqrt{Q-q}\lambda_l + \sqrt{q}x_l) - \kappa \right) \left[\frac{1}{\sqrt{K}} \sum_l c_l \left(\sqrt{P-p}h_l + \frac{t}{\sqrt{q}}x_l + \frac{T-t}{\sqrt{Q-q}}\lambda_l + \sqrt{\gamma}z_l \right) g'(\sqrt{Q-q}\lambda_l + \sqrt{q}x_l) \right]^2} \quad (\text{C.64d})$$

$$\times e^{-\frac{1}{2} \ell' \left(\frac{1}{\sqrt{K}} \sum_l c_l g(\sqrt{Q-q}\lambda_l + \sqrt{q}x_l) - \kappa \right) \frac{1}{\sqrt{K}} \sum_l c_l \left(\sqrt{P-p}h_l + \frac{t}{\sqrt{q}}x_l + \frac{T-t}{\sqrt{Q-q}}\lambda_l + \sqrt{\gamma}z_l \right)^2 g''(\sqrt{Q-q}\lambda_l + \sqrt{q}x_l)} \quad (\text{C.64e})$$

where $\gamma \equiv p - \frac{t^2}{q} - \frac{(T-t)^2}{Q-q}$. As already mentioned in the main text, the values for the overlaps Q, q and their conjugates, are obtained by solving the saddle point equations for the quenched free entropy associated with the equilibrium distribution of the weights, $\phi^{(r)}$, in Equation (4.40), for the finite β case, and in Equation (4.51), for the zero temperature case. The equilibrium order parameters have been computed under the assumption of large width, $K \rightarrow \infty$. We can, then, use the saddle point equation to compute the final values of T, t, P, p , and their duals.

C.1.1 Saddle point equations conjugated parameters

The optimization of ϕ can be performed using the saddle point method [Mézard et al., 1987]. This amounts at solving the following system of equations:

$$P = -2 \frac{\partial \mathcal{G}_S}{\partial \hat{P}} \quad p = -2 \frac{\partial \mathcal{G}_S}{\partial \hat{p}} \quad T = \frac{\partial \mathcal{G}_S}{\partial \hat{T}} \quad t = -\frac{\partial \mathcal{G}_S}{\partial \hat{t}} \quad (\text{C.65})$$

$$\hat{P} = -z - 2 \frac{\partial \mathcal{G}_E}{\partial P} \quad \hat{p} = -2 \frac{\partial \mathcal{G}_E}{\partial p} \quad \hat{T} = \frac{\partial \mathcal{G}_E}{\partial T} \quad \hat{t} = -\frac{\partial \mathcal{G}_E}{\partial t} \quad (\text{C.66})$$

The saddle point equations obtained by taking derivatives with respect to conjugated order parameters provide an exact expression for P, p, T, t :

$$p = \frac{\hat{p} + Q(\hat{T} - \hat{t})^2 + 2(Q - q)\hat{t}(\hat{T} - \hat{t})}{(\hat{p} + \hat{P})^2} \quad (\text{C.67a})$$

$$P = \frac{1}{\hat{p} + \hat{P}} + \frac{\hat{p} + Q(\hat{T} - \hat{t})^2 + 2(Q - q)\hat{t}(\hat{T} - \hat{t})}{(\hat{p} + \hat{P})^2} \quad (\text{C.67b})$$

$$t = \frac{\hat{t}(Q - q) + Q(\hat{T} - \hat{t}) - (Q - q)(\hat{T} - \hat{t})}{\hat{p} + \hat{P}} \quad (\text{C.67c})$$

$$T = \frac{(Q - q)\hat{t} + Q(\hat{T} - \hat{t})}{\hat{p} + \hat{P}}. \quad (\text{C.67d})$$

C.1.2 Large width limit

As already introduced in section 4.2.1, the infinite width limit $K \rightarrow \infty$ is needed to get a closed form for the free entropy of the model. Here, we introduce the main steps required for performing the computation of $\mathcal{G}_E$ in the infinite width limit.

Let's start by introducing some quantities useful for enlightening the notation:

$$f^{(0)}(M, N, O) \equiv e^{-\frac{1}{2}\ell''(O)M^2 - \frac{1}{2}\ell'(O)N} \quad (\text{C.68})$$

where M, N, O are defined as follows:

$$M \equiv \frac{1}{\sqrt{K}} \sum_l c_l \left(\sqrt{P - p} h_l + \frac{t}{\sqrt{q}} x_l + \frac{T - t}{\sqrt{Q - q}} \lambda_l + \sqrt{\gamma} z_l \right) g' \left(\sqrt{Q - q} \lambda_l + \sqrt{q} x_l \right) \quad (\text{C.69a})$$

$$N \equiv \frac{1}{\sqrt{K}} \sum_l c_l \left(\sqrt{P - p} h_l + \frac{t}{\sqrt{q}} x_l + \frac{T - t}{\sqrt{Q - q}} \lambda_l + \sqrt{\gamma} z_l \right)^2 g'' \left(\sqrt{Q - q} \lambda_l + \sqrt{q} x_l \right) \quad (\text{C.69b})$$

$$O \equiv \frac{1}{\sqrt{K}} \sum_l c_l g \left(\sqrt{Q - q} \lambda_l + \sqrt{q} x_l \right) - \kappa \quad (\text{C.69c})$$

We start by applying the central limit theorem on the variables h_l ; to do so, we define

$$a_l \equiv \frac{t}{\sqrt{q}} x_l + \frac{T - t}{\sqrt{Q - q}} \lambda_l + \sqrt{\gamma} z_l. \quad (\text{C.70})$$

The argument of the log term in the G_E , in Equation (4.39), reads

$$\begin{aligned}
I &\equiv \int \prod_l Dh_l f^{(0)}(M, N, O) \\
&= \int \prod_l Dh_l \int \frac{dM d\hat{M}}{2\pi} \frac{dN d\hat{N}}{2\pi} e^{iM\hat{M} + iN\hat{N}} f^{(0)}(M, N, O) \\
&\times e^{-\frac{i\hat{M}}{\sqrt{K}} \sum_l c_l (\sqrt{P-p}h_l + a_l) g'(\sqrt{Q-q}\lambda_l + \sqrt{q}x_l)} e^{-\frac{i\hat{N}}{\sqrt{K}} \sum_l c_l (\sqrt{P-p}h_l + a_l)^2 g''(\sqrt{Q-q}\lambda_l + \sqrt{q}x_l)} \\
&= \int \frac{dM d\hat{M}}{2\pi} \frac{dN d\hat{N}}{2\pi} f^{(0)}(M, N, O) e^{i\hat{M}(M-M^{(0)}) + i\hat{N}(N-N^{(0)}) - \frac{\Delta_0}{2}\hat{M}^2 - \frac{\Xi_0}{2}\hat{N}^2 - \Omega_0\hat{M}\hat{N}}.
\end{aligned} \tag{C.71}$$

where the parameters $M^{(0)}$ and $N^{(0)}$ are obtained integrating respectively M and N over the Gaussian measure $\prod_l Dh_l$. They represents the gaussian mean of M and N (over h_l), and $\Delta_0, \Xi_0, \Omega_0$ are the associated variances and covariances. We are now going to use the general identity:

$$\begin{aligned}
&\int \prod_r \frac{dh_r d\hat{h}_r}{2\pi} \frac{dv d\hat{v}}{2\pi} f(\{h_r\}, v) e^{i\hat{h}_r(h_r - M_r) + i\hat{v}(v - N) - \frac{1}{2} \sum_{rs} \Delta_{rs} \hat{h}_r \hat{h}_s - \frac{\Xi}{2} \hat{v}^2 - \hat{v} \sum_r \Omega_r \hat{h}_r} \\
&\simeq \int \prod_r \frac{dh_r d\hat{h}_r}{2\pi} Dv f(\{h_r\}, N + \sqrt{\Xi}v) e^{i \sum_r \hat{h}_r (h_r - M_r - \frac{v}{\sqrt{\Xi}} \Omega_r) - \frac{1}{2} \sum_{rs} (\Delta_{rs} - \frac{\Omega_r \Omega_s}{\Xi}) \hat{h}_r \hat{h}_s} \\
&= \int Dv \prod_r Dh_r f\left(\left\{M_r + \frac{\Omega_r}{\sqrt{\Xi}} v + \sum_s (\sqrt{A})_{rs} h_s\right\}, N + \sqrt{\Xi}v\right)
\end{aligned} \tag{C.72}$$

with $A_{rs} \equiv \Delta_{rs} - \frac{\Omega_r \Omega_s}{\Xi}$ to rewrite the integral inside the logarithm in Equation (4.39) as

$$I = \int D\lambda_0 Ds_0 f^{(0)}\left(M^{(0)} + \frac{\Omega_0}{\sqrt{\Xi_0}} s_0 + \sqrt{\Delta_0 - \frac{(\Omega_0)^2}{\Xi_0}} \lambda_0, N^{(0)} + \sqrt{\Xi_0} s_0, O\right) \tag{C.73}$$

$$\equiv f^{(1)}(M^{(0)}, N^{(0)}, O). \tag{C.74}$$

Now that we have dealt with the logarithmic term inside the energetic term in Equation (4.39), we can integrate over the variables z_l , using the general result (C.72) which is valid for any function f . Performing the integration, we obtain

$$\begin{aligned}
&e^{-\beta\ell(O)} \int \prod_l Dz_l \ln f^{(1)}(M^{(0)}, N^{(0)}, O) \\
&= e^{-\beta\ell(O)} \int D\lambda_1 Ds_1 \ln f^{(1)}\left(M^{(1)} + \frac{\Omega_1}{\sqrt{\Xi_1}} s_1 + \sqrt{\Delta_1 - \frac{(\Omega_1)^2}{\Xi_1}} \lambda_1, N^{(1)} + \sqrt{\Xi_1} s_1, O\right) \\
&\equiv f^{(2)}(M^{(1)}, N^{(1)}, O),
\end{aligned} \tag{C.75}$$

where the new parameters $M^{(1)}, N^{(1)}$ are the means with respect to the Gaussian measure $\prod_l Dz_l$, and $\Delta^{(1)}, \Xi^{(1)}, \Omega^{(1)}$ are the associated variances and covariances. Now, we are going

to integrate over the λ_l variables. This time, we need to also extract the variable O , obtaining

$$\begin{aligned}
& \frac{1}{\mathcal{N}} \int \prod_l D\lambda_l f^{(2)}(M^{(1)}, N^{(1)}, O) \\
&= \frac{1}{\mathcal{N}} \int Dx_1 Dy_1 Dy_2 \\
&\times f^{(2)} \left(M^{(2)} + \frac{\Psi_0}{\sqrt{\Phi_0}} x_1 + \sum_{s=1}^2 \sqrt{J_{1s}} y_s, N^{(2)} + \frac{\Sigma_0}{\sqrt{\Phi_0}} x_1 + \sum_{s=1}^2 \sqrt{J_{2s}} y_s, O^{(2)} + \sqrt{\Phi_0} x_1 \right) \\
&\equiv f^{(3)}(M^{(2)}, N^{(2)}, O^{(2)})
\end{aligned} \tag{C.76}$$

where J is now a matrix with the following structure

$$J \equiv \begin{pmatrix} \Delta_2 - \frac{\Psi_0^2}{\Phi_0} & \Omega_2 - \frac{\Psi_0 \Sigma_0}{\Phi_0} \\ \Omega_2 - \frac{\Psi_0 \Sigma_0}{\Phi_0} & \Xi_2 - \frac{\Sigma_0^2}{\Phi_0} \end{pmatrix} \tag{C.77}$$

and the normalization is given by

$$\mathcal{N} \equiv \int \prod_l D\lambda_l e^{-\beta \ell \left(\frac{1}{\sqrt{K}} \sum_l c_l g(\sqrt{Q-q}\lambda_l + \sqrt{q}x_l) - \kappa \right)} \simeq \int Dx_1 e^{-\beta \ell (O^{(2)} + \sqrt{\Phi_0} x_1 - \kappa)}. \tag{C.78}$$

