SISSA
Scuola
Internazionale
Superiore di
Studi Avanzati

Mathematics Area - PhD course in
Mathematical Analysis, Modelling, and Applications

Learning Beyond the Gaussian Approximation

Algorithmic Hardness, Neural Network Dynamics and Independent Components

Candidate:
Lorenzo Bardone

Advisor:
Prof. Sebastian Goldt

Academic Year 2024-2025



Abstract

Neural networks excel at discovering statistical patterns in high-dimensional data sets. In practice, higher-order cumulants, which quantify the non-Gaussian correlations between three or more variables, are particularly important for the performance of neural networks. In this thesis we study the efficiency of neural networks at extracting features from higher-order cumulants. We first discuss the fundamental statistical and computational limits of recovering a single non-Gaussian direction, by computing the number of samples n required to reliably distinguish between inputs from the spiked cumulant model and isotropic Gaussian inputs. Our findings confirm the existence of a wide statistical-to-computational gap: statistical distinguishability requires $n \gtrsim d$ samples, while distinguishing the two distributions in polynomial time requires $n \gtrsim d^2$ samples for the class of algorithms covered by the low degree conjecture. Numerical experiments show that neural networks do indeed learn to distinguish the two distributions with quadratic sample complexity. Then, to further bridge the gap between theory and practice, we show that correlations between latent variables along the directions encoded in different input cumulants speed up learning from higher-order cumulants, reaching linear sample complexity. We show this effect analytically by deriving nearly sharp thresholds for the number of samples required by a single neuron to weakly-recover these directions using online SGD from a random start in high dimensions. Our analytical results are confirmed in simulations of two-layer neural networks and unveil a new mechanism for hierarchical learning in neural networks.

Finally we connect to feature learning of deep networks via the proxy of independent component analysis (ICA), a simple unsupervised method that seeks the most non-Gaussian projections of its inputs. The motivation is that ICA finds features that are similar to the first-layer filters learned by deep convolutional networks on classification tasks. This similarity suggests that ICA provides a simple, yet principled model for studying feature learning. Here, we prove that FastICA, the most popular ICA algorithm, requires at least $n \gtrsim d^4$ samples to recover a single non-Gaussian direction from d-dimensional inputs on a simple synthetic data model. We show that vanilla online SGD outperforms FastICA, and prove that the optimal sample complexity $n \gtrsim d^2$ can be reached by smoothing the loss, albeit in a data-dependent way. We conclude by discussing how this picture extends to different distributions and architectures, such as generative diffusion models.

Contents

Contents **iii**

Notation **1**

1 Introduction **2**
1.1 Background and motivation . . . 2
1.2 Single and multi-index models . . . 5
1.3 Overview of the thesis . . . 7
1.4 Organization of the manuscript . . . 10

2 Algorithmic hardness **12**
2.1 Introduction . . . 12
2.2 The data models . . . 14
2.3 How many samples do we need to learn? . . . 15
2.4 Learning from HOCs with neural networks and random features . . . 22
2.5 Concluding perspectives . . . 26
2.A Experimental details . . . 28
2.B Mathematical details on the hypothesis testing problems . . . 29
2.C Details on the replica analysis . . . 46

3 Neural Network dynamics **48**
3.1 Introduction . . . 48
3.2 Results and insights: an informal overview . . . 48
3.3 Rigorous analysis of the perceptron: the non-Gaussian information exponent . . . 52
3.4 Discussion: hierarchies of learning with neural networks . . . 57
3.5 Concluding perspectives . . . 60
3.A Dynamical analysis . . . 62
3.B Details on the experiments . . . 72

4 Feature learning from non-Gaussian inputs: the case of Independent Component Analysis **75**
4.1 Introduction . . . 75
4.2 Setup . . . 77
4.3 Theoretical Results . . . 80
4.4 The search phase of FastICA on images . . . 87
4.5 FastICA provably recovers a non-Gaussian direction at linear sample complexity on the "subspace model" . . . 88
4.6 Concluding perspectives . . . 89

4.A Experimental details . . . 91
4.B Properties and definitions . . . 92
4.C Measures of non-Gaussianity for ICA . . . 93
4.D FastICA . . . 95
4.E Smoothing the landscape . . . 100
4.F The subspace model turns out to be a spiked cumulant model after preprocessing . . 103
4.G Additional numerics . . . 110

5 Concluding perspectives **112**
5.1 Overview of the recurring themes . . . 112
5.2 Perspectives . . . 113
5.3 Final remarks . . . 116

Bibliography **117**

A Notions on Hermite polynomials **129**
A.1 Links with cumulant theory . . . 132

Notation

We use the convention $0 \in \mathbb{N}$. $n \in \mathbb{N}$ denotes the number of samples, d the dimensionality of each data point x. The letters k, m usually denote free natural parameters. Let $[n] := \{1, \dots, n\}$, $\mu \in [n]$ is an index that ranges on the samples and $i \in [d]$ usually ranges on the data dimension. Throughout we use the standard notations $o, O, \Theta, \Omega, \omega$ for asymptotics. We recall them here for sequences, but they generalize in the straight forward way in the case of functions.
Let $\{a_k\}_{k\in\mathbb{N}}$, $\{b_k\}_{k\in\mathbb{N}}$ be two real valued sequences. Then:

$$
\begin{aligned}
a_k \in o(b_k) &\iff \lim_{k\to\infty} \frac{a_k}{b_k} = 0 \\
a_k \in O(b_k) &\iff \exists\, C > 0 \in \mathbb{R} \quad \forall k > k_0 \quad |a_k| \leq C|b_k| \\
a_k \in \Theta(b_k) &\iff \exists\, C_1, C_2 > 0 \quad \forall k > k_0 \quad C_1 b_k \leq a_k \leq C_2 b_k \\
a_k \in \Omega(b_k) &\iff \exists\, C > 0 \in \mathbb{R} \quad \forall k > k_0 \quad |a_k| \geq C|b_k| \\
a_k \in \omega(b_k) &\iff \lim_{k\to\infty} \frac{|a_k|}{|b_k|} = \infty
\end{aligned}
$$

Occasionally we use the shorthands $a_n \ll b_n$ for $a_n = o(b_n)$, $a_n \lesssim b_n$ for $a_n = O(b_n)$ and $a_n \asymp b_n$ or $a_n \approx b_n$ for $a_n = \Theta(b_n)$.

Chapter 1

Introduction

1.1 Background and motivation

The theory of deep learning aims to build a mathematical framework that can answer questions that come from practice. Some examples are: *What makes one neural network design better suited to a task than another? Why do some learning rules succeed where others fail? Are networks actually learning, or do they just "regurgitate" memorized material? On which features of their inputs do neural networks rely on to make predictions?*. These questions introduce the three main ingredients of deep learning: *the architecture, the optimization algorithm and the data distribution*. The first two are easier to understand: *architecture* indicates how many artificial neurons there are and the way in which they are connected: it impacts how the information is propagated inside the network and the operations that the input undergoes before becoming output. *Optimization algorithm* refers to the way in which training is carried out; how the weights inside the network are adjusted to map a given input to a specified output.

At first sight, it may seem there is nothing missing: after all, once the network has been built and trained, then all the work has been done and it is ready for deployment. However, here lies the key issue: the reason why after training the network *can actually work* in practice, i.e. it is able to predict the correct label of inputs that it never saw before, is something that concerns the *probability distribution* of the data.

To better illustrate this, consider the example in fig. 1.1. In this case, the training data consists only of 16 equispaced samples taken from an EEG signal. Both a two-layer neural network and a polynomial interpolator manage to fit these data points perfectly. However, looking at how the lines behave outside the training nodes, we see a stark difference. The neural network solution is always very simple, and matches much more closely the noisier ground truth line, than the wobbly polynomial interpolator, which explodes at the boundary of the interval.

The aim of this example is not to try to elevate neural network algorithms above standard polynomial interpolations: a piecewise linear algorithm would have performed almost as well as the neural network on this simple one-dimensional example. The point is that to assess the performance of a method, one needs to look at the *generalization error*, the mistakes made on inputs that were note seen during training. Hence, by definition, the *generalization error* depends on the whole procedure (often probabilistic) that generates data and labels. If instead of an electroencephalogram, fig. 1.1 depicted the values of a high degree polynomial, then the Lagrange polynomial interpolator would have given better results than the neural network. So why are neural networks widely used in practice, when Lagrange polynomial interpolation on equispaced nodes is usually discouraged? There is a purely empirical answer: on most instances that can be encountered in practice, data

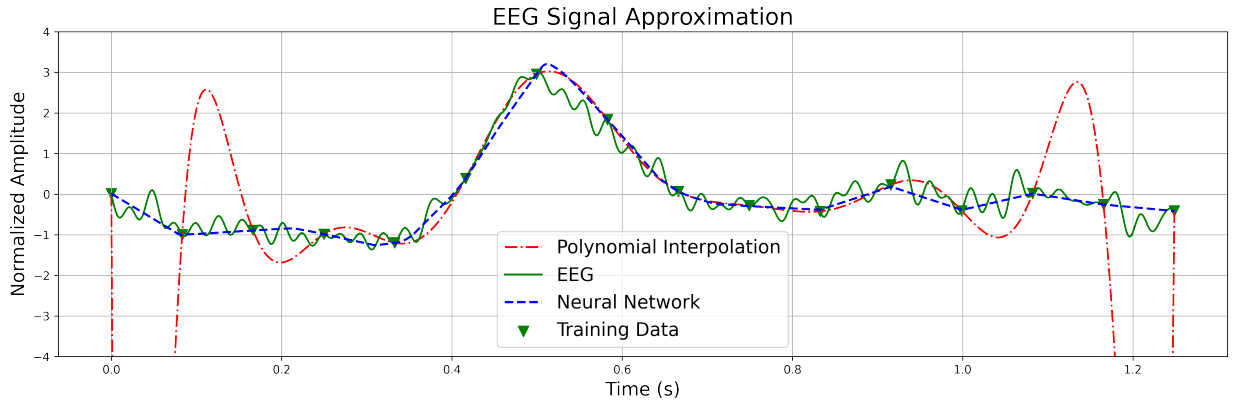

Figure 1.1: Electroencephalogram signal (green) fitted by a two-layer fully connected neural network (blue), and Lagrange interpolation polynomial (red).

behaves more similarly to the the green and blue lines in fig. 1.1, than the red one. However, if we want to build a theory of deep learning, we cannot be satisfied with just this answer. We want to be able to make predictions on which methods will work on which data distributions in advance, before investing resources in building and training them. Hence, it is not enough to study *architecture* and *optimization algorithm* but we need to couple them with models of data distributions that allow to assess how the *generalization error* is impacted by specific properties of the network, the training regime, and the *statistical properties of the inference task*. In this way it is possible to find and isolate which properties of the three ingredients are vital for fruitful synergies.

It follows from the previous discussion that to fully specify the framework of a theoretical investigation, one needs to describe the assumptions or choices made for each ingredient. Combinatorially many approaches are possible, and each choice, in order to be mathematically explainable, must involve some simplification in each of the three dimensions. Regarding the architecture, a lot of research is focusing on *two-layer fully connected neural networks* (2LFCNN or simply 2LNN), i.e. a map of the form

$$\mathbb{R}^d \ni x \to \sum_{i=1}^{P} a_i \sigma \left(\sum_{j=1}^{d} W_{ij} x_j \right) \tag{1.1}$$

where $(a, W) \in \mathbb{R}^P \times \mathbb{R}^{Pd}$ are the trainable weights, and σ is a non-linear *activation* (or *link*) function. The reason behind the interest in this specific architecture is that 2LNN are the simplest networks that exhibit two core properties of deep learning:

- **Feature learning**: The possibility to dynamically change W in eq. (1.1) during training allows to select the right linear projections (features) of the data for the task at hand. This intuitively means that deep networks can "autonomously decide what to look at". As we will see in section 1.2, algorithms based on feature learning are provably more efficient than others on a wide class of high-dimensional inference problems. Key questions that drive theoretical research in the field are related to *feature learning*, some examples that will be relevant in the following are: *which features are the best for which tasks? Under which conditions are networks able to learn them? Are there harder features to learn than others? Is there a particular order in which features are dynamically learnt?*

- **Universal approximation**: under the condition of non-polynomial activation σ, two-layer neural networks can *closely approximate* almost all function. More precisely, let us denote

networks of the form eq. (1.1) as $f_{(a,W)} : \mathbb{R}^d \to \mathbb{R}$. Suppose σ is monotone, going to 0 for $x \to -\infty$, and to 1 for $x \to +\infty$, then for any compact set $K \subseteq \mathbb{R}^d$, target function $f^* \in \mathcal{C}(K)$ and $\epsilon > 0$, there exist $P, \bar{a} \in \mathbb{R}^P, \bar{W} \in \mathbb{R}^{Pd}$ such that $\sup_{x \in K} |f_{(\bar{a},\bar{W})}(x) - f^*(x)|| \leq \epsilon$ (this is the version from [1], the hypotheses on σ and the continuity of f^* can be relaxed, see also Cybenko [2] and Barron [3] for different versions of the result).

Universal approximation results provide another perspective to look at the interplay between architecture, algorithms and data: from the expressivity point of view, 2LNN are enough for any application. If taken wide enough, they can well approximate any function. So why, then, would we need different architectures? Part of the answer is in the fact that approximation theorems do not give a recipe on how to find the solution $(\bar{a}, \bar{W})$, so *optimization algorithms* are needed, and differences in the network architecture will impact the training dynamics. The other half of the answer is provided again by fig. 1.1: there may be many different functions that fit f^* in the training set, but the crucial point is how they behave on new data. There is evidence that common algorithms used to optimize deep networks, such as *stochastic gradient descent* (SGD) or *Adam* [4] tend to favor simple solutions (see for instance [5]), while the architecture of the network can bias the solution in diverse ways, for example introducing desired symmetries (e.g. translation invariance for convolutional networks).

A common way to theoretically model learning with two-layer neural networks is to consider the *teacher-student* setup: fix d and P, and consider a general distribution $\mathcal{P}(x, y)$. Let the network $f^* = f_{(a^*,W^*)}$ be the one that minimizes a loss function, for instance in the mean squared case we have,

$$(a^*, W^*) = \text{argmin}_{a,W}\ L(a, w) = \text{argmin}_{a,W} \mathbb{E}_{\mathcal{P}}[|f_{(a,W)}(x) - y|^2]. \tag{1.2}$$

This means that the network f^* is the best possible mapping $x \to y$ that a 2LNN can ever learn. The *teacher-student* setup consists in approximating $\mathcal{P}$ as first sampling x via the marginal $\mathcal{P}_X$ and then approximating $P(y|x)$ via $f^*(x)+$ noise. In this way the optimization problem becomes:

$$\text{argmin}_{a,W} \mathbb{E}_{\mathcal{P},\text{noise}}[|f_{a,W}(x) - f^*(x) - \text{noise}|^2] \tag{1.3}$$

where f^* is the teacher and f is the student; usually the two networks are taken to have the same number of parameters.

The convenience of this operation becomes evident when one models P_X to be a particularly simple distribution, usually a standard Gaussian in $\mathbb{R}^d$: under the hypothesis of gaussian label noise the intractable high-dimensional integral in eq. (1.2) became a tamer Gaussian integral:

$$\text{argmin}_{a,W} \underset{(x,z)\sim\mathcal{N}(0,\mathbb{1}_{d+1})}{\mathbb{E}}[|f_{a,W}(x) - f^*(x) - \epsilon z|^2]. \tag{1.4}$$

This allows to focus the analysis on properties of the architecture eq. (1.1) and of the task f^*, by choosing a particular initial structure of the teacher's weights (a^*, W^*).

This approach proved to be very fruitful, a first line of work analysed online training dynamics on a simplified version of eq. (1.1), called *committee machines* (second-layer weights constantly equal to 1) started in [6], where the high-dimensional ($d \to \infty$) training dynamics were reduced to a finite dimensional dynamical system in terms of *order parameters*, allowing for an almost exhaustive analysis of the evolution of interesting quantities associated with the inference task, such as the generalization error. This approach was then extended to more and more general networks in [7–9]. In the next section we will overview a specific class of models which focus on the problem of *low-rank detection and estimation* of a key subspace, which will be important to introduce and motivate the content of this thesis.

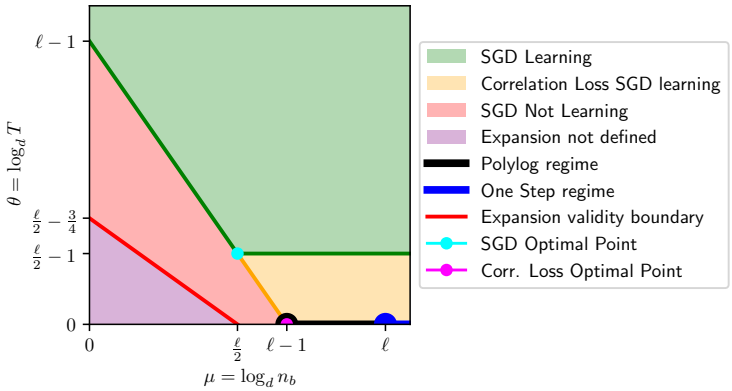

Figure 1.2: This extract from [15] illustrates the relationship between the number of steps T and the batch size n_b for SGD on single index models. Note that the total sample complexity is the product $n = n_b T$. Along the axes we have the limiting cases discussed in this section: *polylog*-batch size SGD along the y-axis and *polylog*-steps GD on the x-axis, of which batch one SGD and single step GD are a particular cases. The plot is drawn considering optimal choices of the learning rate for each regime and also illustrates how MSE loss does not allow to have the optimal sample complexity for large batch sizes and few steps regime, whereas correlation loss (see chapter 3) does. For further details, see [15]

1.2 Single and multi-index models

In this section we provide a brief overview of *Gaussian single and multi index models*, which are teacher-student models as in eq. (1.4) that involve a low-rank feature matrix W^* inside f^*, i.e.

$$f^*(x) = g^*\left(w_1^* \cdot x, w_2^* \cdot x, \dots, w_r^* \cdot x\right) \tag{1.5}$$

with $r = O_d(1)$, and $g^* : \mathbb{R}^r \to \mathbb{R}$ a regular function (the precise assumptions depend on the exact analysis one needs to carry out, for the illustrative purposes of this section we will take $g^* \in \mathcal{C}^\infty(\mathbb{R}^d \to \mathbb{R})$).

We start by focusing on the *single index* case, that amounts to considering $r = 1$. On these models we investigate the *sample complexity*, which is the minimum scaling of the number of samples n (as a function of d) that a student needs to see in order to learn the hidden direction $w^* := w_1^*$.

In the case of polynomial g^*, algorithms not based on feature learning, such as kernels or random features (see [10–12]), have a sample complexity with a poor dependence on the degree of g^*: if the degree is k, then it is necessary to have $n \gtrsim d^k$ samples in order to fully learn σ (see [13, 14]). More in general for d^k samples will learn the best poly-k approximation of g^*, even in the case of non-polynomial g^*.

Turning to feature learning algorithms, the simplest case is the one of a single neuron w trying to find the hidden direction via online SGD; setup considered in [16]. In this scenario the invariant that regulates the sample complexity is not the degree of the link function, but the *information exponent* ℓ, which is the degree of the first non trivial term in the Hermite decomposition of g^* (for a more formal definition, see definition A.3). After choosing the best learning rate, the sample complexity for *weak recovery* ,i.e. non-negligible overlap of the student with the hidden direction $\frac{w \cdot w^*}{||w|| ||w^*||} = \Theta(1)$, is:

- if $\ell = 0$ or 1, then sample complexity is linear, $n \gtrsim d$

- if $\ell = 2$ then $n \gtrsim d \log d$.
- if $\ell \geq 3$ then $n \gtrsim d^{\ell-1}$.

It is possible to further speed up these timescales with the *smoothing trick* first introduced in [17] and then developed for single index models in [18] (see also chapter 4 for more details), which allows to reach $n \gtrsim d^{\ell/2}$.

The sample complexities described, are for the case of *online* batch one stochastic gradient descent, which means that at each step the weights are updated using only one new data sample. It is possible also to go to the opposite regime, with few samples, but with very large batches, [19][20] followed these approaches, that were later generalized for a more general trade off number of steps to batches dimension in [15]. In fig. 1.2 we can see the complete picture. Note that the single step SGD, although often employed for theoretical studies due to its analytic tractability, is suboptimal and requires overall d^ℓ samples. However, by employing $polylog(d)$ gradient updates and *correlation loss* it is possible to achieve recovery with batches of $d^{\ell-1}$ samples, matching the sample complexity of online SGD.

Finally, it was recently found that the invariant given by the information exponent works only due to particular symmetries of the labels, and by applying random label transformation it is possible to lower ℓ. This leads to a new invariant, called *generative exponent* k [21], which is the minimum information exponent of the functions in the set $\{\mathcal{T} \circ f^* \mid \mathcal{T} \circ f^* \in L^2(\mathbb{R}, \mathcal{P}_X)\}$. The generative exponent can be even considerably lower than the information one. For example the information exponent of polynomials can be arbitrarily large, while their generative exponent is at most two. To achieve the sample complexities predicted by the generative exponent, one needs to either transform the labels, or change the structure of the weight updates, so that they escape the regime of *correlational queries* (CSQ): a regime in which the optimization algorithm can acquire new information only through a prescribed, *correlational* coupling between inputs and labels, that does not allow to break the symmetries that can slow down learning (for more details on CSQ see section 1.2 in [21]).

An effective and surprising way to produce the speed up consists in repeating batches ([22], [23]), which is a strategy commonly used in practice, mostly due to the scarcity of data, but previously was not believed to carry particular benefits, it was often believed that the samples complexities predicted by the *information exponent* would extend to this case with just marginal improvements to the constants in front of the asymptotic scalings.

However, the correlations between different times of the learning dynamics that reusing batches introduces, allows to change the structure of weight updates in such a way that the symmetries causing the information exponent to increase are not impacting anymore, allowing to reach a regime in which learning depends only on the *generative exponent*. See [22] and [23] for details on repeating batches, or also [24] for an overview of other algorithms that allow to interpolate between information and generative exponents.

Turning now to the case of *multi-index* models, in the case of a function g^* that does not couple much the different directions, then the situation is almost identical to the rank one case: if the student network is expressive enough, and under reasonable conditions on the training regime, each direction w_i is learned with the sample complexity predicted by the information/generative exponent of its activation σ_i^*. However, there are many cases in which the interdependence between the different arguments of g^* can produce speed-up effects, called the *staircase phenomenon* Dandi *et al.* [20], Abbé *et al.* [25, 26] and Bietti *et al.* [27]. We will illustrate it with an example inspired from [20]. Consider the two teacher functions

$$f_1^*(x) = x_1 + x_1 x_2, \qquad f_2^*(x) = x_1 x_2$$

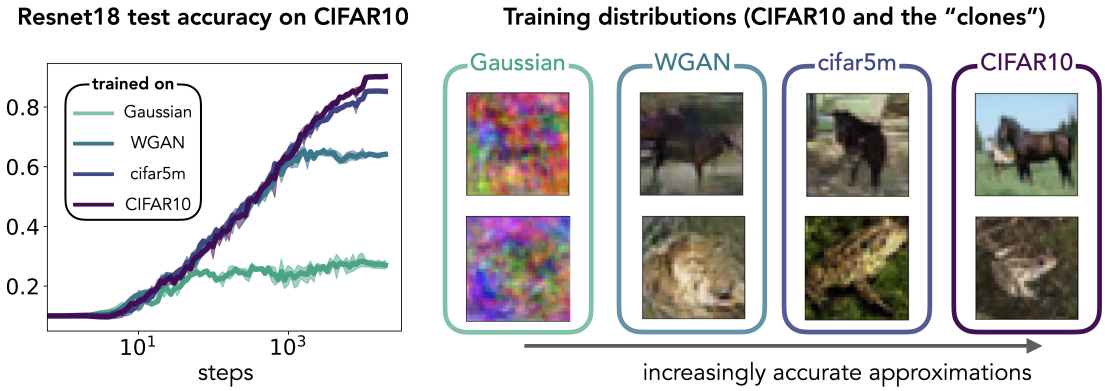

Figure 1.3: Experiment from [28], where a deep network was trained on different clones of CIFAR10 and then tested on a classification task on the original dataset. On the left the accuracy of the performance, portrayed as training progressed, on the right samples from the cloned distributions.

They are multi-index model with $w_1 = e_1$ and $w_2 = e_2$. Performing full batch GD with linear sample complexity, for two steps and two students f_1, f_2 of the form eq. (1.1), with enough neurons, we will observe that in the first step f_1 will learn the direction e_1, since the function $x \to x_1$ has information exponent one. On the contrary, after one step, f_2 will not have learned anything since $x_1 x_2$ has information exponent two. Performing now the second step, f_2 will stay stuck at trivial performance, while f_1 learns also the second direction e_2. Indeed the function $x \to x_1 x_2$, which has information exponent two by itself, when the network has already learned x_1, it becomes of the same difficulty of learning $x \to x_2$, which is learnable in one step of GD. Remarkably, in this way the network learns in two steps a function that would have required an extensive number of steps to be learned.

This was by no means an exhaustive introduction to single and multi index models, for more details we refer to the sources cited in this section. There will also be more in depth discussions of some of the topics introduced here in section 3.4.1.1.

1.3 Overview of the thesis

The main topic of this thesis is to better understand what happens when it is not possible to pass from (1.2) to eq. (1.4). In particular, when it is not possible to assume the data to be Gaussian. Motivations for this line of research come from real data: experiments in [28], that we report in fig. 1.3, show what happens when a network is trained on approximated clones of CIFAR10 and then tested on the true distribution. The green line in the plot is the *best gaussian approximation*: from all the classes in CIFAR10 the empirical mean and covariance matrices were computed and then the clones were constructed by sampling from the Gaussian distribution with such mean and covariance. While in the initial stage of learning this does not seem to impact the accuracy, as training goes on, the performance of the network trained on the Gaussian clones quickly saturates and by the end of learning there is a delta of around 60 percentage points. This is evidence that a key part of the information that networks exploit to solve real world tasks is encoded in the non-Gaussian statistics of data. To be more precise, we define as *non-Gaussian statistics* all the properties of a data distribution that cannot be captured by a Gaussian approximation. A possible representation of these quantities comes from the concept of *cumulant*: let $X \in \mathbb{R}^d$ be a random variable and $\Phi_X(\xi) := \mathbb{E}\left[e^{\xi \cdot X}\right]$ be its *moment generating function*. The cumulants are defined as the

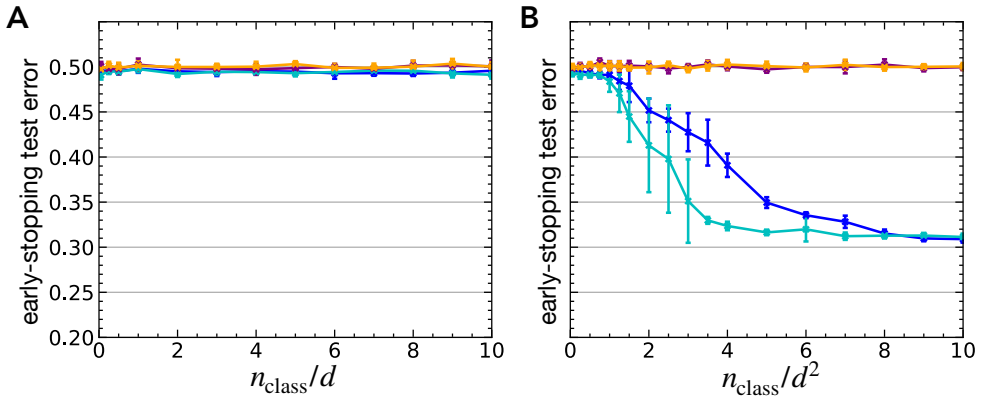

Figure 1.4: **Learning the spiked cumulant task with neural networks and random features.** Test accuracy of random features (orange and red) and early-stopping test accuracy of two-layer ReLU neural networks (blue and light blue) on the spiked cumulant task eq. (2.3) with linear and quadratic sample complexity ($n_{\text{class}} \asymp d$ in **A** and $n_{\text{class}} \asymp d^2$ in **B** where $d = 40$ for blue and red lines, $d = 100$ for light blue and orange). More details in section 2.4.2.

coefficients of the expansion of $K_X(\xi) := \log \Phi_X(\xi)$. The first two are mean vector and covariance matrix, which are the only ones that a Gaussian approximation can capture. All the cumulants of order larger than two (corresponding to the coefficients of the expansion of K_X with degree ≥ 3) are called *higher-order cumulants (HOCs)*, and are one of the main objects of study of this thesis. We want to understand how hard it is to learn from them, both from an abstract perspective, and from the point of view of neural network dynamics. Then, we want to investigate how HOCs in data distributions affect feature learning of deep networks.

In this thesis we will present these themes in three steps, that we overview in the following subsections.

1.3.1 Algorithmic hardness

In chapter 2 we will first study the algorithmic baselines of learning from HOCs. We introduce the *spiked cumulant model*, a variant of the *spiked Wishart* model, that differs from standard Gaussian only via high-order cumulants (≥ 3) of the distribution. In order to learn from HOCs one needs to first be able to detect them, i.e., be able to distinguish a distribution with non-trivial HOCs from a distribution with only low-order correlations. Hence, we will study the problem of hypothesis testing in the high-dimensional limit: given n samples from a $d-$dimensional distribution, is it possible to devise a statistical test that will find whether the samples were taken from the *spiked cumulant model* or they were just standard Gaussian noise? Is it possible to have a probability of correct distinguishability that converges to 1 as $n, d \to \infty$ with a fixed scaling $n \asymp d^\theta$? We answer these questions via the method of likelihood ratio and low-degree polynomials (section 2.3). The main results in this direction are:

- We provide an explicit formula for the likelihood ratio norm, and prove that if $\theta < 1$, it is bounded, whereas if $\theta > 1$ it diverges in the high-dimensional limit. This implies that linear sample complexity is the transition between an inference problem that is information theoretically impossible, to a problem that is statistically solvable.

- We provide an alternative proof of the *low degree analysis* from [29], which confirms the presence of a *statistical-to-computational* gap for the problem: if we restrict to polynomial-time

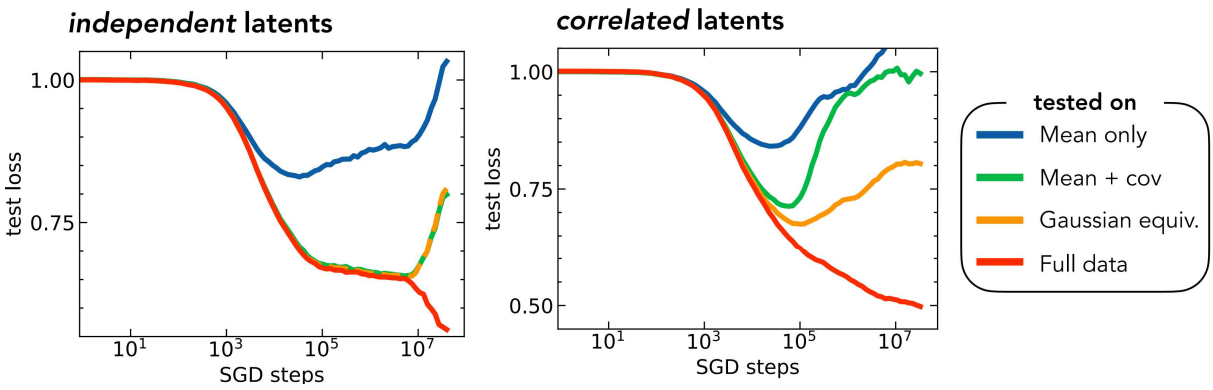

Figure 1.5: **Correlated latent variables speed up learning of neural networks. Left** Test error of a two-layer neural network trained on the mixed cumulant model (MCM) of eq. (3.1). The MCM is a binary classification tasks where the inputs in the two classes have a different mean, a different covariance, and different higher-order cumulants (HOCs). We show the test error on the full data set (red) and on several "censored" data sets: a test set where only the mean of the inputs is different in each class (blue) a test set where mean and covariance are different (green), and a Gaussian mixture that is fitted to the true data set (orange). The neural networks learn distributions of increasing complexity: initially, only the difference means matter, as the blue and red curves coincide; later, the network learns about differences at the level of the covariance, and finally at the level of higher-order cumulants. **Right** Same setup, but here the latent variables corresponding to the covariance and the cumulants of the inputs are correlated, leading to a significant speed-up of learning from HOCs (the red and orange line separate after $\gtrsim 10^4$ steps, rather than $\gtrsim 10^6$ steps). *Parameters:* $d = 128$, Network width $= 512$, ReLU activation function. More details in fig. 3.2.

algorithms, then, under the *low-degree conjecture* (conjecture 2.5), it is required to have $n > d^2$ to be able to solve the problem.

We finally experimentally investigate the performance of 2LNNs and random features on this model: looking at fig. 1.4 we can see that at linear sample complexity the classification accuracies do no better than random guess, while at quadratic sample complexity only the 2LNN manages to learn, showing separation with respect to random features.

Chapter 2 is based on [30], and is the result of a collaboration with Eszter Székely, Federica Gerace and Sebastian Goldt.

1.3.2 Neural network dynamics

In chapter 3 we start from the observation in fig. 1.3, deep networks seem to learn from non-Gaussian statistics faster than the quadratic sample complexity predicted in section 1.3.1. Hence, we will generalize the *spiked cumulant model* to better account for the complexity of real-world distributions, in which low and higher order statistics are not isolated as in the spiked cumulant model, but can interact. We do this in the *mixed cumulant model* via the introduction of spikes in the low order cumulants (as in spiked Wishart model). Then it is possible to model the interdependence of low and high order cumulants by introducing positive correlations in the latent variables corresponding to low and high-order spiked directions. Mathematically, we then prove that these latent correlations can speed up learning with neural networks (see proposition 3.4 and proposition 3.5).

We also illustrate experimentally in fig. 1.5 the result training a two-layer neural network on a classification task based on distinguishing samples of the *mixed cumulant model* from standard

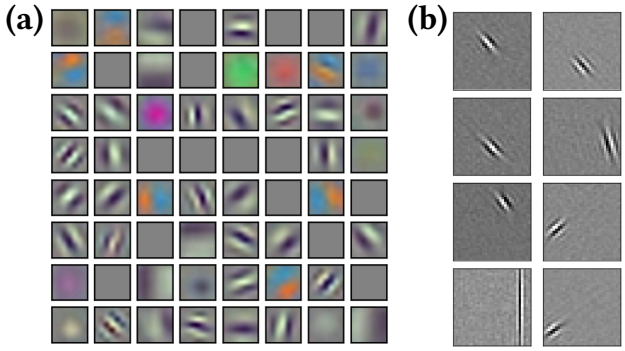

Figure 1.6: **Independent component analysis yields similar filters as deep convolutional neural networks. (a)** The first-layer convolutional filters extracted from an AlexNet trained on ImageNet show the hallmarks of Gabor filters [32], like orientation, localisation, and bandwidth-specificity. **(b)** Independent components found with online SGD on 100 000 64×64 patches extracted from ImageNet, showing more localized versions of Gabor filters.

Gaussian noise. We refer to the caption of fig. 1.5 for a detailed description of the experiment.