The index (2) indicates the variables obtained after the integration over the $\{\lambda_l\}_{l \in \{1, \dots, K\}}$ variables. We finally integrate over the $\{x_l\}_{l \in \{1, \dots, K\}}$ variables

$$\begin{aligned}
\mathcal{G}_E &= \int \prod_l Dx_l f^{(3)}(M^{(2)}, N^{(2)}, O^{(2)}) \\
&= \int Dx_2 Dz_1 Dz_2 f^{(3)} \left(\frac{\Psi_1}{\sqrt{\Phi_1}} x_2 + \sum_{s=1}^2 \sqrt{\tilde{J}_{1s}} z_s, \frac{\Sigma_1}{\sqrt{\Phi_1}} x_2 + \sum_{s=1}^2 \sqrt{\tilde{J}_{2s}} z_s, \sqrt{\Phi_1} x_2 \right),
\end{aligned} \tag{C.79}$$

where

$$\tilde{J} \equiv \begin{pmatrix} \Delta_3 - \frac{\Psi_1^2}{\Phi_1} & \Omega_3 - \frac{\Psi_1 \Sigma_1}{\Phi_1} \\ \Omega_3 - \frac{\Psi_1 \Sigma_1}{\Phi_1} & \Xi_3 - \frac{\Sigma_1^2}{\Phi_1} \end{pmatrix} \tag{C.80}$$

and again the index (3) indicates the variables obtained after the integration over the $\prod_l Dx_l$.

Collecting all the results we get

$$\begin{aligned}
\mathcal{G}_E = & \int \prod_{s=0}^1 Dx_s \prod_{s=1}^2 Dy_s Dz_s \frac{e^{-\beta\ell(\sqrt{\Phi_0}x_0 + \sqrt{\Phi_1}x_1 - \kappa)}}{\int Dx_1 e^{-\beta\ell(\sqrt{\Phi_0}x_0 + \sqrt{\Phi_1}x_1 - \kappa)}} \int Ds_1 D\lambda_1 \\
& \times \ln \int Ds_0 D\lambda_0 f^{(0)} \left(\frac{\Omega_0}{\sqrt{\Xi_0}} s_0 + \frac{\Omega_1}{\sqrt{\Xi_1}} s_1 + \sqrt{\Delta_0 - \frac{\Omega_0^2}{\Xi_0}} \lambda_0 + \sqrt{\Delta_1 - \frac{\Omega_1^2}{\Xi_1}} \lambda_1 \right. \\
& + \sum_{r=1}^2 \sqrt{J_{1r}} y_r + \sum_{r=1}^2 \sqrt{\tilde{J}_{1r}} z_r + \frac{\Psi_0}{\sqrt{\Phi_0}} x_0 + \frac{\Psi_1}{\sqrt{\Phi_1}} x_1, \\
& \left. \sqrt{\Xi_0} s_0 + \sqrt{\Xi_1} s_1 + \sum_{r=1}^2 \sqrt{J_{2r}} y_r + \sum_{r=1}^2 \sqrt{\tilde{J}_{2r}} z_r + \frac{\Sigma_0}{\sqrt{\Phi_0}} x_0 + \frac{\Sigma_1}{\sqrt{\Phi_1}} x_1, \sqrt{\Phi_0} x_0 + \sqrt{\Phi_1} x_1 \right)
\end{aligned} \tag{C.81}$$

where

$$f^{(0)}(M, N, O) \equiv e^{-\frac{M^2}{2} \ell''(O) - \frac{1}{2} \ell'(O) N} \tag{C.82}$$

We have 10 integrals to perform, but they can be reduced to 2 by implementing rotations and algebraic manipulations. In the following, we report the main steps necessary for performing such a reduction of the number of integrations, divided into different steps.

1st step We rotate over the variable y_1, y_2 and z_1, z_2 in order to remove the dependence on y_2 and z_2 on the second argument of $f^{(0)}$, obtaining

$$\begin{aligned}
\mathcal{G}_E = & \int \prod_{s=0}^1 Dx_s \prod_{s=1}^2 Dy_s Dz_s \frac{e^{-\beta\ell(\sqrt{\Phi_0}x_0 + \sqrt{\Phi_1}x_1 - \kappa)}}{\int Dx_1 e^{-\beta\ell(\sqrt{\Phi_0}x_0 + \sqrt{\Phi_1}x_1 - \kappa)}} \int Ds_1 D\lambda_1 \\
& \times \ln \int Ds_0 D\lambda_0 f^{(0)} \left(\frac{\Omega_0}{\sqrt{\Xi_0}} s_0 + \frac{\Omega_1}{\sqrt{\Xi_1}} s_1 + \sqrt{\Delta_0 - \frac{\Omega_0^2}{\Xi_0}} \lambda_0 + \sqrt{\Delta_1 - \frac{\Omega_1^2}{\Xi_1}} \lambda_1 \right. \\
& + \frac{J_{12}}{\sqrt{J_{22}}} y_1 + \sqrt{\frac{\det J}{J_{22}}} y_2 + \frac{\tilde{J}_{12}}{\sqrt{\tilde{J}_{22}}} z_1 + \sqrt{\frac{\det \tilde{J}}{\tilde{J}_{22}}} z_2 + \frac{\Psi_0}{\sqrt{\Phi_0}} x_0 + \frac{\Psi_1}{\sqrt{\Phi_1}} x_1 \\
& \left. , \sqrt{\Xi_0} s_0 + \sqrt{\Xi_1} s_1 + \sqrt{J_{22}} y_1 + \sqrt{\tilde{J}_{22}} z_1 + \frac{\Sigma_0}{\sqrt{\Phi_0}} x_0 + \frac{\Sigma_1}{\sqrt{\Phi_1}} x_1, \sqrt{\Phi_0} x_0 + \sqrt{\Phi_1} x_1 \right)
\end{aligned} \tag{C.83}$$

2nd step We perform a rotation over y_1 and z_1 and (new) y_1 and s_1 , so that we can remove both z_1 and s_1 from the second argument of $f^{(0)}$.

$$\begin{aligned} \mathcal{G}_E = & \int \prod_{s=0}^1 Dx_s \prod_{s=1}^2 Dy_s Dz_s \frac{e^{-\beta\ell(\sqrt{\Phi_0}x_0 + \sqrt{\Phi_1}x_1 - \kappa)}}{\int Dx_1 e^{-\beta\ell(\sqrt{\Phi_0}x_0 + \sqrt{\Phi_1}x_1 - \kappa)}} \int Ds_1 D\lambda_1 \\ & \times \ln \int Ds_0 D\lambda_0 f^{(0)} \left(\frac{\Omega_0}{\sqrt{\Xi_0}} s_0 + B s_1 + \sqrt{\Delta_0 - \frac{\Omega_0^2}{\Xi_0}} \lambda_0 + \sqrt{\Delta_1 - \frac{\Omega_1^2}{\Xi_1}} \lambda_1 \right. \\ & + \frac{\Omega_1 + J_{12} + \tilde{J}_{12}}{\sqrt{\Xi_1 + J_{22} + \tilde{J}_{22}}} y_1 + \sqrt{\frac{\det J}{J_{22}}} y_2 + A z_1 + \sqrt{\frac{\det \tilde{J}}{\tilde{J}_{22}}} z_2 + \frac{\Psi_0}{\sqrt{\Phi_0}} x_0 + \frac{\Psi_1}{\sqrt{\Phi_1}} x_1, \\ & \left. \sqrt{\Xi_0} s_0 + \sqrt{\Xi_1 + J_{22} + \tilde{J}_{22}} y_1 + \frac{\Sigma_0}{\sqrt{\Phi_0}} x_0 + \frac{\Sigma_1}{\sqrt{\Phi_1}} x_1, \sqrt{\Phi_0} x_0 + \sqrt{\Phi_1} x_1 \right) \end{aligned} \quad (\text{C.84})$$

where A and B expressions are defined in Equation (C.91).

3d step We can simplify over 4 integrals by doing a rotation over the variables $\lambda_1, z_1, s_1, y_2, z_2$. We call this new variable z

$$\begin{aligned} \mathcal{G}_E = & \int Dx_0 Dx_1 \frac{e^{-\beta\ell(\sqrt{\Phi_0}x_0 + \sqrt{\Phi_1}x_1 - \kappa)}}{\int Dx_1 e^{-\beta\ell(\sqrt{\Phi_0}x_0 + \sqrt{\Phi_1}x_1 - \kappa)}} \int Dy_1 Dz \\ & \times \ln \int Ds_0 D\lambda_0 f^{(0)} \left(\frac{\Omega_0}{\sqrt{\Xi_0}} s_0 + \sqrt{\Delta_0 - \frac{\Omega_0^2}{\Xi_0}} \lambda_0 + \frac{\Omega_1 + J_{12} + \tilde{J}_{12}}{\sqrt{\Xi_1 + J_{22} + \tilde{J}_{22}}} y_1 \right. \\ & + \sqrt{\eta} z + \frac{\Psi_0}{\sqrt{\Phi_0}} x_0 + \frac{\Psi_1}{\sqrt{\Phi_1}} x_1, \\ & \left. \sqrt{\Xi_0} s_0 + \sqrt{\Xi_1 + J_{22} + \tilde{J}_{22}} y_1 + \frac{\Sigma_0}{\sqrt{\Phi_0}} x_0 + \frac{\Sigma_1}{\sqrt{\Phi_1}} x_1, \sqrt{\Phi_0} x_0 + \sqrt{\Phi_1} x_1 \right), \end{aligned} \quad (\text{C.85})$$

with

$$\eta \equiv \Delta_1 - \frac{\Omega_1^2}{\Xi_1} + \frac{\det J}{J_{22}} + \frac{\det \tilde{J}}{\tilde{J}_{22}} + A^2 + B^2. \quad (\text{C.86})$$

We also conveniently perform a rotation over s_0 and λ_0 , obtaining

$$\begin{aligned} \mathcal{G}_E = & \int Dx_0 Dx_1 \frac{e^{-\beta\ell(\sqrt{\Phi_0}x_0 + \sqrt{\Phi_1}x_1 - \kappa)}}{\int Dx_1 e^{-\beta\ell(\sqrt{\Phi_0}x_0 + \sqrt{\Phi_1}x_1 - \kappa)}} \int Dy_1 Dz \\ & \times \ln \int Ds_0 D\lambda_0 f^{(0)} \left(\sqrt{\Delta_0} s_0 + \frac{\Omega_1 + J_{12} + \tilde{J}_{12}}{\sqrt{\Xi_1 + J_{22} + \tilde{J}_{22}}} y_1 + \sqrt{\eta} z + \frac{\Psi_0}{\sqrt{\Phi_0}} x_0 + \frac{\Psi_1}{\sqrt{\Phi_1}} x_1, \right. \\ & \left. \frac{\Omega_0}{\sqrt{\Delta_0}} s_0 + \sqrt{\Xi_0 - \frac{\Omega_0^2}{\Delta_0}} \lambda_0 + \sqrt{\Xi_1 + J_{22} + \tilde{J}_{22}} y_1 + \frac{\Sigma_0}{\sqrt{\Phi_0}} x_0 + \frac{\Sigma_1}{\sqrt{\Phi_1}} x_1, \sqrt{\Phi_0} x_0 + \sqrt{\Phi_1} x_1 \right). \end{aligned} \quad (\text{C.87})$$

4th step We shall now integrate over s_0 and λ_0 using the fact that $f^{(0)}(M, N, O) \equiv e^{-\frac{M^2}{2}\ell''(O) - \frac{N}{2}\ell'(O)}$. We have

$$\begin{aligned} \int Ds_0 D\lambda_0 e^{-\frac{1}{2}\ell''(O)(\sqrt{\Delta_0}s_0+C)^2 - \frac{\ell'(O)}{2}\left(\frac{\Omega_0}{\sqrt{\Delta_0}}s_0 + \sqrt{\Xi_0 - \frac{\Omega_0^2}{\Delta_0}}\lambda_0 + D\right)} \\ = e^{-D\frac{\ell'(O)}{2} + \frac{\ell'(O)^2}{8}\left(\Xi_0 - \frac{\Omega_0^2}{\Delta_0}\right) - \frac{C^2}{2}\ell''(O)} e^{\frac{\left(\frac{\Omega_0}{2\sqrt{\Delta_0}}\ell'(O) + \ell''(O)C\sqrt{\Delta_0}\right)^2}{2(1+\Delta_0\ell''(O))}} \frac{1}{\sqrt{1+\Delta_0\ell''(O)}}. \end{aligned} \quad (\text{C.88})$$

with $C \equiv \frac{\Omega_1 + J_{12} + \tilde{J}_{12}}{\sqrt{\Xi_1 + J_{22} + \tilde{J}_{22}}}y_1 + \sqrt{\eta}z + \frac{\Psi_0}{\sqrt{\Phi_0}}x_0 + \frac{\Psi_1}{\sqrt{\Phi_1}}x_1$, $D \equiv \sqrt{\Xi_1 + J_{22} + \tilde{J}_{22}}y_1 + \frac{\Sigma_0}{\sqrt{\Phi_0}}x_0 + \frac{\Sigma_1}{\sqrt{\Phi_1}}x_1$ and $O \equiv \sqrt{\Phi_0}x_0 + \sqrt{\Phi_1}x_1 - \kappa$.

5th step We shall now integrate over y_1 and z , obtaining the final expression for the energetic term in ϕ

$$\begin{aligned} \mathcal{G}_E = & -\frac{1}{2} \left\langle \ell'(O) \left(\frac{\Sigma_0}{\sqrt{\Phi_0}}x_0 + \frac{\Sigma_1}{\sqrt{\Phi_1}}x_1 \right) \right\rangle - \frac{1}{2} \left\langle \ln(1 + \Delta_0\ell''(O)) \right\rangle \\ & + \frac{1}{8} \left(\Xi_0 - \frac{\Omega_0^2}{\Delta_0} \right) \left\langle \ell'(O)^2 \right\rangle - \frac{1}{2} \left\langle \frac{\ell''(O)}{1 + \Delta_0\ell''(O)} \left[\left(\frac{\Psi_0}{\sqrt{\Phi_0}}x_0 + \frac{\Psi_1}{\sqrt{\Phi_1}}x_1 \right)^2 + \Gamma \right] \right\rangle \\ & + \frac{\Omega_0^2}{8\Delta_0} \left\langle \frac{\ell'(O)^2}{1 + \Delta_0\ell''(O)} \right\rangle + \frac{\Omega_0}{2} \left\langle \frac{\ell'(O)\ell''(O) \left(\frac{\Psi_0}{\sqrt{\Phi_0}}x_0 + \frac{\Psi_1}{\sqrt{\Phi_1}}x_1 \right)}{1 + \Delta_0\ell''(O)} \right\rangle, \end{aligned} \quad (\text{C.89})$$

where $O \equiv \sqrt{\Phi_0}x_0 + \sqrt{\Phi_1}x_1 - \kappa$ and

$$\langle \bullet \rangle \equiv \int Dx_1 \frac{\int Dx_0 e^{-\beta\ell(\sqrt{\Phi_0}x_0 + \sqrt{\Phi_1}x_1 - \kappa)} \bullet}{\int Dx_0 e^{-\beta\ell(\sqrt{\Phi_0}x_0 + \sqrt{\Phi_1}x_1 - \kappa)}}. \quad (\text{C.90})$$

Moreover

$$\Gamma \equiv \eta + \frac{(\Omega_1 + J_{12} + \tilde{J}_{12})^2}{\Xi_1 + J_{22} + \tilde{J}_{22}} \quad (\text{C.91a})$$

$$= \Delta_1 - \frac{\Omega_1^2}{\Xi_1} + \frac{\det J}{J_{22}} + \frac{\det \tilde{J}}{\tilde{J}_{22}} + A^2 + B^2 + \frac{(\Omega_1 + J_{12} + \tilde{J}_{12})^2}{\Xi_1 + J_{22} + \tilde{J}_{22}} \quad (\text{C.91b})$$

$$A = \frac{1}{\sqrt{J_{22} + \tilde{J}_{22}}} \left(\sqrt{\frac{J_{22}}{\tilde{J}_{22}}} \tilde{J}_{12} - \sqrt{\frac{\tilde{J}_{22}}{J_{22}}} J_{12} \right) \quad (\text{C.91c})$$

$$B = \frac{1}{\sqrt{\Xi_1 + J_{22} + \tilde{J}_{22}}} \left(\sqrt{\frac{J_{22} + \tilde{J}_{22}}{\Xi_1}} \Omega_1 - \sqrt{\frac{\Xi_1}{J_{22} + \tilde{J}_{22}}} (J_{12} + \tilde{J}_{12}) \right). \quad (\text{C.91d})$$

Note that the dependence on Ω_1 disappears

$$-\frac{\Omega_1^2}{\Xi_1} + B^2 + \frac{(\Omega_1 + J_{12} + \tilde{J}_{12})^2}{\Xi_1 + J_{22} + \tilde{J}_{22}} = \frac{(J_{12} + \tilde{J}_{12})^2}{J_{22} + \tilde{J}_{22}} \quad (\text{C.92})$$

so Γ can be simplified as follows

$$\begin{aligned}
\Gamma &\equiv \Delta_1 + \frac{\det J}{J_{22}} + \frac{\det \tilde{J}}{\tilde{J}_{22}} + \frac{1}{J_{22} + \tilde{J}_{22}} \left(\sqrt{\frac{J_{22}}{\tilde{J}_{22}}} \tilde{J}_{12} - \sqrt{\frac{\tilde{J}_{22}}{J_{22}}} J_{12} \right)^2 + \frac{(J_{12} + \tilde{J}_{12})^2}{J_{22} + \tilde{J}_{22}} \\
&= \Delta_1 + \frac{\det J}{J_{22}} + \frac{\det \tilde{J}}{\tilde{J}_{22}} + \frac{1}{J_{22} + \tilde{J}_{22}} \left(\frac{J_{22}}{\tilde{J}_{22}} \tilde{J}_{12}^2 + \frac{\tilde{J}_{22}}{J_{22}} J_{12}^2 - 2 \tilde{J}_{12} J_{12} \right) + \frac{(J_{12} + \tilde{J}_{12})^2}{J_{22} + \tilde{J}_{22}} \quad (\text{C.93}) \\
&= \Delta_1 + \frac{\det J}{J_{22}} + \frac{\det \tilde{J}}{\tilde{J}_{22}} + \frac{1}{J_{22} + \tilde{J}_{22}} \left(\left(\frac{J_{22}}{\tilde{J}_{22}} + 1 \right) \tilde{J}_{12}^2 + \left(\frac{\tilde{J}_{22}}{J_{22}} + 1 \right) J_{12}^2 \right) \\
&= \Delta_1 + \frac{J_{11} J_{22} - J_{12}^2}{J_{22}} + \frac{\tilde{J}_{11} \tilde{J}_{22} - \tilde{J}_{12}^2}{\tilde{J}_{22}} + \frac{\tilde{J}_{12}^2}{\tilde{J}_{22}} + \frac{J_{12}^2}{J_{22}},
\end{aligned}$$

obtaining finally

$$\Gamma \equiv \Delta_1 + J_{11} + \tilde{J}_{11} = \Delta_1 + \Delta_2 + \Delta_3 - \frac{\Psi_0^2}{\Phi_0} - \frac{\Psi_1^2}{\Phi_1}.$$