Chapter 3 is based on [31], in collaboration with Sebastian Goldt.

1.3.3 Independent components

In chapter 4 we turn to the analysis of an unsupervised method designed for the detection and retrieval of non-Gaussian directions in the data, the *independent component analysis* (ICA); which extract features from real data that closely resemble the ones learned autonomously by deep neural networks when trained on classification tasks on ImageNet fig. 1.6. We study the high-dimensional limit of the popular optimization algorithm FastICA, proving sub-optimality in terms of sample complexity. More precisely, in theorem 4.4, we show it requires a sample complexity of $n \gtrsim d^4$ in order to find the spiked direction after one step. There is a wide gap between this sample complexity and the optimal quadratic one discussed in section 1.3.1. Hence we conclude that FastICA, which works well in common low-dimensional setting, is unfit for the high-dimensional regime. We then turn to other algorithms, managing to speed up the optimization of ICA by employing online SGD, modified by adapting the *smoothing trick* introduced in Damian *et al.* [18], reaching the optimal quadratic sample complexity.

Next, we connect back to ImageNet data, verifying experimentally the presence of the search phase predicted by the theoretical results. By contrast, we observe that running FastICA in a subspace of fixed dimension $d = O_D(1)$ yields non-Gaussian directions in the D-dimensional image ambient space (section 4.4.1). We finally develop this insight with the introduction of a *subspace model* where we prove that strong recovery of a non-Gaussian direction is possible at linear sample complexity when running ICA in a suitable principal subspace (section 4.5)

Chapter 4 is based on [33] and [34], in collaboration with Fabiola Ricci and Sebastian Goldt.

1.4 Organization of the manuscript

The topics introduced are detailed in chapters 2-4. Each chapter has its own appendix, for additional proof details, background material or specifics of the experiments. Finally, in chapter 5 we discuss the results presented and also consider possible extensions towards generative diffusion models.

In appendix A we collected theoretical background material on Hermite polynomials, that are needed for the analyses in the main chapters of the thesis. Along with classical properties of the polynomials, some more technical and novel lemmas are presented. The content of this appendix was developed in collaboration with Fabiola Ricci.

Chapter 2

Algorithmic hardness

2.1 Introduction

Discovering statistical patterns in high-dimensional data sets is the key objective of machine learning. In a classification task, the differences between inputs in different classes arise at different statistical levels of the inputs: two different classes of images will usually have different means, different covariances, and different higher-order cumulants (HOCs), which describe the non-Gaussian part of the correlations between three or more pixels. While differences in the mean and covariance allow for rudimentary classification, Refinetti *et al.* [28] (that was recalled in fig. 1.3) recently highlighted the importance of HOCs for the performance of neural networks: when they removed the HOCs per class of the CIFAR10 training set, the test accuracy of various deep neural networks dropped by up to 65 percentage points. The importance of higher-order cumulants (HOCs) for classification in general and the performance of neural networks in particular raises some fundamental questions: what are the fundamental limits of learning from HOCs, i.e. how many samples n ("sample complexity") are required to extract information from the HOCs of a data set? How many samples are required when using a *tractable* algorithm? And how do neural networks and other machine learning methods like random features compare to those fundamental limits?

In this chapter, we study these questions by analysing a series of binary classification tasks. In one class, inputs $x \in \mathbb{R}^d$ are drawn from a normal distribution with zero mean and identity covariance. These inputs are therefore isotropic: the distribution of the high-dimensional points projected along a unit vector in any direction in $\mathbb{R}^d$ is a standard Gaussian distribution. Furthermore, all the higher-order cumulants (of order $p \geq 3$) of the inputs are identically zero. Inputs in the second class are also isotropic, except for one special direction $u \in \mathbb{R}^d$ in which their distribution is different. This direction u is often called a "spike", and it can be encoded in different cumulants: for example, we could "spike" the covariance by drawing inputs from a Gaussian with mean zero and covariance $\mathbb{1} + \beta uu^\top$; the signal-to-noise ratio $\beta > 0$ would then control the variance of $\lambda = \langle u, x \rangle$. Likewise, we could spike a higher-order cumulant of the distribution and ask: what is the minimal number of samples required for a neural network trained with SGD to distinguish between the two input classes? This simple task serves as a proxy for more generic tasks: a neural network cannot be able to *extract* information from a given cumulant if it cannot even recognise that it is different from an isotropic one.

We can obtain the fundamental limits of detecting spikes at different levels of the cumulant hierarchy by considering the hypothesis test between a null hypothesis (the isotropic multivariate normal distribution with zero mean and identity covariance) and an alternative "spiked" distribution. We can then compare the sample complexity of neural networks to the number of samples necessary

to distinguish the two distributions using unlimited computational power, or efficiently using algorithms that run in polynomial time. A second natural comparison for neural networks are random features or kernel methods. Since the discovery of the neural tangent kernel [35], and the practical success of kernels derived from neural networks [36, 37], there has been intense interest in establishing the advantage of neural networks with *learnt* feature maps over classical methods with *fixed* feature maps, like kernel machines. The role of higher-order correlations for the relative advantage of these methods has not been studied yet.

In the following, we first introduce some fundamental notions around hypothesis tests and in particular the low-degree method [38–41], which will be a key tool in our analysis, using the classic spiked Wishart model for sample covariance matrices [42, 43]. For spiked cumulants, the existing literature (see section 2.1.1 for a complete discussion) establishes the presence of a wide statistical-to-computational gap in this model. Our **main contributions** are then as follows:

- We deepen the understanding of the statistical-to-computational gap for learning from higher-order correlations on the statistical side by showing that at *unbounded computational power*, a number of samples *linear* in the input dimension is required to reach *statistical distinguishability*. We prove this by explicitly computing the norm of the *likelihood ratio*, see lemma 2.2.
- On the algorithmic side, SQ bounds and previous low-degree analyses [29, 44], showed that the sample complexity of learning from HOCs is instead *quadratic* for a wide class of polynomial-time algorithms (section 2.3.2). Here, we provide a different, more direct proof of such a bound.
- Using these fundamental limits on learning from HOCs as benchmarks, we show numerically that neural networks learn the mixed cumulant model efficiently, while random features do not, revealing a large separation between the two methods (sections 2.4.1 and 2.4.2).
- We finally show numerically that the distinguishability in a simple model for images [45] is precipitated by a cross-over in the behaviour of the higher-order cumulants of the inputs (section 2.4.3).

2.1.1 Further related work

Detecting spikes in high-dimensional data There is an enormous literature on statistical-to-computational gaps in the detection and estimation of variously structured principal components of high-dimensional data sets. This problem has been studied for the Wigner and Wishart ensembles of random matrices [42, 46–56] and for spiked tensors [57–65]. For comprehensive reviews of the topic, see refs. [66–69]. While the samples in these models are often non-Gaussian, depending on the prior distribution over the spike u, the spike appears already at the level of the covariance of the inputs. Here, we instead study a high-dimensional data model akin to the one used by Wang & Lu [70] to study online independent component analysis (ICA). This data model can be interpreted as having an additional whitening step to pre-process the inputs, which is a common pre-processing step in ICA [71], hence inputs have identity covariance even when cumulants of order $p \geq 3$ are spiked. Wang & Lu [70] proved the existence of a scaling limit for online ICA, but did not consider the sample complexity of recovering the spike/distinguishing the distributions, which is the focus of this chapter.

NGCA, Gaussian pancakes and low-degree polynomials Related models have been introduced under the name of *Non-Gaussian Component Analysis* (NGCA) [72], and studied from

the point of view of *statistical query* (SQ) complexity in a sequence of papers [21, 73–77]. In particular [44] nicknames *Gaussian pancakes* a class of models that includes the *spiked cumulant model* presented here. The SQ bounds found in [44] and refined in the subsequent works (see Diakonikolas *et al.* [78] and references therein) could be used, together with SQ-LDLR equivalence [79], to provide estimates on low-degree polynomials. Finally, [29] proves very general bounds on LDLR norm; our theorem 2.6 provides an alternative derivation of these bounds, in a setting that is closer to the experiments in section 4.

Separation between neural networks and random features The discovery of the neural tangent kernel by Jacot *et al.* [35] and the flurry of results on linearised neural networks [35, 36, 80–86] has triggered a new wave of interest in the differences between what neural networks and kernel methods can learn efficiently. While statistical separation results have been well-known for a long time [87], recent work focused on understanding the differences between random features and neural networks *trained by gradient descent* both with wide hidden layers [37, 88–92] or even with just a few hidden neurons [93, 94]. At the heart of the data models in all of these theoretical models is a hidden, low-dimensional structure in the task, either in input space (for mixture classification) or in the form of single- or many-index target functions. The impact of higher-order input correlations in mixture classification tasks on the separation between random features and neural networks has not been directly studied yet.

Reproducibility We provide code for all of our experiments, including routines to generate the various synthetic data sets we discuss, on GitHub https://github.com/eszter137/data-driven-separation.

2.2 The data models

Throughout the chapter, we consider binary classification tasks where high-dimensional inputs $x^\mu = (x_i^\mu) \in \mathbb{R}^d$ have labels $y^\mu = \pm 1$. The total number of training samples is $2n$, i.e. we have n samples per class. For the class $y^\mu = -1$, inputs $x^\mu = z^\mu$, where $z^\mu \sim_{\text{iid}} \mathcal{N}(0, \mathbb{1}_d)$ and $\mathcal{N}$ denotes the normal distribution. For the class $y^\mu = 1$ instead, we consider "spiked" input models as follows.

The Gaussian case The simplest spiked model is the spiked Wishart model from random matrix theory [42, 43], in which

$$x^\mu = \sqrt{\frac{\beta}{d}} g^\mu u + z^\mu, \quad g^\mu \sim_{\text{iid}} \mathcal{N}(0, 1), \tag{2.1}$$

where $u = (u_i)$ is a d-dimensional vector with norm $\|u\| = \sqrt{d}$ whose elements are drawn element-wise i.i.d. according to some probability distribution $\mathcal{P}_u$. In this model, inputs are Gaussian and indistinguishable from white noise except in the direction of the "spike" u, where they have variance $1 + \beta$, where $\beta > 0$ is the *signal-to-noise ratio*. By construction, all higher-order cumulants are zero.

The spiked cumulant model To study the impact of HOCs, we draw inputs in the "spiked" class from the data model used by Wang & Lu [70] to study online independent component analysis (ICA). First, we sample inputs

$$\tilde{x}^\mu = \sqrt{\frac{\beta}{d}} g^\mu u + z^\mu, \qquad g^\mu \sim_{\text{i.i.d.}} p_g, \tag{2.2}$$

as in the Wishart model, but crucially sample the latent variables g^μ from some non-Gaussian distribution $p_g(g^\mu)$, say, the Rademacher distribution; see assumption 2.1 for a precise statement. For any non-Gaussian p_g, the resulting inputs $\tilde{x}^\mu$ have a non-trivial fourth-order cumulant proportional to $\kappa_4^g u^{\otimes 4}$, where $\kappa_4^g \equiv \mathbb{E}\,(g^\mu)^4 - 3\mathbb{E}\,(g^\mu)^2$ is the fourth cumulant of the distribution of g^μ. We fix the mean and variance of p_g to be zero and one, respectively, so the covariance of inputs has a covariance matrix $\Sigma = \mathbb{1}_d + {}^{\beta}/_{d} uu^\top$. To avoid trivial detection of the spike from the covariance, the key ingredient of the spiked cumulant model is that we whiten the inputs in that class, so that the inputs are finally given by

$$x^\mu = S\tilde{x}^\mu, \qquad S = \mathbb{1} - \frac{\beta}{1+\beta+\sqrt{1+\beta}}\frac{uu^\top}{d}, \tag{2.3}$$

with the whitening matrix S (see section 2.B.3.3). Hence inputs x^μ are isotropic Gaussians in all directions except u, where they are a weighted sum of g^μ and $\langle z, u\rangle$. The whitening therefore changes the interpretation of β: rather than being a signal-to-noise ratio, as in the spiked Wishart model, here β controls the quadratic interpolation between the distributions of g^μ and z in the direction of u (see eq. (2.55) in the appendix). This leaves us with a data set where inputs in both classes have an average covariance matrix that is the identity, which means that PCA or linear neural networks [95–98] cannot detect any difference between the two classes.

2.3 How many samples do we need to learn?

Given a data set sampled from the spiked cumulant model, we can now ask: how many samples does a statistician need to reliably detect whether inputs are Gaussian or not, i.e. whether HOCs are spiked or not? This is equivalent to the hypothesis test between n i.i.d. samples of the isotropic normal distribution in $\mathbb{R}^d$ as the null hypothesis $\mathbb{Q}_{n,d}$, and n i.i.d. samples of the spiked cumulant model eq. (2.3) as the alternative hypothesis $\mathbb{P}_{n,d}$. In section 2.3.1 we will first consider the problem from a statistical point of view, assuming to have *unbounded computational power* and no restrictions on the distinguishing algorithms. Then, in section 2.3.2 we will use the *low-degree method* to understand how the picture changes when we restrict to algorithms whose running time is at most polynomial in the space dimension d.

2.3.1 Statistical distinguishability: LR analysis

Recall the notion of *strong asymptotic distinguishability*: two sequences of probability measures are strongly distinguishable if it is possible to design statistical tests that can classify correctly which of the two distributions a sample was drawn from with probabilities of type I and II errors that converge to 0 (see section 2.B.2 for the precise definition). Using this definition of distinguishability, we will ask what is the *statistical sample complexity exponent*, i.e. the minimum θ such that in the high-dimensional limit $d \to \infty$, if $n \asymp d^\theta$, then $\mathbb{P}_{n,d}$ and $\mathbb{Q}_{n,d}$ are strongly distinguishable (with no constraints on the complexity of statistical tests).

A useful quantity to consider is the *likelihood ratio* (LR) of probability measures, which is defined as

$$L(x) := \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}}(x). \tag{2.4}$$

Computing the LR norm $||L||^2 := \mathbb{E}_{\mathbb{Q}}[L^2]$ is an excellent tool to probe for distinguishability: if $(\mathbb{P}_{n,d})$ and $(\mathbb{Q}_{n,d})$ are strongly distinguishable, then $\|L_{n,d}\| \to \infty$ (this is the well-known *second moment method for distinguishability*, see proposition 2.9 in the appendix for the precise statement). In the

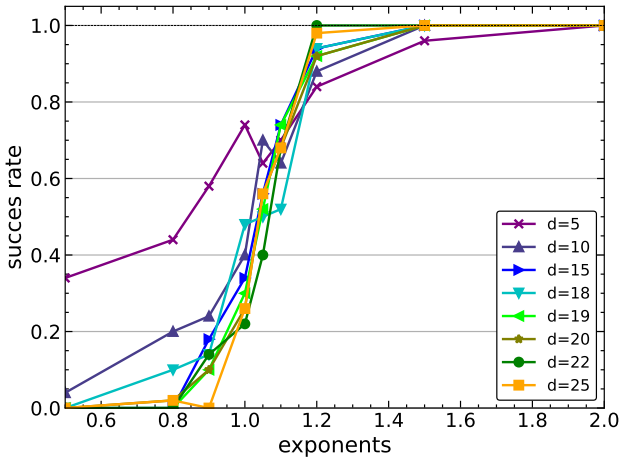

Figure 2.1: **The performance of an exhaustive-search algorithm corroborates the presence of a phase transition for $\theta = 1$, as suggested by lemma 2.2.** Success rate of an exponential-time search algorithm over all the possible spikes in the d-hypercube as a function of the exponent θ that quantifies as $n = d^\theta$ the samples used in the log-likelihood test (2.12), in the $g \sim$Radem(1/2) case.

following we will apply this method, finding a formula for the LR norm and then study its limit as a function of θ. Here and throughout, we will denote the data matrix by $\underline{x} = (x^\mu)^\top_{1,\dots,n}$; in general matrices of size $n \times d$ will be denoted with underlined letters; see section 2.B.1 for a complete summary of our notation. We will use Hermite polynomials, denoted by $(h_k)_k$, see appendix A for details. We assume that the spike u is drawn from a known prior $\mathcal{P}(u)$, and that the scalar latent variables $(g^\mu)_{\mu=1,\dots,n}$ are drawn i.i.d. from a distribution p_g with the following properties:

Assumption 2.1 (Assumption on latent variables g^μ). We assume that the one-dimensional probability distribution of the non-Gaussian random variable $p_g(g)$ is an even distribution, $p_g(g = \mathrm{d}x) = p_g(-g = -\mathrm{d}x)$, with mean 0 and variance 1, and that it satisfies the following requirement on the growth of its Hermite coefficients:

$$\mathbb{E}\left[h_m(g)\right] \leq \Lambda^m m! \tag{2.5}$$

where $\Lambda > 0$ is a positive constant that does not depend on m. Finally, we assume that p_g has tails that cannot go to 0 slower than a standard Gaussian, $\mathbb{E}[\exp(g^2/2)] < +\infty$.

A detailed discussion of these assumptions can be found in section 2.B.3, as well as a proof that they are satisfied by a wide class of distributions including all the compactly supported distributions with mean 0 and variance 1 (some concrete examples are p_g=Rademacher(1/2) or $p_g = \mathrm{Unif}(-\sqrt{3}, \sqrt{3})$).

Under these assumption we can compute the following formula for the LR norm.

Lemma 2.2. *Suppose that u has i.i.d. Rademacher prior and that the non-Gaussian distribution p_g satisfies assumption 2.1. Then the norm of the total LR is given by*

$$\|L_{n,d}\|^2 = \sum_{j=0}^{d} \binom{d}{j} \frac{1}{2^d} f\left(\beta, \frac{2j}{d} - 1\right)^n, \tag{2.6}$$

where f is defined as the following average over two independent replicas $g_u, g_v \sim g$ of g:

$$f(\beta, \lambda) := \mathop{\mathbb{E}}_{g_u, g_v} \left[\frac{1+\beta}{\sqrt{(1+\beta)^2 - \beta^2\lambda^2}} e^{-\frac{(1+\beta)\left((1+\beta)(g_u^2+g_v^2) - 2\beta(g_u g_v)\lambda\right)}{2(1+\beta)^2 - 2\beta^2\lambda^2} + \frac{g_u^2+g_v^2}{2}} \right] \tag{2.7}$$

We provide a streamlined version of the proof of lemma 2.2 here, leaving all the detailed computations in section 2.B.4.

Proof. Since the samples are independent, the total LR factorises as

$$L(\underline{y}) = \mathop{\mathbb{E}}_{u}\left[\prod_{\mu=1}^{n} l(y^\mu|u)\right], \tag{2.8}$$

where the sample-wise likelihood ratio is

$$l(y|u) = \frac{p_x(y|u)}{p_z(y)} = \mathop{\mathbb{E}}_{g\sim p_g}\left[\sqrt{1+\beta}\exp\left(-\frac{1+\beta}{2}\left(g - \sqrt{\frac{\beta}{(1+\beta)d}}y\cdot u\right)^2 + \frac{g^2}{2}\right)\right]. \tag{2.9}$$

To compute the norm of the LR, we consider two independent replicas of the spike, u and v, to get

$$\|L_{n,d}\|^2 = \mathop{\mathbb{E}}_{\underline{y}\sim\mathbb{Q}}\left[\mathop{\mathbb{E}}_{u}\left[\prod_{\mu=1}^{n} l(y^\mu|u)\right]\mathop{\mathbb{E}}_{v}\left[\prod_{\mu=1}^{n} l(y^\mu|v)\right]\right]. \tag{2.10}$$

The hard part now is to simplify the high-dimensional integrals in this expression, it can be done because the integrand is almost completely symmetric, and the only asymmetries lie on the subspace spanned by u and v.

$$\|L_{n,d}\|^2 = \mathop{\mathbb{E}}_{u,v}\left[\mathop{\mathbb{E}}_{g_u,g_v}\left[\frac{1+\beta}{\sqrt{(1+\beta)^2-\beta^2\left(\frac{u\cdot v}{d}\right)^2}}e^{-\frac{(1+\beta)\left((1+\beta)(g_u^2+g_v^2)-2\beta(g_u g_v)\left(\frac{u\cdot v}{d}\right)\right)}{2(1+\beta)^2-2\beta^2\left(\frac{u\cdot v}{d}\right)^2}+\frac{g_u^2+g_v^2}{2}}\right]^n\right]$$

Note that the integrand depends on u and v only trough $\frac{u\cdot v}{d} =: \lambda$. So, using that the prior on u, v is i.i.d. Rademacher, the outer expectation can be transformed in a one dimensional expectation over λ, leading to (2.6). □

The formula has key consequences in two directions.

The case $\theta < 1$

Lemma 2.3. *If p_g satisfies assumptions 2.1 and $\theta < 1$, then there exists C such that:*

$$||L_{n,d}|| \le C \qquad \forall n, d$$

Proof. We first prove that we have that

$$f(\beta,\lambda) \le 1 + C_\beta\lambda^2, \qquad \forall\ \lambda \in [-1,1] \tag{2.11}$$

To do it, first note that $f\beta, 0) = 1$, and $f(\beta,\cdot)$ is bounded on $[-1,1]$ thanks to the sub-Gaussianity of g:

$$\begin{aligned}
|f(\beta,\lambda)| &\le \mathbb{E}_{g_u,g_v}\left[e^{\frac{1}{2}\left(g_u^2+g_v^2\right)}\right]\sup_{g_u,g_v}\left(\frac{1+\beta}{\sqrt{(1+\beta)^2-\beta^2\lambda^2}}e^{-\frac{(1+\beta)\left((1+\beta)(g_u^2+g_v^2)-2\beta(g_u g_v)\lambda\right)}{2(1+\beta)^2-2\beta^2\lambda^2}}\right)\\
&\le C\frac{1+\beta}{\sqrt{(1+\beta)^2-\beta^2\lambda^2}}\\
&\le C_1\left(1+C_2\lambda^2\right)
\end{aligned}$$

So we just need to prove that, up to changes on C_2, we can take $C_1 = 1$. To do that it is sufficient to study the behavior around $\lambda = 0$:

$$\frac{\partial}{\partial\lambda} f(\beta, 0) = \mathbb{E}\left[\frac{\beta}{1+\beta} g_u g_v\right] = 0$$

So there is a neighborhood of 0 such that $f(\beta, 0) = 1 + C\lambda^2 + o(\lambda^2)$. Hence we can take a suitable constant C_β so that eq. (2.11) holds.

Now we can apply eq. (2.11) to (2.6), to get

$$\begin{aligned}
||L_{n,d}||^2 &\leq \sum_{j=0}^{d} \binom{d}{j} \frac{1}{2^d} \left(1 + C_\beta \left(\frac{2j}{d} - 1\right)^2\right)^n \\
&= \sum_{j=0}^{d} \binom{d}{j} \frac{1}{2^d} \sum_{k=0}^{n} \binom{n}{k} C_\beta^k \left(\frac{2j}{d} - 1\right)^{2k} \\
&= \sum_{k=0}^{n} \binom{n}{k} \left(\frac{C_\beta}{d^2}\right)^k \mathbb{E}\left[Y_d^{2k}\right]
\end{aligned}$$

Where Y_d is a random variable distributed as the sum of d independent Rademacher with parameter $1/2$. So we can apply the central limit theorem on $\frac{Y_d}{\sqrt{d}}$ to get that $\frac{\mathbb{E}[Y_d^{2k}]}{d^k} \to (2k-1)!!$. Moreover, the moment generating functions of the Y_d are $\psi_d(t) = \cosh\left(t/\sqrt{d}\right)^d$, since $\cosh(t) \leq e^{t^2/2}$ we have that $\psi_d(t) \leq e^{t^2/2} = \psi_{gaussian}(t)$, for all $d \in \mathbb{N}$, hence by Khintchine's inequality (corollary 2.6.4 in [99]) the moments can be estimated by the Gaussian ones, and we get:

$$\begin{aligned}
||L_{n,d}||^2 &\leq C \sum_{k=0}^{n} \binom{n}{k} \left(\frac{C_\beta}{d}\right)^k (2k-1)!! \\
&\leq C_1 \sum_{k=0}^{n} \left(\frac{C_2 n}{d}\right)^k
\end{aligned}$$

where C_1 and C_2 are constants independent of n, d. So now we can substitute that $n \asymp d^\theta$, and since $\theta < 1$ the series converges for all d, hence the LR norm is bounded. □

So lemma 2.3 states that the LR norm is bounded for $\theta < 1$, which confirms that below linear sample complexity, *strong distinguishability is impossible.*

The case $\theta = 1$ On the other hand, eq. (2.6) implies that whenever there exists $\tilde{\lambda}$ such that $f(\beta, \tilde{\lambda}) > 1$, we find that $\|L_{n,d}\| \geq \frac{f(\beta,\tilde{\lambda})^n}{2^d}$. Thus, the LR norm diverges as soon as n grows as $n \asymp d^\theta$ with any $\theta > 1$. In section 2.B.4.1 we detail as an example the case in which $g \sim$Rademacher$(1/2)$ where the norm $\|L_{n,d}\|$ even diverges for $\theta = 1$ and $d \asymp \gamma n$ for some $\gamma > 0$. So, besides the intrinsic value of providing an exact formula for the LR of the spiked cumulant model, lemma 2.2 implies the presence of a phase transition at $\theta = 1$ for the *strong statistical distinguishability.*

A complementary approach that substantiates the presence of the statistical phase transition at $\theta = 1$ can be seen in fig. 2.1, where we perform a maximum log-likelihood test along $u \cdot x^\mu$ for *all* the possible spikes u in the d-dimensional hypercube using the formula for the LR conditioned on the spike,

$$\sum_{\mu=1}^{n} \log\left(\frac{p_x(x^\mu|u)}{p_z(x^\mu)}\right) = \sum_{\mu=1}^{n} \log \mathop{\mathbb{E}}_{g} \sqrt{1+\beta} \exp\left(-\frac{1+\beta}{2}\left(g - \sqrt{\frac{\beta}{(1+\beta)d}} x^\mu \cdot u\right)^2 + \frac{g^2}{2}\right), \qquad (2.12)$$

(see eq. (2.58) in section 2.B.4 for the derivation of this equation) and output the most likely u. Note that due to the exponential complexity in d of the algorithm, it is unfeasible to reach large values for this parameter. However, even at small d values, the success rate for retrieving the correct spike has a very steep increase at around $\theta = 1$, as predicted by our analysis of the LR norm in lemma 2.2.

2.3.2 Computational distinguishability: LDLR analysis

We now compare the statistical threshold of section 2.3.1 with the *computational sample complexity exponent* that quantifies the sample complexity of detecting non-Gaussianity with an efficient algorithm that runs in a time that is polynomial in the input dimension. The algorithmic sample complexity can be analysed rigorously for a wide class of algorithms using the *low-degree method* [38–41, 69]. The low-degree method arose from the study of the sum-of-squares hierarchy [38] and rose to prominence when Hopkins & Steurer [40] demonstrated that the method can capture the Kesten–Stigum threshold for community detection in the stochastic block model [100–102]. In the case of hypothesis testing, the key quantity is the *low-degree likelihood ratio* (LDLR) [41, 69].

Definition 2.4 (Low-degree likelihood ratio (LDLR))**.** Let $D \geq 0$ be an integer. The low-degree likelihood ratio of degree D is defined as

$$L^{\leq D} := \mathcal{P}^{\leq D} L \tag{2.13}$$

where $\mathcal{P}^{\leq D}$ projects L onto the space of polynomials of degree up to D, parallel to the Hilbert space structure defined by the scalar product $\langle f, g \rangle_{L^2(\Omega,\mathbb{Q})} := \mathbb{E}_{x\sim\mathbb{Q}}[f(x)g(x)]$.

The idea of this method is that among degree-D polynomials, $L^{\leq D}$ captures optimally the difference between $\mathbb{P}$ and $\mathbb{Q}$, and this difference can be quantified by the norm $\left\|L^{\leq D}\right\| = \left\|L^{\leq D}\right\|_{L^2(\Omega_n,\mathbb{Q}_n)}$. Hence in analogy to the *second moment method* used in section 2.3.1, we can expect low-degree polynomials to be able to distinguish $\underline{\mathbb{P}}$ from $\underline{\mathbb{Q}}$ only when $\left\|L_n^{\leq D(n)}\right\| \underset{n}{\to} \infty$, where $D(n)$ is a monotone sequence diverging with n. Indeed, the following (informal) conjecture from Hopkins [41] states that this criterion is valid not only for polynomial tests, but for all polynomial-time algorithms:

Conjecture 2.5. *For two sequences of measures $\mathbb{Q}_N, \mathbb{P}_N$ indexed by N, suppose that (i) $\mathbb{Q}_N$ is a product measure; (ii) $\mathbb{P}_N$ is* sufficiently *symmetric with respect to permutations of its coordinates; and (iii) $\mathbb{P}_N$ is* robust *with respect to perturbations with small amount of random noise. If $\|L_N^{\leq D}\| = O(1)$ as $N \to \infty$ and for $D \geq (\log N)^{1+\varepsilon}$, for some $\varepsilon > 0$, then there is no polynomial-time algorithm that strongly distinguishes the distributions $\mathbb{Q}$ and $\mathbb{P}$.*

Even though this conjecture is still not proved in general, its empirical confirmation on many benchmark problems has made it an important tool to probe questions of computational complexity, see theorem 2.11 in the appendix for a simple example.

We will now compute LDLR norm estimates for the spiked cumulant model, so that the application of conjecture 2.5 will help to understand the *computational sample complexity* of this model.

Theorem 2.6 (LDLR for spiked cumulant model)**.** *Suppose that $(u_i)_{i=1,\dots,d}$ are drawn i.i.d. from the symmetric Rademacher distribution. If the non-Gaussian distribution p_g satisfies assumption 2.1, then the following lower and upper bounds hold:*

- *Let $D \in \mathbb{N}$ such that $D/4 \leq n$, then:*

$$\left\|L_{n,d}^{\leq D}\right\|^2 \geq \sum_{m=0}^{\lfloor D/4 \rfloor} \binom{n}{m} \binom{d+1}{2}^m \left(\frac{\beta^2 \kappa_4^g}{\sqrt{4!} d^2 (1+\beta)^2}\right)^{2m}. \tag{2.14}$$

  where κ_4^g is the fourth-order cumulant of g.

- *Conversely, for any D, n, d:*

$$\left\|L_{n,d}^{\leq D}\right\|^2 \leq 1 + \sum_{m=1}^{D} \frac{C_m}{d^m} \sup_{k \leq m} \left(\mathbb{E}\left[h_k(g)\right]^{2m/k}\right) \binom{n}{\lfloor m/4 \rfloor} \binom{d}{\lfloor m/2 \rfloor} \tag{2.15}$$

  where $C_m := \left(\frac{\beta}{1+\beta}\right)^m \binom{\lfloor m/4 \rfloor \lfloor m/2 \rfloor + m - 1}{m}$.

Corollary 2.7 (Asymptotics of LDLR bounds)**.** *Assume the hypotheses of theorem 2.6. Let $0 < \varepsilon < 1$ and assume $D(n) \asymp \log^{1+\varepsilon}(n)$. Take $n, d \to \infty$, with the scaling $n \asymp d^\theta$ for $\theta > 0$. Estimate* (2.14) *implies that for n, d large enough, the following lower bound holds:*

$$\left\|L_{n,d}^{\leq D(n)}\right\|^2 \geq \left(\frac{1}{\lfloor D(n)/4 \rfloor} \left(\frac{\beta^2 \kappa_4^g}{(1+\beta)^2}\right)^2 \frac{n}{d^2}\right)^{\lfloor D(n)/4 \rfloor} \tag{2.16}$$

Conversely, (2.15) *leads to:*

$$\left\|L_{n,d}^{\leq D(n)}\right\|^2 \leq 1 + \sum_{m=1}^{D(n)} \left(\frac{\Lambda^2 \beta}{1+\beta}\right)^m m^{4m} \left(\frac{n}{d^2}\right)^{m/4} \tag{2.17}$$

Taken together, (2.16) *and* (2.17) *imply the presence of a critical regime for $\theta_c = 2$, and describe the behavior of $\left\|L_{n,d}^{\leq D}\right\|$ for all $\theta \neq \theta_c$*

$$\lim_{n,d \to \infty} \left\|L_{n,d}^{\leq D(n)}\right\| = \begin{cases} 1 & 0 < \theta < 2 \\ +\infty & \theta > 2 \end{cases} \tag{2.18}$$

This theorem could be derived with different constants from lemma 26 and proposition 8 in Dudeja & Hsu [29]. Here we derive a more direct argument tailored to the *spiked cumulant model*.

2.3.3 Proof of theorem 2.6 and corollary 2.7

In the following we will present the proofs of 2.6 and 2.7. Here we give a streamlined prof, with all the key ideas, leaving the most tedious and long details in 2.B.5.

The starting point of the argument is the observation that since the null hypothesis is white Gaussian noise, the space $L^2(\mathbb{R}^{nd}, \langle \cdot, \cdot \rangle)$ has an explicit orthogonal basis in the set of multivariate Hermite polynomials $(H_{\underline{\boldsymbol{\alpha}}})$. The matrix-multi-index $\underline{\boldsymbol{\alpha}} \in \mathbb{N}^{nd}$ denotes the degree of the Hermite polynomial which is computed for each entry of the data matrix; see section 2.B.1 for a detailed explanation. We can then expand the LDLR norm in this basis (see lemma 2.13 for more details) to write:

$$\|L_{n,d}^{\leq D}\|^2 = \sum_{|\underline{\boldsymbol{\alpha}}| \leq D} \frac{\langle L, H_{\underline{\boldsymbol{\alpha}}} \rangle^2}{\underline{\boldsymbol{\alpha}}!} = \sum_{|\underline{\boldsymbol{\alpha}}| \leq D} \frac{1}{\underline{\boldsymbol{\alpha}}!} \mathop{\mathbb{E}}_{\underline{x} \sim \underline{\mathbb{P}}} [H_{\underline{\boldsymbol{\alpha}}}(\underline{x})]^2 \tag{2.19}$$

From now on the two bounds are treated separately.

Lower bound The idea is to bound the LDLR from below by computing the sum 2.19 only over a restricted set of addends $\mathcal{I}_m$ which we can compute explicitly. In particular, we consider the set $\mathcal{I}_m$ of all the polynomials with degree $4m$ in the data matrix, which are at most of order 4 in each individual sample x^μ. Then we use that the expectation of such Hermite polynomials conditioned on u can be split into m integrals of degree-4 Hermite polynomials in d variables. In this way we can use our knowledge of the fourth-order cumulant of x to compute those expectations (see 2.82), so we find that

$$\|L_{n,d}^{\leq D}\|^2 \geq \sum_{m}^{\lfloor D/4 \rfloor} \sum_{|\underline{\boldsymbol{\alpha}}| \in \mathcal{I}_m} \frac{1}{\underline{\boldsymbol{\alpha}}!} \mathop{\mathbb{E}}_{\underline{x} \sim \underline{\mathbb{P}}} [H_{\underline{\boldsymbol{\alpha}}}(\underline{x})]^2 \geq \sum_{m}^{\lfloor D/4 \rfloor} \sum_{\underline{\boldsymbol{\alpha}} \in \mathcal{I}_m} \left(\frac{\beta^2 \kappa_4^g}{\sqrt{4!} d^2 (1+\beta)^2} \right)^{2m} . \tag{2.20}$$

Thanks to this manipulation, the bound does not depend on $\underline{\boldsymbol{\alpha}}$ explicitly, so we can complete the bound by estimating the cardinality of $\mathcal{I}_m$. Thanks to the fact that we have n i.i.d. copies of variable x at disposal, the class of polynomials $\mathcal{I}_m$ has size that grows at least as $\binom{n}{m}\binom{d+1}{2}^m$, allowing to reach the lower bound in the statement.