C.1.3 Effective order parameters

We report here the definitions of the effective order parameters for the variables

$$M \equiv \left(\sqrt{P - ph} + \frac{t}{\sqrt{q}} x + \frac{T - t}{\sqrt{Q - q}} \lambda + \sqrt{\gamma} z \right) g' \left(\sqrt{Q - q} \lambda + \sqrt{q} x \right) \quad (\text{C.94a})$$

$$N \equiv \left(\sqrt{P - ph} + \frac{t}{\sqrt{q}} x + \frac{T - t}{\sqrt{Q - q}} \lambda + \sqrt{\gamma} z \right)^2 g'' \left(\sqrt{Q - q} \lambda + \sqrt{q} x \right) \quad (\text{C.94b})$$

$$O \equiv g \left(\sqrt{Q - q} \lambda + \sqrt{q} x \right), \quad (\text{C.94c})$$

that are

$$\Delta_0 \equiv \langle \langle M^2 \rangle_h \rangle - \langle \langle M \rangle_h^2 \rangle; \quad (\text{C.95a})$$

$$\Xi_0 \equiv \langle \langle N^2 \rangle_h \rangle - \langle \langle N \rangle_h^2 \rangle; \quad (\text{C.95b})$$

$$\Omega_0 \equiv \langle \langle MN \rangle_h \rangle - \langle \langle M \rangle_h \langle N \rangle_h \rangle; \quad (\text{C.95c})$$

$$\Phi_0 \equiv \langle \langle O^2 \rangle_\lambda \rangle_x - \langle \langle O \rangle_\lambda^2 \rangle_x; \quad (\text{C.95d})$$

$$\Phi_1 \equiv \langle \langle O \rangle_\lambda^2 \rangle_x - \langle \langle O \rangle_\lambda \rangle_x^2; \quad (\text{C.95e})$$

$$\Psi_0 \equiv \langle \langle \langle M \rangle_{hz} O \rangle_\lambda \rangle_x - \langle \langle M \rangle_{hz\lambda} \langle O \rangle_\lambda \rangle_x; \quad (\text{C.95f})$$

$$\Psi_1 \equiv \langle \langle M \rangle_{hz\lambda} \langle O \rangle_\lambda \rangle_x - \langle M \rangle_{hz\lambda x} \langle O \rangle_{\lambda x}; \quad (\text{C.95g})$$

$$\Sigma_0 \equiv \langle \langle \langle N \rangle_{hz} O \rangle_\lambda \rangle_x - \langle \langle N \rangle_{hz\lambda} \langle O \rangle_\lambda \rangle_x; \quad (\text{C.95h})$$

$$\Sigma_1 \equiv \langle \langle N \rangle_{hz\lambda} \langle O \rangle_\lambda \rangle_x - \langle N \rangle_{hz\lambda x} \langle O \rangle_{\lambda x}; \quad (\text{C.95i})$$

$$\Delta_1 \equiv \langle \langle M \rangle_h^2 \rangle_{z,\lambda,x} - \langle \langle M \rangle_{h,z}^2 \rangle_{\lambda,x}; \quad (\text{C.95j})$$

$$\Delta_2 \equiv \langle \langle M \rangle_{h,z}^2 \rangle_{\lambda,x} - \langle \langle M \rangle_{h,z,\lambda}^2 \rangle_x; \quad (\text{C.95k})$$

$$\Delta_3 \equiv \langle \langle M \rangle_{h,z,\lambda}^2 \rangle_x - \langle M \rangle_{h,z,\lambda,x}^2. \quad (\text{C.95l})$$

Notice that if we have not specified the variable subscript in the notation $\langle \bullet \rangle$ the average is taken with respect to all the possible remaining Gaussian variables (z , λ , x). Note also that we only need the sum of Δ_1 , Δ_2 , Δ_3 which is

$$\tilde{\Delta} \equiv \Delta_1 + \Delta_2 + \Delta_3 = \langle \langle M \rangle_h^2 \rangle_{z,\lambda,x} - \langle M \rangle_{h,z,\lambda,x}^2. \quad (\text{C.96})$$

The final expression of the effective order parameter is reported in equation (C.97).

$$\Delta_0 \equiv (P - p) \int Dx g'(\sqrt{Q}x)^2 \quad (\text{C.97a})$$

$$\Xi_0 \equiv 2(P - p) \int Dx \left[P + p + \frac{2T^2}{Q}(x^2 - 1) \right] g''(\sqrt{Q}x)^2 \quad (\text{C.97b})$$

$$\Omega_0 = \frac{T(P - p)}{Q} \int Dx (x^2 - 1) g'(\sqrt{Q}x)^2 \quad (\text{C.97c})$$

$$\Psi_0 = \frac{T}{\sqrt{Q}} \int Dx x g'(\sqrt{Q}x) g(\sqrt{Q}x) \quad (\text{C.97d})$$

$$- \int Dx Dy f_1(x, y) g'(\sqrt{Q}x) g\left(\frac{q}{\sqrt{Q}}x + \sqrt{Q - \frac{q^2}{Q}}y\right) \quad (\text{C.97e})$$

$$\Psi_1 = -T \left[\int Dx g''(\sqrt{Q}x) \right] \left[\int Dx g(\sqrt{Q}x) \right] \quad (\text{C.97f})$$

$$+ \int Dx Dy f_1(x, y) g'(\sqrt{Q}x) g\left(\frac{q}{\sqrt{Q}}x + \sqrt{Q - \frac{q^2}{Q}}y\right) \quad (\text{C.97g})$$

$$\Sigma_0 = \int Dx \left[P - \frac{T^2}{Q} + \frac{T^2}{Q}x^2 \right] g''(\sqrt{Q}x) g(\sqrt{Q}x) \quad (\text{C.97h})$$

$$- \int Dx Dy f_2(x, y) g''(\sqrt{Q}x) g\left(\frac{q}{\sqrt{Q}}x + \sqrt{Q - \frac{q^2}{Q}}y\right) \quad (\text{C.97i})$$

$$\Sigma_1 = \int Dx Dy f_2(x, y) g''(\sqrt{Q}x) g\left(\frac{q}{\sqrt{Q}}x + \sqrt{Q - \frac{q^2}{Q}}y\right) \quad (\text{C.97j})$$

$$- \left[\int Dx g(\sqrt{Q}x) \right] \left[\int Dx \left(P - \frac{T^2}{Q} + \frac{T^2}{Q}x^2 \right) g''(\sqrt{Q}x) \right]$$

$$\tilde{\Delta} = \int Dx \left(p - \frac{T^2}{Q} + \frac{T^2}{Q}x^2 \right) g'(\sqrt{Q}x)^2 - \left(T \int Dx g''(\sqrt{Q}x) \right)^2 \quad (\text{C.97k})$$

where the functions f_1 and f_2 are defined as:

$$f_1(x, y) = \frac{T}{\sqrt{Q}}x - \frac{qT - tQ}{\sqrt{Q(Q^2 - q^2)}}y$$

$$f_2(x, y) = P - p + \gamma + \frac{(qT - tQ)^2}{q(Q^2 - q^2)} + \left(\frac{T}{\sqrt{Q}}x - \frac{qT - tQ}{\sqrt{Q(Q^2 - q^2)}}y \right)^2.$$

Negative perceptron case

We now consider the case of the negative perceptron model. This case can be directly reconstructed from the previous computation by simply choosing the activation function to be an identity function, i.e., $g(x) = -x$. In this case, the effective order parameters reduce to

$$\Delta_0 = P - p \quad \tilde{\Delta} = p \quad (\text{C.98a})$$

$$\Psi_0 = T - t \quad \Psi_1 = t \quad (\text{C.98b})$$

$$\Phi_0 = Q - q \quad \Phi_1 = q \quad (\text{C.98c})$$

$$\Xi_0 = \Omega_0 = \Sigma_0 = \Sigma_1 = 0 \quad (\text{C.98d})$$

$$\Gamma = \tilde{\Delta} - \frac{\Psi_0^2}{\Phi_0} - \frac{\Psi_1^2}{\Phi_1} = \gamma = p - \frac{t^2}{q} - \frac{(T - t)^2}{Q - q}. \quad (\text{C.98e})$$

The free entropy function ϕ can then be written as

$$\phi = \mathcal{G}_I + \mathcal{G}_S + \alpha \mathcal{G}_E \quad (\text{C.99a})$$

$$\mathcal{G}_I = \frac{1}{2} P \hat{P} + \frac{1}{2} \hat{p} p + \hat{t} t - \hat{T} T + \frac{z - \lambda}{2} P \quad (\text{C.99b})$$

$$\mathcal{G}_S = \frac{1}{2} \ln \frac{1}{\hat{P} + \hat{p}} + \frac{1}{\hat{P} + \hat{p}} \left[\frac{\hat{p}}{2} + \frac{Q}{2} (\hat{T} - \hat{t})^2 + \hat{t} (\hat{T} - \hat{t}) (Q - q) \right] \quad (\text{C.99c})$$

$$\mathcal{G}_E = -\frac{1}{2} \langle \ln \eta(x_0, x_1) \rangle \quad (\text{C.99d})$$

$$- \frac{1}{2} \left\langle \frac{\ell''(\sqrt{Q - q} x_0 + \sqrt{q} x_1 - \kappa)}{\eta(x_0, x_1)} \left[\left(\frac{t}{\sqrt{q}} x_1 + \frac{T - t}{\sqrt{Q - q}} x_0 \right)^2 + \gamma \right] \right\rangle \quad (\text{C.99e})$$

$$\eta(x_0, x_1) \equiv 1 + (P - p) \ell''(\sqrt{Q - q} x_0 + \sqrt{q} x_1 - \kappa), \quad (\text{C.99f})$$

where

$$\langle \bullet \rangle \equiv \int D x_1 \frac{\int D x_0 e^{-\beta \ell(\sqrt{Q - q} x_0 + \sqrt{q} x_1 - \kappa)} \bullet}{\int D x_0 e^{-\beta \ell(\sqrt{Q - q} x_0 + \sqrt{q} x_1 - \kappa)}}. \quad (\text{C.100})$$