Upper bound The first important part is the rewriting of the expectations of Hermite polynomials.

Lemma 2.8. *Let $x = S\left(\sqrt{\beta/d} g u + z\right)$ be a spiked cumulant random variable. Then for any $\boldsymbol{\alpha} \in \mathbb{N}^d$, with $|\boldsymbol{\alpha}| = m$, we have that:*

$$\langle l(\cdot|u), H_{\boldsymbol{\alpha}} \rangle = \mathop{\mathbb{E}}_{x \sim \mathbb{P}(\cdot|u)} [H_{\boldsymbol{\alpha}}(x)] = \frac{T_{m,g}}{d^{m/2}} u^{\boldsymbol{\alpha}} \tag{2.21}$$

where $T_{m,g}$ is a coefficient defined as:

$$T_{m,g} = \left(\frac{\beta}{1+\beta} \right)^{m/2} \mathbb{E}\left[h_m(g) \right] \tag{2.22}$$

Proof. It is a direct computation using lemma A.8, and properties of Hermite polynomials, for all the computations see section 2.B.5. □

Using the fact that inputs are sampled i.i.d. from $\mathbb{P}(\cdot|u)$ and substituting this result into 2.19, we find that

$$\|L_{n,d}^{\leq D}\|^2 = \sum_{|\underline{\boldsymbol{\alpha}}| \leq D} \frac{\left(\prod_{\mu=1}^n T_{|\boldsymbol{\alpha}^\mu|,g} \right)^2}{\underline{\boldsymbol{\alpha}}! d^{|\underline{\boldsymbol{\alpha}}|}} \mathop{\mathbb{E}}_{u \sim \mathcal{P}(u)} [u^{\underline{\boldsymbol{\alpha}}}]^2 . \tag{2.23}$$

To obtain an upper bound, we will show that many addends in this sum are equal to zero and estimate the remaining ones.

For each $m \in \mathbb{N}$, we can denote $\mathcal{A}_m$ the set of matrix-multi-indices $|\underline{\boldsymbol{\alpha}}| = m$ that give non-zero contributions in 2.23, so that we can write:

$$\|L_{n,d}^{\leq D}\|^2 = \sum_{m=0}^{D} \sum_{\underline{\boldsymbol{\alpha}} \in \mathcal{A}_m} \frac{\left(\prod_{\mu=1}^n T_{|\boldsymbol{\alpha}^\mu|,g} \right)^2}{\underline{\boldsymbol{\alpha}}! d^m} \tag{2.24}$$

Now we can use 2.22 to bound the inner terms so that they depend on $\underline{\boldsymbol{\alpha}}$ only through the norm $|\underline{\boldsymbol{\alpha}}| = m$ (cf. 2.99 for details):

$$\begin{aligned} \|L_{n,d}^{\leq D}\|^2 &\leq \sum_{m=0}^{D} \sum_{\underline{\boldsymbol{\alpha}} \in \mathcal{A}_m} \frac{1}{d^m} \left(\frac{\beta}{1+\beta} \right)^m \sup_{k \leq m} \left(\mathbb{E}\left[h_k(g) \right]^{2m/k} \right) \\ &= \sum_{m=0}^{D} \frac{1}{d^m} \left(\frac{\beta}{1+\beta} \right)^m \sup_{k \leq m} \left(\mathbb{E}\left[h_k(g) \right]^{2m/k} \right) \#\mathcal{A}_m \end{aligned} \tag{2.25}$$

The cardinality of $\mathcal{A}_m$ is 1 in case $m = 0$ (which leads to the addend 1 in 2.15). If $m > 0$ it can be bounded considering that on the one hand, since we chose a Rademacher prior over the elements of the spike u_i, the expectation over the prior yields either 1 or 0 depending on whether the exponent of each u_i is even or odd, hence $\underline{\boldsymbol{\alpha}} \in \mathcal{A}_m$ implies that the sum over its columns is even. On the other hand, the whitening of the data means that $T_{|\boldsymbol{\alpha}|,g} = 0$ for $0 < |\boldsymbol{\alpha}| < 4$ (as proved in lemma 2.8), hence $\underline{\boldsymbol{\alpha}} \in \mathcal{A}_m$ implies that the sum over its rows is either 0 or ≥ 4. Summing up, we have the inclusion $\mathcal{A}_m \subseteq \tilde{\mathcal{A}}_m$, with

$$\tilde{\mathcal{A}}_m = \left\{ \underline{\boldsymbol{\alpha}} \in \mathbb{N}^{n\times d} \middle| |\underline{\boldsymbol{\alpha}}| = m, \text{ there are at most } \left\lfloor \frac{m}{4} \right\rfloor \text{ non-zero rows and } \left\lfloor \frac{m}{2} \right\rfloor \text{ non-zero columns} \right\}. \tag{2.26}$$

The cardinality of $\tilde{A}_m$ can be computed explicitly (for details see 2.100 in the appendix), leading to the following bound

$$\#\mathcal{A}_m \leq \#\tilde{A}_m = \binom{\lfloor m/4 \rfloor \lfloor m/2 \rfloor + m - 1}{m} \binom{n}{\lfloor m/4 \rfloor} \binom{d}{\lfloor m/2 \rfloor}, \tag{2.27}$$

which completes the proof of (2.15).

Finally, the proof of corollary 2.7 consists in elementary asymptotics, see section 2.B.5 for all the details.

2.3.4 Statistical-to-computational gaps in the spiked cumulant model

Put together, our results for the statistical and computational sample complexities of detecting non-Gaussianity in lemma 2.2 and theorem 2.6 suggest the existence of three different regimes in the spiked cumulant model as we vary the exponent θ, with a statistical-to-computational gap: for $0 \leq \theta < 1$, the problem is *statistically impossible* in the sense that no algorithm is able to strongly distinguish $\mathbb{P}$ and $\mathbb{Q}$ with so few samples, since the LR norm is bounded. For $1 < \theta < 2$, the norm of the likelihood ratio with $g \sim \text{Rademacher}(1/2)$ diverges (even at $\theta = 1$ for some values of β), the problem is statistically solvable (as validated by the results of exhaustive-search algorithms in fig. 2.1), but conjecture 2.5 suggests no polynomial-time algorithm is able to achieve distinguishability in this regime; this is the so-called *hard phase*. If $\theta > 2$, the problem is solvable in polynomial time by evaluating a polynomial function (fourth-order in each sample) and thresholding; this is the *easy phase*.

The spiked cumulant model leads thus to intrinsically harder classification problems than the spiked Wishart model, where the critical regime is at $\theta = 1$. The proof of theorem 2.6 reveals that this increased difficulty is a direct consequence of the whitening of the inputs in eq. (2.3). Without whitening, degree-2 polynomials would also give contributions to the LDLR (2.19) which would yield linear sample complexity. The difference in sample complexity of the spiked Wishart and spiked cumulant models mirrors the gap between the sample complexity of the best-known algorithms for matrix factorisation, which require linear sample complexity, and tensor PCA for rank-1 spiked tensors of order k [57, 60, 62], where sophisticated spectral algorithms can match the computational lower bound of $d^{k/2}$ samples.

2.4 Learning from HOCs with neural networks and random features

The analysis of the (low-degree) likelihood ratio has given us a detailed picture of the statistical and computational complexity of extracting information from the covariance or the higher-order

cumulants of data. We will now use these fundamental limits to benchmark the sample complexity of two-layer neural networks (2LNN) trained with stochastic gradient descent on a binary discrimination task, where inputs in one class are drawn from the normal distribution $\mathcal{N}(0, \mathbb{1}_d)$, while inputs in the other class are drawn from the spiked Wishart or the spiked cumulant model. In addition, we will also benchmark random feature methods (RF) [10, 12, 103] as a finite-dimensional approximation of kernel methods [10, 12, 103].

In a nutshell, the idea behind our experiments is to first *validate* both 2LNN and RF on the simpler spiked Wishart task and then to apply both methods to the spiked cumulant model, where inputs are generated in a way that mirrors the spiked Wishart: comparing eq. (2.1) with eqs. (2.2) and (2.3), we see that the only differences are the whitening, and the latent distribution p_g. In our experiments, we choose the latent variables to be standard Gaussian for spiked Wishart, and Rademacher for the spiked cumulant – hence the latent variables have matching first and second moments. However, we will see that the spiked cumulant model exhibits a large gap in the sample complexity required for neural networks or random features to learn the problem. We relegate details on the experimental setups such as hyper-parameters to section 2.A.

2.4.1 Spiked Wishart model

We trained **two-layer ReLU neural networks** $\phi_\theta(x) = v^\top \max\left(0, w^T x\right)$ with $m = 5d$ hidden neurons on the spiked Wishart classification task. We show the early-stopping test accuracy of the networks in the linear and quadratic regimes with $n_{\text{class}} \asymp d, d^2$ samples per class in fig. 2.2A and B, resp. Neural networks are able to solve this task, in the sense that their classification error *per sample* is below 0.5, implying strong distinguishability of the whole data matrix. Indeed, some of the hidden neurons converge to the spike u of the covariance matrix, as can be seen from the maximum value of the normalised overlaps $\max_k w_k^\top u / \sqrt{\|w_k\|\|u\|}$ in fig. 2.2C and D, where w_k is the weight vector of the kth hidden neuron. In the linear regime (C), there is an increasingly clear transition from a random overlap for small data sets to a large overlap at large data sets as we increase the input dimension d; in the quadratic regime (D), the neural network recovers the spike almost perfectly.

The **relatively large overlap between hidden neurons and spike at small sample complexities** (fig. 2.2C and D) is due to the fact that we plot the maximum overlaps over a relative large number of hidden neurons $m = 5d$; hence even at initialisation, a few neurons will have large overlaps. We verified that ensuring an initial overlap of only $1/\sqrt{d}$ by explicit orthogonalisation did not change our results on distinguishability, see fig. 2.5. A possible explanation is that the dynamics of the wide network is dominated by the majority of neurons, which do not have a macroscopic overlap.

Meanwhile, we found that **the performance of random features** tends to random guessing at linear sample complexity, where we let the input dimension $d \to \infty$ with m/d and n_{class}/d fixed, while at quadratic sample complexity, random features learn the task, although they perform worse than neural networks. The failure of RF in the linear regime makes sense in light of recent results that suggest that random features in this scaling regime are limited to learning a linear approximation of the target function [14, 104–106], while the LDLR analysis section 2.B.2.3 shows that the target function, i.e. the low-degree likelihood ratio, cannot be efficiently approximated by linear functions. However, these results are, to the best of our knowledge, restricted to the case of Gaussian isotropic inputs.

To ensure that the performance of random features does indeed tend to random guessing, we performed a **replica analysis** following Loureiro *et al.* [107] for mixture classification tasks together with the Gaussian equivalence theorem [13, 108–111] (black lines in fig. 2.2A, details in section 2.C).

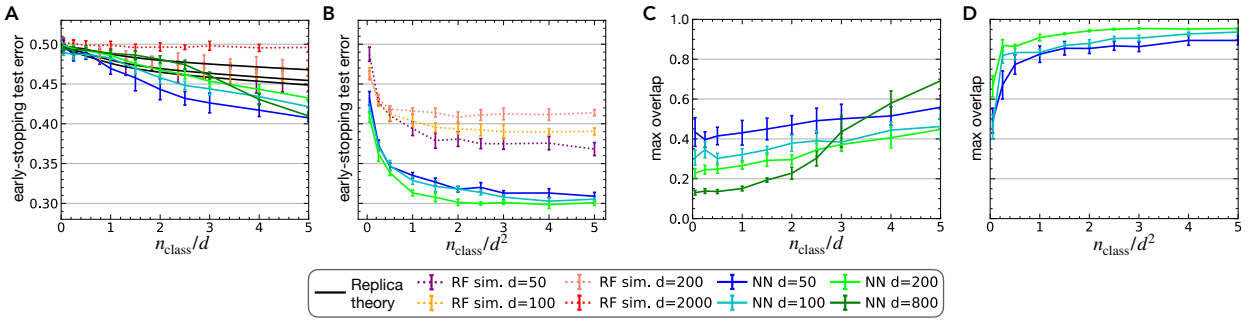

Figure 2.2: **Learning the spiked Wishart task with neural networks and random features. (A,B)** Test accuracy of random features (RF) and early-stopping test accuracy of two-layer ReLU neural networks (NN) on the spiked Wishart task, eq. (2.1), with linear and quadratic sample complexity ($n_{\text{class}} \asymp d$, d^2, respectively, where d is the input dimension). Predictions for the performance of random features obtained using replicas are shown in black. **(C,D)** Maximum normalised overlaps of the networks' first-layer weights with the spike u, eq. (2.1). *Parameters*: $\beta = 5$. Neural nets and random features have $m = 5d$ hidden neurons. Full experimental details in section 2.A.

Replica theory perfectly predicts the performance of RF we obtain in numerical experiments (red-ish dots) for various values of d at linear sample complexity. We thus find a clear separation in the sample complexity required by random features ($n_{\text{class}} \gtrsim d^2$) and neural networks ($n_{\text{class}} \gtrsim d$) to learn the spiked Wishart task. The replica analysis can be extended to the polynomial regime by a simple rescaling of the free energy [112] on several data models, like the vanilla teacher–student setup [113], the Hidden manifold model [108], and the vanilla Gaussian mixture classification (see fig. 2.8). However, we found that for the spiked Wishart model, the Gaussian equivalence theorem which we need to deal with the random feature distribution *fails* at quadratic sample complexity. This might be due to the fact that in this case, the spike induces a dominant direction in the covariance of the random features, and this direction is also key to solving the task, which can lead to a breakdown of Gaussian equivalence [111]. A similar breakdown of the Gaussian equivalence theorem at quadratic sample complexity has been demonstrated recently in teacher–student setups by Cui *et al.* [114] and Camilli *et al.* [115].

2.4.2 Spiked cumulant

Having thus validated the performance of 2LNN and RF on the simpler spiked Wishart model, we turn to the spiked cumulant model and to the question: can neural networks learn higher-order correlations, efficiently? The LDLR analysis predicts that a polynomial-time algorithm requires at least a quadratic number of samples to detect non-Gaussianity, and hence to solve the classification task, and we found indeed that neural networks require at least quadratic sample complexity to solve the task, fig. 2.3A and B. The high values of the maximum overlap between hidden neurons and cumulant spike in the linear regime (compared to $d^{-1/2}$) are again a consequence of choosing the maximum overlap among $m = 5d$ hidden neurons. Random features cannot solve this task even at quadratic sample complexity, since they are limited to a quadratic approximation of the target function [14, 105, 106, 116], but we know from the LDLR analysis that the target function complexity cannot be captured by polynomials with degree lower than four. We thus find an even larger separation in the minimal number of samples required for random features and neural networks to solve tasks that depend on directions which are encoded exclusively in the higher-order

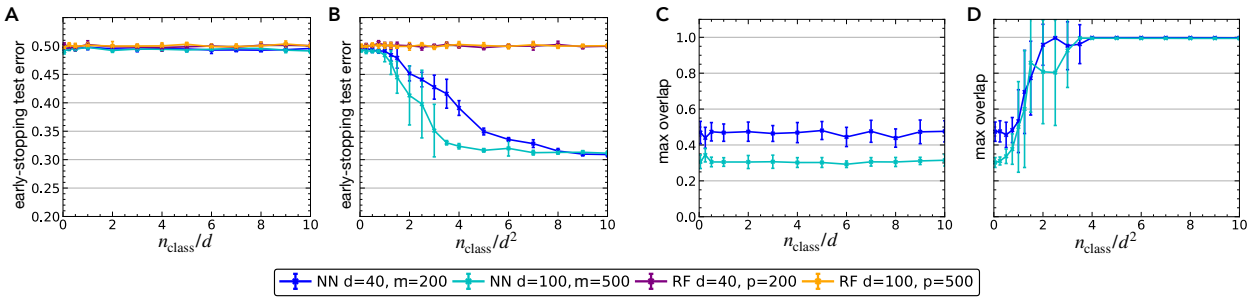

Figure 2.3: **Learning the spiked cumulant task with neural networks and random features. (A, B)** Test accuracy of random features (RF) and early-stopping test accuracy of two-layer ReLU neural networks (NN) on the spiked cumulant task eq. (2.3) with linear and quadratic sample complexity ($n_{\text{class}} \asymp d,\ d^2$, respectively, where d is the input dimension). **(C, D)** Maximum normalised overlaps of the networks' first-layer weights with the spike u, (2.3). *Parameters*: $\beta = 10$. Neural nets and random features have $m = 5d$ hidden neurons, same optimisation as in fig. 2.2. Full experimental details in section 2.A.

cumulants of the inputs.

2.4.3 Phase transitions and neural network performance in a simple model for images

We finally show another example of a separation between the performance of random features and neural networks in the feature-learning regime on a toy model for images that was introduced recently by Ingrosso & Goldt [45], the non-linear Gaussian process (NLGP). The idea is to generate inputs that are (i) translation-invariant and that (ii) have sharp edges, both of which are hallmarks of natural images [117]. We first sample a vector $z \in \mathbb{R}^d$ from a normal distribution with zero mean and covariance $C_{ij} = \mathbb{E}\, z_i z_j = \exp(-|i-j|/\xi)$ to ensure translation-invariance of the inputs, with length scale $\xi > 0$. We then introduce edges, i.e. sharp changes in luminosity, by passing z through a saturating non-linearity like the error function, $x_i = \text{erf}(gz_i)/Z(g)$, where $Z(g)$ is a normalisation factor that ensures that the pixel-wise variance $\mathbb{E}\, x_i^2 = 1$ for all values of the gain $g > 0$. The classification task is to discriminate these "images" from Gaussian inputs with the same mean and covariance, as illustrated in two dimensions in fig. 2.4A. This task is different from the spiked cumulant model in that the cumulant of the NLGP is not low-rank, so there are many directions that carry a signal about the non-Gaussianity of the inputs.

We trained wide two-layer neural networks on this task and interpolated between the feature-learning and the "lazy" regimes using the α-renormalisation trick of Chizat *et al.* [118]. As we increase α, the networks go from feature-learners ($\alpha = 1$) to an effective random feature model and require an increasing amount of data to solve the task, fig. 2.4B. There appears to be a sharp transition from random guessing to non-trivial performance as we increase the number of training samples for all values of α. This transition is preceded by a transition in the behaviour of the higher-order cumulant that was reported by Ingrosso & Goldt [45]. They showed numerically that the CP-factors of the empirical fourth-order cumulant T, defined as the vectors $\hat{u} \in \mathbb{R}^d$ that give the best rank-r approximation $\hat{T} = \sum_{k=1}^{r} \gamma_k \hat{u}_k^{\otimes 4}$ of T [119], localise in space if the data set from which the empirical cumulant is calculated is large enough. Quantifying the localisation of a weight

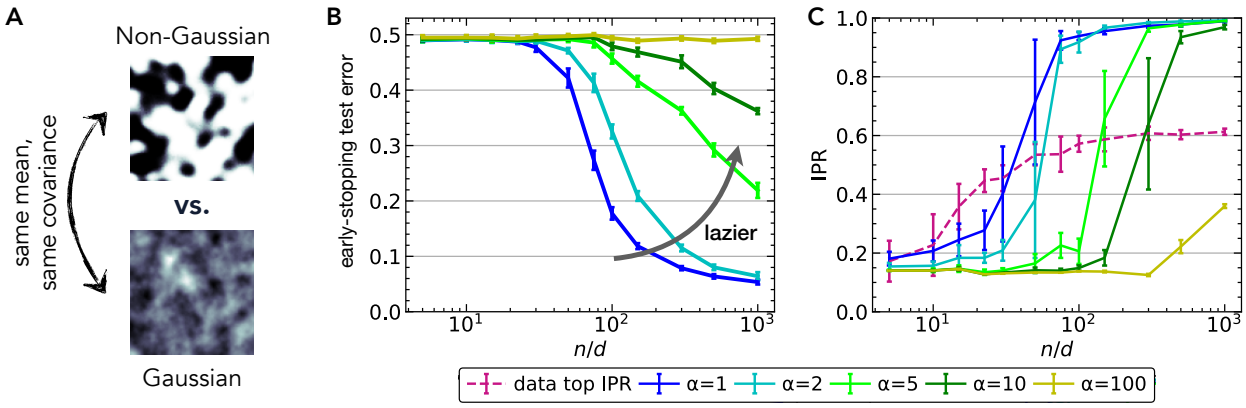

Figure 2.4: **A phase transition in the fourth-order cumulant precedes learning from the fourth cumulant. (A)** We train neural networks to discriminate inputs sampled from a simple non-Gaussian model for images introduced by Ingrosso & Goldt [45] (top) from Gaussians with the same mean and covariance (bottom). **(B)** Test error of two-layer neural networks interpolating between the fully-trained ($\alpha = 1$) and lazy regimes (large α) – see section 2.4.3. **(C)** The localisation of the leading CP-factor of the non-Gaussian inputs (dashed purple line) and the first-layer weights of the trained networks, as measured by the inverse participation ratio (IPR), eq. (2.28). Large IPR denotes a more localised vector w. *Parameters*: $g = 3$, $\xi = 1, d = 20, m = 100$. Full details in section 2.A.

vector w using the *inverse participation ratio*

$$\mathrm{IPR}(w) = \frac{\sum_{i=1}^{d} w_i^4}{\left(\sum_{i=1}^{d} w_i^2\right)^2}, \tag{2.28}$$

we confirm that the leading CP-factors of the fourth-order cumulant localise (purple dashed line in fig. 2.4C). The localisation of the CP-factors occurs with slightly less samples than the best-performing neural network requires to learn ($\alpha = 1$). The weights of the neural networks also localise at a sample complexity that is slightly below the sample complexity for solving the task. The laziest network ($\alpha = 100$), i.e. the one where the first-layer weights move least and which is hence closest to random features, does not learn the task even with a training set containing $n = 10^3 d$ samples when $d = 20$, indicating again a large advantage of feature-learners over methods with fixed feature maps, such as random features.

2.5 Concluding perspectives

Neural networks crucially rely on the higher-order correlations of their inputs to extract statistical patterns that help them solve their tasks. Here, we have studied the difficulty of learning from higher-order correlations in the spiked cumulant model, where the first non-trivial information in the data set is carried by the input cumulants of order 4 and higher. Our LR analysis of the corresponding hypothesis test confirmed that data sampled from the spiked cumulant model could be statistically distinguishable (in the sense that it passes the *second moment method for distinguishability*) from isotropic Gaussian inputs at linear sample complexity, while the number of samples required to strongly distinguish the two distributions in polynomial time scales as $n \gtrsim d^2$ for the class of algorithms covered by the low-degree conjecture [38–41], suggesting the existence of a large statistical-to-computational gap in this problem. Our experiments with neural networks

show that they learn from HOCs efficiently in the sense that they match the sample complexities predicted by the analysis of the hypothesis test, which is in stark contrast to random features, which require a lot more data. In the future, a key challenge will be extend this framework to null hypotheses that go beyond isotropic Gaussian distributions. In chapter 3 we will analyse the *dynamics* of neural networks on spiked cumulant models to understand how neural networks extract information from the higher-order cumulants of realistic data sets efficiently.

Appendix

2.A Experimental details

2.A.1 Figures 2.2 and 2.3

For the spiked Wishart and spiked cumulant tasks, we trained two-layer ReLU neural networks $\phi_\theta(x) = v^\top \max\left(0, w^T x\right)$. The number of hidden neurons $m = 5d$, where d is the input dimension. We train the networks using SGD with a learning rate of 0.002 and a weight-decay of 0.002 for 50 epochs for the spiked Wishart task and for 200 epochs for the spiked cumulant task. The plots show the early-stopping test errors. The results are plotted as averages over 10 random seeds, showing the standard deviation by the error bars.

The random features (RF) models [12, 103] have also a width of $5d$. The ridge regression is performed using scikit-learn [120] with a regularisation of 0.1.

For the spiked datasets, the spikes are from the Rademacher distribution, using a signal-to-noise ratio of 5.0 for the spiked Wishart and 10.0 for the spiked cumulant datasets. For the overlaps between spikes and features overlaps, we plot the highest overlap amongst the incoming weights of the hidden neurons with the spike, calculated as the normalised dot product.

Starting from small initial overlap Since the neural networks have a large number of neurons ($m = 5d$), some of them will have a relatively large overlap with the spike at initialisation, as can be seen in fig. 2.3. To check that this relatively large overlap did not affect our results, we repeated the same set of experiments while enforcing an overlap of all hidden neuron weights with the spike of $1/\sqrt{d}$ by explicit orthogonalisation, as discussed in section 2.4.1. We show the results in fig. 2.5: while the maximum overlaps do indeed decrease for small sample complexities, the qualitative behaviour is unchanged.

2.A.2 Figure 2.4

For the NLGP–GP task, we use the α-scaling trick of Chizat *et al.* [118] to interpolate between feature- and lazy-learning. We define the network function as:

$$\phi_{\mathrm{NN}}(x; v, W, v_0, W_0) = \frac{\alpha_{\mathrm{NTK}}}{K}\left[\sum_j^m v_j \sigma\left(\sum_i^d w_i x_i\right) - \sum_j^m v_{0,j}\sigma\left(\sum_i^d w_{0,i} x_i\right)\right] \tag{2.29}$$

where v_0, w_0 are kept fixed at their initial values. The mean-squared loss is also rescaled by $(1/\alpha_{\mathrm{NTK}}^2)$. Changing α_{NTK} from low to high allows to interpolate between the feature- and the lazy-learning limits as the first-layer weights will move away less from their initial values.

For fig. 2.4, the network has a width of 100 and the optimisation is run by SGD for 200 epochs with a weight-decay of $5 \cdot 10^{-6}$ and learning rate of 0.5. The one-dimensional data vectors have a

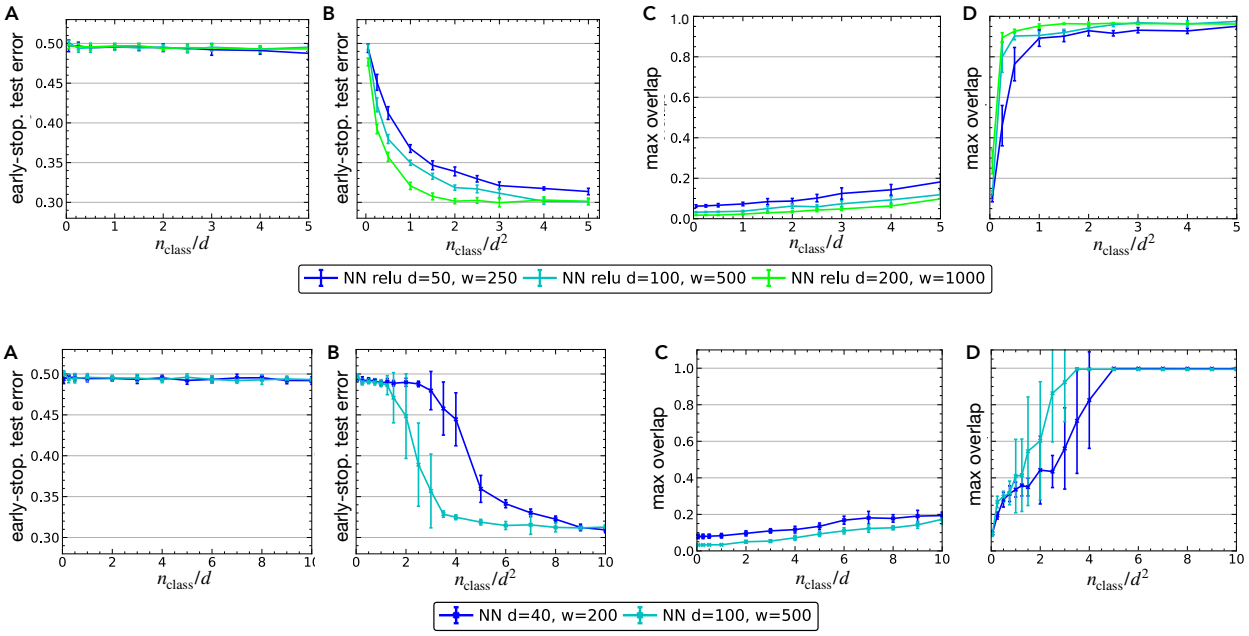

Figure 2.5: **Learning the spiked Wishart and spiked cumulant task, starting from small initial overlaps.** We repeat the neural network experiments on the spiked Wishart (top) and spiked cumulant (bottom) task, see figs. 2.2 and 2.3, while enforcing that all hidden neurons have an overlap of exactly $1/\sqrt{d}$ with the spikes, by simple explicit orthogonalisation. While the maximum overlaps do indeed decrease for small sample complexities, the qualitative behaviour is unchanged. All hyper-parameters as in figs. 2.2 and 2.3, respectively.

length of 20; the correlation length is 1 and the signal-to-noise ratio is set to 3. The error shown is early-stopping test error. The localisation of the neural networks' features and the data's fourth moments is shown by the IPR measure. (We used here a lower length for the data vectors so that the calculations for fourth-order cumulants do not have high memory requirements.) For the neural networks, the highest IPR is shown amongst the incoming features of the hidden neurons at the final state of the training. For the data, the highest IPR is the highest amongst the CP-factors of the fourth cumulants of the nlgp-class using a rank-1 PARAFAC decomposition from the tensorly package [121] .

2.B Mathematical details on the hypothesis testing problems

In this section, we provide more technical details on Hermite polynomials, LR and LDLR, and present the complete proofs of all theorems stated in the main.

2.B.1 Additional notation

Underlined letters $\underline{x}, \underline{\mathbb{P}}, \underline{\mathbb{Q}}$ are used for objects that are $n \times d$ dimensional. In proofs the letter C denotes numerical constants whose precise value may change from line to line, and any dependency is denoted as a subscript. $m!$ denotes factorial and $m!!$ denotes the double factorial (product of all the numbers $k \leq m$ with the same parity as m).

Multi-index notation We will need multi-index notation to deal with d-dimensional functions and polynomials. Bold greek letters are used to denote multi-indices and their components, $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_d) \in \mathbb{N}^d$. The following conventions are adopted:

$$\begin{aligned} |\boldsymbol{\alpha}| &:= \sum_{i=1}^{d} \boldsymbol{\alpha}_i, \quad \boldsymbol{\alpha}! := \prod_{i=1}^{d} \boldsymbol{\alpha}_i!, \quad x^{\boldsymbol{\alpha}} := \prod_{i=1}^{d} x_i^{\boldsymbol{\alpha}_i} \\ \partial_{\boldsymbol{\alpha}} f(x_1, \dots, x_d) &= \frac{\partial}{\partial x^{\boldsymbol{\alpha}}} f := \left(\frac{\partial}{\partial x_1^{\boldsymbol{\alpha}_1}} \cdots \frac{\partial}{\partial x_d^{\boldsymbol{\alpha}_d}} \right) f \end{aligned} \tag{2.30}$$

Since we will have n samples of d-dimensional variables, we will need to consider polynomials and functions in nd variables. To deal with all these variables we introduce matrix-multi-indices (denoted by underlined bold Greek letters $\underline{\boldsymbol{\alpha}}, \underline{\boldsymbol{\beta}}, \dots$). They are $n \times d$ matrices with entries in $\mathbb{N}$ (i.e. elements of $\mathbb{N}^{n \times d}$)

$$\underline{\boldsymbol{\alpha}} := (\boldsymbol{\alpha}_i^{\mu}) = \begin{pmatrix} \boldsymbol{\alpha}_1^1 & \dots & \boldsymbol{\alpha}_d^1 \\ \vdots & \ddots & \vdots \\ \boldsymbol{\alpha}_1^n & \dots & \boldsymbol{\alpha}_d^n \end{pmatrix} \tag{2.31}$$

We denote by $\boldsymbol{\alpha}^{\mu}$ the rows of $\underline{\boldsymbol{\alpha}}$, that are d-dimensional multi-indices.

All the notations (2.30) generalize to matrix-multi-indices in the following way:

$$\begin{aligned} |\underline{\boldsymbol{\alpha}}| &:= \sum_{\mu=1}^{n} |\boldsymbol{\alpha}^{\mu}| = \sum_{\mu=1}^{n} \sum_{i=1}^{d} \boldsymbol{\alpha}_i^{\mu}, \quad \underline{\boldsymbol{\alpha}}! := \prod_{\mu} \boldsymbol{\alpha}^{\mu}! = \prod_{\mu} \prod_{i} \boldsymbol{\alpha}_i^{\mu}!, \\ \underline{x}^{\underline{\boldsymbol{\alpha}}} &:= \prod_{\mu=1}^{n} (x^{\mu})^{\boldsymbol{\alpha}^{\mu}} = \prod_{\mu=1}^{n} \prod_{i=1}^{d} (x_i^{\mu})^{\boldsymbol{\alpha}_i^{\mu}} \\ \partial_{\underline{\boldsymbol{\alpha}}} f(\underline{x}) &:= \left(\frac{\partial}{\partial (x^1)^{\boldsymbol{\alpha}^1}} \cdots \frac{\partial}{\partial (x^n)^{\boldsymbol{\alpha}^n}} \right) f(\underline{x}) \end{aligned} \tag{2.32}$$

2.B.2 More on LR and LDLR

2.B.2.1 Statistical and computational distinguishability

From a statistical point of view, distinguishing two sequences of probability measures $\underline{\mathbb{P}} = (\mathbb{P}_n)_{n \in \mathbb{N}}$ and $\underline{\mathbb{Q}} = (\mathbb{Q}_n)_{n \in \mathbb{N}}$ defined on a sequence of measure spaces $(\Omega_n, \mathcal{F}_n)$ means finding a sequence of tests $f_n : \Omega_n \to \{0, 1\}$, which are measurable functions that indicate whether a given set of inputs was sampled from $\mathbb{P}$ or $\mathbb{Q}$. We will say that the two measures are **statistically distinguishable** in the strong sense if there exists a statistical test f for which

$$\mathbb{Q}_n \left(f_n(x) = 0 \right) \underset{n \to \infty}{\longrightarrow} 1 \quad \text{and} \quad \mathbb{P}_n(f_n(x) = 1) \underset{n \to \infty}{\longrightarrow} 1. \tag{2.33}$$

The strong version of statistical distinguishability requires that the probability of success of the statistical test must tend to 1 as $n \to \infty$, whereas **weak distinguishability** just requires it to be asymptotically greater than $1/2 + \varepsilon$, for any $\varepsilon > 0$. We will call the minimal number of samples required to achieve strong statistical distinguishability the **statistical sample complexity** of the problem. We can obtain a computationally bounded analogue of this definition by restricting the complexity of the statistical test f_n to functions that are computable in a time that is polynomial in the input dimension d. The **computational statistical complexity** is then the minimal number of samples required to achieve **computational distinguishability**.

2.B.2.2 Necessary conditions for distinguishability

A necessary condition for strong distinguishability is based on the *likelihood ratio* (LR) of probability measures, which is defined as

$$L_n(x) := \frac{\mathrm{d}\mathbb{P}_n}{\mathrm{d}\mathbb{Q}_n}(x) \tag{2.34}$$

The likelihood ratio provides a necessary condition for strong distinguishability of $\mathbb{P}$ and $\mathbb{Q}$ via the so-called second moment method:

Proposition 2.9 (Second Moment Method for Distinguishability)**.** *Suppose that $\mathbb{P}_n$ is absolutely continuous with respect to $\mathbb{Q}_n$, and let L_n be the corresponding LR. A necessary condition for strong distinguishability of $\underline{\mathbb{P}}$ from $\underline{\mathbb{Q}}$ is*

$$\|L_n\|^2 := \mathop{\mathbb{E}}_{x\sim\mathbb{Q}_n}[L_n(x)^2] \underset{n\to\infty}{\longrightarrow} +\infty. \tag{2.35}$$

where $\|\cdot\|$ is the norm with respect to the Hilbert space

$$L^2(\Omega_n, \mathbb{Q}_n) = \left\{ f : \Omega_n \to \mathbb{R} \mid \mathbb{E}_{\mathbb{Q}_n}[f^2(x)] < \infty \right\}. \tag{2.36}$$

Proof. The proof is immediate: if $(||L_n||)_n$ were bounded, by Cauchy-Schwartz $\mathbb{Q}(A_n) \to 0$ would imply $\mathbb{P}(A_n) \to 0$. But by the definition of strong asymptotic distinguishability there exists a sequence of events $(A_n)_n$ such that $\mathbb{P}(A_n) \to 1$ and $\mathbb{Q}(A_n) \to 0$; hence $(||L_n||)_n$ must diverge. □

The second moment method has been used to derive statistical thresholds for various high-dimensional inference problems [51, 59, 69, 122]. Note that proposition 2.9 only states a necessary condition; while it is possible to construct counterexamples, they are usually based on rather artificial constructions (like the following example from [69]), and the second moment method is considered a good proxy for strong statistical distinguishability.

Example 2.10. Let $\underline{\mathbb{P}}$ and $\underline{\mathbb{Q}}$ be two strongly distinguishable sequences of probability measures on $(\mathcal{Y}_n, \mathcal{F}_n)_n$. Define $\mathbb{P}'_n$ on $\mathcal{F}_n$ as a measure that with probability $1/2$ follows $\mathbb{P}_n$ and with probability $1/2$ follows $\mathbb{Q}_n$. Letting L'_n be the LR of $\mathbb{P}'_n$ with respect to $\mathbb{Q}_n$. We have that $||L'_n|| \to \infty$, but $\underline{\mathbb{P}'}$ and $\mathbb{Q}$ are not strongly distinguishable.

Proof. The density of $\mathbb{P}'_n$ is:

$$p'_n(y) = \frac{p_n(y) + q_n(y)}{2} \tag{2.37}$$

hence

$$L'_n = \frac{1}{2}(1 + L_n) \tag{2.38}$$

which clearly has the same asymptotic behaviour as L_n.

On the other hand, it cannot be that $\underline{\mathbb{P}'}$ and $\underline{\mathbb{Q}'}$ are strongly distinguishable since for any event A, $\mathbb{P}'_n(A) \geq \frac{1}{2}\mathbb{Q}_n(A)$. □

2.B.2.3 LDLR analysis for spiked Wishart model

In this subsection we show the application of LDLR method to the spiked Wishart model. We obtain the correct BBP threshold for the signal-to-noise ratio even when we restrict ourselves to the class of polynomials of constant degree:

Theorem 2.11 (LDLR for spiked Wishart model)**.** *Suppose the prior on u belongs to the following cases:*

- $(u_i)_{i=1,\dots,d}$ *are i.i.d. and symmetric Rademacher random variables*
- $(u_i)_{i=1,\dots,d}$ *are i.i.d. and* $u_i \sim \mathcal{N}(0,1)$

Let $D \in \mathbb{N}$ and $d, n \to \infty$, with fixed ratio $\gamma := d/n$, then

$$\lim_{d,n\to\infty} \|L^{\leq D}\| = \sum_{k=0}^{D} \frac{(2k-1)!!}{(2k)!!} \frac{\beta^{2k}}{\gamma^k}, \tag{2.39}$$

which, as D increases, stays bounded for $\beta < \beta_c := \sqrt{\gamma} = \sqrt{d/n}$ and diverges for $\beta > \beta_c$.

The distinguishability threshold that we have recovered here is of course the famous BBP phase transition in the spiked Wishart model [42]: if $\beta < \beta_c = \sqrt{d/n}$, the low-degree likelihood ratio stays bounded and indeed a set of inputs drawn from 2.1 is statistically indistinguishable from a set of inputs drawn from $\mathcal{N}(0, \mathbb{1})$. For $\beta > \beta_c$ instead, there is a phase transition for the largest eigenvalue of the empirical covariance of the inputs in the second class, which can be used to differentiate the two classes, and the LDLR diverges.

As mentioned in the main text, a thorough LDLR analysis of a more general model that encompasses the spiked Wishart model was performed by Bandeira *et al.* [123]. Their theorem 3.2 is a more general version of our theorem 2.11, that generalizes it in two directions: it works for a class of subgaussian priors on u that satisfy some concentration property, and also allows for negative SNR (the requirement is $\beta > -1$). For completeness, here we show that this result can be obtained as a straightforward application of a Gaussian additive model.

Proof. We note that the problem belongs the class of *Additive Gaussian Noise models* which is studied in depth by Kunisky *et al.* [69]. In those models the two hypotheses have to be expressed as an additive perturbation of white noise:

- $\mathbb{P}_n$: $\underline{x}_n = y_n + z_n$,
- $\mathbb{Q}_n$: $\underline{x}_n = z_n$,

The spiked Wishart model that we consider belongs to this class. It can be seen by defining $\mathbb{R}^{nd} \ni y_n = \left(\sqrt{\frac{\beta}{d}} g^1 u, \dots, \sqrt{\frac{\beta}{d}} g^n u\right)^\top$. So we can apply theorem 2.6 from Kunisky *et al.* [69], that computes the norm of the LDLR by using two independent replicas of the variable y. Denoting the two replicas by $\hat{y}$ and $\tilde{y}$, we get

$$\begin{aligned}
||L_n^{\leq D}||^2 &= \mathbb{E}\left[\sum_{m=0}^{D} \frac{1}{m!} (\hat{y}\cdot\tilde{y})^m\right] \\
&= \mathbb{E}\left[\sum_{m=0}^{D} \frac{\beta^m}{m! d^m} \left(\sum_{\mu=1}^{n} \hat{g}^\mu \tilde{g}^\mu \hat{u}\cdot\tilde{u}\right)^m\right] \\
&= \sum_{m=0}^{D} \frac{\beta^m}{m! d^m} \mathbb{E}\left[(\hat{u}\cdot\tilde{u})^m \left(\sum_{\mu=1}^{n} \hat{g}^\mu \tilde{g}^\mu\right)^m\right] \\
&= \sum_{m=0}^{D} \frac{\beta^m}{m! d^m} \mathbb{E}\left[\left(\sum_{\mu=1}^{n} \hat{g}^\mu \tilde{g}^\mu\right)^m\right] \mathbb{E}\left[(\hat{u}\cdot\tilde{u})^m\right]
\end{aligned} \tag{2.40}$$

Note now that $\sum_{\mu=1}^{n} \hat{g}^\mu \tilde{g}^\mu$ has the same distribution as $-\sum_{\mu=1}^{n} \hat{g}^\mu \tilde{g}^\mu$, so its distribution is even and all the odd moments are 0. This means that we can reduce to the case $m = 2k$, so we need to study:

$$||L_n^{\leq D}||^2 = \sum_{k=0}^{\lfloor D/2 \rfloor} \frac{\beta^{2k}}{(2k)! d^{2k}} \underbrace{\mathbb{E}\left[\left(\sum_{\mu=1}^{n} \hat{g}^\mu \tilde{g}^\mu\right)^{2k}\right]}_{T_1} \underbrace{\mathbb{E}\left[(\hat{u} \cdot \tilde{u})^{2k}\right]}_{T_2} \tag{2.41}$$

Let us consider the term T_1. Call $Y^\mu := \hat{g}^\mu \tilde{g}^\mu$. The distribution of each of the Y^μ is not Gaussian, but we have that $\mathbb{E}[Y^\mu] = 0$ and $\mathrm{Var}(Y^\mu) = 1$. So by the central limit theorem $S_n := \frac{1}{\sqrt{n}} \sum_{\mu=1}^{n} Y^\mu$ converges to a standard normal in distribution, as $n \to \infty$. Note that the cumulants of S_n can be computed thanks to linearity and additivity of cumulants and are $\kappa_{2k}^{(S_n)} = n^{1-k} \kappa_{2k}^{Y}$. So they all go to 0 except from the variance. Since the moments can be written as a function of the cumulants up to that order, it follows that $\lim_n \mathbb{E}[S_n^{2k}]$ will be the $2k$-th moment of the standard normal distribution, which means that we have the following:

$$\lim_{n \to +\infty} \mathbb{E}\left[\left(\frac{1}{\sqrt{n}} \sum_{\mu=1}^{n} \hat{g}^\mu \tilde{g}^\mu\right)^{2k}\right] = (2k-1)!! \tag{2.42}$$

We turn now to T2 and do the same reasoning. Define $v_i := \hat{u}_i \tilde{u}_i$, both in Rademacher and in Gaussian prior case, we have that the $(v_i)_{i=1,\dots,d}$ is an independent family of random variables that have 0 mean and variance equal to 1. So we can again apply the central limit theorem to get that:

$$\lim_{d \to +\infty} \mathbb{E}\left[\left(\frac{\hat{u} \cdot \tilde{u}}{\sqrt{d}}\right)^{2k}\right] = (2k-1)!! \tag{2.43}$$

Taking the limit for $n, d \to +\infty$ with the constraint $\gamma = d/n$, we have that:

$$\begin{aligned} \lim_{n,d \to \infty} ||L_n^{\leq D}||^2 &= \lim_{n,d \to \infty} \sum_{k=0}^{\lfloor D/2 \rfloor} \frac{\beta^{2k} n^k}{(2k)! d^k} \mathbb{E}\left[\left(\frac{1}{\sqrt{n}} \sum_{\mu=1}^{n} \hat{g}^\mu \tilde{g}^\mu\right)^{2k}\right] \mathbb{E}\left[\left(\frac{\hat{u} \cdot \tilde{u}}{\sqrt{d}}\right)^{2k}\right] \\ &= \sum_{k=0}^{D} \frac{\left(\beta^k (2k-1)!!\right)^2}{(2k)! \gamma^k} = \sum_{k=0}^{D} \frac{(2k-1)!!}{(2k)!!} \frac{\beta^{2k}}{\gamma^k} \end{aligned} \tag{2.44}$$

which is what we wanted to prove.

As a final note we remark that the basic ideas of the arguments in [123] coincide with what exposed above. However, the increased generality of the statement requires the use of the abstract theory of Umbral calculus to generalize the notion of Hermite polynomials to negative SNR cases, as well as more technical work to achieve the bounds on the LDLR projections. □

2.B.3 Details on the spiked cumulant model

Here we will expand on the mathematical details of the spiked cumulant model.

2.B.3.1 Prior distribution on the spike

For the prior distribution on u, $\mathcal{P}$, its role is analogous to the spiked Wishart model, so all the choices commonly used for that model can be considered applicable to this case. Namely symmetric

distributions so that $||\frac{u}{\sqrt{d}}|| \approx 1$ as $d \to \infty$. In the following we will make the computations assuming u_i i.i.d. and with Rademacher prior:

$$u_i \sim \text{Rademacher}(1/2) \tag{2.45}$$

It helps to have i.i.d. components and constant norm $||\frac{u}{\sqrt{d}}|| \equiv 1$. However all the results should hold, with more involved computations, also with the following priors:

- $(u_i)_{i=1,\dots,d}$ are i.i.d. and $u_i \sim \mathcal{N}(0,1)$
- $u \in \text{Unif}(\partial B(0, \sqrt{d})$.

2.B.3.2 Distribution of the non-Gaussianity

As detailed in 2.1, we need the non-Gaussianity to satisfy specific conditions. Some of the requirements are fundamental and cannot be avoided, whereas others could likely be removed or modified with only technical repercussion that do not change the essence of the results.

The most vital assumptions are the first and the last: $\mathbb{E}[g] \neq 0$ would introduce signal in the first cumulant, changing completely the model. It is important to control the tails of the distribution with

$$\mathbb{E}[\exp\Big(g^2/2\Big)] < +\infty, \tag{2.46}$$

a fat-tailed g would make the LR-LDLR technique pointless due to the fact that $L \notin \mathcal{L}^2(\mathbb{R}^d, \mathbb{Q})$ and $||L|| = \infty$ for any n, d. For example it is not possible to use Laplace distribution for g.

On the opposite side, the least important assumptions are that $\text{Var}(g) = 1$ and eq. (2.5); removing the first would just change the formula for whitening matrix S, while eq. (2.5) is a very weak requirement and it is needed just to estimate easily the Hermite coefficients' contribution and reach 2.17, it may be even possible to remove it and try to derive a similar estimate from the assumption on the tails 2.46.

Finally, the requirement of symmetry of the distribution has been chosen arbitrarily thinking about applications: the idea is that the magnitude of the fourth-order cumulant, the kurtosis, is sometimes used as a test for non-Gaussianity, so it is interesting to isolate the contribution given by the kurtosis, and higher-order, even degree, cumulants, by cancelling the contribution of odd-order cumulants. However, it would be interesting to extend the model in the case of a centered distribution with non-zero third-order cumulant. It is likely that the same techniques can be applied, but the final thresholds on θ may differ.

The following lemma gives a criterion of admissibility that ensures $\text{Radem}(1/2)$ and $\text{Unif}(-\sqrt{3}, \sqrt{3})$ (together with many other compactly supported distributions) satisfy 2.1.

Lemma 2.12. *Suppose p_g is a probability distribution, compactly supported in $[-\Lambda, \Lambda]$, $\Lambda \geq 1$, then*

$$\mathop{\mathbb{E}}_{g\sim p_g}[h_m(g)] \leq \Lambda^m m!$$

Proof. Use the notation $h_m(x) = \sum_{k=0}^m a_{m,k} x^k$ and $S_m := \sum_{k=0}^m |a_{m,k}|$. Then we have that

$$\begin{aligned} \mathop{\mathbb{E}}_{g\sim p_g}[h_m(g)] &= \mathop{\mathbb{E}}_{g\sim p_g}\left[\sum_{k=0}^m a_{m,k} g^k\right] \\ &\leq \mathop{\mathbb{E}}_{g\sim p_g}\left[\sum_{k=0}^m |a_{m,k}||g|^k\right] \\ &\leq \Lambda^m S_m \end{aligned} \tag{2.47}$$

So we just need to prove that $S_m \leq m!$, which can be done by induction using A.7. Suppose it true for m, we prove it for $m+1$. By A.7 we have that:

$$|a_{m+1,k}| \leq \begin{cases} |a_{m,1}| & k = 0 \\ |a_{m,k-1}| + (k+1)|a_{m,k+1}| & k > 0 \end{cases} \tag{2.48}$$

Summing on both sides (and using $a_{m,k} = 0$ when $k > m$), we get:

$$\begin{aligned} S_{m+1} &\leq |a_{m,1}| + \sum_{k=1}^{m+1} |a_{m,k-1}| + (k+1)|a_{m,k+1}| \\ S_{m+1} &\leq S_m + \sum_{j=0}^{m} j|a_{m,j}| \\ S_{m+1} &\leq S_m + m\sum_{j=0}^{m} |a_{m,j}| \\ S_{m+1} &\leq (m+1)S_m \end{aligned} \tag{2.49}$$

Hence by application of the inductive hypothesis we get $S_{m+1} \leq (m+1)!$, completing the proof. □

2.B.3.3 Computing the whitening matrix

In this paragraph all the expectations are made assuming u fixed and it will be best to work with its normalized version $\bar{u} = u/\sqrt{d}$. Note that $\mathbb{E}[x] = 0$ and we want also that

$$\begin{aligned} \mathbb{1}_{d\times d} = \mathbb{E}[xx^\top] &= \mathbb{E}[S\left(\sqrt{\beta}g\bar{u} + z\right)\left(\sqrt{\beta}g\bar{u}^\top + z^\top\right)S^\top] \\ &= S\left(\mathbb{1} + \beta\bar{u}\bar{u}^\top\right)S^\top \end{aligned} \tag{2.50}$$

Hence we need

$$S^2 = SS^T = \left(\mathbb{1} + \beta\bar{u}\bar{u}^\top\right)^{-1} = \mathbb{1} - \frac{\beta}{1+\beta}\bar{u}\bar{u}^\top \tag{2.51}$$

So we look for γ such that:

$$(\mathbb{1} + \gamma\bar{u}\bar{u}^\top)^2 = \mathbb{1} - \frac{\beta}{1+\beta}\bar{u}\bar{u}^\top \tag{2.52}$$

By solving the second-order equation we get

$$\gamma_\pm = -\left(1 \pm \frac{1}{\sqrt{1+\beta}}\right) \tag{2.53}$$

We choose the solution with $-$, so that S is positive definite:

$$S = \mathbb{1} - \left(1 - \frac{1}{\sqrt{1+\beta}}\right)\bar{u}\bar{u}^\top = \mathbb{1} - \frac{\beta}{1+\beta+\sqrt{1+\beta}}\frac{uu^\top}{d}. \tag{2.54}$$

Hence we can compute also the explicit expression for x:

$$\begin{aligned} x &= z - \left(1 - \frac{1}{\sqrt{1+\beta}}\right)\bar{u}^\top z\bar{u} + \sqrt{\frac{\beta}{1+\beta}}g\bar{u} \\ x &= \underbrace{z - \bar{u}^\top z\bar{u}}_{z_{\perp u}} + \left(\sqrt{\frac{1}{1+\beta}}\bar{u}^\top z + \underbrace{\sqrt{\frac{\beta}{1+\beta}}}_{\eta}g\right)\bar{u} \\ &= z_{\perp u} + \left(\sqrt{1-\eta^2}\bar{u}^\top z + \eta g\right)\bar{u} \end{aligned} \tag{2.55}$$

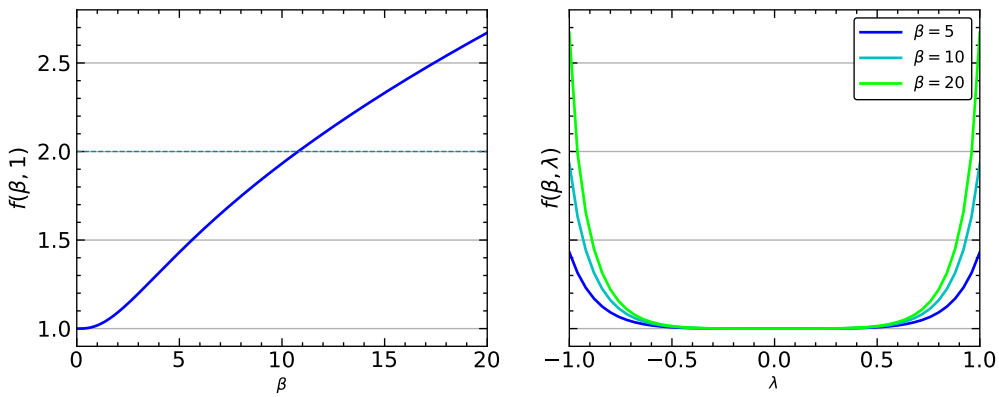

Figure 2.6: Graphs of f, defined in (2.7), when $g \sim \text{Rademacher}(1/2)$.

So x is standard Gaussian in the directions orthogonal to u, whereas in u direction, it is a weighted sum between g and z. Note that it is a quadratic interpolation: the sum of the square of the weights is 1.

2.B.4 Details of the LR analysis for spiked cumulant model

Proof of lemma 2.2. $\tilde{x}^\mu = \sqrt{\frac{\beta}{d}} g^\mu u + z^\mu$, to find the marginal density of x^μ we integrate over the possible values of g

$$p_{\tilde{x}}(y|u) = \mathbb{P}\left(\tilde{x}^\mu \in \mathrm{d}y|u\right) = \mathop{\mathbb{E}}_{g}\left[p_z\left(y - \sqrt{\frac{\beta}{d}} g u\right)\right] \tag{2.56}$$

where p_z is the density of a standard normal d-dimensional variable $z \sim \mathcal{N}(0, \mathbb{1}_d)$. But we are interested in the density of the whitened variable x, $p_x(\cdot|u)$, which can be seen as the push forward of the density of $\tilde{x}$ with respect to the linear transformation S. So

$$p_x(y|u) = p_{\tilde{x}}(S^{-1}y|u)|\det S^{-1}|. \tag{2.57}$$

It is easy to see from 2.3 that $|\det S^{-1}| = \sqrt{1+\beta}$, so we can plug it into 2.56 and after expanding the computations we get to

$$p_x(y|u) = p_z(y)\mathop{\mathbb{E}}_{g}\left[\sqrt{1+\beta}\exp\left(-\frac{1+\beta}{2}\left(g - \sqrt{\frac{\beta}{(1+\beta)d}} y\cdot u\right)^2 + \frac{g^2}{2}\right)\right]. \tag{2.58}$$

So we have found the likelihood ratio for a single sample, conditioned on the spike:

$$l(y|u) = \frac{p_x(y|u)}{p_z(y)} = \mathop{\mathbb{E}}_{g}\left[\sqrt{1+\beta}\exp\left(-\frac{1+\beta}{2}\left(g - \sqrt{\frac{\beta}{(1+\beta)d}} y\cdot u\right)^2 + \frac{g^2}{2}\right)\right]. \tag{2.59}$$

Note that conditioning on u the samples are independent, so we have the following formula:

$$L(\underline{y}) = \mathop{\mathbb{E}}_{u}\left[\prod_{\mu=1}^{n} l(y^\mu|u)\right]. \tag{2.60}$$

So to compute the norm we consider two independent replicas of the spike, u and v, then we switch the order of integration to get

$$\begin{aligned}
||L_{n,d}||^2 &= \mathop{\mathbb{E}}_{\underline{y}\sim\underline{\mathbb{Q}}}\left[\mathop{\mathbb{E}}_{u}\left[\prod_{\mu=1}^{n} l(y^\mu|u)\right]\mathop{\mathbb{E}}_{v}\left[\prod_{\mu=1}^{n} l(y^\mu|v)\right]\right] \\
&= \mathop{\mathbb{E}}_{u,v}\left[\mathop{\mathbb{E}}_{\underline{y}\sim\underline{\mathbb{Q}}}\left[\prod_{\mu=1}^{n} l(y^\mu|u)l(y^\mu|v)\right]\right] \\
&= \mathop{\mathbb{E}}_{u,v}\left[\mathop{\mathbb{E}}_{y\sim\mathbb{Q}}\left[l(y|u)l(y|v)\right]^n\right] \\
&= \mathop{\mathbb{E}}_{u,v}\Bigg[\mathop{\mathbb{E}}_{y\sim\mathbb{Q}}\Bigg(\mathop{\mathbb{E}}_{g_u}\left[\sqrt{1+\beta}\exp\left(-\frac{1+\beta}{2}\left(g_u-\sqrt{\frac{\beta}{(1+\beta)d}}y\cdot u\right)^2+\frac{g_u^2}{2}\right)\right] \\
&\qquad\cdot\mathop{\mathbb{E}}_{g_v}\left[\sqrt{1+\beta}\exp\left(-\frac{1+\beta}{2}\left(g_v-\sqrt{\frac{\beta}{(1+\beta)d}}y\cdot v\right)^2+\frac{g_v^2}{2}\right)\right]\Bigg)^n\Bigg].
\end{aligned} \tag{2.61}$$

Switching the integral over y inside the new integrals over g_u and g_v we can isolate the following integral over y:

$$I := \mathop{\mathbb{E}}_{y\sim\mathbb{Q}}\left[\exp\left(-\frac{1+\beta}{2}\left(g_u-\sqrt{\frac{\beta}{(1+\beta)d}}y\cdot u\right)^2-\frac{1+\beta}{2}\left(g_v-\sqrt{\frac{\beta}{(1+\beta)d}}y\cdot v\right)^2\right)\right]. \tag{2.62}$$

It can be computed by noting the subspace orthogonal to $\{u,v\}$, we just have the integral of a standard normal, and the remaining 2-dimensional integral can be computed explicitly. Since the Rademacher prior implies that $||u|| = ||v|| = \sqrt{d}$, the result depends only on their overlap λ (i.e. $\lambda = \frac{u\cdot v}{d}$), leading to:

$$I = \frac{1}{\sqrt{(1+\beta)^2-\beta^2\lambda^2}}\exp\left(-\frac{1+\beta}{2(1+\beta)^2-2\beta^2\lambda^2}\left((1+\beta)(g_u^2+g_v^2)-2\beta(g_ug_v)\lambda\right)\right) \tag{2.63}$$

If we plug this formula inside 2.61 and rearrange the terms we get an expression that can be written in terms of the density of two bi-dimensional centered Gaussians

$$\begin{aligned}
||L_{n,d}||^2 &= \mathop{\mathbb{E}}_{\lambda}\left[\mathop{\mathbb{E}}_{g_u,g_v}\left[\frac{1+\beta}{\sqrt{(1+\beta)^2-\beta^2\lambda^2}}e^{-\frac{(1+\beta)\left((1+\beta)(g_u^2+g_v^2)-2\beta(g_ug_v)\lambda\right)}{2(1+\beta)^2-2\beta^2\lambda^2}+\frac{g_u^2+g_v^2}{2}}\right]^n\right] \\
&= \mathop{\mathbb{E}}_{\lambda}\left[\mathop{\mathbb{E}}_{g_u,g_v}\left[\mathcal{N}\left((g_u,g_v);\Sigma\right)\mathcal{N}\left((g_u,g_v);\mathbb{1}_{2\times2}\right)^{-1}\right]^n\right].
\end{aligned} \tag{2.64}$$

where

$$\Sigma^{-1} = \frac{1+\beta}{(1+\beta)^2-\beta^2\lambda^2}\begin{pmatrix}1+\beta & -\beta\lambda\\ -\beta\lambda & 1+\beta\end{pmatrix}, \qquad \lambda = \frac{u\cdot v}{d}. \tag{2.65}$$

Now we turn to compute the expectation over λ. Note that since u, v are independent and their components are Rademacher distributed, product of Rademacher is still Rademacher, hence $u\cdot v$ is the sum of d independent Rademacher random variables. Using moreover that the Rademacher distribution is just a linear transformation of the Bernoulli, we can link the distribution of a the overlap λ to a linear transformation of a binomial distribution: $\frac{d}{2}(\lambda+1)\sim\text{Binom}(d,1/2)$.

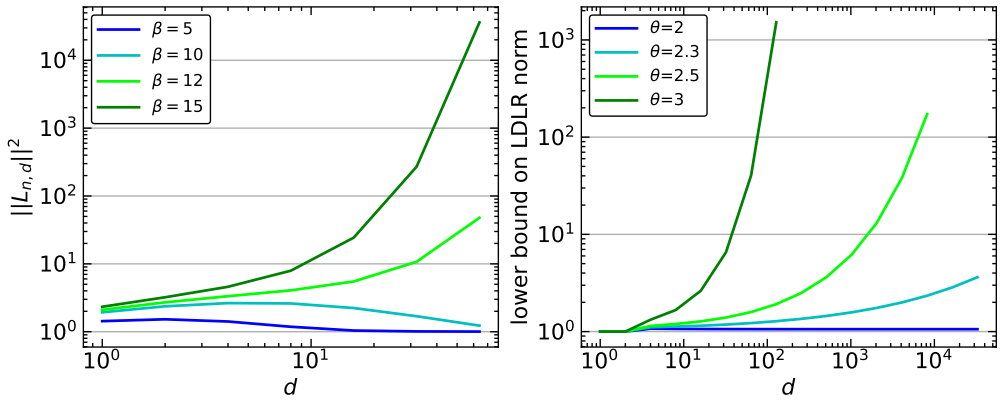

Figure 2.7: On the left, LR norm when $g \sim$ Rademacher(1/2) in the regime $n = \gamma d$ with $\gamma = 1$. When $\beta < \beta_\gamma \approx 10.7$ the likelihood ratio remains bounded, whereas it goes to $+\infty$ for $\beta > \beta_\gamma$. On the right, the lower bound on $||L_{n,d}^{\leq D(n)}||$ given by 2.14 goes to $+\infty$ for $\theta > 2$. Parameters for the plot: $g \sim$Radem(1/2), $\beta = 10$, $D(n) = \log^{3/2}(n)$

We therefore define the auxiliary function

$$f(\beta, \lambda) := \mathop{\mathbb{E}}_{g_u, g_v} \left[\frac{1+\beta}{\sqrt{(1+\beta)^2 - \beta^2\lambda^2}} e^{-\frac{(1+\beta)\left((1+\beta)(g_u^2+g_v^2)-2\beta(g_u g_v)\lambda\right)}{2(1+\beta)^2-2\beta^2\lambda^2} + \frac{g_u^2+g_v^2}{2}} \right] \tag{2.66}$$

so that the LR norm can be rewritten as

$$||L_{n,d}||^2 = \sum_{j=0}^{d} \binom{d}{j} \frac{1}{2^d} f\left(\beta, \frac{2j}{d} - 1\right)^n \tag{2.67}$$

which is what we needed to prove. □

2.B.4.1 Consequences of lemma 2.2

We will analyze in more detail the case in which $g \sim$ Rademacher(1/2). It is a case that is particularly interesting for the point of view of the applications because it amounts to comparing a standard Gaussian with a Gaussian mixture with same mean and covariance. Moreover, with this choice of non-Gaussianity, the technical work simplifies because the troublesome integral over g_u, g_v in 2.7 becomes just a simple sum over 4 possibilities. In this case f can be computed exactly and it is displayed in 2.6. The maximum of $f(\beta, \cdot)$ is attained at $\lambda = \pm 1$ and $f(\cdot, 1)$ is monotonically increasing. Assume that $n \asymp d^\theta$, then a sufficient condition for LR to diverge is $\frac{f(\beta,1)^{d^\theta}}{2^d} \to \infty$, which holds as soon as $\theta > 1$. Moreover, even at linear sample complexity, it is possible to find regimes that ensure divergence of the LR norm, similar to BBP phase transition in spiked Wishart model. Assume that samples and dimensions scale at the same rate as in spiked Wishart model: $n \asymp \frac{d}{\gamma}$, then a sufficient condition for divergence is that

$$\frac{f(\beta, 1)^{d/\gamma}}{2^d} \to \infty$$

which holds if and only if $f(\beta, 1)^{1/\gamma} > 2$. Hence given β, you can always find

$$\gamma_\beta := \frac{\log\left(f(\beta, 1)\right)}{\log 2} \tag{2.68}$$

And for $\gamma > \gamma_\beta$ there is guarantee that $||L_{n,d}|| \to \infty$. Vice-versa, we could also fix γ and define β_γ as only value of β that makes 2.68 true.

It is spontaneous to ask whether this condition for divergence of LR norm is also necessary, making the SNR threshold β_γ the analogous in this model of the threshold $\beta_c = \sqrt{\gamma}$ in spiked Wishart model (2.11). Although a rigorous proof of this is still missing, numerical evidence (2.7) suggests that for $\beta \leq \beta_\gamma$ the LR norm stays indeed bounded.

2.B.5 Detailed proofs for section 2.3.2

First we prove two lemmas that provide useful formulas and estimates.

Lemma 2.13. *We consider a hypothesis testing problem in $\mathbb{R}^d$ where the null hypothesis $\mathbb{Q}$ is the multivariate Gaussian distribution $\mathcal{N}(0, \mathbb{1}_{d\times d})$, while the alternative hypothesis $\mathbb{P}$ is absolutely continuous with respect to it. Then:*

$$L^{\leq D} = \sum_{|\boldsymbol{\alpha}|\leq D} \frac{\langle L, H_{\boldsymbol{\alpha}}\rangle H_{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!} = \sum_{|\boldsymbol{\alpha}|\leq D} \frac{\mathop{\mathbb{E}}_{x\sim\mathbb{P}}[H_{\boldsymbol{\alpha}}(x)]H_{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!} \tag{2.69}$$

Which implies

$$||L^{\leq D}||^2 = \sum_{|\boldsymbol{\alpha}|\leq D} \frac{\langle L, H_{\boldsymbol{\alpha}}\rangle^2}{\boldsymbol{\alpha}!} = \sum_{|\boldsymbol{\alpha}|\leq D} \frac{\mathop{\mathbb{E}}_{x\sim\mathbb{P}}[H_{\boldsymbol{\alpha}}(x)]^2}{\boldsymbol{\alpha}!} \tag{2.70}$$

Proof. First note that

$$\langle L, H_{\boldsymbol{\alpha}}\rangle = \mathop{\mathbb{E}}_{x\sim\mathbb{P}}[H_{\boldsymbol{\alpha}}(x)] \tag{2.71}$$

due to the definition of likelihood ratio and a change of variable in the expectation. Then we can just use the fact that $(H_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha}} \in \mathbb{N}^d$ are an orthogonal basis for $L^2(\mathbb{R}^d, \mathbb{Q})$, and if we consider the Hermite polynomials up to degree D, they are also a basis of the space of polynomials in which we want to project L to get $L^{\leq D}$. Hence the formulas follow by just computing the projection using this basis. □

Note that we set the lemma in $\mathbb{R}^d$, but of course it holds also in $\mathbb{R}^{nd}$ just switching to matrix-multi-index notation. The following lemma provides estimates and formulas for projections of the sample-wise likelihood ratio $l(\cdot|u)$ on the Hermite polynomials.