Overlap Parameters in the $\beta \rightarrow \infty$ case

The overlap parameters in the $\beta \rightarrow \infty$ case are the following:

$$\Delta_0 \equiv (P - p) \int Dx g'(\sqrt{Q}x)^2 \quad (\text{C.101a})$$

$$\Xi_0 \equiv 2(P - p) \int Dx \left[P + p + \frac{2T^2}{Q}(x^2 - 1) \right] g''(\sqrt{Q}x)^2 \quad (\text{C.101b})$$

$$\Omega_0 \equiv \frac{T(P - p)}{Q} \int Dx g'(\sqrt{Q}x)^2 \quad (\text{C.101c})$$

$$\Phi_1 \equiv \mathcal{K}_Q(Q) - \mathcal{K}_Q(0) \quad (\text{C.101d})$$

$$\delta\Psi_0 = \frac{\delta q}{Q} \int Dx \left[Q\delta t + \frac{T}{2}(x^2 - 1) \right] g'(\sqrt{Q}x)^2 \quad (\text{C.101e})$$

$$\Psi_1 = \frac{T}{\sqrt{Q}} \int Dx x g'(\sqrt{Q}x) g(\sqrt{Q}x) - \frac{T}{\sqrt{Q}} \left[\int Dx x g'(\sqrt{Q}x) \right] \left[\int Dx g(\sqrt{Q}x) \right] \quad (\text{C.101f})$$

$$\delta\Sigma_0 = 2\delta q\delta tT \int Dx g''(\sqrt{Q}x)^2 + \delta q \int Dx \left(P + 2\delta tT + \frac{T^2}{Q}(x^2 - 1) \right) g'''(\sqrt{Q}x) g'(\sqrt{Q}x) \quad (\text{C.101g})$$

$$\Sigma_1 = \int Dx \left[P - \frac{T^2}{Q} + \frac{T^2}{Q}x^2 \right] g''(\sqrt{Q}x) g(\sqrt{Q}x) \quad (\text{C.101h})$$

$$- \left[\int Dx g(\sqrt{Q}x) \right] \left[\int Dx \left(P - \frac{T^2}{Q} + \frac{T^2}{Q}x^2 \right) g''(\sqrt{Q}x) \right] \quad (\text{C.101i})$$

$$\tilde{\Delta} = \int Dx \left(p - \frac{T^2}{Q} + \frac{T^2}{Q}x^2 \right) g'(\sqrt{Q}x)^2 - \left(T \int Dx g''(\sqrt{Q}x) \right)^2 \quad (\text{C.101j})$$

$$\Gamma \equiv \tilde{\Delta} - \frac{\Psi_1^2}{\Phi_1} \quad (\text{C.101k})$$

Bibliography

- Emmanuel Abbe, Shuangping Li, and Allan Sly. Binary perceptron: efficient algorithms can find solutions in a rare well-connected cluster. *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 860–873, 2022.
- Beatrice Achilli, Enrico Ventura, Gianluigi Silvestri, Bao Pham, Gabriel Raya, Dmitry Krotov, Carlo Lucibello, and Luca Ambrogioni. Losing dimensions: Geometric memorization in generative diffusion, 2024. URL <https://arxiv.org/abs/2410.08727>.
- Beatrice Achilli, Luca Ambrogioni, Carlo Lucibello, Marc Mézard, and Enrico Ventura. Memorization and generalization in generative diffusion under the manifold hypothesis. *JSTAT*, 2025. URL <https://arxiv.org/abs/2502.09578>.
- David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147–169, 1985. ISSN 0364-0213. doi: [https://doi.org/10.1016/S0364-0213\(85\)80012-4](https://doi.org/10.1016/S0364-0213(85)80012-4). URL <https://www.sciencedirect.com/science/article/pii/S0364021385800124>.
- Ahmed El Alaoui and David Gamarnik. Hardness of sampling solutions from the symmetric binary perceptron. *arXiv preprint arXiv:2407.16627*, 2024.
- Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions, 2023. URL <https://arxiv.org/abs/2303.08797>.
- Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2): 251–276, 1998.
- Brandon Annesi, Enrico Malatesta, and Francesco Zamponi. Exact full-rsb sat/unsat transition in infinitely wide two-layer neural networks. *SciPost Physics*, 18, 04 2025. doi: 10.21468/SciPostPhys.18.4.118.
- Brandon Livio Annesi, Clarissa Lauditi, Carlo Lucibello, Enrico M Malatesta, Gabriele Perugini, Fabrizio Pittorino, and Luca Saglietti. Star-shaped space of solutions of the spherical negative perceptron. *Physical Review Letters*, 131(22):227301, 2023.
- Fabrizio Antenucci, Silvio Franz, Pierfrancesco Urbani, and Lenka Zdeborová. Glassy nature of the hard phase in inference problems. *Phys. Rev. X*, 9:011020, Jan 2019a. doi: 10.1103/PhysRevX.9.011020. URL <https://link.aps.org/doi/10.1103/PhysRevX.9.011020>.
- Fabrizio Antenucci, Florent Krzakala, Pierfrancesco Urbani, and Lenka Zdeborová. Approximate survey propagation for statistical inference. *J. Stat. Mech.*, 2019(0341):023401, 2019b.

- Luca Arnaboldi, Ludovic Stephan, Florent Krzakala, and Bruno Loureiro. From high-dimensional & mean-field dynamics to dimensionless odes: A unifying approach to sgd in two-layers networks. *Neural Information Processing Systems*, 02 2023. doi: 10.48550/arXiv.2302.05882.
- Kiana Asgari, Andrea Montanari, and Basil Saeed. Local minima of the empirical risk in high dimension: General theorems and convex examples, 2025. URL <https://arxiv.org/abs/2502.01953>.
- Antonio Auffinger, Gerard Ben Arous, and Jiri Cerny. Random matrices and complexity of spin glasses. *Communications on Pure and Applied Mathematics*, 66(2), 2012. ISSN 0010-3640.
- Antonio Auffinger, Gérard Ben Arous, and Jiří Černý. Random matrices and complexity of spin glasses. *Communications on Pure and Applied Mathematics*, 66(2):165–201, 2013. doi: <https://doi.org/10.1002/cpa.21422>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.21422>.
- Jinho Baik, Gérard Ben Arous, and Sandrine Péché. Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices. *The Annals of Probability*, 33(5): 1643 – 1697, 2005. doi: [10.1214/009117905000000233](https://doi.org/10.1214/009117905000000233). URL <https://doi.org/10.1214/009117905000000233>.
- Carlo Baldassi and Alfredo Braunstein. A max-sum algorithm for training discrete neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2015(8):P08008, aug 2015. doi: [10.1088/1742-5468/2015/08/P08008](https://dx.doi.org/10.1088/1742-5468/2015/08/P08008). URL <https://dx.doi.org/10.1088/1742-5468/2015/08/P08008>.
- Carlo Baldassi, Alessandro Ingrosso, Carlo Lucibello, Luca Saglietti, and Riccardo Zecchina. Subdominant dense clusters allow for simple learning and high computational performance in neural networks with discrete synapses. *Phys. Rev. Lett.*, 115:128101, Sep 2015. doi: [10.1103/PhysRevLett.115.128101](https://link.aps.org/doi/10.1103/PhysRevLett.115.128101). URL <https://link.aps.org/doi/10.1103/PhysRevLett.115.128101>.
- Carlo Baldassi, Christian Borgs, Jennifer T. Chayes, Alessandro Ingrosso, Carlo Lucibello, Luca Saglietti, and Riccardo Zecchina. Unreasonable effectiveness of learning neural networks: From accessible states and robust ensembles to basic algorithmic schemes. *Proceedings of the National Academy of Sciences*, 113(48):E7655–E7662, 2016a. doi: [10.1073/pnas.1608103113](https://www.pnas.org/doi/abs/10.1073/pnas.1608103113). URL <https://www.pnas.org/doi/abs/10.1073/pnas.1608103113>.
- Carlo Baldassi, Alessandro Ingrosso, Carlo Lucibello, Luca Saglietti, and Riccardo Zecchina. Local entropy as a measure for sampling solutions in constraint satisfaction problems. *Journal of Statistical Mechanics: Theory and Experiment*, 2016(2):023301, 2016b.
- Carlo Baldassi, Fabrizio Pittorino, and Riccardo Zecchina. Shaping the learning landscape in neural networks around wide flat minima. *Proceedings of the National Academy of Sciences*, 117(1):161–170, 2020.
- Carlo Baldassi, Enrico M Malatesta, Gabriele Perugini, and Riccardo Zecchina. Typical and atypical solutions in nonconvex neural networks with discrete and continuous weights. *Physical Review E*, 108(2):024310, 2023.

- Afonso S. Bandeira, Jameson Cahill, Dustin G. Mixon, and Aaron A. Nelson. Saving phase: Injectivity and stability for phase retrieval. *Applied and Computational Harmonic Analysis*, 37(1):106–125, 2014. ISSN 1063-5203. doi: <https://doi.org/10.1016/j.acha.2013.10.002>. URL <https://www.sciencedirect.com/science/article/pii/S1063520313000936>.
- Damien Barbier, Carlo Lucibello, Luca Saglietti, Florent Krzakala, and Lenka Zdeborová. The phase diagram of compressed sensing with ℓ_0 -norm regularization. *Journal Of Statistical Mechanics-theory And Experiment*, 2024. URL <https://arxiv.org/abs/2408.08319>.
- Jean Barbier, Florent Krzakala, Nicolas Macris, Léo Miolane, and Lenka Zdeborová. Optimal errors and phase transitions in high-dimensional generalized linear models. *Proceedings of the National Academy of Sciences*, 116(12):5451–5460, 2019.
- E. Barkai, D. Hansel, and I. Kanter. Statistical mechanics of a multilayered neural network. *Phys. Rev. Lett.*, 65:2312–2315, Oct 1990. doi: 10.1103/PhysRevLett.65.2312. URL <https://link.aps.org/doi/10.1103/PhysRevLett.65.2312>.
- E. Barkai, D. Hansel, and H. Sompolinsky. Broken symmetries in multilayered perceptrons. *Phys. Rev. A*, 45:4146–4161, Mar 1992. doi: 10.1103/PhysRevA.45.4146. URL <https://link.aps.org/doi/10.1103/PhysRevA.45.4146>.
- Mohsen Bayati and Andrea Montanari. The dynamics of message passing on dense graphs, with applications to compressed sensing. *IEEE Transactions on Information Theory*, 57(2):764–785, 2011.
- Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116:201903070, 07 2019. doi: 10.1073/pnas.1903070116.
- Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):192–236, 1974. doi: 10.1111/j.2517-6161.1974.tb00999.x.
- Giulio Biroli and Marc Mézard. Generative diffusion in very large dimensions. *Journal of Statistical Mechanics: Theory and Experiment*, 2023(9):093402, September 2023. ISSN 1742-5468. doi: 10.1088/1742-5468/acf8ba. URL <http://dx.doi.org/10.1088/1742-5468/acf8ba>.
- Giulio Biroli, Tony Bonnaire, Valentin de Bortoli, and Marc Mézard. Dynamical regimes of diffusion models. *Nature Communications*, 15(1), November 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-54281-3. URL <http://dx.doi.org/10.1038/s41467-024-54281-3>.
- Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- Tony Bonnaire, Giulio Biroli, and Chiara Cammarota. From zero to hero: How local curvature at artless initial conditions leads away from bad minima, 2024. URL <https://arxiv.org/abs/2403.02418>.
- Léon Bottou. Large-scale machine learning with stochastic gradient descent. *Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT)*, pages 177–186, 2010.