Proof of lemma 2.8. Recall that by (2.9)

$$l(y|u) = \mathop{\mathbb{E}}_{g}\left[\sqrt{1+\beta}\exp\left(-\frac{1}{2}\left(\sqrt{1+\beta}g - \sqrt{\frac{\beta}{d}}y\cdot u\right)^2 + \frac{g^2}{2}\right)\right]. \tag{2.72}$$

Note that this expression, thanks to 2.46 that bounds the integral, is differentiable infinitely many times in the y variable. It can be quickly proven by induction on $|\boldsymbol{\alpha}| = m$ (using the recursive definition of Hermite polynomials A.6) that:

$$\begin{aligned}\partial_{\boldsymbol{\alpha}} l(y|u) = &\left(\frac{\beta}{d}\right)^{m/2} u^{\boldsymbol{\alpha}}\cdot \\ &\cdot \mathop{\mathbb{E}}_{g}\left[\sqrt{1+\beta}\; h_m\left(\sqrt{1+\beta}g - \sqrt{\frac{\beta}{d}}y\cdot u\right)\exp\left(-\frac{1}{2}\left(\sqrt{1+\beta}g - \sqrt{\frac{\beta}{d}}y\cdot u\right)^2 + \frac{g^2}{2}\right)\right]\end{aligned} \tag{2.73}$$

Hence we can again use 2.46 together with the fact that

$$\sup_g \left| h_m\left(\sqrt{1+\beta}g - \sqrt{\frac{\beta}{d}}y\cdot u\right)\exp\left(-\frac{1}{2}\left(\sqrt{1+\beta}g - \sqrt{\frac{\beta}{d}}y\cdot u\right)^2\right)\right| < \infty$$

to deduce that the hypothesis of lemma A.8 are met for any $\boldsymbol{\alpha}\in\mathbb{N}^d$, leading to:

$$\langle l(\cdot|u), H_{\boldsymbol{\alpha}}\rangle = \mathop{\mathbb{E}}_{x\sim\mathbb{P}(\cdot|u)}[H_{\boldsymbol{\alpha}}(x)] = \mathop{\mathbb{E}}_{y\sim\mathbb{Q}}[\partial_{\boldsymbol{\alpha}} l(y|u)] \tag{2.74}$$

This, and 2.73 already prove 2.21. Now to compute the exact value of $T_{m,g}$ we need to take the expectation with respect to $y\sim\mathbb{Q}=\mathcal{N}(0,\mathbb{1}_{d\times d})$. Note that by the choice of Rademacher prior on u, we know that $||u||=\sqrt{d}$. So, conditioning on u, $y\cdot\frac{u}{\sqrt{d}}\sim\mathcal{N}(0,1)$. Hence switching the expectations in 2.73, we get:

$$\begin{aligned}
T_{m,g} &= \beta^{m/2}\mathop{\mathbb{E}}_g\left[\int_{-\infty}^{\infty}\frac{\mathrm{d}z}{\sqrt{2\pi}}\sqrt{1+\beta}\; h_m\left(\sqrt{1+\beta}g-\sqrt{\beta}z\right)e^{-\frac{1}{2}\left(\sqrt{1+\beta}g-\sqrt{\beta}z\right)^2+\frac{g^2-z^2}{2}}\right]\\
&= \beta^{m/2}\mathop{\mathbb{E}}_g\left[\int_{-\infty}^{\infty}\frac{\mathrm{d}z}{\sqrt{2\pi}}\sqrt{1+\beta}\; h_m\left(\sqrt{1+\beta}g-\sqrt{\beta}z\right)\exp\left(-\frac{1}{2}\left(\sqrt{\beta}g-\sqrt{1+\beta}z\right)^2\right)\right]\\
&\overset{\tilde{z}=\sqrt{1+\beta}z}{=} \beta^{m/2}\mathop{\mathbb{E}}_g\left[\int_{-\infty}^{\infty}\frac{\mathrm{d}\tilde{z}}{\sqrt{2\pi}}\; h_m\left(\sqrt{1+\beta}g-\sqrt{\frac{\beta}{1+\beta}}\tilde{z}\right)\exp\left(-\frac{1}{2}\left(\sqrt{\beta}g-\tilde{z}\right)^2\right)\right]
\end{aligned} \tag{2.75}$$

Now we use A.8 on

$$\begin{aligned}
x &= \frac{\beta}{\sqrt{1+\beta}}g - \sqrt{\frac{\beta}{1+\beta}}\tilde{z}\\
y &= \sqrt{1+\beta}g - \frac{\beta}{\sqrt{1+\beta}}g = \frac{g}{\sqrt{1+\beta}}
\end{aligned} \tag{2.76}$$

applying also the translation change of variable $\hat{z}=\tilde{z}-\sqrt{\beta}\,g$ we get:

$$\mathop{\mathbb{E}}_{y\sim\mathbb{Q}}[\partial_{\boldsymbol{\alpha}} l(y|u)] = \left(\frac{\beta}{d}\right)^{m/2} u^{\boldsymbol{\alpha}}\mathop{\mathbb{E}}_g\left[\sum_{k=0}^{m}\binom{m}{k}h_k\left(\frac{g}{\sqrt{1+\beta}}\right)\left(-\sqrt{\frac{\beta}{1+\beta}}\right)^{m-k}\mathop{\mathbb{E}}_{\hat{z}\sim\mathcal{N}(0,1)}[\hat{z}^{m-k}]\right] \tag{2.77}$$

Now recall the formula for the moments of the standard Gaussian (see for instance section 3.9 in [124]):

$$\mathop{\mathbb{E}}_{\hat{z}\sim\mathcal{N}(0,1)}[\hat{z}^{m-k}] = \begin{cases}(m-k-1)!! & \text{if } m-k \text{ is even}\\ 0 & \text{if } m-k \text{ is odd}\end{cases} \tag{2.78}$$

Plugging this formula inside and changing the summation index $2j := m-k$ we get

$$T_{m,g} = \beta^{m/2}\sum_{j=0}^{\lfloor m/2\rfloor}\binom{m}{2j}\left(\frac{\beta}{1+\beta}\right)^j (2j-1)!!\mathbb{E}\left[h_{m-2j}\left(\frac{g}{\sqrt{1+\beta}}\right)\right]. \tag{2.79}$$

Note that A.9 with $x=\frac{g}{\sqrt{1+\beta}}$ and $\gamma=\sqrt{1+\beta}$ gives the following rewriting of $h_m(g)$:

$$h_m(g) = \sum_{j=0}^{\lfloor m/2\rfloor}(\sqrt{1-\beta})^{m-2j}(\beta)^j\binom{m}{2j}(2j-1)!!h_{m-2j}\left(\frac{g}{\sqrt{1+\beta}}\right) \tag{2.80}$$

which is almost the same expression as in 2.79. This allows simplify everything, leading to 2.22. □

Note that lemma 2.8 together with 2.1 on g imply:

$$m = 2 \text{ or } m \text{ odd} \Longrightarrow T_{m,g} = 0, \tag{2.81}$$

So, apart from $m = 0$ which gives the trivial contribution $+1$, the first non zero contributions to the LDLR norm is at degree $m = 4$:

$$\mathop{\mathbb{E}}_{x\sim\mathbb{P}(\cdot|u)}[H_{\boldsymbol{\alpha}}(x)] = \frac{\beta^2}{(1+\beta)^2}\kappa_4^g \frac{u^{\boldsymbol{\alpha}}}{d^2} \tag{2.82}$$

From now on we will consider separately the lower and the upper bounds.

Lower bound The idea of the proof is to start from

$$||L_{n,d}^{\leq D}||^2 = \sum_{\substack{\underline{\boldsymbol{\alpha}}\in\mathbb{N}^{nd} \\ |\underline{\boldsymbol{\alpha}}|\leq D}} \frac{\langle L, H_{\underline{\boldsymbol{\alpha}}}\rangle^2}{\underline{\boldsymbol{\alpha}}!} \tag{2.83}$$

and to estimate it from below by picking only few terms in the sum. So we restrict to particular sets of matrix-multi-indices for which we can exploit 2.82. Let $m \in \mathbb{N}$ such that $4m \leq D$, define

$$\mathcal{I}_m = \left\{\underline{\boldsymbol{\alpha}} \in \left(\mathbb{N}^d\right)^n \,\middle|\, |\underline{\boldsymbol{\alpha}}| = 4m,\ |\boldsymbol{\alpha}^\mu| \in \{0,4\}\ \forall\ 1 \leq \mu \leq n\right\} \tag{2.84}$$

$\mathcal{I}_m$ is non empty since $m < n$ and for each $\underline{\boldsymbol{\alpha}} \in \mathcal{I}_m$ we can enumerate all the indices $\mu_1, \ldots, \mu_m$ such that $\boldsymbol{\alpha}^{\mu_i} \neq 0$.

Now we go on to compute the term $\langle L, H_{\underline{\boldsymbol{\alpha}}}\rangle$ for $\underline{\boldsymbol{\alpha}} \in \mathcal{I}_m$:

$$\begin{aligned}\langle L, H_{\underline{\boldsymbol{\alpha}}}\rangle &= \int_{\mathcal{U}} \mathop{\mathbb{E}}_{\underline{x}\sim\mathbb{P}(\cdot|u)^{\otimes n}} [H_{\underline{\boldsymbol{\alpha}}}(\underline{x})]\mathrm{d}\mathcal{P}(u) \\ &= \int_{\mathcal{U}} \mathop{\mathbb{E}}_{\underline{x}\sim\mathbb{P}(\cdot|u)^{\otimes n}} \left[\prod_{i=1}^m H_{\boldsymbol{\alpha}^{\mu_i}}(x^{\mu_i})\right] \mathrm{d}\mathcal{P}(u)\end{aligned} \tag{2.85}$$

Since the samples are independent conditionally on u, we can split the inner expectation along the m contributing directions:

$$\langle L, H_{\underline{\boldsymbol{\alpha}}}\rangle = \int_{\mathcal{U}} \prod_{i=1}^m \mathop{\mathbb{E}}_{x\sim\mathbb{P}(\cdot|u)} [H_{\boldsymbol{\alpha}^{\mu_i}}(x^{\mu_i})]\mathrm{d}\mathcal{P}(u) \tag{2.86}$$

Now we need to compute the inner d-dimensional expectation. For that we use that $|\boldsymbol{\alpha}^{\mu_i}| = 4$, so, recalling the notation $\eta = \sqrt{\frac{\beta}{1+\beta}}$, we can apply 2.82 to get for each i

$$\mathop{\mathbb{E}}_{x\sim\mathbb{P}(\cdot|u)} [H_{\boldsymbol{\alpha}^{\mu_i}}(x^{\mu_i})] = \frac{\eta^4}{d^2}\kappa_4^g u^{\alpha^{\mu_i}}. \tag{2.87}$$

So the resulting integral can be written in the following way:

$$\langle L, H_{\underline{\boldsymbol{\alpha}}}\rangle = \left(\frac{\eta^4\kappa_4^g}{d^2}\right)^m \int_{\mathcal{U}} \prod_{j=1}^d u_j^{\gamma_j} \mathrm{d}\mathcal{P}(u) \tag{2.88}$$

where for each $j \in \{1, \ldots, d\}$, $\gamma_j := \sum_\mu \boldsymbol{\alpha}_j^\mu$. So we also have that $\sum_j \gamma_j = 4m$.

Now we take the expectation with respect to $\mathcal{P}(u)$, and use the fact that the components are i.i.d. Rademacher so the result depends on the parity of the γ_j in the following way:

$$\langle L, H_{\underline{\boldsymbol{\alpha}}}\rangle = \begin{cases} \left(\frac{\eta^4\kappa_4^g}{d^2}\right)^m & \text{if all } (\gamma_j)_{j=1,\dots,n} \text{ are even} \\ 0 & \text{if there is at least one } \gamma_j \text{ which is an odd number} \end{cases} \tag{2.89}$$

Hence, if we restrict to the set:

$$\tilde{\mathcal{I}}_m = \left\{ \underline{\boldsymbol{\alpha}} \in \mathcal{I}_m | \forall j \in \{1,\dots,d\} \sum_{\mu=1}^{n} \underline{\boldsymbol{\alpha}}_j^\mu \text{ is even} \right\} \tag{2.90}$$

We have that all the indices belonging to $\tilde{\mathcal{I}}_m$ give the same contribution:

$$\langle L, H_{\underline{\boldsymbol{\alpha}}}\rangle^2 = \left(\frac{\eta^4}{d^2}\kappa_4^g\right)^{2m} \tag{2.91}$$

Also, note that inside $\mathcal{I}_m$, $\boldsymbol{\alpha}! \leq (4!)^m$, so get the following estimates

$$\begin{aligned} ||L_{n,d}^{\leq D}||^2 &= \sum_{|\underline{\boldsymbol{\alpha}}|\leq D} \frac{\langle L, H_{\underline{\boldsymbol{\alpha}}}\rangle^2}{\underline{\boldsymbol{\alpha}}!} \\ &\geq \sum_{m=0}^{\lfloor D/4\rfloor} \sum_{\underline{\boldsymbol{\alpha}}\in\tilde{\mathcal{I}}_m} \frac{\langle L, H_{\underline{\boldsymbol{\alpha}}}\rangle^2}{\underline{\boldsymbol{\alpha}}!} \\ &\geq \sum_{m=0}^{\lfloor D/4\rfloor} \sum_{\underline{\boldsymbol{\alpha}}\in\tilde{\mathcal{I}}_m} \left(\frac{\eta^4\kappa_4^g}{\sqrt{4!}d^2}\right)^{2m} \\ &\geq \sum_{m=0}^{\lfloor D/4\rfloor} \#\tilde{\mathcal{I}}_m \left(\frac{\eta^4\kappa_4^g}{\sqrt{4!}d^2}\right)^{2m} \end{aligned} \tag{2.92}$$

Now we just need to estimate the cardinality of $\tilde{\mathcal{I}}_m$. A lower bound can be provided by considering

$$\hat{\mathcal{I}}_m = \left\{ \underline{\boldsymbol{\alpha}} \in \mathcal{I}_m | \ \forall\ \mu \in [n]\ \forall\ j \in [d]\ \underline{\boldsymbol{\alpha}}_j^\mu \text{ is even} \right\} \tag{2.93}$$

Clearly $\hat{\mathcal{I}}_m \subseteq \tilde{\mathcal{I}_m}$ and $\#\hat{\mathcal{I}_m} = \binom{n}{m}\binom{d+1}{2}^m$ because first we can pick the $n-m$ rows of $\underline{\boldsymbol{\alpha}}$ that will be $0 \in \mathbb{N}^d$, which can be done in $\binom{n}{m}$. Then, for each non-zero row, we need to pick two columns (with repetitions) in which to place a 2 and leave all the other entries as 0, that can be done in $\binom{d+1}{2}^m$ ways.

So plugging the lower bound in the previous estimate, we reach the inequality that we wanted to prove:

$$||L_{n,d}^{\leq D}||^2 \geq \sum_{m=0}^{\lfloor D/4\rfloor} \binom{n}{m}\binom{d+1}{2}^m \left(\frac{\eta^4\kappa_4^g}{\sqrt{4!}d^2}\right)^{2m} = \sum_{m=0}^{\lfloor D/4\rfloor} \binom{n}{m}\binom{d+1}{2}^m \left(\frac{\beta^2\kappa_4^g}{\sqrt{4!}d^2(1+\beta)^2}\right)^{2m} \tag{2.94}$$

Upper bound We start from (2.19) using the following rewriting:

$$\begin{aligned} ||L_{n,d}^{\leq D}||^2 &= \sum_{|\underline{\boldsymbol{\alpha}}|\leq D} \frac{\langle L, H_{\underline{\boldsymbol{\alpha}}}\rangle^2}{\underline{\boldsymbol{\alpha}}!} \\ &= \sum_{|\underline{\boldsymbol{\alpha}}|\leq D} \frac{\mathop{\mathbb{E}}_{\underline{x}\sim\mathbb{P}} \left[H_{\underline{\boldsymbol{\alpha}}}(\underline{x})\right]^2}{\underline{\boldsymbol{\alpha}}!} \\ &= \sum_{|\underline{\boldsymbol{\alpha}}|\leq D} \frac{1}{\underline{\boldsymbol{\alpha}}!} \mathop{\mathbb{E}}_{u\sim\mathcal{P}(u)} \left[\prod_{\mu=1}^{n} \mathop{\mathbb{E}}_{x^\mu\sim\mathbb{P}(\cdot|u)} \left[H_{\boldsymbol{\alpha}^\mu}(x^\mu)\right]\right]^2 \end{aligned} \tag{2.95}$$

Now use 2.8 and plug in the formulas for the inner expectations.

$$\begin{aligned} ||L_{n,d}^{\leq D}||^2 &= \sum_{|\underline{\boldsymbol{\alpha}}|\leq D} \frac{1}{\underline{\boldsymbol{\alpha}}!} \mathop{\mathbb{E}}_{u\sim\mathcal{P}(u)} \left[\prod_{\mu=1}^{n} \frac{T_{|\boldsymbol{\alpha}^\mu|,g}}{d^{|\boldsymbol{\alpha}^\mu|/2}} u^{\boldsymbol{\alpha}^\mu}\right]^2 \\ &= \sum_{|\underline{\boldsymbol{\alpha}}|\leq D} \frac{\left(\prod_{\mu=1}^{n} T_{|\boldsymbol{\alpha}^\mu|,g}\right)^2}{\underline{\boldsymbol{\alpha}}! d^{|\underline{\boldsymbol{\alpha}}|}} \mathop{\mathbb{E}}_{u\sim\mathcal{P}(u)} \left[u^{\underline{\boldsymbol{\alpha}}}\right]^2 \end{aligned} \tag{2.96}$$

Now we use our prior assumption that $u_i \overset{\text{i.i.d.}}{\sim} \text{Rad}(1/2)$. Note that odd moments of u_i are equal to 0 and even moments are equal to 1, so (denoting by $\chi_A(\cdot)$ the indicator function of set A):

$$\mathop{\mathbb{E}}_{u\sim\mathcal{P}(u)} \left[u^{\underline{\boldsymbol{\alpha}}}\right] = \prod_{i=1}^{d} \mathop{\mathbb{E}}_{u_i\sim\text{Rademacher}(1/2)} \left[u_i^{\sum_{\mu=1}^{n} \boldsymbol{\alpha}_i^\mu}\right] = \chi_{\left\{\sum_{\mu=1}^{n} \boldsymbol{\alpha}_i^\mu \text{ is even for all } i\right\}}(\underline{\boldsymbol{\alpha}}) \tag{2.97}$$

Now the key point of the proof: this last formula, together with (2.81), implies that most of the addends in the sum in (2.96) are zero. Set $|\underline{\boldsymbol{\alpha}}| = m$, then the set of multi-indices that could give a non-zero contribution is

$$\mathcal{A}_m = \left\{ \underline{\boldsymbol{\alpha}} \in \mathbb{N}^{n\times d} \middle| |\underline{\boldsymbol{\alpha}}| = m,\ \forall\ \mu \in [n], \underline{\boldsymbol{\alpha}}^\mu = 0 \text{ or } |\underline{\boldsymbol{\alpha}}^\mu| > 4, \text{ and } \forall\ i \in [d] \sum_{\mu=1}^{n} \boldsymbol{\alpha}_i^\mu \text{ is even} \right\} \tag{2.98}$$

Using this fact together with (2.22) we get:

$$\begin{aligned} ||L_{n,d}^{\leq D}||^2 &= \sum_{m=0}^{D} \sum_{\underline{\boldsymbol{\alpha}}\in\mathcal{A}_m} \frac{\left(\prod_{\mu=1}^{n} T_{|\boldsymbol{\alpha}^\mu|,g}\right)^2}{\underline{\boldsymbol{\alpha}}! d^{|\underline{\boldsymbol{\alpha}}|}} \\ &\leq \sum_{m=0}^{D} \sum_{\underline{\boldsymbol{\alpha}}\in\mathcal{A}_m} \frac{1}{d^m} \prod_{\mu=1}^{n} \left(\left(\frac{\beta}{1+\beta}\right)^{|\boldsymbol{\alpha}^\mu|} \mathbb{E}\left[h_{|\boldsymbol{\alpha}^\mu|}(g)\right]^2\right) \\ &\leq \sum_{m=0}^{D} \sum_{\underline{\boldsymbol{\alpha}}\in\mathcal{A}_m} \frac{1}{d^m} \left(\frac{\beta}{1+\beta}\right)^m \sup_{k\leq m} \left(\mathbb{E}\left[h_k(g)\right]^{2m/k}\right) \\ &= \sum_{m=0}^{D} \frac{1}{d^m} \left(\frac{\beta}{1+\beta}\right)^m \sup_{k\leq m} \left(\mathbb{E}\left[h_k(g)\right]^{2m/k}\right) \#\mathcal{A}_m \end{aligned} \tag{2.99}$$

In the third step we used the following inequality

$$\prod_{\mu} \mathbb{E}\left[h_{|\boldsymbol{\alpha}^\mu|}(g)\right]^2 \leq \prod_{\mu} \sup_{k\leq m} \left(\mathbb{E}\left[h_k(g)\right]^{1/k}\right)^{2|\boldsymbol{\alpha}^\mu|} = \sup_{k\leq m} \mathbb{E}\left[h_k(g)\right]^{2m/k}$$

That holds since $\underline{\boldsymbol{\alpha}} \in \mathcal{A}_m$ hence $\sum_\mu |\boldsymbol{\alpha}^\mu| = m$.

It is hard to compute exactly the cardinality of $\mathcal{A}_m$, but we can estimate it by considering the inclusion $\mathcal{A}_m \subseteq \tilde{\mathcal{A}}_m$ defined as

$$\tilde{\mathcal{A}}_m = \left\{ \underline{\boldsymbol{\alpha}} \in \mathbb{N}^{n\times d} \middle| |\underline{\boldsymbol{\alpha}}| = m, \text{ there are at most } \left\lfloor \frac{m}{4} \right\rfloor \text{ non-zero rows and } \left\lfloor \frac{m}{2} \right\rfloor \text{ non-zero columns} \right\} \tag{2.100}$$

Assume now $m > 0$ (the case $m = 0$ can be treated separately, leading to the addend $+1$ in (2.15)). To compute the cardinality of $\tilde{\mathcal{A}}_m$ we just have to multiply the different contributions

- There are $\binom{n}{\lfloor m/4 \rfloor}$ possibilities for the non-zero rows
- There are $\binom{d}{\lfloor m/2 \rfloor}$ possibilities for the non-zero columns
- once restricted to a $\lfloor m/4 \rfloor \times \lfloor m/2 \rfloor$ we have to place the units to get to norm m. It is the counting problem of placing m units inside the $\lfloor m/4 \rfloor \lfloor m/2 \rfloor$ matrix entries. The possibilities are

$$\binom{\lfloor m/4 \rfloor \lfloor m/2 \rfloor + m - 1}{m} \tag{2.101}$$

So we get the following estimate for the cardinality of $\mathcal{A}_m$

$$\#\mathcal{A}_m \leq \binom{n}{\lfloor m/4 \rfloor} \binom{d}{\lfloor m/2 \rfloor} \binom{\lfloor m/4 \rfloor \lfloor m/2 \rfloor + m - 1}{m} \tag{2.102}$$

Plugging into (2.99) we reach the final formula (2.15), which completes the proof.

Proof of corollary 2.7 Now we turn to the proof of 2.7 on the asymptotic behavior of the bound

$$\binom{n}{m} = \frac{n^m}{m!} + O(n^{m-1}) \tag{2.103}$$

$$\binom{d+1}{2}^m \geq \frac{d^{2m}}{2^m} \tag{2.104}$$

So we have that:

$$\begin{aligned} ||L_{n,d}^{\leq D(n)}||^2 &\geq \sum_{m=0}^{\lfloor D/4 \rfloor} \left(\frac{n^m}{m!} + O(n^{m-1}) \right) \left(\frac{d^{2m}}{2^m} + O(1) \right) \left(\frac{\beta^2 \kappa_4^g}{\sqrt{4!} d^2 (1+\beta)^2} \right)^{2m} \\ &= \left(\sum_{m=0}^{\lfloor D/4 \rfloor} \frac{1}{m!} \left(\frac{\beta^2 \kappa_4^g}{\sqrt{2}\sqrt{4!}(1+\beta)^2} \right)^{2m} \left(\frac{n}{d^2} \right)^m \right) + O\left(\sum_m \frac{n^{m-1}}{m! d^{2m}} \left(\frac{\beta^2 \kappa_4^g}{\sqrt{2}\sqrt{4!}(1+\beta)^2} \right)^{2m} \right) \end{aligned} \tag{2.105}$$

Then we lower-bound the sum by considering just the last term.

$$||L_{n,d}^{\leq D(n)}||^2 \geq \frac{1}{\lfloor D/4 \rfloor!} \left(\frac{\beta^2 \kappa_4^g}{\sqrt{2}\sqrt{4!}(1+\beta)^2} \right)^{2\lfloor D/4 \rfloor} \left(\frac{n}{d^2} \right)^{\lfloor D/4 \rfloor} + o\left(\frac{1}{\lfloor D/4 \rfloor!} \left(\frac{Cn}{d^2} \right)^{\lfloor D/4 \rfloor} \right) \tag{2.106}$$

So by using $k! \leq k^k$ on $\lfloor D/4 \rfloor$ and picking n, d large enough so that numerical constants and the $o(\dots)$ become negligible for the estimate, we get (2.16).

Plugging in the scaling $n \asymp d^\theta$ and $D(n) \asymp \log^{1+\varepsilon}(n)$ it is clear that what decides the behaviour of the sequence is the term $\frac{n}{d^2}$.

- If $0 < \theta \leq 2$, $\frac{n}{d^2} \to 0$ and so (2.16) does not provide information on the divergence of the LDLR norm.
- If $\theta > 2$ it is $\frac{n}{d^2} \to \infty$ faster that the logarithmic term at denominator, so the right hand side in (2.16) diverges at ∞, proving the second regime in (2.18)

Now let us turn to proving (2.17). We are in the regime $m \leq D << \min(n, d)$, hence we can use the following estimates of binomial coefficients and factorial

$$\begin{aligned} \binom{n}{\lfloor m/4 \rfloor} &\leq n^{m/4} \\ \binom{d}{\lfloor m/2 \rfloor} &\leq d^{m/2} \\ \binom{\lfloor m/4 \rfloor \lfloor m/2 \rfloor + m - 1}{m} &\leq \left(m^2/8\right)^m \\ m! &\leq m^m \end{aligned} \tag{2.107}$$

on 2.15 to get

$$||L_{n,d}^{\leq D}||^2 \leq 1 + \sum_{m=1}^{D} \left(\frac{\beta}{1+\beta}\right)^m m^{2m} \sup_{k \leq m} \left(\mathbb{E}\left[h_k(g)\right]^{2m/k}\right) \left(\frac{n}{d^2}\right)^{m/4} \tag{2.108}$$

Now we use estimate the term depending on the Hermite coefficients of g, using the assumption 2.5:

$$\begin{aligned} \sup_{k \leq m} \left(\mathbb{E}\left[h_k(g)\right]^{2m/k}\right) &\leq \sup_{k \leq m} \left((\Lambda^k k!)^{2m/k}\right) \\ &\leq \Lambda^{2m} \sup_{k \leq m} \left(k^{2m}\right) \\ &\leq \Lambda^{2m} m^{2m} \end{aligned} \tag{2.109}$$

Plugging into the estimate for the LDLR we get (2.17)

$$||L_{n,d}^{\leq D}||^2 \leq 1 + \sum_{m=1}^{D} \left(\frac{\Lambda^2 \beta}{1+\beta}\right)^m m^{4m} \left(\frac{n}{d^2}\right)^{m/4} \tag{2.110}$$

By letting $n \asymp d^\theta$, for $\theta > 0$ and $D(n) \asymp \log^{1+\varepsilon}(n)$, we can see that the bound goes to 1 when $\theta < 2$, proving the first regime in (2.18).

2.B.6 Limitations of our theoretical analysis

The main limitation of the theoretical portion of our work is that it relies on the assumption that the null hypothesis is standard i.i.d. Gaussian noise. Since this assumption is central to the large body of work analysing hypothesis tests [69], a very interesting future direction is to develop tools for analysing the case of a different null hypothesis. On the technical side, while the assumption of Rademacher prior on u is not essential and could be easily generalized to other isotropic distributions, assumption 2.1 is a key requirement to carry out the proofs. As is generally the case in high-dimensional statistics, our results do not constitute a complete proof of the existence of a statistical-to-computational gap: our argument relies on the low-degree conjecture 2.5, and the divergence of the norm of the likelihood ratio is only a necessary condition for *strong distinguishability*.

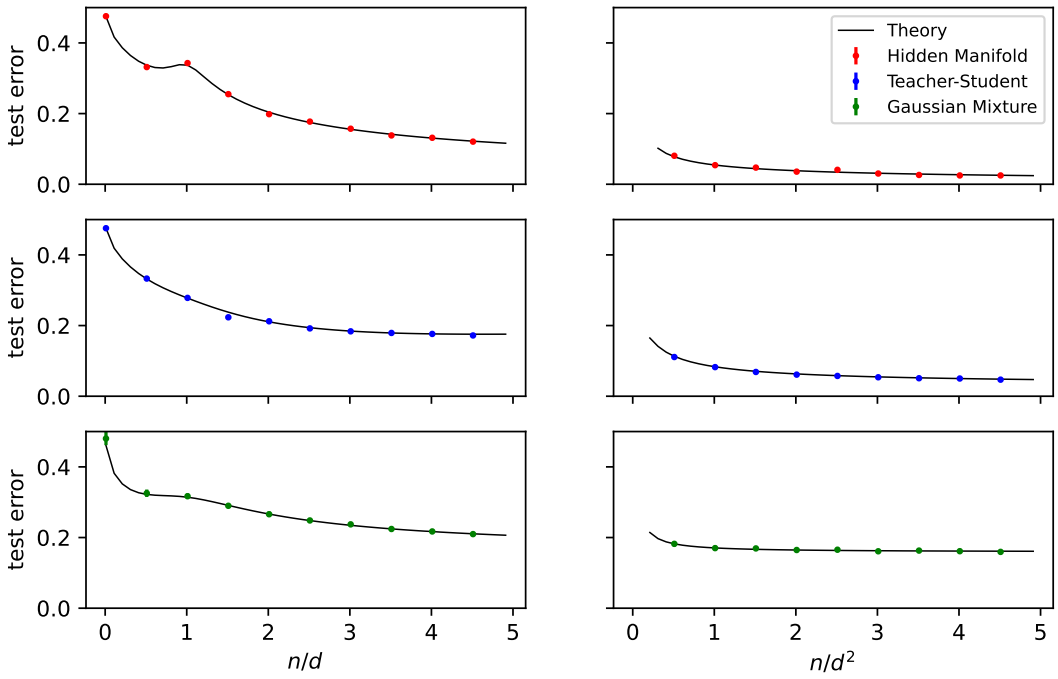

Figure 2.8: **Linear and quadratic sample regimes for different synthetic data models.** (*Right*) Generalization error of the hidden manifold model (top), the teacher-student setup (center) and a mixture of two Gaussians as a function of the ratio of the number of samples and the input dimension. (*Left*) Same except that the number of samples scales with the square of the input dimension. The solid black line corresponds to the replica theory prediction while the coloured dots display the outcome of the numerical simulations averaged over 10 different seeds. In all the panels, $d = 1000$ and $d = 20$ for linear and quadratic sample regimes respectively, $\lambda = 0.01$ for both the teacher–student setup and the hidden manifold model while $\lambda = 0.1$ for Gaussian mixtures. In the Gaussian mixture case, $\mu_{\pm} = (\pm 1, 0, ..., 0) \in \mathbb{R}^d$ while the covariance matrices are both isotropic and, in particular, both equal to the identity matrix: $\Sigma_{\pm} = \mathbb{I}$.

In fact, there are no techniques at the moment that can prove that average-case problems require super-polynomial time in the hard phase, even if we assume $P \neq NP$ [69]. So our work should be seen as providing rigorous evidence for a statistical-to-computational gap in the spiked cumulant model.

2.C Details on the replica analysis

We analytically compute the generalisation error of a random feature model trained on the Gaussian mixture classification task of 2.B.2.3 by exploiting the Gaussian equivalence theorem for dealing with structured data distributions [13, 108–111]. In particular, we use equations derived by Loureiro *et al.* [107] that describe the generalisation error of random features on Gaussian mixture classification using the replica method of statistical physics. The equations describe the high-dimensional limit where n, d, and the size of the hidden layer $p \to \infty$ while their ratios stay finite, $\alpha = n/d$ and

$\gamma = d/p \sim O(1)$. In this regime, the generalisation error is a function of only scalar quantities, i.e.

$$\epsilon_g = 1 - \rho_+ \mathbb{E}_\xi \left[\text{sign}\left(m_+^\star + \sqrt{q_+^\star}\xi + b^\star \right) \right] + \rho_- \mathbb{E}_\xi \left[\text{sign}\left(m_-^\star + \sqrt{q_-^\star}\xi + b^\star \right) \right], \tag{2.111}$$

where $\xi \sim \mathcal{N}(0,1)$, $\rho_\pm$ defines the fraction of data points belonging to class $y = \pm 1$ while $m_\pm^\star$, $q_\pm^\star$ and $b^\star$ are the so-called overlap parameters and bias term respectively. Their value for every size of the training set can be determined by solving the following optimisation problem [107]:

$$f_\beta = \underset{\{q_k, m_k, V_k, \hat{q}_k, \hat{m}_k, \hat{V}_k, b\}_{k=+,-}}{\text{extr}} \left[\sum_{k=+,-} -\frac{1}{2}\left(\hat{q}_k V_k - q_k \hat{V}_k \right) + m_k \hat{m}_k + \lim_{p\to\infty} \frac{1}{p} \Psi_s + \alpha\gamma \Psi_e \right]. \tag{2.112}$$

where the entropic potential Ψ_s can be expressed as a function of the means $\boldsymbol{\mu}_\pm \in \mathbb{R}^p$ and covariances $\Sigma_\pm \in \mathbb{R}^{p\times p}$ of the hidden-layer activations of the random features model, while the energetic potential Ψ_e depends on the specific choice of the loss function $\ell(\cdot)$.

Solving the optimization problem in 2.112, leads to a set of coupled saddle-point equations for the overlap parameters and the bias term. This set of equations is nothing but a special case of the equations already derived in [107], that is high-dimensional classification of a mixture of two Gaussians with random features, except for the typo in the self-consistency equation of the bias term

$$b = \left(\sum_{k=\pm} \frac{\rho_k}{V_k} \right)^{-1} \sum_{k=\pm} \rho_k \mathbb{E}_\xi \left[\eta_k - m_k \right], \tag{2.113}$$

with η_k being the extremizer

$$\eta_k = \underset{\lambda}{\text{argmin}} \left[\frac{\left(\lambda - \sqrt{q}_k \xi - m_k - b \right)^2}{2V_+} + \ell(y_k, \lambda) \right]. \tag{2.114}$$

and $\ell(\cdot)$ being any convex loss function. Analogously to the more general setting of [107], the Gaussian Equivalence Theorem allows to express the entropic potential Ψ_s as a function of the means $\boldsymbol{\mu}_k \in \mathbb{R}^p$ and covariances $\Sigma_k \in \mathbb{R}^{p\times p}$ of the random features

$$\begin{aligned} \Psi_s = &\lim_{p\to\infty} \frac{1}{p} \left(\hat{q}_+ \boldsymbol{\mu}_+ + \hat{q}_- \boldsymbol{\mu}_- \right)^t \left(\lambda \mathbb{I}_p + \hat{V}_+ \Sigma_+ + \hat{V}_- \Sigma_- \right)^{-1} \left(\hat{q}_+ \boldsymbol{\mu}_+ + \hat{q}_- \boldsymbol{\mu}_- \right) + \\ &+ \lim_{p\to\infty} \frac{1}{p} \text{Tr}\left(\left(\hat{q}_+ \Sigma_+ + \hat{q}_- \Sigma_- \right) \left(\lambda \mathbb{I}_p + \hat{V}_+ \Sigma_+ + \hat{V}_- \Sigma_- \right)^{-1} \right); \end{aligned} \tag{2.115}$$

while the energetic potential Ψ_e depends on the specific choice of the loss function $\ell(\cdot)$

$$\Psi_e = -\frac{1}{2} \mathbb{E}_\xi \left[\sum_{k=\pm} \rho_k \frac{\left(\eta_k - \sqrt{q}_k \xi - m_k - b \right)^2}{2V_k} + \ell(y_k, \eta_k) \right]. \tag{2.116}$$

As discussed in 2.4.1, the Gaussian Equivalence Theorem breaks in the quadratic sample regime if the data are sampled from the spiked Wishart model. Interestingly, this is not the case for the Hidden Manifold model. Indeed, as shown in fig. 2.8, we still get a perfect match between the Gaussian theory and the numerical simulations even in the quadratic sample regime. The theoretical prediction can be easily derived by rescaling the free-energy associated to the Hidden Manifold model as in [109] by a factor $1/d^2$. This is the same trick already proposed in [112] for teacher–student settings and displayed in the middle panel of fig. 2.8.