- Alfredo Braunstein and Riccardo Zecchina. Learning by message passing in networks of discrete synapses. *Physical review letters*, 96(3):030201, 2006.
- S. Brooks, A. Gelman, G. Jones, and X.L. Meng. *Handbook of Markov Chain Monte Carlo*. Chapman & Hall/CRC Handbooks of Modern Statistical Methods. Taylor & Francis, 2011. ISBN 978-1-4200-7941-8. URL <https://books.google.com/books?id=Pon2mAEACAAJ>.
- Nicola Bulso and Yasser Roudi. Restricted boltzmann machines as models of interacting variables, 2021. URL <https://arxiv.org/abs/2103.15917>.
- Emmanuel J. Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.
- Emmanuel J. Candes, Xiaodong Li, and Mahdi Soltanolkotabi. Phase retrieval via wirtinger flow: Theory and algorithms. *IEEE Transactions on Information Theory*, 61(4):1985–2007, April 2015. ISSN 1557-9654. doi: 10.1109/tit.2015.2399924. URL <http://dx.doi.org/10.1109/TIT.2015.2399924>.
- Miguel Á. Carreira-Perpiñán and Geoffrey Hinton. On contrastive divergence learning. In Robert G. Cowell and Zoubin Ghahramani, editors, *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, volume R5 of *Proceedings of Machine Learning Research*, pages 33–40. PMLR, 06–08 Jan 2005. URL <https://proceedings.mlr.press/r5/carreira-perpignan05a.html>. Reissued by PMLR on 30 March 2021.
- Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. *Semi-Supervised Learning*. MIT press, 2006.
- Patrick Charbonneau, Enzo Marinari, Giorgio Parisi, Federico Ricci-terseghi, Gabriele Sicuro, Francesco Zamponi, and Marc Mezard. *Spin glass theory and far beyond: replica symmetry breaking after 40 years*. World Scientific, 2023.
- Sanjay Chawla, Preslav Nakov, Ahmed Ali, Wendy Hall, Issa Khalil, Xiaosong Ma, Husrev Taha Sencar, Ingmar Weber, Michael Wooldridge, and Ting Yu. Ten years after imagenet: A 360 perspective on ai, 2022. URL <https://arxiv.org/abs/2210.01797>.
- Lénaïc Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2019. Curran Associates Inc.
- Anna Choromanska, Mikael Henaff, Michael Mathieu, Gerard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. *Artificial Intelligence and Statistics*, pages 192–204, 2015.
- C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968. doi: 10.1109/TIT.1968.1054142.
- Simona Cocco and Rémi Monasson. Adaptive cluster expansion for inferring boltzmann machines with noisy data. *Physical Review Letters*, 106(9):090601, 2011. doi: 10.1103/PhysRevLett.106.090601.

- Patrick L Combettes and Vianney R Wajs. Proximal splitting methods in signal processing. *Applied and Computational Harmonic Analysis*, 22(3):269–300, 2007.
- Gregory F. Cooper. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial Intelligence*, 42(2):393–405, 1990. ISSN 0004-3702. doi: [https://doi.org/10.1016/0004-3702\(90\)90060-D](https://doi.org/10.1016/0004-3702(90)90060-D). URL <https://www.sciencedirect.com/science/article/pii/000437029090060D>.
- Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3): 273–297, September 1995. ISSN 0885-6125. doi: 10.1023/A:1022627411411. URL <https://doi.org/10.1023/A:1022627411411>.
- Guido Cossu, Luigi Del Debbio, Tommaso Giani, Ava Khamseh, and Michael Wilson. Machine learning determination of dynamical parameters: The ising model case. *Phys. Rev. B*, 100:064304, Aug 2019. doi: 10.1103/PhysRevB.100.064304. URL <https://link.aps.org/doi/10.1103/PhysRevB.100.064304>.
- Thomas M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, EC-14(3):326–334, 1965.
- Hugo Cui, Luca Saglietti, and L. Zdeborová. Large deviations for the perceptron model and consequences for active learning. *IOP science Machine Learning and Technologies*, abs/1912.03927, 2021.
- Ingrid Daubechies, Michel Defrise, and Christine Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraints. *Communications on Pure and Applied Mathematics*, 57, 11 2004. doi: 10.1002/cpa.20042.
- Yann N. Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS’14, page 2933–2941, Cambridge, MA, USA, 2014. MIT Press.
- Aurelien Decelle, Cyril Furtlehner, and Beatriz Seoane. Equilibrium and non-equilibrium regimes in the learning of restricted boltzmann machines. *Proceedings of the 35th International Conference on Neural Information Processing Systems*, 2021.
- Aurélien Decelle and Cyril Furtlehner. Restricted boltzmann machine: Recent advances and mean-field theory. *Chinese Physics B*, 30(4):040202, 2021. doi: 10.1088/1674-1056/abd160. URL https://cpb.iphy.ac.cn/EN/abstract/article_123418.shtml.
- Aurélien Decelle, Cyril Furtlehner, Alfonso De Jesús Navas Gómez, and Beatriz Seoane. Inferring effective couplings with restricted Boltzmann machines. *SciPost Phys.*, 16: 095, 2024. doi: 10.21468/SciPostPhys.16.4.095. URL <https://scipost.org/10.21468/SciPostPhys.16.4.095>.
- J.E. Dennis and R.B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, 1983.

- Loc Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *Proceedings of the 34th International Conference on Machine Learning*, ICML'17, page 1019–1028, 2017.
- David Donoho and Jared Tanner. Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1906):4273–4293, 2009.
- David L. Donoho, Arian Maleki, and Andrea Montanari. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences*, 106(45):18914–18919, 2009. doi: 10.1073/pnas.0909892106. URL <https://www.pnas.org/doi/abs/10.1073/pnas.0909892106>.
- Victor Dotsenko, Silvio Franz, and Marc Mézard. Partial annealing and overfrustration in disordered systems. *J. Phys. A: Math. Gen.*, 27(2351):2351–2365, 1994.
- Iain Dunning, Joey Huchette, and Miles Lubin. JuMP: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017. doi: 10.1137/15M1020575.
- Ahmed El Alaoui and Andrea Montanari. An information-theoretic view of stochastic localization. *IEEE Transactions on Information Theory*, 68(11):7423–7426, 2022.
- Ahmed El Alaoui, Andrea Montanari, and Mark Sellke. Sampling from the sherrington-kirkpatrick gibbs measure via algorithmic stochastic localization. *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 323–334, 2022.
- Ronen Eldan. Thin shell implies spectral gap up to polylog via a stochastic localization scheme. *Geometric and Functional Analysis*, 23(2):532–569, 2013.
- Richard S. Ellis. *Entropy, Large Deviations, and Statistical Mechanics*. Springer, New York, 1985.
- A. Engel and C. Van den Broeck. *Statistical Mechanics of Learning*. Cambridge University Press, 2001.
- David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Pearson, Upper Saddle River, NJ, 2 edition, 2011.
- Stanislav Fort and Surya Ganguli. Emergent properties of the local geometry of neural loss landscapes. *ArXiv*, abs/1910.05929, 2019. URL <https://api.semanticscholar.org/CorpusID:204509527>.
- Silvio Franz and Giorgio Parisi. The simplest model of jamming. *Journal of Physics A: Mathematical and Theoretical*, 49(14):145001, 2016.
- Silvio Franz, Giorgio Parisi, Pierfrancesco Urbani, and Francesco Zamponi. Universal Spectrum of Normal Modes in Low-Temperature Glasses: an Exact Solution. *Proceedings of the National Academy of Sciences of the United States of America*, 112:14539, 2015. URL <https://hal.science/hal-01237684>. 6 pages, 3 figures, submitted to PNAS.

- Silvio Franz, Giorgio Parisi, Maksim Sevelev, Pierfrancesco Urbani, and Francesco Zamponi. Universality of the SAT-UNSAT (jamming) threshold in non-convex continuous constraint satisfaction problems. *SciPost Phys.*, 2:019, 2017. doi: 10.21468/SciPostPhys.2.3.019. URL <https://scipost.org/10.21468/SciPostPhys.2.3.019>.
- Marylou Gabri  , Eric W. Tramel, and Florent Krzakala. Training restricted boltzmann machines via the thouless-anderson-palmer free energy. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*, NIPS’15, page 640–648, Cambridge, MA, USA, 2015. MIT Press.
- Marylou Gabri  , Surya Ganguli, Carlo Lucibello, and Riccardo Zecchina. Neural networks: From the perceptron to deep nets. In *Spin Glass Theory and Far Beyond: Replica Symmetry Breaking After 40 Years*, pages 477–497. World Scientific, 2023.
- David Gamarnik. The overlap gap property: A topological barrier to optimizing over random structures. *Proceedings of the National Academy of Sciences*, 118(41):e2108492118, 2021. doi: 10.1073/pnas.2108492118. URL <https://www.pnas.org/doi/abs/10.1073/pnas.2108492118>.
- David Gamarnik, Eren C. Kızıldağ, Will Perkins, and Changji Xu. Algorithms and barriers in the symmetric binary perceptron model. *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 576–587, 2022. doi: 10.1109/FOCS54457.2022.00061.
- Elizabeth Gardner. The space of interactions in neural network models. *Journal of Physics A: Mathematical and General*, 21(1):257, 1988.
- Elizabeth Gardner and Bernard Derrida. Optimal storage properties of neural network models. *J. Phys. A: Math. Gen.*, 21(271):271–284, 1988.
- Stuart Geman, Elie Bienenstock, and Ren   Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992. doi: 10.1162/neco.1992.4.1.1.
- Davide Ghio, Yatin Dandi, Florent Krzakala, and Lenka Zdeborov  . Sampling with flows, diffusion, and autoregressive neural networks from a spin-glass perspective. *Proceedings of the National Academy of Sciences*, 121(27):e2311810121, 2024.
- Sebastian Goldt, Madhu S Advani, Andrew M Saxe, Florent Krzakala, and Lenka Zdeborov  . Dynamics of stochastic gradient descent for two-layer neural networks in the teacher–student setup*. *Journal of Statistical Mechanics: Theory and Experiment*, 2020 (12):124010, dec 2020. doi: 10.1088/1742-5468/abc61e. URL <https://dx.doi.org/10.1088/1742-5468/abc61e>.
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL <http://www.deeplearningbook.org>. Book in preparation for MIT Press.
- Priya Goyal, Piotr Doll  r, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *CoRR*, 2018. URL <https://arxiv.org/abs/1706.02677>.