Chapter 3

Neural Network dynamics

3.1 Introduction

Neural networks excel at learning rich representations of their data, but which parts of a data set are actually important for them? From a statistical point of view, we can decompose the data distribution into cumulants, which capture correlations between groups of variables. The first cumulant is the mean, the second describes pair-wise correlations, and *higher-order cumulants* (HOCs) encode correlations between three or more variables. In image classification, HOCs are particularly important: on CIFAR10, removing HOCs from the training distribution incurs a drop in test accuracy of up to 65 % for DenseNets, ResNets, and Vision transformers [28], as we also saw in fig. 1.3.

While HOCs are important for the performance of neural networks, extracting information from HOCs with stochastic gradient descent (SGD) is computationally hard, as presented for the case of the spiked cumulant model in chapter 2: where we showed that the number of samples required to strongly distinguish two classes of inputs $x \in \mathbb{R}^d$ whose distributions have the same mean and covariance, but different fourth-and higher-order cumulants, scales as $n \gtrsim d^2$ for the wide class of polynomial-time algorithms covered by the low-degree conjecture [38–41].

The theoretical difficulty of learning from HOCs is in apparent contradiction to the importance of HOCs for the performance of neural networks, and the speed with which neural networks pick them up [28, 45, 125, 126], lead us to the following question:

How do neural networks extract information from higher-order input correlations efficiently*?*

In this chapter, we show that neural networks can learn efficiently from higher-order cumulants by exploiting **correlations between the latent variables of type $u \cdot x$ corresponding to input cumulants of different orders**.

3.2 Results and insights: an informal overview

3.2.1 Deep convolutional neural networks recover the principal subspace of their inputs before or when they find non-Gaussian directions

In fig. 3.1 we consider the three different deep convolutional networks trained on a classification task on ImageNet. We plot the excess kurtosis of the input dotted with the first layer weights. We see that networks manage to access the HOCs between 10^3 and 10^4 steps, which is roughly when

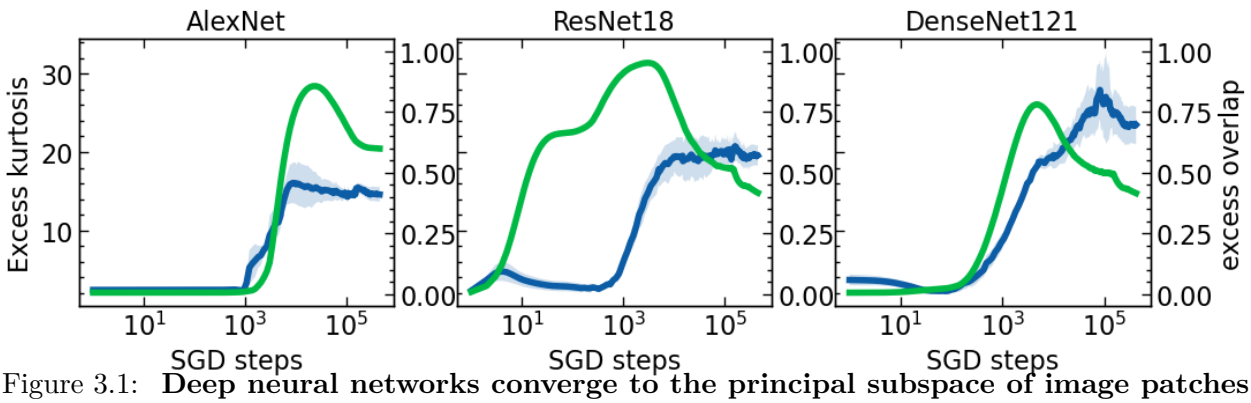

Figure 3.1: **Deep neural networks converge to the principal subspace of image patches before or when they find non-Gaussian directions.** For three deep convolutional neural networks trained on ImageNet, we plot the overlap with the principal subspace of image patches, (eq. (3.50), blue), obtained via PCA, and the excess kurtosis of the first-layer representations, which measures their non-Gaussianity (eq. (3.51), green). Details in section 3.B.1.

features start to form in the convolutional filters (we will discuss this in more detail in chapter 4), but it is also a seemingly low sample complexity if we expect the *quadratic sample complexity* discussed in chapter 2. A possible explanation on how the networks may learn faster than in the theoretical spiked cumulant model is that ImageNet data, along with HOCs, contains also low-order structure (mean, covariance), that could interact with higher order cumulants, possibly speeding up their learning. Indeed, if we compare the learning of excess kurtosis (blue line) with the learning of the principal components of the data (green line) we see that *deep networks recover the principal subspace before or when they find the non-Gaussian direction.* In the following we develop a theoretical model to investigate how this interdependence may arise.

3.2.2 The mixed cumulant model (MCM)

We illustrate how correlated latent variables speed up learning by introducing a simple model for the data, the mixed-cumulant model (MCM). The MCM is a binary discrimination task where the signals that differentiate the two classes are carried by different input cumulants. Specifically, we draw n data points $x^\mu = (x_i^\mu) \in \mathbb{R}^d$ with $\mu = 0, 1, \dots, n$ either from the isotropic Gaussian distribution $\mathbb{Q}_0$ (label $y^\mu = -1$) or from a *planted distribution* $\mathbb{Q}_{\text{plant}}$ (label $y^\mu = 1$). Under $\mathbb{Q}_0$, inputs have zero mean, an isotropic covariance equal to the identity $\mathbb{1}$, and all HOCs are zero. Under $\mathbb{Q}_{\text{plant}}$, the first few cumulants each carry a signal that distinguishes the inputs from those in $\mathbb{Q}_0$: a non-zero mean $m \in \mathbb{R}^d$, a covariance which is isotropic except in the direction $u \in \mathbb{R}^d$, and higher-order cumulants proportional to $v^{\otimes k}, k \geq 2$.

We sample an input from $\mathbb{Q}_{\text{plant}}$ by first drawing an i.i.d. Gaussian vector z^μ and two scalar latent variables, the normally distributed $\lambda^\mu \sim \mathcal{N}(0,1)$ and ν^μ, which is drawn from a non-Gaussian distribution. For concreteness, we will assume that $\nu^\mu = \pm 1$ with equal probability. Both latent variables are independent of z. Then, for $y^\mu = 1$, we have

$$x^\mu = \overbrace{\underbrace{\beta_m m}_{\text{mean}} + \underbrace{\sqrt{\beta_u}\lambda^\mu u}_{\text{covariance}}}^{\text{Gaussian}} + \underbrace{S(\sqrt{\beta_v}\nu^\mu v + z^\mu)}_{\text{HOCs}}. \tag{3.1}$$

where $\beta_i \geq 0$ are the signal-to-noise ratios associated to the three directions, or "spikes", m, u, v. The spikes are fixed and drawn uniformly from the unit sphere. We will sometimes force them to be

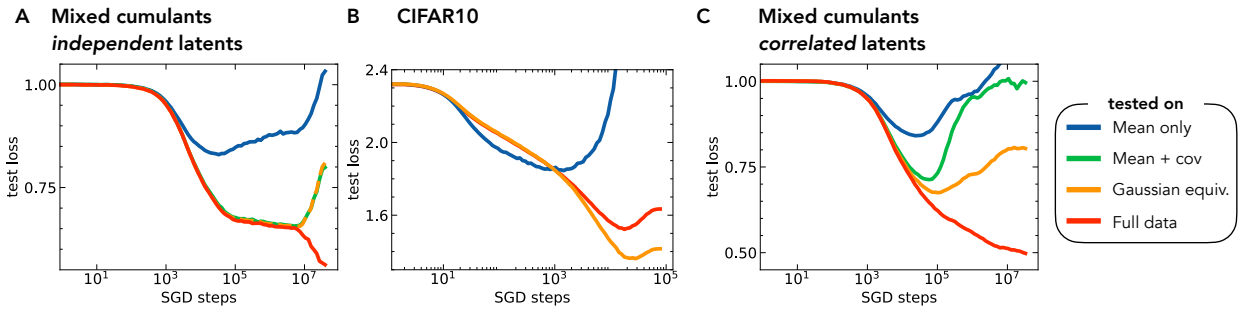

Figure 3.2: **Correlated latent variables speed up learning of neural networks. A** Test error of a two-layer neural network trained on the mixed cumulant model (MCM) of eq. (3.1) with signal-to-noise ratios $\beta_m = 1, \beta_u = 5, \beta_v = 10$. The MCM is a binary classification tasks where the inputs in the two classes have a different mean, a different covariance, and different higher-order cumulants (HOCs). We show the test error on the full data set (red) and on several "censored" data sets: a test set where only the mean of the inputs is different in each class (blue, $\beta_m = 1, \beta_u = \beta_v = 0$), a test set where mean and covariance are different (green, $\beta_m = 1, \beta_u = 5, \beta_v = 0$), and a Gaussian mixture that is fitted to the true data set (orange). The neural networks learn distributions of increasing complexity: initially, only the difference means matter, as the blue and red curves coincide; later, the network learns about differences at the level of the covariance, and finally at the level of higher-order cumulants. **B** Test loss of a two-layer neural network trained on CIFAR10 and evaluated on CIFAR10 (red), a Gaussian mixture with the means fitted to CIFAR10 (blue) and a Gaussian mixture with the means and covariance fitted on CIFAR10 (orange). **C** Same setup as in **A**, but here the latent variables corresponding to the covariance and the cumulants of the inputs are correlated, leading to a significant speed-up of learning from HOCs (the red and orange line separate after $\gtrsim 10^4$ steps, rather than $\gtrsim 10^6$ steps). *Parameters:* $\beta_m = 1, \beta_u = 5, \beta_v = 10, d = 128, m = 512$ hidden neurons, ReLU activation function. Full details in section 3.B.2.

orthogonal to one another. If $\beta_v = 0$, it is easy to verify that inputs are Gaussian with mean $\beta_m m$ and covariance $\mathbb{1} + \beta_u u u^\top$. If $\beta_v > 0$, inputs are non-Gaussian but the presence of the whitening matrix

$$S = \mathbb{1} - \frac{\beta_v}{1 + \beta_v + \sqrt{1 + \beta_v}} v v^\top \tag{3.2}$$

removes the direction v from the covariance matrix, so that v cannot be recovered from the input covariance if the latent variables λ^ν, ν^μ are uncorrelated.

3.2.3 Neural networks take a long time to learn the cumulant spike in the vanilla MCM

We show the performance of a two-layer neural network trained on the MCM model in fig. 3.2A with signal-to-noise ratios $\beta_m = 1, \beta_u = 5, \beta_v = 10$ and independent latent variables $\lambda^\mu \sim \mathcal{N}(0, 1)$ and $\nu^\mu = \pm 1$ with equal probability (red line). We can evaluate which of the three directions have been learnt by the network at any point in time by *evaluating* the same network on a reduced test set where only a subset of the spikes are present. Testing the network on a test set where the only difference between the two classes $\mathbb{Q}_0$ and $\mathbb{Q}_{\text{plant}}$ are the mean of the inputs (blue, $\beta_u = \beta_v = 0$) or the mean and covariance ($\beta_v = 0$), we find that two-layer networks learn about the different directions in a sequential way, learning first about the mean, then the covariance, and finally the higher-order cumulants. This is an example of the distributional simplicity bias [28, 45, 126, 127].

However, note that the periods of sudden improvement, where the networks discovers a new direction in the data, are followed by long plateaus where the test error does not improve. The plateaus notably delay the learning of the direction v that is carried by the higher-order cumulants. We also see an "overfitting" on the censored test sets: as the network discovers a direction that is not present in the censored data sets, this appears as overfitting on the censored tests.

3.2.4 Correlated latent variables speed up learning non-Gaussian directions

As it stands, the MCM with orthogonal spikes and independent latent variables is a poor model for real data: neural networks show long plateaus between learning the Gaussian and non-Gaussian part of the data on the MCM, but if we train the same network on CIFAR10, we see a smooth decay of the test loss, see fig. 3.2B. Moreover, testing a network trained on CIFAR10 on a Gaussian mixture with the means fitted to CIFAR10 (blue) and a Gaussian mixture with the means and covariance fitted on CIFAR10 (orange) shows that the network goes smoothly and quickly from Gaussian to the non-Gaussian part of the data, without a plateau.

A natural idea improve the MCM, i.e. to make the loss curves of a neural network trained on the MCM resemble more the dynamics observed on CIFAR10, is to correlate the covariance and cumulant spikes: instead of choosing them to be orthogonal to each other, one could give them a finite overlap $u \cdot v = \rho$. However, a detailed analysis of the SGD dynamics in section 3.3.2 will show that this does *not* speed up learning the non-Gaussian part of the data; instead, the crucial ingredient to speed up learning of non-Gaussian correlations is the correlation between latent variables. Setting for example

$$\nu^\mu = \text{sign}(\lambda^\mu), \tag{3.3}$$

we obtain the generalisation dynamics shown in red in fig. 3.2C: the plateaus have disappeared, and we get a behaviour that is very close to real data: a single exponential decay of the test loss.

3.2.5 An analysis of a single neuron quantifies the speed-up of correlated latents

We can make our experimental observations for two-layer networks rigorous in the simplest model of a neural network a single neuron $f(w, x) = \sigma(w \cdot x)$ trained using online projected stochastic gradient descent (also known as the spherical perceptron). At each step t of the algorithm, we sample a tuple (x_t, y_t) from the MCM (3.1) and update the weight w_t according to the following rule:

$$w_t = \begin{cases} w_0 \sim \text{Unif}\left(\mathbb{S}^{d-1}\right) & t = 0 \\ \tilde{w}_t = w_{t-1} - \frac{\delta}{d}\nabla_{\text{sph}}\mathcal{L}\left(w_t, (x_t, y_t)\right) & t \geq 1 \\ w_t = \frac{\tilde{w}_t}{||\tilde{w}_t||}. \end{cases} \tag{3.4}$$

Here, ∇_{sph} is the spherical gradient defined by $\nabla_{\text{sph}} f(w) = (\mathbb{1} - ww^\top)\nabla f(w)$. We follow Damian *et al.* [18] in using the *correlation loss* for our analysis,

$$\mathcal{L}(w, (x, y)) = 1 - yf(w, x). \tag{3.5}$$

In high dimensions, the typical overlap of the weight vector at initialisation and, say, the cumulant spike scales as $w_0 \cdot v \simeq d^{-1/2}$. We will say we the perceptron has learnt the direction v if we have "weakly recovered" the spike, i.e. when the overlap $\alpha_v \equiv w \cdot v \sim O(1)$. This transition from diminishing to macroscopic overlap marks the exit of the search phase of stochastic gradient descent, and it often requires most of the runtime of online SGD [16].

In this setup, we can give a precise characterisation of the sample complexity of learning a single spike (either in the covariance or in the cumulant) and of learning in the presence of two spikes with

independent or correlated latent variables. By looking at stochastic gradient descent, we can also make quantitative statements on the optimal learning rates to achieve weak recovery quickly. For simplicity, our perceptron analysis does not consider the spike in the mean, i.e. $\beta_m = 0$ throughout. This assumption is mainly to enhance the mathematical tractability of the model and we expect most of the following to hold also in the case $\beta_m \neq 0$. Our main theoretical results are then as follows:

Learning a single direction: If only the covariance *or* the cumulant are spiked, i.e. either $\beta_u = 0$ or $\beta_v = 0$, the analysis of Ben Arous *et al.* [16] applies directly and we find that projected online SGD requires $n \gtrsim d \log^2 d$ samples to learn the covariance spike u, but $n \gtrsim d^3$ samples to learn the cumulant spike; see proposition 3.2 and proposition 3.3 for the precise scaling of the learning rate δ. This result establishes that in isolation, learning from the higher-order cumulant has a much higher sample complexity than learning from the covariance.

Independent latent variables: In the mixed cumulant model with spikes in the covariance and the HOCs, proposition 3.4 shows that if $n \leq d^3$, it is impossible to have weak recovery of the cumulant spike for any learning rate $\delta_d = o\left(1/d\right)$ *if* the latent variables λ and ν are independent – hence learning from HOCs remains hard. For larger learning rates $1/d \leq \delta_d = o(1)$, the SGD noise becomes dominant after d/δ^2 steps and our analysis works only up to that horizon; we discuss this in section 3.3.2.

Correlated latent variables: If instead both spikes are present and their latent variables have a positive correlation $\mathbb{E}[\lambda^\mu \nu^\mu] > 0$ (fixed, independent of d), proposition 3.5 shows that $n \gtrsim d \log^2 d$ samples are *sufficient to weakly recover both spikes* with the optimal learning rate $\delta \approx 1/\log d$. Moreover, even for sub-optimal learning rates $\delta = o\left(1/d\right)$, the time to reach weak recovery of the cumulant spike is $d^2 \log^2 d$, which is still faster that in the uncorrelated case. Proposition 3.5 also shows that the speed-up in terms of sample complexity required for weak recovery happens even when the amount of correlation is small compared to the signal carried by the cumulant spike. In other words, it does not affect the minimum landscape of the loss: if the global minimum is close to the cumulant spike, introducing correlations between the latents will not move that minimum to the covariance spike.

We give a full summary of the known results and our contributions in table 3.1. In the following, we give precise statements of our theorems in section 3.3 and discuss our results in the context of the wider literature, and the recently discussed "staircase phenomenon", in section 3.4.

3.3 Rigorous analysis of the perceptron: the non-Gaussian information exponent

We now present a detailed analysis of the simplest model where correlated latents speed up learning from higher-order cumulants in the data, the spherical perceptron. Before stating the theorems, we present the main ingredients of the analysis informally.

The key idea of our analysis is borrowed from Ben Arous *et al.* [16]: during the *search phase* of SGD, while the overlaps $\alpha_u = u \cdot w, \alpha_v = v \cdot w$ of the weight w with the covariance and cumulant spikes are small, i.e. $o(1)$ with respect to d, the dynamics of spherical SGD is driven by the low-order terms of the polynomial expansion of the *population loss* $\mathcal{L}(w) = \mathbb{E}[\mathcal{L}(w, (x, y)]$. In our case of a mixture classification task, it is useful to rewrite the expectation using the *likelihood ratio* between

the isotropic and the planted distributions $L(x) := {}^{\mathrm{d}\mathbb{Q}_{\mathrm{plant}}}/_{\mathrm{d}\mathbb{Q}_0}(x)$ such that all averages are taken with respect to the simple, isotropic Gaussian distribution:

$$\mathcal{L}(w) = 1 + \frac{1}{2}\mathbb{E}_{\mathbb{Q}_0}[\sigma(w \cdot x)] - \frac{1}{2}\mathbb{E}_{\mathbb{Q}_0}\left[L(x)\sigma(w \cdot x)\right]. \tag{3.6}$$

We can then expand the loss in Hermite polynomials and prove that the loss depends only on α_u, α_v and can be expanded in the following way:

$$\mathcal{L}(w) = \ell(\alpha_u, \alpha_v) = \sum_{i,j=0}^{\infty} c_{ij}^L c_{i+j}^\sigma \alpha_u^i \alpha_v^j, \tag{3.7}$$

see lemma 3.8 in the appendix. The degree of the lowest order term in this expansion is called the *information exponent* k of the loss and it rigorously determines the duration of the search phase before SGD *weakly recovers* the key directions [16, 20], in the sense that for any time-dependent overlap $\alpha(t)$, there exists an $\eta > 0$ such that defining $\tau_\eta := \min\left\{t \geq 0 \middle| \ |\alpha(t)| \geq \eta\right\}$, we have

$$\lim_{d\to\infty} \mathbb{P}\left(\tau_\eta \leq n\right) = 1. \tag{3.8}$$

Since the time in online SGD is equivalent to the number of samples, the information exponent governs the sample complexity for weak recovery. Here, we are interested in finding out how the correlation of the latent variables changes the information exponent, and which consequences this change has on the duration of search phase of projected SGD as $d \to \infty$. Considering only the terms that give relevant contribution for the search phase, eq. (3.7) applied to the MCM model with $\beta_m = 0$ becomes (see section 3.A.2.1 for more details):

$$\ell(\alpha_u, \alpha_v) = -\left(c_{20}\alpha_u^2 + c_{11}\alpha_u\alpha_v + c_{04}\alpha_v^4\right) \tag{3.9}$$

where $c_{20}, c_{04} > 0$ as long as $\beta_u, \beta_v > 0$, whereas $c_{11} > 0$ if and only if the latent variables are positively correlated: $\mathbb{E}[\lambda\nu] > 0$. Note that switching on this correlation does not change the overall information exponent, which is still 2, but the *mixed term* $\alpha_u\alpha_v$ strongly impacts the direction v, along which $\partial_v \ell = c_{11}\alpha_u + 4c_{04}\alpha_v^3$ has degree 3 if $c_{11} = 0$ whereas it has degree 1 in case of positive correlation. This means that *positive correlation of latent variables lowers the information exponent along the non-Gaussian direction.*

It is not straightforward to link these changes of the population-loss series expansion to actual changes in SGD dynamics. The randomness of each sample comes into play and it must be estimated thanks to a careful selection of the learning rate δ (see proposition 3.4 and proposition 3.5). However, at the risk of oversimplification, we could imagine that the dynamic is updated descending the spherical gradient of the population loss, leading to the following system of ODEs that approximately hold when $\alpha_u, \alpha_v \leq \eta$

$$\begin{cases} \dot{\alpha}_u(t) = 2c_{20}\alpha_u + c_{11}\alpha_v + O(\eta^2) \\ \dot{\alpha}_v(t) = c_{11}\alpha_u + 4c_{04}\alpha_v^3 - 2c_{20}\alpha_u^2\alpha_v + O(\eta^4) \end{cases} \tag{3.10}$$

where the last term in the second equation is due to the distorting effect of the spherical gradient.

We can see that the behaviour of α_u is not affected too much by the presence of correlation (the integral lines starting at $\alpha_u(0) \approx d^{-1/2}$ reach order 1 in $t = d\log^2 d$). On the contrary the dynamic of α_v changes completely: if $c_{11} > 0$, α_v will grow at the same pace as α_u, otherwise it will be orders of magnitude slower (there is even the possibility that it will never escape 0, since the term $-2c_{20}\alpha_u^2\alpha_v$ could push it back too fast). Before stating the theorems that make these intuitions precise, we have to state our main assumptions on the activation function:

Model	Step size	Weak recovery u	Weak recovery v
Covariance spike only	$\delta = O(1)$, $\delta \approx \frac{1}{\log d}$	$n \ll d \log d$, $n \gg d \log^2 d$	not present
Cumulants spike only	$\delta = o\left(\frac{1}{d}\right)$, $\delta \approx \frac{1}{d}$	not present	$n \ll d^3$ $n \gg d^3 \log^2 d$
Two spikes, independent latents	$\delta \approx \frac{1}{\log d}$	$n \gg d \log^2 d$	$n \ll d \log^2 d$
	$\delta = o\left(\frac{1}{d}\right)$, $\delta \approx \frac{1}{d}$	$n \gg d^2 \log d$	$n \ll d^3$
Two spikes, correlated latents	$\delta \approx \frac{1}{\log d}$	$n \gg d \log^2 d$	$n \gg d \log^2 d$
	$\delta \approx \frac{1}{d}$	$n \gg d^2 \log d$	$n \gg d^2 \log d$

Table 3.1: **Summary of positive and negative results on learning from higher-order cumulants.** We summarise the sample complexities required to reach weak recovery of the covariance spike u and the cumulant spike v in the different settings discussed in section 3.3. Negative results mean that weak recovery is impossible below these sample complexities at the given learning rates. Positive results guarantee weak recovery with at least as many samples as stated. The first two lines are direct applications of results from Ben Arous *et al.* [16] for data models with a single spike. The mixed cumulant model with independent latent variables is a direct corollary of these results. The notation $\delta \approx f(d)$ for the positive results means that $\delta = o(f(d))$, but the best sample complexity is achieved when δ is as close as possible to its bound. Note that for independent latents with large learning rate $\delta \approx 1/\log d$, our techniques only allow studying a time horizons up to $t \approx d \log^2 d$, after which the randomness of the SGD updates becomes too large to be controlled, see section 3.3.2 for more details.

Assumption 3.1 (Assumption on the student activation function σ). We require that the activation function $\sigma \in \mathscr{C}^1(\mathbb{R})$ fulfils the following conditions:

$$\begin{aligned} &\mathop{\mathbb{E}}_{z\sim\mathcal{N}(0,1)}[\sigma(z)h_2(z)] > 0, \mathop{\mathbb{E}}_{z\sim\mathcal{N}(0,1)}[\sigma(z)h_4(z)] < 0, \\ &\sup_{w\in\mathbb{S}^{d-1}} \mathbb{E}\left[\sigma'(w\cdot x)^4\right] \le C, \sup_{w\in\mathbb{S}^{d-1}} \mathbb{E}\left[\sigma'(w\cdot x)^{8+\iota}\right] \le C, \\ &\text{and} \sum_{k=0} k \mathop{\mathbb{E}}_{z\sim\mathcal{N}(0,1)}[h_k(z)\sigma(z)] < \infty. \end{aligned} \tag{3.11}$$

Here, $(h_i)_i$ are normalised Hermite polynomials (see appendix A) and $\iota > 0$.

Note that smoothed versions of the ReLU activation function satisfy these conditions. We remark that all eq. (3.11) are important requirements, activation functions that do not satisfy one or more of these conditions could lead to different dynamics. On the other hand the assumption $\sigma \in \mathscr{C}^1(\mathbb{R})$ is done for convenience and differentiability a.e. should be enough; for instance all the simulations of the present work were done with the ReLU activation function and there is complete agreement with the theoretical predictions.

3.3.1 Learning a single direction

To establish the baselines for weak recovery in single index models, where inputs carry only a covariance spike or a spike in the higher-order cumulants, we apply the results of Ben Arous *et al.* [16] directly to find the following SGD learning timescales:

Proposition 3.2 (Covariance spike only)**.** *Considering only the spike in the covariance the generative distribution is* $y \sim$ *Rademacher*$(1/2)$*,*

$$\begin{aligned} y = 1 \;&\Rightarrow\; x^\mu = \sqrt{\beta}_u \lambda^\mu u + z^\mu, && \lambda^\mu \sim \mathcal{N}(0,1) \\ y = -1 \;&\Rightarrow\; x^\mu = z^\mu, && z^\mu \sim \mathcal{N}(0, \mathbb{1}_d) \end{aligned}$$

A spherical perceptron that satisfies assumption 3.1 and is trained on the correlation loss (3.5) *using online SGD* (3.4) *has the following result concerning the overlap with the hidden direction* $\alpha_{u,t} := u \cdot w_t$*:*

- *if* $\frac{1}{\log^2 d} \ll \delta_d \ll \frac{1}{\log d}$ *and* $n \gg d \log^2 d$ *then there is* ***strong recovery*** *and*

$$|\alpha_{u,t}| \to 1 \qquad \text{in probability and in } \mathscr{L}_p \; \forall p \geq 1$$

- *if* $\delta_d = O(1)$ *and* $n \leq d \log d$ *then it is impossible to learn anything:*

$$|\alpha_{u,t}| \to 0 \qquad \text{in probability and in } \mathscr{L}_p \; \forall p \geq 1$$

Proposition 3.3 (Cumulant spike only)**.** *Considering only the spike in the cumulants, letting* $(y^\mu)_\mu, (\nu^\mu)_\mu$ *be i.i.d. Rademacher*$(1/2)$*,* $(z^\mu)_\mu$ *be i.i.d* $N(0, \mathbb{1}_d)$ *and* S *as in* (3.2)*, the generative distribution is:*

$$\begin{aligned} y^\mu = 1 \;&\Rightarrow\; x^\mu = S\left(\sqrt{\beta}_v \nu^\mu v + z^\mu\right) \\ y^\mu = -1 \;&\Rightarrow\; x^\mu = z^\mu, \end{aligned}$$

A spherical perceptron that satisfies assumption 3.1 and is trained on the correlation loss (3.5) *using online SGD* (3.4) *has the following result concerning the overlap with the hidden direction* $\alpha_{v,t} := u \cdot v_t$*:*

- *if* $\frac{1}{d^2 \log^2 d} \ll \delta_d \ll \frac{1}{d \log d}$ *and* $n \gg d^3 \log^2 d$ *then there is* ***weak recovery*** *of the cumulant spike in the sense of eq.* (3.8)*;*

- *if* $\delta_d = o(\frac{1}{d})$ *and* $n \ll d^3$ *then it is impossible to learn anything:*

$$|\alpha_{v,t}| \to 0 \qquad \text{in probability and in } \mathscr{L}_p \; \forall p \geq 1$$

Note that the thresholds found in proposition 3.3 coincide with the ones found for tensor-PCA for a order 4 tensor (see proposition 2.8 in [16]), which is exactly the order of the first non trivial cumulant of $\mathbb{Q}_{\text{plant}}$ in the spiked-cumulant model. We will also see what happens in case of larger step size than what considered in proposition 3.3.

3.3.2 Learning two directions

The following two propositions apply to a perceptron trained on the MCM model of eq. (3.1) with $\beta_m = 0$, $\nu^\mu \sim \text{Radem}(1/2)$. We first state a negative result: in the case of *independent* latent variables, the cumulant direction cannot be learned faster than in the one dimensional case (proposition 3.3).

Proposition 3.4. *Consider a spherical perceptron that satisfies assumption 3.1 and is trained on the MCM* (3.1) *with correlation loss* (3.5) *using online SGD* (3.4)*, with λ^μ independent of ν^μ. Let $\delta_d = o(1)$ be the step size. As long as $n \ll \min\left(\frac{d}{\delta_d^2}, d^3\right)$, we have that*

$$\lim_{d\to\infty} \sup_{t\le n} |\alpha_{v,t}| = 0$$

where the limit holds in $\mathscr{L}^p$ for every $p \ge 1$.

The condition $n \ll {}^{d}/_{\delta_d^2}$ is due to *sample complexity horizon* in online SGD. As already pointed out by Ben Arous *et al.* [16], it turns out that after a certain number of steps, when ${}^{n\delta_d^2}/_{d}$ becomes large, the noise term in the equations dominates the drift term and the dynamics starts to become completely random (mathematically, Doob's inequality (3.30) fails to provide a useful estimate on the noise after this point). So our results can prove the absence of weak recovery only up to the *sample complexity horizon*; after that our definition of *weak recovery* is not useful anymore: due to the increased randomness, it is possible that the weight vector attains, by pure chance, a finite overlap with the spiked directions, but that would be forgotten very easily, not leading to meaningful learning. If instead the learning rate is small enough, $\delta_d = o(\frac{1}{d})$, then the horizon becomes large enough to include $n \ll d^3$, which is the same regime of proposition 3.3.

Now we state the positive results: the covariance spike can be weakly recovered in the same sample complexity as the one-dimensional case (proposition 3.2). However, if the latent variables have positive correlation, the same sample complexity is also sufficient to weakly recover the cumulant spike.

Proposition 3.5. *In the setting described, let the total number of samples be $n = \theta_d d$, with $\theta_d \gtrsim \log^2 d$ and growing at most polynomially in d. The step size $(\delta_d)_d$ is chosen to satisfy:*

$$\frac{1}{\theta_d} \ll \delta_d \ll \frac{1}{\sqrt{\theta_d}} \tag{3.12}$$

Projected SGD reaches ***weak recovery*** *of the covariance spike in the sense of eq.* (3.8) *in a time $\tau_u \le n$. Moreover, if the latent variables have positive correlation: $\mathbb{E}[\lambda\nu] > 0$, conditioning on having matching signs at initialisation: $\alpha_{u,0}\alpha_{v,0} > 0$,* ***weak recovery is reached also for the cumulant spike*** *v* ***in a time*** *$\tau_v \le n$.*

The initialisation assumption $\alpha_{u,0}\alpha_{v,0} > 0$ means that the second part of proposition 3.5 can be applied on half of the runs on average. This requirement could be fundamental: in case of mismatched initialisation, the correlation of the latent variables would push both $\alpha_{u,t}$ and $\alpha_{v,t}$ towards 0, slowing down the process. Note that the dependence of SGD dynamics on initialisation is a recurring phenomenon that arises with complex tasks, it is for example a well-known effect when training a neural network on inputs on a XOR-like Gaussian mixture task [94, 128]. It is likely that this initialisation problem is specifically due to having a single neuron network. In a two-layer neural network, overparametrisation usually helps since it is sufficient that there is a neuron with the right initialisation signs to drive the learning in the right direction [94]. For instance, the simulations we performed for fig. 3.2 with wide two-layer networks did not exhibit initialisation problems.

Remark 3.6. Optimising in θ in proposition 3.5, we get that if $\frac{1}{\log^2 d} \ll \delta_d \ll \frac{1}{\log d}$, then $n \gg d \log^2 d$ guarantees to have weak recovery. This optimal scaling of δ_d does not coincide with the best learning rate scaling for proposition 3.4, since the sample complexity horizon is very short. However, by taking the slightly sub-optimal learning rate $\delta_d = \frac{1}{d \log d}$, we get that on one hand proposition 3.4

applies and we can infer that, with independent latent variables, it takes at least d^3 samples/steps to learn direction v. On the other hand, by proposition 3.5 we know that when the latent variables have (even a small) correlation, in $n \approx d^2 \log^2 d$ projected SGD attains weak recovery of both spikes.

This completes our analysis of the search phase of SGD, which established rigorously how correlated latent variables speed up the weak recovery of the cumulant spike. We finally note that this analysis can only be applied to the *search phase* of SGD. The expansion in eq. (3.7) breaks down as soon as α_u, α_v become macroscopic, since there are infinitely many terms that are not negligible along the non-Gaussian direction v in that case. Hence, even though the correlation between latent variables is key in the early stages of learning, it can become a sub-leading contribution in the later stages of SGD where the direction of weight updates is greatly influenced by terms due to HOCs. A thorough analysis of the descent phase after weak recovery will require different techniques: the SGD dynamical system involving the complete expression of the population loss should be studied, we leave this for future work.

However, the following simple example conveys an idea of the possible dynamics in this regime. Consider a sequence (β_u^m, β_v^m) such that $(\beta_u^0, \beta_v^0) = (1, 0)$ and $\lim_{m\to\infty}(\beta_u^m, \beta_v^m) = (0, 1)$. The global minimum of the population loss will move continuously from $w = u$ to $w = v$. However, looking at the dynamics of eq. (3.10), the change will be only at the level of the coefficients, and as long as none of them vanishes, the search phase dynamics will be qualitatively the same. So changing the signal-to-noise ratios in this regime will affect only the the descent phase of the problem.

3.4 Discussion: hierarchies of learning with neural networks

3.4.1 Learning functions of increasing complexity and the staircase phenomenon

There are several well-known hierarchies that characterise learning in neural networks. In a seminal contribution, Saad & Solla [6, 129] characterised a "specialisation" transition where two-layer neural networks with a few hidden neurons go from performing like an effective (generalised) linear model to a non-linear model during training with online SGD. Notably, this transition occurs after a long plateau in the generalisation error. A similar learning of functions of increasing complexity was shown experimentally in convolutional neural networks by Kalimeris *et al.* [130].

More recently, Abbé *et al.* [25, 26] studied the problem of learning a target function over binary inputs that depend only on a small number of coordinates. They showed that two-layer neural networks can learn k-sparse functions (i.e. those functions that depend only on k coordinates) with sample complexity $n \gtrsim d$, rather than the $n \gtrsim d^k$ sample complexity of linear methods operating on a fixed feature space, if the target functions fulfil a (merged) *staircase* property. Abbé *et al.* [131] extended this analysis to study the saddle-to-saddle dynamics of two-layer neural networks for specific target link functions. Similar saddle-to-saddle dynamics have been described by Jacot *et al.* [132] in linear networks and by Boursier *et al.* [133] for two-layer ReLU networks.

A detailed description of the staircase phenomenon when learning a multi-index target function over Gaussian inputs was given by Dandi *et al.* [20], who studied feature learning via a few steps of single-pass, large-batch, gradient descent[1]. Their analysis showed how subsequent steps of gradient descent allow for learning new perpendicular directions of the target function if those directions are linearly connected to the previously learned directions. Bietti *et al.* [27] also consider multi-index target function over Gaussian inputs, but instead look at the gradient flow for two-layer neural networks and provide a complete picture of the timescales of the ensuing saddle-to-saddle dynamics.

[1] Note that a single step of gradient descent was studied in a similar setting by Damian *et al.* [19] and Ba *et al.* [134] without analysing the staircase phenomenon.

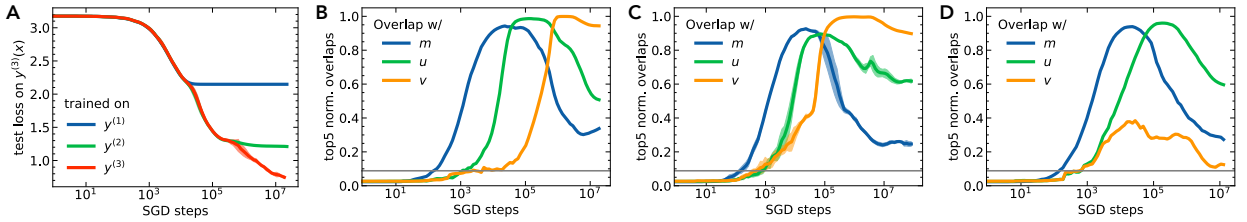

Figure 3.3: **Staircases in the teacher-student setup. A** Test accuracy of the same two-layer neural networks as in fig. 3.2 evaluated on the degree-4 target function $y^*(x)$ (3.13) during training on the target functions $y^{(1)}(x) = h_1(m \cdot x)$ (blue), $y^{(2)}(x) = h_1(m \cdot x) + h_2(u \cdot x)$ (green), and the teacher function eq. (3.13) (red). Inputs are drawn from the standard multivariate Gaussian distribution. **B-D** We show the the average of the top-5 largest normalised overlaps $w_k \cdot u$ of the weights of the kth hidden neuron w_k and the three directions that need to be learnt for three different target functions: the teacher function in eq. (3.13) (**B**), the same teacher with inputs that have a covariance $\mathbb{1} + uv^\top + vu^\top$ (**C**), and a teacher with mixed terms, eq. (3.14) **(D)**. The dashed black line is at $d^{-1/2}$, the threshold for weak recovery. *Parameters:* Simulation parameters as in fig. 3.2: $d = 128, m = 512$ hidden neurons, ReLU activation function. Full details in section 3.B.2.

Berthier *et al.* [135] analyse the gradient flow of two-layer neural networks when both layers are trained simultaneously. They perform a perturbative expansion in the (small) learning rate of the second layer and find the timescales over which the neural network learns the Hermite coefficients of the target function sequentially.

3.4.1.1 Relation between the mixed cumulant and teacher-student models

Given these results, it is natural to ask how the staircases in teacher-student setups relate to the hierarchical learning in the mixed-cumulant model. A teacher model with three "spikes" (or teacher weight vectors) where each spike is learnt with a different sample complexity, analogously to the uncorrelated MCM, is the function

$$y^*(x) = h_1(m \cdot x) + h_2(u \cdot x) + h_4(v \cdot x), \tag{3.13}$$

where h_k is the kth Hermite polynomial (see appendix A for details) and the inputs x are drawn i.i.d. from the standard Gaussian distribution with identity covariance. In the language of Ben Arous *et al.* [16], a single-index model with activation h_k has information exponent k, so taken individually, the three target functions in eq. (3.13) with directions m, u and v would be learnt with $n \gtrsim d, d\log^2 d$, and d^3 samples by a spherical perceptron using online SGD, exactly as in our mixed cumulant model with independent latent variables. We show the sequential learning of the different directions of such a teacher for a two-layer neural network in fig. 3.3A. In particular, we plot the test error with respect to the teacher $y^*(x)$ (3.13) for a two-layer network trained directly on the teacher (red), and for two-layer networks trained on only the first / the first two Hermite polynomials of fig. 3.3 (blue and green, respectively). Their performance suggests that the three directions m, v, u are learnt sequentially, which is further supported by showing the maximum normalised overlap of the hidden neurons with the different spikes shown in fig. 3.3B.

How can we speed up learning of the direction u with information exponent 4 in a way that is analogous to the correlated latents? The latent variables of the MCM correspond to the pre-activations of the target function in the sense that both are low-dimensional projections of the inputs that determine the label. In the standard teacher-student setup, where inputs are drawn

i.i.d. from a multivariate normal distribution with identity covariance, and target functions are of two-layer networks of the form $y^*(x) = \sum_k v_k \sigma_k(u_k \cdot x)$, it is not possible to have correlated pre-activations. Instead, we can speed up learning of the direction v by introducing a mixed term in the target function,

$$y^*(x) = h_1(m \cdot x) + h_1(u \cdot x)h_1(v \cdot x) + h_2(u \cdot x) + h_4(v \cdot x), \tag{3.14}$$

as is apparent from the overlap plot in fig. 3.3C. Dandi *et al.* [20] and Bietti & Bach [136] discuss the speed-up of learning this type of target function in the one-step and gradient flow frameworks, respectively, whereas our analysis focuses on online stochastic gradient descent with non-Gaussian inputs. An alternative for correlating the pre-activations is to keep the target function eq. (3.13) while drawing inputs from a normal distribution with covariance $\mathbb{1} + uv^\top$, and we see in fig. 3.3D that it does indeed speed up learning of the cumulant direction. This result is similar to the setting of Mousavi-Hosseini *et al.* [137], who found that "spiking" of the input covariance can speed up learning a single-index model with a two-layer network.

So in summary, we see that the accelerated learning requires fine-tuning between the target function and the input structure. The mixed cumulant model does not require an explicit target function and instead directly highlights the importance of the correlation in the latent variables of the inputs to distinguish different classes of inputs.

Generative exponent and repeating batches Another important difference of eq. (3.14) with the *mixed cumulant model* comes into play when considering the *generative exponent* introduced in [21]: if we allow for random transformations of the labels, or batch repetition ([22, 23]), then even eq. (3.13) can be fully learned in linear sample complexity (since the generative exponent of any polynomial is at most two). On the contrary the *spiked cumulant model* cannot have generative exponent lower than four: with binary labels it is impossible to find a transformation T such that $E[h_{1/2/3}(x)|T(y)] \neq 0$ (which is the requirement to have generative exponent less than four).

3.4.2 The role of Gaussian fluctuations with correlated latents

Correlation between latent variables changes the covariance of the inputs compared to independent latents. A quick calculation shows that the covariance of the inputs under the planted distribution $C_{uv} \equiv \underset{\mathbb{Q}_{\text{plant}}}{\text{cov}}\,(x, x)$ becomes

$$C_{uv} = \mathbb{1} + \beta_u uu^\top + \sqrt{\frac{\beta_u \beta_v}{1+\beta_v}}\, \mathbb{E}\, \lambda\nu \left(uv^\top + vu^\top\right). \tag{3.15}$$

In other words, we can weakly recover the cumulant spike by computing the leading eigenvector of the covariance of the inputs. Does that mean that the speed-up in learning due to correlated latents is simply due to the amount of information about the cumulant spike v that is exposed in the covariance matrix?

We test this hypothesis by taking the two-layer neural networks trained on the full MCM model and evaluating them on an equivalent Gaussian model, where we replace inputs from the planted distribution $\mathbb{Q}_{\text{plant}}$ of the MCM with samples from a multivariate normal distribution with mean $\beta_m u_m$ and covariance C_{uv}. We plot the corresponding test losses for independent and correlated latents in orange in fig. 3.2. When latent variables are independent, the Gaussian approximation breaks down after after a long plateau at $\approx 10^7$ steps, precisely when the network discovers the non-Gaussian fluctuations due to the cumulant spike. For correlated latent variables

on the other hand, the Gaussian approximation of the data only holds for $\approx 10^4$ steps. This suggests that the presence of the cumulant spike in the covariance gives some of the neurons a finite overlap $w_k \cdot v$ with the cumulant spike, and this initial overlap speeds up the recovery of the cumulant spike using information from the higher-order cumulants of the data which is inaccessible at those time scales when latent variables are uncorrelated. The presence of correlated latent variables therefore genuinely accelerates the learning from higher-order, non-Gaussian correlations in the inputs.

3.4.3 Further related work

Learning polynomials of increasing degree with kernel methods A series of works analysing the performance of kernel methods [84, 88, 89, 105, 112, 138, 139] revealed that kernels (or linearised neural networks in the "lazy" regime [35, 36, 80, 82, 118]) require $n \gtrsim d^\ell$ samples to learn the ℓth Hermite polynomial approximation of the target function. Here, we focus instead on the feature learning regime where neural networks can access features from higher-order cumulants much faster.

Tensor PCA with side information From the perspective of unsupervised learning, the MCM model is closely related to models studied in random matrix theory. When $\beta_m = \beta_\nu = 0$, the data distribution boils down to the well known Spiked Wishart model, that exhibits the *BBP phase transition* [140], which predicts thresholds on β_c for detection, at linear sample complexity $n \approx d$. If instead only $\beta_\nu \neq 0$, the MCM model corresponds to a tensor PCA problem, where one seeks to recover a signal $\xi \in \mathbb{R}^d$ from a noisy order-p tensor $T = \xi^{\otimes p} + \Delta$ with suitable noise tensor Δ (that in most of the cases is assumed to be Gaussian distributed). Indeed, in the MCM model the HOCs spike v could be retrieved by performing tensor PCA on the empirical kurtosis tensor.

The notion of correlated latent variables that we consider in the present work is reminiscent to the concept of *side information* that Richard & Montanari [57] considered for tensor PCA. The side information would be an additional source of information on the spike via a Gaussian channel, $y = \gamma\xi + g$, where g is a Gaussian noise vector and $\gamma > 0$; using this estimate as an initialisation for AMP leads to a huge improvement in estimation. This joint model can be seen as a rank-1 version of a topic modelling method analysed by Anandkumar *et al.* [141]. In a similar vein, Sarao Mannelli *et al.* [142] introduced the spiked matrix-tensor model, in which the statistician tries to recover ξ via the observation of a spiked matrix $M \propto \xi\xi^\top + \Delta^{(1)}$ and an order-p tensor $T \propto \xi^{\otimes p} + \Delta^{(2)}$ with appropriately scaled noise matrix / tensor $\Delta^{(1)}$ and $\Delta^{(2)}$, respectively. They analyse the optimisation landscape of the problem and the performance of the Langevin algorithm [142] and of gradient descent [143, 144]. The main difference to our model is that these works consider recovering a *single* direction that spikes both the matrix and the tensor; we consider the case where two orthogonal directions are encoded as principal components of the covariance matrix and the higher-order cumulant tensors of the data.

3.5 Concluding perspectives

To achieve good performance, neural networks need to unwrap the higher-order correlations of their training data to extract features that are pertinent for a given task. We have shown that neural networks exploit correlations between the latent variables corresponding to these directions to speed up learning. In particular, our analysis of the spherical perceptron showed that correlations between the latent variables corresponding to the directions of two cumulants of order p and $q > p$, respectively, will speed up the learning of the direction from the qth cumulant by lowering its information exponent.

Our results open up several research directions. First, it will be intriguing to extend our analysis from single-index to shallow models. For two-layer neural networks, a key difficulty in applying mean-field techniques to analyse the impact of data structure on the dynamics of learning is the breakdown of the Gaussian equivalence principle [13, 108–111, 145] that we discussed in section 3.4.2. In the presence of non-Gaussian pre-activations $w_k \cdot x$, it is not immediately clear what the right order parameters are that completely capture the dynamics. An analysis of the dynamics in the one-step framework of Dandi *et al.* [20] and Ba *et al.* [134] is another promising approach to tackle the dynamics of two-layer networks. Other intriguing directions include determining the hardness of learning from higher-order cumulants when re-using data [22] and from the perspective of generative exponents [21]. For deeper neural networks, it is important to investigate the role of different layers in processing the HOCs of the data (even numerically), for example along the lines of Fischer *et al.* [146]. Finally, it will be intriguing to investigate how correlated latent variables emerge both in real data and in (hierarchical) models of synthetic data.

Appendix

3.A Dynamical analysis

3.A.1 Expansion lemmas

In this section we prove technical versions of the multi-variate Hermite expansions that are needed for the proofs of proposition 3.4 and proposition 3.5.

Lemma 3.7. *Suppose $g \in \mathscr{L}^2(\mathbb{R}, \mathcal{N}(0,1))$ and $w \in \mathbb{R}$, with $||w|| = 1$, then $f : \mathbb{R}^d \to \mathbb{R}$ defined as $f(x) := g(w \cdot x)$ belongs to $\mathscr{L}^2(\mathbb{R}^d, \mathcal{N}(0, \mathbb{1}_d))$ and has Hermite coefficients:*

$$C^f_{\boldsymbol{\alpha}} := \mathop{\mathbb{E}}_{z\sim\mathcal{N}(0,\mathbb{1}_d)} [H_{\boldsymbol{\alpha}}(z) f(z)] = \mathop{\mathbb{E}}_{z\sim\mathcal{N}(0,1)} \left[h_{|\boldsymbol{\alpha}|}(z) g(z)\right] \prod_{i=1}^{d} w_i^{\boldsymbol{\alpha}_i}, \quad \boldsymbol{\alpha} \in \mathbb{N}^d \tag{3.16}$$

Suppose now $g \in \mathscr{L}^2(\mathbb{R}^2, \mathcal{N}(0, \mathbb{1}_2))$ and $u, v \in \mathbb{R}$, with $||u|| = ||v|| = 1$ and $u \cdot v$, then $f : \mathbb{R}^d \to \mathbb{R}$ defined as $f(x) := g(u \cdot x, v \cdot x)$ belongs to $\mathscr{L}^2(\mathbb{R}^d, \mathcal{N}(0, \mathbb{1}_d))$ and has Hermite coefficients:

$$C^f_{\boldsymbol{\alpha}} = \sum_{i+j=|\boldsymbol{\alpha}|} \mathop{\mathbb{E}}_{z\sim\mathcal{N}(0,\mathbb{1}_2)} \left[H_{(i,j)}(z) g(z)\right] \prod_{m=1}^{d} \left(\sum_{\substack{i_m+j_m=\boldsymbol{\alpha}_m \\ \sum_m i_m = i, \sum_m j_m = j}} u_m^{i_m} v_m^{j_m} \right) \tag{3.17}$$

First we will present the expansion procedure informally, and then in lemma 3.8 we will see the exact statement that will be used in the subsequent proofs.

Assume the learning algorithm is a perceptron $f(w, x) = \sigma(w \cdot x)$, it can be expanded in Hermite basis with appropriate coefficients:

$$f(w, x) = \sum_{k=0}^{\infty} \frac{c_k^\sigma}{k!} h_k(w \cdot x)$$

where

$$c_k^\sigma = \mathop{\mathbb{E}}_{z\sim\mathcal{N}(0,1)} [\sigma(z) h_k(z)] \tag{3.18}$$

Suppose that the data distribution is the MCM as described in section 3.2 (for simplicity we will assume that $\beta_m = 0$; there are only the covariance and HOCs spikes). To be able to carry on the expansion it is useful to consider (as done in [69] and [147]) the *likelihood ratio* of $\mathbb{Q}_{\text{plant}}$ with respect to $\mathbb{Q}_0$:

$$L(x) := \frac{\mathrm{d}\mathbb{Q}_{\text{plant}}(x)}{\mathrm{d}\mathbb{Q}_0(x)} \tag{3.19}$$

Note that since $\mathbb{Q}_{\text{plant}}$ is a perturbation of a standard Gaussian only along the directions u, v; then the likelihood ratio L will depend only on the projections $y_u := x \cdot u, y_v = x \cdot v$. Assuming u, v

to be orthogonal and the non Gaussianity ν to have a distribution that belongs to $\mathscr{L}^2(\mathbb{R}, \mathcal{N}(0,1))$, we can have an expansion in Hermite basis:

$$L(x) = \sum_{i,j=0}^{\infty} \frac{c_{ij}^L}{i!j!} h_i(y_u) h_j(y_v)$$

where

$$c_{ij}^L = \mathop{\mathbb{E}}_{(z_1,z_2)\sim\mathcal{N}(0,\mathbb{1}_{2\times 2})} [L(u\cdot z_1 + v\cdot z_2) h_i(z_1) h_j(z_2)] \tag{3.20}$$

So assuming to use a *correlation loss*:

$$\mathcal{L}(w,(x,y)) = 1 - yf(w,x)$$

The population loss is:

$$\begin{aligned}
\mathcal{L}(w) &= \mathbb{E}_{x,y}[\mathcal{L}(w,(x,y)] = 1 + \frac{1}{2}\mathbb{E}_{\mathbb{Q}_0}[\sigma(w\cdot x)] - \frac{1}{2}\mathbb{E}_{\mathbb{Q}_0}\left[L(x)\sigma(w\cdot x)\right] \\
&= \mathbb{E}_{x,y}[\mathcal{L}(w,(x,y)] = 1 + \frac{1}{2}\mathop{\mathbb{E}}_{z\sim\mathcal{N}(0,1)}[\sigma(z)] - \frac{1}{2}\mathbb{E}_{\mathbb{Q}_0}\left[L(x)\sigma(w\cdot x)\right] \\
&= 1 + \frac{c_0^\sigma}{2} - \frac{1}{2}\mathbb{E}_{\mathbb{Q}_0}\left[\left(\sum_{i,j=0}^{\infty}\frac{c_{ij}^L}{i!j!}h_i(x\cdot u)h_j(x\cdot v)\right)\left(\sum_{k=0}^{\infty}\frac{c_k^\sigma}{k!}h_k(w\cdot x)\right)\right]
\end{aligned} \tag{3.21}$$

The following lemma deals with the exchanging the integral and the series.

Lemma 3.8. *Let $\mathbb{P}$ be such that the likelihood ratio is square integrable $L \in \mathscr{L}^2\left(\mathbb{R}^d, \mathbb{Q}_d\right)$ and depends on x only through its projection along 2 orthogonal directions u, v:*

$$L(x) = l(u\cdot x, v\cdot x) = \sum_{i,j=0}^{\infty}\frac{c_{ij}^L}{i!j!}h_i(x\cdot u)h_j(x\cdot v) \tag{3.22}$$

Suppose $\sigma \in \mathscr{L}^2(\mathbb{R},\mathbb{Q}) \cap \mathscr{C}^1(\mathbb{R})$, with derivative $\sigma' \in \mathscr{L}^2(\mathbb{R},\mathbb{Q}) \cap \mathscr{C}^0(\mathbb{R})$, and expansions

$$\sigma(y) = \sum_{k=0}^{\infty}\frac{c_k^\sigma}{k!}h_k(y) \tag{3.23}$$

then, defining $\alpha_u := u\cdot w$ and $\alpha_v := v\cdot w$, the following identity holds

$$\mathbb{E}_{\mathbb{Q}_0}\left[L(x)\sigma(w\cdot x)\right] = \sum_{i,j}^{\infty}\frac{c_{ij}^L}{i!j!}\frac{c_{i+j}^\sigma}{(i+j)!}\alpha_u^i\alpha_v^j \tag{3.24}$$

Moreover, if $\sum_k k c_k^\sigma < \infty$, the population loss $\mathcal{L}(w) \in \mathscr{C}^1(\mathcal{C},\mathbb{R})$, where $\mathcal{C} := \left\{w \in \mathbb{S}^{d-1} | \alpha_u < \frac{1}{2}, \alpha_v < \frac{1}{2}\right\}$ and it is also possible to switch expectations and derivatives, to get the expression:

$$\begin{aligned}
&\nabla\mathcal{L}(w) = \nabla\mathbb{E}_{\mathbb{Q}}\left[L(x)\mathcal{L}(w,(x,y))\right] = \mathbb{E}_{\mathbb{Q}}\left[L(x)\nabla\mathcal{L}(w,(x,y))\right] = \\
&= -\frac{1}{2}\left[\left(\sum_{i=1}^{\infty} i\frac{c_{i0}^L c_i^\sigma}{(i!)^2}\alpha_u^{i-1}\right)u + \left(\sum_{j=1}^{\infty} j\frac{c_{0j}^L c_j^\sigma}{(j!)^2}\alpha_v^{j-1}\right)v + \left(\sum_{i,j=1}^{\infty}\frac{c_{ij}^L c_{i+j}^\sigma}{i!j!(i+j)!}\alpha_u^{i-1}\alpha_v^{j-1}\left(i\alpha_v u + j\alpha_u v\right)\right)\right]
\end{aligned} \tag{3.25}$$

Proof. First focus on (3.24). We know that both L and $\sigma(w\cdot_)$ belong to $\mathscr{L}^2\left(\mathbb{R}^d,\mathbb{Q}_0\right)$, hence we can see the integral in (3.24) as the $\mathscr{L}^2$ inner product, and compute it using the series expansion (we avoid writing the factorials at denominator) and lemma 3.7.

$$\begin{aligned}
\mathbb{E}_{\mathbb{Q}_0}\left[L(x)\sigma(w\cdot x)\right] &= \sum_{\boldsymbol{\alpha}\in\mathbb{N}^d} c^L_{\boldsymbol{\alpha}} c^{\sigma(w\cdot)}_{\boldsymbol{\alpha}} = \sum_{k=0}^{\infty}\sum_{|\boldsymbol{\alpha}|=k} c^L_{\boldsymbol{\alpha}} c^\sigma_k \prod_{l=1}^d w_l^{\boldsymbol{\alpha}_l} \\
&= \sum_k \sum_{i+j=k} c^L_{ij} c^\sigma_k \sum_{|\boldsymbol{\alpha}|=k}\prod_{m=1}^d w_m^{\boldsymbol{\alpha}_m}\left(\sum_{\substack{i_m+j_m=\boldsymbol{\alpha}_m \\ \sum_m i_m=i,\sum_m j_m=j}} u_m^{i_m} v_m^{j_m}\right) \\
&= \sum_k \sum_{i+j=k} c^L_{ij} c^\sigma_k \prod_{m=1}^d \left(\sum_{\substack{\sum_m i_m=i \\ \sum_m j_m=j}} (u_m w_m)^{i_m}(v_m w_m)^{j_m}\right) \\
&= \sum_k \sum_{i+j=k} c^L_{ij} c^\sigma_k \alpha_u^i \alpha_v^j
\end{aligned}$$

Moreover it can be verified directly that, under the assumption that $\sum_k k c_k^\sigma < \infty$, and $w\in\mathcal{C}$ the series in (3.25) converges uniformly, hence also the second part of the statement holds. □

3.A.2 Properties of the MCM model

In this section we will apply the Hermite expansion to the Mixed Cumulant Model and note the key properties that will allow to prove the Propositions of section 3.3.

3.A.2.1 Hermite coefficients

Starting from the single direction models we have that:

- If $\beta_m = \beta_v = 0$ only the covariance spike survives and the model is Gaussian, hence the population loss is:
$$\mathcal{L}(w) = \ell(\alpha_u) = 1 - \frac{\beta_u c_2^\sigma}{4}\alpha_u^2 \tag{3.26}$$
- if $\beta_m = \beta_u = 0$ only the cumulants spike survives. To compute the Hermite coefficients we can rely on lemma 13 form [147] to get:
$$\mathcal{L}(w) = \ell(\alpha_v) = 1 - \frac{1}{2}\sum_{j\geq 3}\frac{c_j^\sigma}{j!}\left(\frac{\beta_v}{1+\beta_v}\right)^{j/2}\mathbb{E}[h_j(\nu)]\alpha_v^j \tag{3.27}$$
So if ν =Radem(1/2) we have that the leading term is
$$\mathcal{L}(w) = \ell(\alpha_v) = 1 + \frac{c_4^\sigma}{4!}\left(\frac{\beta_v}{1+\beta_v}\right)^2\alpha_v^4 + o(\alpha_v^4)$$
Note the change of sign, due to $\mathbb{E}[h_4(\nu)] = -2$. This is why we need to ask $c_4^\sigma < 0$ to have that the population loss is decreasing for small α_v.

- consider now the case with both spike and *independent* latent variables λ, ν. It is easy to see that, thanks to the independence, the coefficients split $c_{ij}^L = c_i^{cov} c_j^{cumulant}$, so the leading terms are:

$$\mathcal{L}(w) = \ell(\alpha_u, \alpha_v) = 1 - \frac{1}{2} \underbrace{\frac{\beta_u c_2^\sigma}{4}}_{c_{20}} \alpha_u^2 - \underbrace{\frac{-c_4^\sigma}{4!} \left(\frac{\beta_v}{1+\beta_v} \right)^2}_{c_{04}} \alpha_v^4 + o(\alpha_v^4) \tag{3.28}$$

- in case of latent variables with positive correlation, we need to add mixed terms, the leading one is $c_{11} = c_2^\sigma \sqrt{\frac{\beta_u \beta_v}{1+\beta_v}} \mathbb{E}[\lambda \nu]$. So the first few terms of the expansion of the population loss are:

$$\mathcal{L}(w) = \ell(\alpha_u, \alpha_v) = 1 - \frac{1}{2} \left(c_{20} \alpha_u^2 + c_{11} \alpha_u \alpha_v + c_{04} \alpha_v^4 \right) + o(\alpha_u \alpha_v) \tag{3.29}$$

  note that the presence of $c_{11} > 0$ makes the term $c_{04} \alpha_v^4$ to become a sub-leading contribution, lowering the information exponent on the v direction.

3.A.2.2 Assumption on the sample wise error

An important quantity for proving the Propositions in section 3.3 is the *sample-wise error*

$$H_d^\mu(w) := \mathcal{L}_d\left(w, (x^\mu, y^\mu)\right) - \mathcal{L}_d(w)$$

we will ask the same assumptions of [16]:

$$\begin{aligned}
&\sup_{w \in \mathbb{S}^{d-1}} \mathbb{E}\left[\left(\nabla_{sph} H_d(w) \cdot u \right)^2 \right] \leq C_1 \\
&\sup_{w \in \mathbb{S}^{d-1}} \mathbb{E}\left[\left(\nabla_{sph} H_d(w) \cdot v \right)^2 \right] \leq C_1 \\
&\sup_{w \in \mathbb{S}^{d-1}} \mathbb{E}\left[||\nabla_{sph} H_d(w)||^{4+\iota} \right] \leq C_2 d^{4+\iota/2} \qquad \text{for some } \iota > 0
\end{aligned}$$

simple calculations ensure that it is sufficient to ask that

- $\mathbb{Q}_{\text{plant}}$ has finite moments up to the 8-th order,which is satisfied when taking $\nu \sim$Rademacher$(1/2)$.
- the following holds:

$$\begin{aligned}
\sup_{w \in \mathbb{S}^{d-1}} \mathbb{E}\left[\sigma'(w \cdot x)^4 \right] \leq C \\
\sup_{w \in \mathbb{S}^{d-1}} \mathbb{E}\left[\sigma'(w \cdot x)^{8+2\iota} \right] \leq C
\end{aligned}$$

  which we have assumed in (3.11)

3.A.3 Proof for the SGD analysis

Here we will provide proofs for the Propositions presented in section 3.3. All of them rely heavily on the thorough analysis of spherical perceptron dynamics proved in [16].

Note that proposition 3.2 is a well known result, very close to the examples provided in section 2 of [16], and we can verify that a straightforward application of theorems 1.3 and 1.4 from [16], in the case $k = 2$, proves it.

A different situation happens for proposition 3.3, the spiked cumulant model is quite recent so its SGD dynamics have still to be investigated. However it is very quick to verify that also in this case it is possible to apply theorems 1.3 and 1.4 from [16]. Assumption B in [16] is met thanks to section 3.A.2.2. The monotonicity assumption could be harder to check, but since we are interested only in the search phase, it is sufficient to satisfy that $\mathcal{L}(w) = \ell(\alpha_v)$ is monotone only in $(0, \rho)$ for any $\rho > 0$ (assumption A_ρ introduced in section 3.1 of [16]), which is clear from (3.27). Hence we can apply the case $k = 4$ of theorems 1.3 and 1.4 from [16], to conclude the proof.

3.A.3.1 Proof of proposition 3.4

This proposition can be considered a Corollary of *Theorem 1.4* from [16], the only differences is that the data distribution has an additional direction u and that we are considering also lower learning rates $\delta = o(1)$ instead of $\delta = o(\frac{1}{d})$ (note that at page 10 in [16] this extension to larger learning rates is already mentioned).

On one hand we will verify that thanks to the fact that the auxiliary variable λ, ν are independent, then the additional direction u makes no impact in the dynamic of the overlap α_v. On the other hand, the condition $n \leq \frac{d}{\delta_d^2}$ ensures that we are always inside the time horizon considered in [16]. We will discuss how to deal with these two changes in the next two paragrphs.

Direction u can be neglected Since $\nabla_{sph}\mathcal{L}(w_t, (x^t, y^t))$ is orthogonal to w_t we have that $|\tilde{\alpha}_{v,t}| = |v \cdot \tilde{w}_t| \geq |\alpha_{v,t}|$. Hence we have that if $\alpha_{v,t} > 0$, then

$$\begin{aligned}\alpha_{v,t} \leq \tilde{\alpha}_{v,t} &= \alpha_{v,t-1} - \frac{\delta}{d}\nabla_{sph}\mathcal{L}(w_{t-1}, (x^t, y^t)) \cdot v \\ &= \alpha_{v,t-1} - \frac{\delta}{d}\nabla_{sph}\mathcal{L}(w_{t-1}) \cdot v - \frac{\delta}{d}\nabla_{sph}H^t(w_{t-1}) \cdot v\end{aligned}$$

Since the sample-wise error satisfies the same properties required in [16] (as checked in section 3.A.2.2) we just need to ensure that the drift term also can be handled in the same way as in [16]. There, the monotonicity assumption on the population loss allowed to estimate $-v \cdot \nabla_{sph}\mathcal{L}(w) \leq C\alpha_v^3$ for $\alpha_v > 0$. So the aim will be to get a similar bound. Note that we can assume α_v smaller than η, hence we can apply (3.28) and lemma 3.8 to get:

$$\begin{aligned}-\nabla_{sph}\mathcal{L}(w) \cdot v &= \left((v - \alpha_v w)^\top \nabla L(w)\right. \\ &= \partial_v \ell - \alpha_v \left(\alpha_u \partial_u \ell + \alpha_v \partial_v \ell\right) \\ &= \ 4c_{04}\alpha_v^3 - c_{20}\alpha_u^2\alpha_v + o(\eta^3) \\ &\leq 5c_{04}\alpha_v^3\end{aligned}$$

We can now use this inequality to reach the same estimates as in p 37 of [16], and carry on with their proof, in the case $\delta = o(\frac{1}{d})$. We only need to show that we can take larger step size, at the cost of taking $n \ll \frac{d}{\delta_d^2}$.

Larger step size We note that in the proof of *Theorem 1.4* from [16] the main use of the assumptions $\delta = o(\frac{1}{d})$ was to ensure that $\frac{n}{d}\delta_d^2 = o(1)$ which is required to use the following form of Doob's maximal inequality for martingales:

$$\sup_{w_0 \in \mathbb{S}^{d-1}} \mathbb{P}_{w_0}\left(\max_{t \leq n} \frac{\delta}{d}\left|\sum_{j=0}^{t-1} \nabla_{sph}H^{j+1}(w_j) \cdot u\right| \geq r\right) \leq \frac{2n\delta^2 C}{d^2 r^2}, \tag{3.30}$$

where $\mathbb{P}_{w_0}$ means that we are conditioning that the initial weight is w_0, C comes from section 3.A.2.2 and $r > 0$ is a free parameter. To carry out the proof, the noise level needs to be of order $\frac{1}{\sqrt{d}}$, hence we want to be able to take $r \approx d^{-1/2}$ and have that the probability in eq. (3.30) goes to 0, which is guaranteed by the condition $\frac{n}{d}\delta_d^2 = o(1)$. Hence we can follow [16] p 38 and get that the trajectory will be below η for all $t \leq \tilde{t} := C_\eta \frac{d^2}{\delta}$, which is larger than our horizon, completing the proof.

3.A.3.2 Proof of proposition 3.5

We will focus on the case of correlated latent variables and prove weak recovery of both spikes. This will be a proof also of the weak recovery of the covariance spike in the case of independent latent variable (which is actually much simpler and is essentially a straight-forward verification of [16] theorem 1.3 to our slightly different setting).

First note that we can restrict to initialisation with positive overlap, $\alpha_{u,0}, \alpha_{v,0} > 0$: if for example the initialisation was $\alpha_{u,0} < 0$ then we can just substitute $\hat{u} := -u$, $\hat{\lambda} = -\lambda$ so that the sign changes cancel each other and $\hat{x} = x$ but $\hat{\alpha}_{u,0} = -\alpha_{u,0}$. In the case of uncorrelated latents, the distribution would not change: $\hat{x} \sim x$, due to the symmetry of the latent variables and independents of the two directions, hence the new problem is equivalent to the previous one.

In case of positive correlation of the latent variables, we need to use our assumption that there is no sign mismatch in the initialisation. Unlike the uncorrelated case, we cannot change the sign of one of the axes without changing also the other, since it would flip the sign of the correlation. So there are only two cases: either $\alpha_{u,0}, \alpha_{v,0} > 0$, which is the desired initialisation; or $\alpha_{u,0}, \alpha_{v,0} < 0$, where we can flip the signs along *both the spiked directions*:

$$\hat{u} = -u,\ \hat{\lambda} = -\lambda, \quad \hat{v} = -v,\ \hat{\mu} = -\mu$$

So that $\hat{x} = x$ and $(\hat{\lambda}, \hat{\nu}) \sim (\lambda, \nu)$ keeping a positive correlation and $\hat{\alpha}_{u/v,0} = -\alpha_{u/v,0}$.

So from now on we can assume $\alpha_{u,0}, \alpha_{v,0} > 0$. Hence, for d large enough, $w_0 \in K_\eta := \{w : (\alpha_u, \alpha_v) \in [0, \eta]^2\}$, for $\eta > 0$ small, so we are able to use the expansions from section 3.A.2.1.

Recall from the statement that we assume $n = \theta_d d$, with $\theta_d \gtrsim \log^2 d$, and choose the step size $\delta = \delta_d$ such that $1/\theta_d \ll \delta \ll 1/\sqrt{\theta_d}$. To prove weak recovery, we will show that there are $\eta_u, \eta_v \leq \eta$ such that, defining the τ_u, τ_v as the first times in which $\alpha_{u,t} \geq \eta_u$ and $\alpha_{v,t} \geq \eta_v$, then the probability that $\tau_u, \tau_v \leq n$ goes to 1 as $d \to \infty$.