- Louis Grenioux, Maxence Noble, Marylou Gabri  , and Alain Olivier Durmus. Stochastic localization via iterative posterior sampling, 2024. URL <https://arxiv.org/abs/2402.10758>.
- Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition, 2009. URL <http://www-stat.stanford.edu/~tibs/ElemStatLearn/>.
- Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002. doi: 10.1162/089976602760128018.
- Geoffrey E. Hinton. A practical guide to training restricted boltzmann machines. *Neural Networks: Tricks of the Trade: Second Edition*, pages 599–619, 2012. doi: 10.1007/978-3-642-35289-8_32. URL https://doi.org/10.1007/978-3-642-35289-8_32.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- Haiping Huang and Yoshiyuki Kabashima. Origin of the computational hardness for learning with binary synapses. *Physical Review E*, 90(5):052813, 2014.
- Jiaoyang Huang and Horng-Tzer Yau. Dynamics of deep neural networks and neural tangent hierarchy. In Hal Daum   III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4542–4551. PMLR, 13–18 Jul 2020. URL <https://proceedings.mlr.press/v119/huang201.html>.
- Yukito Iba. The nishimori line and bayesian statistics. *Journal of Physics A: Mathematical and General*, 32(21):3875–3888, January 1999. ISSN 1361-6447. doi: 10.1088/0305-4470/32/21/302. URL <http://dx.doi.org/10.1088/0305-4470/32/21/302>.
- Arthur Jacot, Franck Gabriel, and Cl  ment Hongler. Neural tangent kernel: convergence and generalization in neural networks. In *NIPS 2018*, 2018. doi: 10.1145/3406325.3465355. URL <https://doi.org/10.1145/3406325.3465355>.
- A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, September 1999. ISSN 0360-0300. doi: 10.1145/331499.331504. URL <https://doi.org/10.1145/331499.331504>.
- Stanis  aw Jastrz  bski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Y. Bengio, and Amos Storkey. Three factors influencing minima in sgd. In *ICANN2018*, 11 2017. doi: 10.48550/arXiv.1711.04623.
- Abhijith Jayakumar, Andrey Lokhov, Sidhant Misra, and Marc Vuffray. Discrete distributions are learnable from metastable samples, 10 2024.
- Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4037–4058, 2020.
- Jason Johnson, Praneeth Netrapalli, and Michael Chertkov. Learning planar ising models. *Journal of Machine Learning Research*, 17, 11 2010.

- Iain M. Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *The Annals of Statistics*, 29(2):295 – 327, 2001. doi: 10.1214/aos/1009210544. URL <https://doi.org/10.1214/aos/1009210544>.
- Ian T Jolliffe. *Principal Component Analysis*. Springer, 2002.
- Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *ICLR*, 2017.
- D.P. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015.
- Scott Kirkpatrick, C. Daniel Gelatt, and Mario P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- Werner Krauth and Marc Mézard. Storage capacity of memory networks with binary couplings. *Journal de Physique*, 50(20):3057–3066, 1989.
- Florent Krzakala, Federico Ricci-Tersenghi, Lenka Zdeborova, Riccardo Zecchina, Eric W. Tramel, and Leticia F. Cugliandolo. *Statistical Physics, Optimization, Inference, and Message-Passing Algorithms: Lecture Notes of the Les Houches School of Physics: Special Issue, October 2013*. Oxford University Press, December 2015. ISBN 978-0-19-874373-6. doi: 10.1093/acprof:oso/9780198743736.001.0001. URL <https://doi.org/10.1093/acprof:oso/9780198743736.001.0001>.
- L. D. Landau and E. M. Lifshitz. *Statistical Physics, Part 1*, volume 5 of *Course of Theoretical Physics*. Pergamon Press, Oxford, 3 edition, 1980. ISBN 0-08-028075-4.
- Yann Lecun, Leon Bottou, Genevieve Orr, and Klaus-Robert Müller. *Efficient BackProp*. Springer, 03 2002. ISBN 978-3-540-49430-0.
- Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *NIPS’18: Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018. URL <https://arxiv.org/abs/1712.09913>.
- Xinyan Li, Qilong Gu, Yingxue Zhou, Tiancong Chen, and Arindam Banerjee. Hessian based analysis of sgd for deep nets: Dynamics and generalization. *SIAM Publications*, 2019. URL <https://arxiv.org/abs/1907.10732>.
- Zhenyu Liao and Michael W. Mahoney. Hessian eigenspectra of more realistic nonlinear models. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS ’21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.
- Giacomo Livan, Marcel Novaes, and Pierpaolo Vivo. *Introduction to Random Matrices - Theory and Practice*, volume 26. Springer, 12 2017. ISBN 978-3-319-70883-6. doi: 10.1007/978-3-319-70885-0.

- Andrey Y. Lokhov, Marc Vuffray, Sidhant Misra, and Michael Chertkov. Optimal structure and parameter learning of ising models. *Science Advances*, 4(3):e1700791, 2018. doi: 10.1126/sciadv.1700791. URL <https://www.science.org/doi/abs/10.1126/sciadv.1700791>.
- Junjie Ma, Ji Xu, and Arian Maleki. Approximate message passing for amplitude based optimization. In *International Conference on Machine Learning*, pages 3365–3374. PMLR, 2018.
- Antoine Maillard, Bruno Loureiro, Florent Krzakala, and Lenka Zdeborová. Phase retrieval in high dimensions: Statistical and computational phase transitions, 2020.
- Stefano Sarao Mannelli, Giulio Biroli, Chiara Cammarota, Florent Krzakala, Pierfrancesco Urbani, and Lenka Zdeborová. Complex dynamics in simple neural networks: Understanding gradient flow in phase retrieval. *NeurIPS*, 2020. URL <https://arxiv.org/abs/2006.06997>.
- Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, 1999.
- Charles H. Martin and Michael W. Mahoney. Implicit self-regularization in deep neural networks: evidence from random matrix theory and implications for learning. *J. Mach. Learn. Res.*, 22(1), January 2021. ISSN 1532-4435.
- V A Marčenko and L A Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457, apr 1967. doi: 10.1070/SM1967v001n04ABEH001994. URL <https://dx.doi.org/10.1070/SM1967v001n04ABEH001994>.
- Peter McCullagh and John A. Nelder. *Generalized Linear Models*. Chapman and Hall/CRC, 2nd edition, 1989.
- Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, 1943. doi: 10.1007/bf02478259.
- Song Mei and Andrea Montanari. Generalization error of random features regression: Precise asymptotics and double descent curve. *Communications on Pure and Applied Mathematics*, 2021. later published in Proceedings of the National Academy of Sciences, 117(56):35136–35146, 2020.
- Marc Mézard and A. Montanari. *Information, Physics and computation*. Oxford University Press, 2002.
- Marc Mezard and Andrea Montanari. *Information, Physics, and Computation*. Oxford University Press, Inc., USA, 2009. ISBN 019857083X.
- Marc Mézard, Giorgio Parisi, Nicolas Sourlas, Gérard Toulouse, and Miguel A. Virasoro. Replica symmetry breaking and the nature of the spin glass phase. *J. Phys. France*, 45(5):843–854, 1984.

- Marc Mézard, Giorgio Parisi, and Miguel Angel Virasoro. *Spin glass theory and beyond: An Introduction to the Replica Method and Its Applications*, volume 9. World Scientific Publishing Company, 1987.
- Francesca Mignacco, Florent Krzakala, Pierfrancesco Urbani, and Lenka Zdeborová. Dynamical mean-field theory for stochastic gradient descent in gaussian mixture classification. *Advances in Neural Information Processing Systems*, 33:9540–9550, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6c81c83c4bd0b58850495f603ab45a93-Paper.pdf.
- Francesca Mignacco, Pierfrancesco Urbani, and Lenka Zdeborová. Stochasticity helps to navigate rough landscapes: comparing gradient-descent-based algorithms in the phase retrieval problem. *Machine Learning: Science and Technology*, 2(3):035029, jul 2021. doi: 10.1088/2632-2153/ac0615. URL <https://dx.doi.org/10.1088/2632-2153/ac0615>.
- R. P. Millane. Phase retrieval in crystallography and optics. *J. Opt. Soc. Am. A*, 7(3): 394–411, Mar 1990. doi: 10.1364/JOSAA.7.000394. URL <https://opg.optica.org/josaa/abstract.cfm?URI=josaa-7-3-394>.
- Tom M Mitchell. *Machine learning*, volume 1. McGraw-hill New York, 1997.
- Tom Michael Mitchell. The need for biases in learning generalizations, 2007. URL <https://api.semanticscholar.org/CorpusID:3237155>.
- Marco Mondelli and Andrea Montanari. Fundamental Limits of Weak Recovery with Applications to Phase Retrieval. *Foundations of Computational Mathematics*, 19(3): 703–773, June 2019. ISSN 1615-3383. doi: 10.1007/s10208-018-9395-y. URL <https://doi.org/10.1007/s10208-018-9395-y>.
- Andrea Montanari. Sampling, diffusions, and stochastic localization. *arXiv preprint arXiv:2305.10690*, 2023.
- Andrea Montanari and Jose Pereira. Which graphical models are difficult to learn? In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009. URL https://proceedings.neurips.cc/paper_files/paper/2009/file/22fb0cee7e1f3bde58293de743871417-Paper.pdf.
- Andrea Montanari and Yuchen Wu. Posterior sampling in high dimension via diffusion processes. *arXiv preprint arXiv:2304.11449*, 2023.
- Faruck Morcos, Andrea Pagnani, Bryan Lunt, Arianna Bertolino, Debora S. Marks, Chris Sander, Riccardo Zecchina, José N. Onuchic, Terence Hwa, and Martin Weigt. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, 108(49):E1293–E1301, 2011. doi: 10.1073/pnas.1111471108. URL <https://www.pnas.org/doi/abs/10.1073/pnas.1111471108>.
- Marc Mézard, Giorgio Parisi, and Riccardo Zecchina. Analytic and algorithmic solution of random satisfiability problems. *Science*, 297(5582):812–815, 2002.

- J. A. Nelder and R. W. M. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3):370–384, 1972. ISSN 00359238, 23972327. URL <http://www.jstor.org/stable/2344614>.
- Alexander Nemirovski and David Yudin. Prox methods and variational inequalities. *Mathematics of Operations Research*, 29(1):1–28, 2004.
- Yurii Nesterov. *Introductory Lectures on Convex Optimization*. Springer, 2013.
- H Nishimori. Exact results and critical properties of the ising model with competing interactions. *Journal of Physics C: Solid State Physics*, 13(21):4071, jul 1980. doi: 10.1088/0022-3719/13/21/012. URL <https://dx.doi.org/10.1088/0022-3719/13/21/012>.
- Hidetoshi Nishimori. *Statistical physics of spin glasses and information processing*. Oxford Science Publications, 2001.
- Maxence Noble, Louis Grenioux, Marylou Gabri  , and Alain Oliviero Durmus. Learned reference-based diffusion sampling for multi-modal distributions. *ICLR 2025*, 2024.
- Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer series in operations research and financial engineering. Springer, New York, NY, 2. ed. edition, 2006. ISBN 978-0-387-30303-1. URL http://gso.gbv.de/DB=2.1/CMD?ACT=SRCHA&SRT=YOP&IKT=1016&TRM=ppn+502988711&sourceid=fbw_bibsonomy.
- Tomoyuki Obuchi, Yoshinori Nakanishi-Ohno, Masato Okada, and Yoshiyuki Kabashima. Statistical mechanical analysis of sparse linear regression as a variable selection problem. *Journal of Statistical Mechanics: Theory and Experiment*, 2018(10):103401, 2018.
- Giorgio Parisi. Towards a mean field theory for spin glasses. *Phys. Lett*, 73(A):203–205, 1979.
- Giorgio Parisi. Order parameter for spin glasses. *Phys. Rev. Lett*, 50:1946–1948, 1983.
- Ayush Parti. Diffusion models: A beginners guide, 2024. URL <https://pareto.ai/blog/diffusion-models>.
- Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
- Karl Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901.
- Fabrizio Pittorino, Carlo Lucibello, Christoph Feinauer, Gabriele Perugini, Carlo Baldassi, Elizaveta Demyanenko, and Riccardo Zecchina. Entropic gradient descent algorithms and wide flat minima. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124015, dec 2021. doi: 10.1088/1742-5468/ac3ae8. URL <https://dx.doi.org/10.1088/1742-5468/ac3ae8>.
- M. Potters and J.P. Bouchaud. *A First Course in Random Matrix Theory: for Physicists, Engineers and Data Scientists*. Cambridge University Press, 2020. ISBN 978-1-108-48808-2. URL <https://books.google.it/books?id=jdsCEAAQBAJ>.

- Yixuan Qiu, Lingsong Zhang, and Xiao Wang. Unbiased contrastive divergence algorithm for training energy-based latent variable models. In *International Conference on Learning Representations*, 2020. URL <https://api.semanticscholar.org/CorpusID:209316195>.
- Sundeeep Rangan. Generalized approximate message passing for estimation with random linear mixing. *2011 IEEE International Symposium on Information Theory Proceedings*, pages 2168–2172, 2011.
- Sundeeep Rangan, Philip Schniter, and Alyson K. Fletcher. Vector approximate message passing. *2017 IEEE International Symposium on Information Theory (ISIT)*, page 1588–1592, 2017. doi: 10.1109/ISIT.2017.8006797. URL <https://doi.org/10.1109/ISIT.2017.8006797>.
- Pradeep Ravikumar, Martin J. Wainwright, and John D. Lafferty. High-dimensional Ising model selection using L1-regularized logistic regression. *The Annals of Statistics*, 38(3):1287 – 1319, 2010. doi: 10.1214/09-AOS691. URL <https://doi.org/10.1214/09-AOS691>.
- Federico Ricci-Tersenghi and Guilhem Semerjian. On the cavity method for decimated random constraint satisfaction problems and the analysis of belief propagation guided decimation algorithms. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(09):P09001, 2009.
- Federico Ricci-Tersenghi, Guilhem Semerjian, and Lenka Zdeborová. Typology of phase transitions in bayesian inference problems. *Physical Review E*, 99(4), April 2019. ISSN 2470-0053. doi: 10.1103/physreve.99.042109. URL <http://dx.doi.org/10.1103/PhysRevE.99.042109>.
- Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models . In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, Los Alamitos, CA, USA, June 2022. IEEE Computer Society. doi: 10.1109/CVPR52688.2022.01042. URL <https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.01042>.
- Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.
- David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by backpropagating errors. *Nature*, 323(6088):533–536, 1986.
- David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. *Learning representations by back-propagating errors*, page 696–699. MIT Press, Cambridge, MA, USA, 1988. ISBN 0262010976.
- Luca Saglietti, Yue M Lu, and Carlo Lucibello. Generalized approximate survey propagation for high-dimensional estimation. *Journal of Statistical Mechanics: Theory*

- and Experiment*, 2020(12):124003, dec 2020. doi: 10.1088/1742-5468/abc62c. URL <https://doi.org/10.1088%2F1742-5468%2Fabc62c>.
- Levent Sagun, Leon Bottou, and Yann Lecun. Singularity of the hessian in deep learning. *ICLR 2016*, 11 2016. doi: 10.48550/arXiv.1611.07476.
- Levent Sagun, Utku Evci, V. Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the hessian of over-parametrized neural networks. In *ICLR 2018*, 06 2017. doi: 10.48550/arXiv.1706.04454.
- Stefano Sarao-Mannelli, Giulio Biroli, Chiara Cammarota, Florent Krzakala, Pierfrancesco Urbani, and Lenka Zdeborova. Complex dynamics in simple neural networks: Understanding gradient flow in phase retrieval. *2020 Conference on Neural Information Processing Systems - NeurIPS 2020*, Dec 2020. URL <https://hal.science/hal-02983829>.
- Vitor Sessak and Rémi Monasson. Small-correlation expansions for the inverse Ising problem. *Journal of Physics A: Mathematical and Theoretical*, 42(5):055001, 2009. doi: 10.1088/1751-8113/42/5/055001.
- H. Sebastian Seung, Haim Sompolinsky, and Naftali Tishby. Statistical mechanics of learning from examples. *Physical Review A*, 45(8):6056, 1992.
- Silvio Franz and Giorgio Parisi. Recipes for metastable states in spin glasses. *J. Phys. I France*, 5(11):1401–1415, 1995. doi: 10.1051/jp1:1995201. URL <https://doi.org/10.1051/jp1:1995201>.
- Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *International conference on machine learning*, pages 2256–2265, 2015.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, January 2014. ISSN 1532-4435.
- Davide Straziota and Luca Saglietti. Isolating the hard core of phaseless inference: the phase selection formulation, 2025. URL <https://arxiv.org/abs/2502.04282>.
- Davide Straziota, Elizaveta Demyanenko, Carlo Baldassi, and Carlo Lucibello. Generative diffusion for perceptron problems: statistical physics analysis and efficient algorithms, 2025. URL <https://arxiv.org/abs/2502.16292>.
- Ilya Sutskever and Tijmen Tieleman. On the convergence properties of contrastive divergence. In Yee Whye Teh and Mike Titterton, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 789–795, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL <https://proceedings.mlr.press/v9/sutskever10a.html>.
- Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer, London, 2010.
- Albert Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2005. ISBN 0-89871-576-0.

- Tijmen Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1064–1071, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605582054. doi: 10.1145/1390156.1390290. URL <https://doi.org/10.1145/1390156.1390290>.
- Andrey N. Tikhonov and Vasilij Y. Arsenin. *Solutions of ill-posed problems*. V. H. Winston & Sons, Washington, D.C.: John Wiley & Sons, New York, 1977. Translated from the Russian, Preface by translation editor Fritz John, Scripta Series in Mathematics.
- Gašper Tkačik, Olivier Marre, Dario Amodei, Elad Schneidman, William Bialek, and Michael J. Berry, II. Searching for collective behavior in a large network of sensory neurons. *PLOS Computational Biology*, 10(1):1–23, 01 2014. doi: 10.1371/journal.pcbi.1003408. URL <https://doi.org/10.1371/journal.pcbi.1003408>.
- Hugo Touchette. The large deviation approach to statistical mechanics. *Physics Reports*, 478:1–69, 2009.
- E. Tsang. *Foundations of Constraint Satisfaction*. Computation in cognitive science. Academic Press, 1993. ISBN 978-0-12-701610-8. URL <https://books.google.it/books?id=TnxQAAAAMAAJ>.
- V. Vapnik and A. Chervonenkis. *Uniform Convergence of Relative Frequencies of Events to Their Probabilities*. Springer, 1971.
- Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer: New York, 1999.
- Francisco Vargas, Will Grathwohl, and Arnaud Doucet. Denoising diffusion samplers. In *The Eleventh International Conference on Learning Representations, 2023*, 2023a.
- Francisco Vargas, Shreyas Padhy, Denis Blessing, and Nikolas Nüsken. Transport meets variational inference: Controlled monte carlo diffusions. *Workshop on New Frontiers in Learning, Control, and Dynamical Systems at the International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA, 2023*, 2023b.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6000–6010, 2017.
- Enrico Ventura, Beatrice Achilli, Gianluigi Silvestri, Carlo Lucibello, and Luca Ambrogioni. Manifolds, random matrices and spectral gaps: The geometric phases of generative diffusion. In *ICLR 2025*, 10 2024. doi: 10.48550/arXiv.2410.05898.
- Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.
- Marc Vuffray, Sidhant Misra, Andrey Lokhov, and Michael Chertkov. Interaction screening: Efficient and sample-optimal learning of ising models. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates,

- Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/861dc9bd7f4e7dd3cccd534d0ae2a2e9-Paper.pdf.
- Marc Vuffray, Sidhant Misra, and Andrey Y Lokhov. Efficient learning of discrete graphical models*. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124017, jan 2022. doi: 10.1088/1742-5468/ac3aea. URL <https://dx.doi.org/10.1088/1742-5468/ac3aea>.
- Paul J. Werbos. Generalization of backpropagation to recurrent neural networks. *Proceedings of the Institute of Electrical and Electronics Engineers (IEEE)*, 63(3):453–462, 1974.
- Eugene Paul Wigner. On the distribution of the roots of certain symmetric matrices. *Annals of Mathematics*, 67:325, 1958. URL <https://api.semanticscholar.org/CorpusID:123682463>.
- Lenka Zdeborová and Florent Krzakala. Statistical physics of inference: Thresholds and algorithms. *Advances in Physics*, 65(5):453–552, 2016.
- Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM*, 64(3):107–115, February 2021. ISSN 0001-0782. doi: 10.1145/3446776. URL <https://doi.org/10.1145/3446776>.