Weak recovery of the covariance spike Given $\gamma > 0$, let $E_\gamma := \left\{w \mid \alpha_u \geq \frac{\gamma}{\sqrt{d}}, \alpha_v \geq \frac{\gamma}{\sqrt{d}}\right\}$. Since we consider the limit $d \to \infty$ and $w_0 \sim$Unif$(\mathbb{S}^{d-1})$, we can apply Poincaré lemma and have that for any $\varepsilon > 0$ we can find $\gamma > 0$ such that for all d sufficiently large

$$\mathbb{P}(w_0 \in E_\gamma) \geq 1 - \varepsilon \tag{3.31}$$

So, fix $\varepsilon > 0$ and from now on we will assume that $w \in E_\gamma$ for some $\gamma > 0$ sufficiently small (or d sufficiently large) so that (3.31) holds.

The first step is to prove that the *difference inequality* from *Proposition 4.1* from [16] holds also in our case for $(\alpha_{u,t})_{t>0}$. So starting from the update equation (3.4) and taking the scalar product by u we get

$$\begin{aligned}\tilde{\alpha}_{u,t+1} &= \alpha_{u,t} - \frac{\delta}{d} u \cdot \nabla_{sph} \mathcal{L}(w, (x_t, y_t)) \\ &= \alpha_{u,t} - \frac{\delta}{d} u \cdot \left(\nabla_{sph} \mathcal{L}(w) + \nabla_{sph} H^t(w)\right)\end{aligned}$$

Where we used the definition of *sample-wise loss* $H^t(w) = \mathcal{L}(w,(x_t,y_t)) - \mathcal{L}(w)$. Now we need to normalise $\alpha_{u,t} = {\tilde{\alpha}_{u,t}}/{||\tilde{\alpha}_{u,t}||}$ but instead we just estimate the denominator, using the fact that if $\alpha_u, \alpha_v \leq \frac{1}{2}$ by lemma 3.8 $\mathcal{L} \in \mathscr{C}^1(\mathcal{C},\mathbb{R})$. So restricting in $\mathcal{K}_\eta$ for any $\eta < \frac{1}{2}$ we have that there exists $A > 0$:

$$\sup_{w\in\mathcal{K}_\eta} |\nabla_{sph}\mathcal{L}(w)| \leq A \tag{3.32}$$

and letting

$$\Theta_t := \left|\frac{1}{\sqrt{d}}\nabla_{sph}H^t(w_{t-1})\right|^2 \tag{3.33}$$

It is straightforward to estimate:

$$1 \leq |\tilde{\alpha}_{u,t}| \leq 1 + \underbrace{\delta^2\left(\frac{A}{d^2} + \frac{\Theta_t}{d}\right)}_{\zeta}$$

Then we can use the chain of inequalities: $\frac{1}{|\tilde{\alpha}_{u,t}|} \geq \frac{1}{1+\zeta} \geq 1-\zeta$ to get:

$$\begin{aligned}\alpha_{u,t} \geq \alpha_{u,t-1} - \frac{\delta}{d}\nabla_{sph}\mathcal{L}(w)\cdot u - \frac{\delta}{d}u\cdot\nabla_{sph}H^t(w_{t-1}) - \delta^2\left(\frac{A}{d^2}+\frac{\Theta_t}{d}\right)|\alpha_{u,t-1}| \\ - \delta^3\left(\frac{A}{d^2}+\frac{\Theta_t}{d}\right)\left(\frac{1}{d}|\nabla_{sph}\mathcal{L}(w)\cdot u| + \frac{1}{d}|u\cdot\nabla_{sph}H^t(w_{t-1})|\right)\end{aligned} \tag{3.34}$$

Now, one of the key ideas employed in [16] is to split the random term

$$\delta^2\frac{\Theta_t}{d}|\alpha_{u,t-1}| = \delta^2\frac{\Theta_t\mathbb{1}_{\Theta_t\leq\hat{\Theta}}}{d}|\alpha_{u,t-1}| + \delta^2\frac{\Theta_t\mathbb{1}_{\Theta_t\geq\hat{\Theta}}}{d}|\alpha_{u,t-1}| \tag{3.35}$$

Where $\hat{\Theta}$ is a real parameter to be tuned carefully. So we can regroup the terms in (3.34) in two families (for the sake of brevity, we use the notation $\Lambda := |\nabla_{sph}\mathcal{L}(w)\cdot u| + |u\cdot\nabla_{sph}H^t(w_{t-1})|$):

$$\begin{aligned}\alpha_{u,t} \geq \alpha_{u,t-1} \overbrace{-\frac{\delta}{d}\nabla_{sph}\mathcal{L}(w_{t-1})\cdot u - \delta^2\frac{\Theta_t\mathbb{1}_{\Theta_t\leq\hat{\Theta}}}{d}|\alpha_{u,t-1}|}^{F_1} + \\ \underbrace{-\frac{\delta}{d}u\cdot\nabla_{sph}H^t(w_{t-1}) - \delta^2\left(\frac{A}{d^2}+\frac{\Theta_t\mathbb{1}_{\Theta_t\geq\hat{\Theta}}}{d}\right)|\alpha_{u,t-1}| - \frac{\Lambda\delta^3}{d}\left(\frac{A}{d^2}+\frac{\Theta_t}{d}\right)}_{F_2}\end{aligned} \tag{3.36}$$

The family F_2 is made up by terms that are either always negligible, or that can cause problems only if *very unlikely realisations* of H and Θ happen. They are estimated in lemmas 4.3 and 4.5 of [16], and, since our assumption 3.A.2.2 implies assumption B in [16], we are able to replicate those estimates also in our case.

On the contrary, to handle F_1 we need to make an adjustment: [16] relied on their formula (4.12), that was derived from the monotonicity and mono-dimensionality of the population loss, which we do not have. However, consider the gradient of the population loss in direction u. Applying lemma 3.8 and substituting the values from section 3.A.2.1 we have that $-u\nabla_{sph}\mathcal{L}(w) = \frac{1}{2}(c_{20}\alpha_u + c_{11}\alpha_v) + o(\eta)$, since both $c_{20} = 2\beta_u c_2^\sigma > 0$ and $c_{11} > 0$, taking η small enough and assuming $(\alpha_u,\alpha_v) \in \mathcal{K}_\eta$ we have that:

$$\frac{1}{4}(c_{20}\alpha_u + c_{11}\alpha_v) \leq -u\nabla_{sph}\mathcal{L}(w) \leq c_{20}\alpha_u + c_{11}\alpha_v \tag{3.37}$$

Hence, if we assume $\alpha_v \geq 0$, we get $-u\nabla_{sph}\mathcal{L}(w) \geq \frac{1}{4}c_{20}\alpha_u$, which is exactly what we needed to replicate equation (4.12) in [16], so as long as $\alpha_{u/v,t-1} > 0$ we have that:

$$\alpha_{u,t} \geq \alpha_{u,t-1} - \frac{\delta}{d}\left(\frac{c_{20}}{4} - \delta\Theta_t \mathbb{1}_{\Theta_t \leq \hat{\Theta}}\right)\alpha_{u,t-1} + F_2 \tag{3.38}$$

Hence we are back on the setting studied in [16], and using that $\delta \lesssim \frac{1}{\log d}$ we can apply also lemma 4.2 and proposition 4.4 in [16].

So, conditioning on positivity of $(\alpha_{v,t})_{t\leq\tau}$ we can apply to $(\alpha_{u,t})_t$ *Proposition 4.1* from [16], that implies

$$\begin{gathered}\lim_{d\to\infty}\inf_{w_0\in E_{\gamma/\sqrt{d}}}\mathbb{P}_{w_0}\left(\alpha_{u,t} \geq \frac{\alpha_{u,0}}{2} + \frac{\delta}{8d}\sum_{j=0}^{t-1} c_{20}\alpha_{u,j} \forall\ t \leq \tau \,\middle|\, \alpha_{v,t} \geq 0\ \forall t \leq \tau\right) = 1 \\ \text{where } \tau \text{ is a stopping time defined as} \qquad \tau := \inf\left\{t \leq n \middle|\ \alpha_{u,t} \leq \frac{\gamma}{2\sqrt{d}} \text{ or } \alpha_{u,t} \geq \eta\right\}\end{gathered} \tag{3.39}$$

where $\mathbb{P}_{w_0}$ denotes the probability conditioned on starting from w_0.

Recall discrete Gronwall inequality

$$x_n \geq a + b\sum_{i=0}^{n-1} x_i \quad \Longrightarrow \quad x_n \geq a\,(1+b)^n\,, \qquad a, b > 0$$

Using it, we get that, if the event in (3.39) happens, then

$$\alpha_{u,t} \geq \frac{\alpha_{u,0}}{2}\left(1 + \frac{\delta c_{20}}{8d}\right)^t \qquad \forall t \leq \tau \tag{3.40}$$

So, recalling that we are conditioning to be in E_γ, so $\alpha_{u,0} \geq \frac{\gamma}{\sqrt{d}}$, (3.40) proves that $\alpha_{u,t}$ will have surpassed η_u after a number of steps t_u^*:

$$t_u^* = \frac{\log\left(2\eta_u\sqrt{d}\right) - \log\gamma}{\log\left(1 + \frac{\delta c_{20}}{8d}\right)} \approx C\frac{d}{\delta}\log d \tag{3.41}$$

This proves weak recovery of the covariance spike in the time required to end the first part of proposition 3.5.

Weak recovery of the cumulant spike Now turn to $(\alpha_{v,t})_t$. Starting from the equivalent of (3.36), we cannot apply the same reasoning this time because

$$-v\cdot\nabla_{sph}\mathcal{L}(w) = c_{11}\alpha_u(1+o(\eta)) + 4c_{04}\alpha_v^3(1+o(\eta)) \qquad \alpha_u, \alpha_v \leq \eta. \tag{3.42}$$

There is no linear term in α_v, hence we cannot regroup the term $\delta\Theta_t\mathbb{1}_{\Theta_t\leq\hat{\Theta}}|\alpha_{v,t-1}|$ as did in (3.38).

We will instead prove a the following:

$$\alpha_{v,t} \geq \frac{\tilde{\gamma}}{2\sqrt{d}}\left(1 + \frac{\delta\min(c_{20}, c_{11})}{8d}\right)^t \tag{3.43}$$

for some $\tilde{\gamma}$ small enough. To show it, we apply lemma 4.3, 4.5 from [16], and (3.42), to get that:

$$\begin{gathered}\lim_{d\to\infty}\inf_{w_0\in E_{\gamma/\sqrt{d}}}\mathbb{P}_{w_0}\left(\alpha_{v,t} \geq \frac{\alpha_{v,0}}{2} + \frac{\delta}{4d}\sum_{j=0}^{t-1} c_{11}\alpha_{u,j} + 4c_{04}\alpha_{v,j}^3 - \frac{\delta^2}{d}\mathbb{1}_{\Theta_j\leq\hat{\Theta}}\alpha_{v,j} \quad \forall\ t \leq \tilde{\tau}\right) = 1 \\ \text{where } \tilde{\tau} \text{ is a stopping time defined as} \qquad \tilde{\tau} := \inf\left\{t \leq n \middle|\ \alpha_{v,t} \leq \frac{\gamma}{2\sqrt{d}} \text{ or } \alpha_{v,t} \geq \eta\right\}\end{gathered}$$

We need to divide in 2 cases to be able to carry one the estimates: suppose first that $\alpha_{u,s} \geq \alpha_{v,s}$ for all $s \leq t$. Hence $\delta^2 \alpha_{v,t} \leq \delta^2 \alpha_{u,t}$ hence we can also apply proposition 4.4 to get:

$$\begin{gathered}\lim_{d\to\infty} \inf_{w_0 \in E_{\gamma/\sqrt{d}}} \mathbb{P}_{w_0}\left(\alpha_{v,t} \geq \frac{\alpha_{v,0}}{2} + \frac{\delta}{8d}\sum_{j=0}^{t-1} c_{11}\alpha_{u,j} \quad \forall\ t \leq \mathcal{S}_u\right) = 1 \\ \text{where } \mathcal{S}_u \text{ is defined as } \mathcal{S}_u := \inf\left\{t \leq n \middle|\ \alpha_{v,t} \leq \frac{\tilde{\gamma}}{2\sqrt{d}} \text{ or } \alpha_{v,t} \geq \eta \text{ or } \alpha_{v,t} \geq \alpha_{u,t}\right\}\end{gathered} \tag{3.44}$$

This implies that as long as α_v is below α_u, it satisfies the same kind of Gronwall inequality, with different coefficients, leading to:

$$\alpha_{v,t} \geq \frac{\alpha_{v,0}}{2}\left(1 + \frac{\delta c_{11}}{8d}\right)^t \qquad t \leq \mathcal{S}_u$$

so it satisfies (3.43) for this selection of times. Now note that we actually proved a stronger property. If at any time t: $\frac{\tilde{\gamma}}{2\sqrt{d}}\left(1 + \frac{\delta \min(c_{11}, c_{20})}{8d}\right)^t \leq \alpha_{v,t} \leq \alpha_{u,t}$, then we know that same reasoning applies and, calling $\mathcal{S}_u^{(2)}$the next instant such that with $\alpha_{v,\mathcal{S}_u^{(2)}} \geq \alpha_{u,\mathcal{S}_u^{(2)}}$ we get that for all times $t \leq s \leq \mathcal{S}_u^{(2)}$:

$$\alpha_{v,s} \geq \frac{\alpha_{v,t}}{2}\left(1 + \frac{\delta c_{11}}{8d}\right)^{s-t}$$

hence $\alpha_{v,s}$ satisfies (3.43).

Consider now the other case: if at any instant t we have that $\alpha_{v,t-1} \geq \alpha_{u,t-1}$ we want to prove that it cannot happen that $\alpha_{v,t} \leq \frac{\tilde{\gamma}}{2\sqrt{d}}\left(1 + \frac{\delta \min(c_{20}, c_{11})}{8d}\right)^t$. To do this we use an estimate on the update equation (it can be easily verified that comes from the application of lemma 4.3,4.5 to formula (4.11) of [16], with their choice $\hat{\Theta} = d^{\frac{1}{2} - \frac{1}{4}\iota}$):

$$\begin{aligned}\alpha_{v,t} &\geq \left(1 - \frac{\delta^2}{d^{1/2+1/4\iota}}\right)\alpha_{v,t-1} + \frac{c_{11}\delta}{8d}\alpha_{u,t-1} - \frac{\tilde{\gamma}}{5\sqrt{d}} \\ &\geq \left(1 - \frac{\delta^2}{d^{1/2+1/4\iota}} + \frac{c_{11}\delta}{8d}\right)\alpha_{u,t-1} - \frac{\tilde{\gamma}}{5\sqrt{d}} \\ &\geq \left(1 - O\left(d^{-1/2}\right)\right)\frac{\alpha_{u,0}}{2}\left(1 + \frac{\delta c_{20}}{8d}\right)^t - \frac{\tilde{\gamma}}{5\sqrt{d}}\end{aligned}$$

Which verifies our requirement by taking $\tilde{\gamma}$ small enough with respect to γ and d large enough.

Hence we verified that conditioning on the the events (3.31) and (3.39) (that have arbitrarily large probabilities), also inequality (3.43) holds. This means that we can drop all the positivity requirements and have that, as long as $0 < \alpha_{u,t}, \alpha_{v,t} \leq \eta$, then with high probability $(\alpha_{u,t})_t$ satisfies (3.40) until it reaches η_u and $(\alpha_{v,t})_t$ satisfies (3.43) until it reaches η_v, hence we need to take n larger than the maximum of those bounds, which is (3.43):

$$n > t_v^* = \frac{\log\left(2\eta\sqrt{d}\right) - \log\tilde{\gamma}}{\log\left(1 + \frac{\delta \min(c_{20}, c_{11})}{8d}\right)} \approx C\frac{d}{\delta \min(c_{20}, c_{11})}\log d \tag{3.45}$$

Where the first constant C is just a number, does not depend on $\eta, \tilde{\gamma}$, for d large enough. Substituting the constraint $\delta \ll \frac{1}{\log d}$ (that we used in order to apply proposition 4.1 from [16]) we get that $n \gtrsim d\log^2 d$ is a sufficient sample complexity for weak recovery if δ is as close as possible to the

upper bound $\frac{1}{\log d}$. If instead we take a smaller learning rate $\delta \approx \frac{1}{d^2}$ we have that if $n \gg d^2 \log d$ weak recovery is guaranteed, as shown in table 3.1.

So the proposition is proved under the assumption that all the trajectories live in K_η with η sufficiently small. To conclude we need to make sure that it cannot happen that one of $(\alpha_{u/v,t})_t$ surpasses η while the other is still close to 0 and far away from $\eta_{u/v}$. Note that up until now we did not choose explicitly values for $\eta_{u/v}$, so it will be sufficient to take them way smaller than η to be sure that there will be no problems.

To be more precise, we will use the following bound on the maximum progress achieved in a fixed time interval.

Bound on maximum progress Supposing $\alpha_{u,t} > 0$ (all that we will do here works also for $\alpha_{v,t}$), since the direction of the spherical gradient is tangent to the sphere it follows that $||\tilde{w}_{u,t}|| > 1$, hence

$$\alpha_{u,t} \leq \tilde{\alpha}_{u,t} = \alpha_{u,t-1} - \frac{\delta}{d}\nabla_{sph}\mathcal{L}(w_{t-1}) \cdot u - \frac{\delta}{d} u \cdot \nabla_{sph} H^t(w_{t-1}) \tag{3.46}$$

Applying this inequality iteratively in a time interval $[t_1, t_2]$, assuming $\alpha_{u,t} > 0$ in all the interval, we get that

$$\alpha_{u,t_2} \leq \alpha_{u,t_1} - \sum_{j=t_1}^{t_2-1} \frac{\delta}{d}\nabla_{sph}\mathcal{L}(w_j) \cdot u + \frac{\delta}{d} u \cdot \nabla_{sph} H^{j+1}(w_j)$$

To estimate the martingale term we use the estimate from lemma 4.5 in [16] (which is an application of Doob maximal inequality):

$$\sup_{T \leq n} \sup_{w_0 \in \mathbb{S}^{d-1}} \mathbb{P}_{w_0}\left(\max_{t \leq T} \frac{1}{\sqrt{T}} \left| \sum_{j=0}^{t-1} \nabla_{sph} H^{j+1}(w_j) \cdot u \right| \geq r \right) \leq \frac{2C_1}{r^2}, \tag{3.47}$$

where $r > 0$ is a positive parameter and C_1 comes from assumption 3.A.2.2. Together with (3.32), we have that with probability larger than $1 - \frac{2C_1}{r^2}$

$$\alpha_{u,t_2} - \alpha_{u,t_1} \leq \frac{\delta}{Ad}(t_2 - t_1) + \frac{\delta}{d}\sqrt{t_2 - t_1} r \tag{3.48}$$

assume that at t_1 we have that $\alpha_{u,t_1} \geq \eta_1$ and that at $\alpha_{u,t_2} = \eta_2$ while $\alpha_{u,t} \geq \eta_1$ for all $t \in [t_1, t_2]$, choosing $r = \sqrt{t_2 - t_1}$ we get that with probability larger than $1 - \frac{2C_1}{t_2 - t_1}$

$$t_2 - t_1 \geq C(\eta_2 - \eta_1)\frac{d}{\delta} \gg C(\eta_2 - \eta_1) d \log d \tag{3.49}$$

so in the limit $d \to \infty$ with probability converging to 1 $t_2 - t_1 \gg d \log d$.

Application of the bound Suppose that at time t_2 $\alpha_{u,t_2} = \eta =: \eta_u$, hence there exists t_1 such that $\alpha_{u,t} \geq \frac{\eta}{2}$ for all $t \in [t_1, t_2]$. By (3.49), $t_2 - t_1 \geq \frac{C\eta}{2} d \log d$. If $\alpha_{v,t}$ is larger than $\alpha_{u,t}$ for any t in this interval, the statement is already verified by taking $\eta_v = \eta_u$, so assume $\alpha_{v,t} \leq \alpha_{u,t}$ for $t \in [t_1, t_2]$. We can apply eq. (3.44) to get that with probability converging to 1 in the limit $d \to \infty$:

$$\alpha_{v,t_2} \geq \frac{\alpha_{v,t_1}}{2} + \frac{\delta}{8d} \sum_{j=t_1}^{t_2-1} c_{11} \alpha_{u,j} \geq c_{11} \frac{\delta \eta_u}{16 d}(t_2 - t_1) \geq C\eta_u^2$$

where the constant C does not depend on d, hence we can take $\eta_v := C\eta_u^2$ there is weak recovery for the cumulant spike v even in the "unlucky" case where the overlap along u reaches η very fast.

The vice versa can be done analogously: assume that at time t_2 $\alpha_{v,t_2} = \eta =: \eta_v$. Again, let t_1 such that $\alpha_{v,t} \geq \frac{\eta}{2}$ for all $t \in [t_1, t_2]$. By eq. (3.49), $t_2 - t_1 \geq \frac{C\eta}{2} d \log d$. Now note that by (3.37) $-u \cdot \nabla_{\text{sph}} L(w) \geq \frac{c_{11}\alpha_{v,t}}{4}$, so the same reasoning as before applies leading to weak recovery with $\eta_u := C\eta_v^2$.

3.B Details on the experiments

3.B.1 Figure 3.1

Training We trained three deep convolutional neural networks, AlexNet [148], DenseNet121 [149] and Resnet18 [150] on the ImageNet 1000 class classification task [151]. We used standard training recipes for all three networks, training with SGD for 90 epochs. We employ a cosine learning rate schedule with initial learning rate of 0.1 for Resnet18 and DenseNet, and 0.01 for AlexNet. We further chose momentum of 0.9, weight decay of 10^{-4}, and mini-batch size of 256. During training, we randomly cropped images to 224×224 pixels, and randomly flipped them horizontally. For all three architectures, we made three runs with different random number seeds, and took 50 logarithmically spaced snapshots of the weights during each training run.

Subspace overlap We first verified whether the first-layer convolutional filters at the end of training were predominantly located in a subspace spanned by the leading principal components. To this end, we first computed the principal components of patches sampled at random from ImageNet, which look roughly like Fourier components in two dimensions see the left plot in fig. 3.4. We then determined the excess overlap of the final weights at the end of training w^* with the principal subspace. To this end, we vectorised the convolutional filters $w^* d$, which have dimension d, and computed the excess overlap with a matrix of the leading k eigenvectors $U \in \mathbb{R}^{k \times d}$ as

$$m = 1 - \frac{\left\| w - U^\top U w \right\|_2}{\|w\|_2} - k/d, \tag{3.50}$$

where the factor of k/d is the overlap of the filters with a *random* basis of rank k. The excess overlap m thus captures how much of the vector w is captured by the basis U in excess of what would be captured by a random basis of the same size. We plot the excess overlap m against the relative rank k/d for the filters of the three networks in fig. 3.4 There is a peak around $k = .4d$. We then plotted the excess overlap of the filters with the fixed subspace of the leading $k = .4d$ principal components for each snapshot of the networks, yielding the green curves in fig. 3.1.

Excess kurtosis For each weight snapshot, we extracted the weights for each first-layer convolution and computed the dot-products $s_\mu^k = w^k \cdot x^\mu$ between the weight of the kth convolution for randomly sampled patches from ImageNet x^μ. We normalised the s_μ^k for each neuron (i.e. fixed k) to have variance 1, and computed the excess kurtosis of the resulting s_μ^k,

$$\kappa_{\text{ex}}(s) = \mathbb{E}s^4 - 3. \tag{3.51}$$

We then averaged over all the neurons to obtain the curves shown in fig. 3.1(b).

3.B.2 Experiments with two-layer networks

Training In all our experiments with two-layer neural networks on *synthetic* data, be it on the mixed cumulant model or in the teacher-student setup, we trained two-layer neural networks using

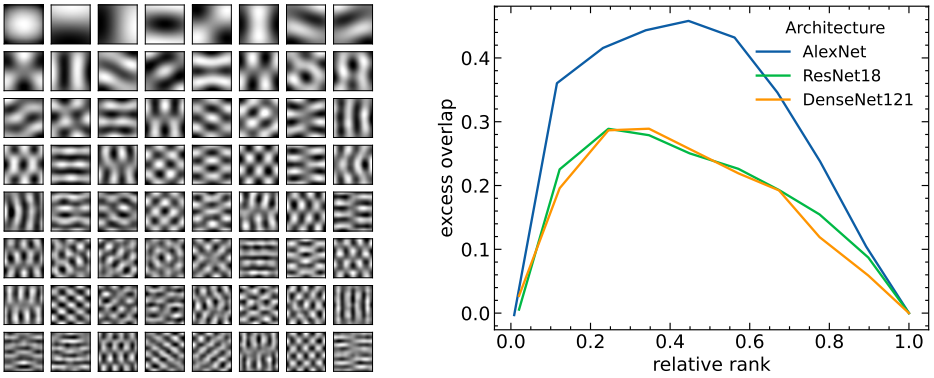

Figure 3.4: **The final filters of deep neural networks are concentrated in the subspace spanned by the principal components.** *(Left)* The first 64 principal components of patches of 11×11 pixels (the size of AlexNet filters) sampled randomly from ImageNet images. Filters are shown here for the red channel and shown in greyscale; filters for the other channels look the same. *(Right)* For three deep convolutional neural networks trained on ImageNet, we plot the excess overlap, eq. (3.50), of the final first-layer filters with the principal subspace of increasing rank, normalised by input dimension since filters in the three architectures have different sizes. The excess overlap captures how much of the filters is contained by the leading principal components compared to a random basis of the same rank.

online stochastic gradient with mini-batch size 1. Note that we used different learning rates for the first and second layer, with the second layer learning rate $\eta_2 = \epsilon\eta_1$ and $\epsilon = 0.01$ unless otherwise noted. It has been noted several times that rescaling the second-layer learning rate in this way ensures convergence to a well-defined mean-field limit [135, 152]

3.B.2.1 Figure 3.2

CIFAR10 We trained a two-layer neural network with $m = 32^2 * 4 = 4096$ hidden neurons on grayscale CIFAR10 images to ensure the same ratio between hidden neurons and input dimension as in our experiments with synthetic data. We set the mini-batch size to 128, weight decay to $5 \cdot 10^{-4}$, and momentum to 0.9. Images were transformed to grayscale using the pyTorch conversion function. We trained for 200 epochs on the cross-entropy loss and used a cosine learning rate scheduler. The final test accuracy of the networks was just below 50%.

Synthetic data distributions We trained the two-layer neural network on a data set sampled from the mixed cumulant model with signal-to-noise ratios $\beta_m = 1, \beta_u = 5, \beta_v = 10$. The test error on this data set is shown in red ("Full data"). For the independent latent variables, we used $\lambda^\mu \sim \mathcal{N}(0, 1)$ and $\nu^\mu = \pm 1$ with equal probability. For correlated latent variables, we set $\nu^\mu = \text{sign}(\lambda^\mu)$. We also show the test loss of the same network evaluated on the following censored data sets:

- $\beta_m = 1, \beta_u = 0, \beta_v = 0$ (blue, "mean only")
- $\beta_m = 1, \beta_u = 5, \beta_v = 0$ (green, "mean + cov")

- A Gaussian equivalent model (orange), where we sample inputs for the spiked class from a Gaussian distribution with mean $\beta_m m$ and a covariance

$$\mathrm{cov}_{\mathbb{P}}(x,x) = \mathbb{1} + \beta_u u u^\top + \sqrt{\beta_u \beta_v (1-\gamma)^2}\, \mathbb{E}\, \lambda\nu \left(u v^\top + v u^\top \right) \tag{3.52}$$

3.B.2.2 Teacher-student setup (figure 3.3)

We trained the same two-layer network with the same hyperparameters as in fig. 3.2. Defining the pre-activations $\lambda_i = \langle u_i, x \rangle$ for the three spikes u_i, the tasks were:

A inputs $x \sim \mathcal{N}(0, \mathbb{1})$, teacher

$$y(x) = h_1(\lambda_1) + h_2(\lambda_2) + h_4(\lambda_3) \tag{3.53}$$

B For the simulation shown in A, We computed the absolute normalised overlaps $|\langle w_k, u_i \rangle| / \|w_k\|$ of all the first-layer weights w_k with a given spike, and took the average over the five highest values.

C Same as in B, except that we sampled inputs from a normal distribution with zero mean and covariance $\mathbb{1} + \gamma(uv^\top + vu^\top)$, which introduces correlations between the pre-activations λ_2 and λ_3 of the teacher model.

D Same as in B, except that we trained on isotropic Gaussian inputs $\mathcal{N}(0, \mathbb{1})$ and instead added a co-linear cross-term between spikes u and v, akin to the cross-term that is found in our expansion of the mixed cumulant task,

$$y(x) = h_1(\lambda_1) + h_1(\lambda_2) h_1(\lambda_3) + h_2(\lambda_2) + h_4(\lambda_3) \tag{3.54}$$

Chapter 4

Feature learning from non-Gaussian inputs: the case of Independent Component Analysis

4.1 Introduction

The practical success of deep neural networks is generally attributed to their ability to learn the most relevant input features directly from data [153]. A classic example for this feature learning are deep convolutional neural networks (CNNs), which recover Gabor filters [32] like the ones shown in fig. 4.1(a) in their first layer when trained on sets of natural images like ImageNet [148, 154, 155]. Since Gabor filters are localised in both space and frequency domains, they are effective for capturing edges and textures in natural images [156]. How these filters emerge from training a deep network end-to-end on natural images with stochastic gradient descent remains a key question for the theory of neural networks.

Deep neural networks are not the only approach to learning Gabor filters directly from data. Instead, it is well-known that independent component analysis (ICA) [71, 117, 157–159], a simple, unsupervised learning algorithm, learns filters that are similar to the first-layer filters of a deep CNN when applied to patches of natural images, see fig. 4.1(b). The key idea of ICA is to identify directions of maximal non-Gaussianity in input space, suggesting that the non-Gaussian structure in data plays a crucial role in shaping the representations learnt by deep neural networks [28]. More precisely, given a set of n zero-mean inputs $\mathcal{D} = \{x^1, x^2, \dots, x^n\}$, ICA seeks the direction $w^* \in \mathbb{R}^d$ that maximises the non-Gaussianity of the projections of the inputs $s := w \cdot x$,

$$w^* := \underset{\|w\|=1}{\arg\max}\ \mathbb{E}_{\mathcal{D}}\, G(w \cdot x), \tag{4.1}$$

where the contrast function G is a measure of the non-Gaussianity of the projection s, for example its excess kurtosis $G(s) = s^4 - 3$. For ICA, inputs are always pre-whitened, so ICA can be seen as a refinement of standard principal component analysis, which instead seeks the projection of the data that maximises its variance, i.e. $G(s) = s^2$. On images, PCA yields a completely different set of filters which is spatially extended and oscillating due to the approximate translation-invariance of the image patches [156], see fig. 4.1(c). A closer look at the dynamics of the CNNs trained on the standard ImageNet dataset [151] reveals further similarities between ICA and deep CNNs. In fig. 4.1(d), we plot the excess kurtosis of the first-layer projections $s^k_\mu = w^k \cdot x^\mu$ of ImageNet patches x^μ along the weights $\{w^k\}_{k=1,\dots,K}$ of the K first-layer filters of three different CNNs during

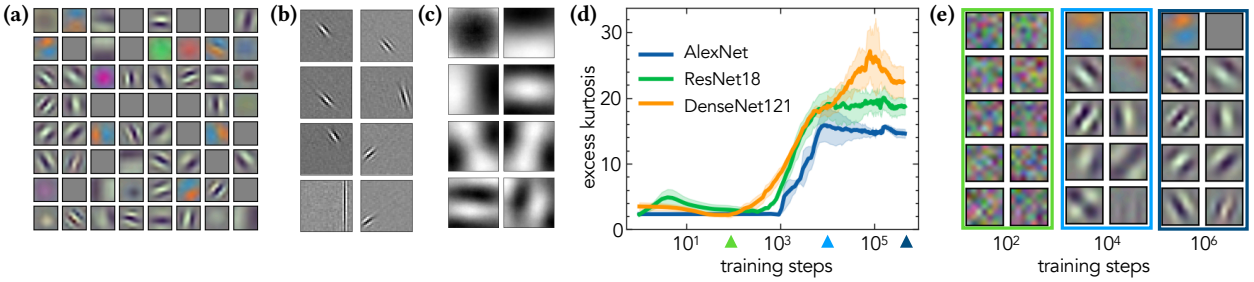

Figure 4.1: **Independent component analysis yields similar filters as deep convolutional neural networks. (a)** The first-layer convolutional filters extracted from an AlexNet trained on ImageNet show the hallmarks of Gabor filters [32], like orientation, localisation, and bandwidth-specificity. **(b)** Independent components found with online SGD on 100 000 64×64 patches extracted from ImageNet. **(c)** Leading principal components of the same ImageNet patches. **(d)** Excess kurtosis of the representations of the first convolutional layer of three deep CNNs during training on ImageNet. For each filter, we compute the dot product s between its weights and patches sampled from ImageNet, then compute its excess kurtosis averaged over image patches and neurons. **(e)** Snapshots of a subset of convolutional filters of AlexNet at various points during training. As the excess kurtosis of representations increases, prototypes of Gabor filters emerge, which are then sharpened during the remainder of training. Full experimental details in section 4.A.1.

training, averaged over patches and neurons. If the projections s_μ^k are normally distributed, the excess kurtosis is (close to) zero, as is the case early during training when weights are close to their (uniform) initialisation. In all three networks, the non-Gaussianity of projections increases sharply between 10^3 and 10^4 steps, before plateauing. Interestingly, the moment in which the deep CNN focuses on non-Gaussian projections is precisely the moment in which the Gabor filters form, see fig. 4.1(e).

Taken together, the similarity between filters obtained by deep CNNs and ICA and the dynamics of the excess kurtosis of first-layer representations of deep CNNs suggest that ICA can serve as a simplified, yet principled model for studying feature learning from non-Gaussian inputs.

In this work, we develop a quantitative theory of feature learning from non-Gaussian inputs by providing a sharp analysis of the sample complexity of two algorithms for Independent Component Analysis, namely FastICA, the most popular ICA algorithm used in practice, and SGD, which is used to train deep neural networks.

Early theoretical works on independent component analysis focused mostly on the speed of convergence of different algorithms for finding w^* in the classical regime of fixed input dimension and large number of samples; the most popular algorithm for performing ICA, FastICA [71], derives its name from the fact that it converges with a quadratic [160] rather than linear rate. However, ICA struggles in modern applications where inputs tend to be high-dimensional; for example, its performance is highly sensitive to its initial conditions [161, 162]. For algorithms running in high dimensions, the main bottleneck is typically not the speed of convergence, but the length of the initial search phase before the algorithm recovers any trace of a signal [163, 164]. The focus of this chapter is therefore in establishing lower bounds on the sample complexity required by FastICA and SGD for escaping the search phase.

Recently, Auddy & Yuan [162] showed that in high dimensions, $n \gtrsim d/\varepsilon^2$ samples are necessary and sufficient to ensure the existence of estimates of the non-Gaussian directions with an error up to ε when given unlimited computational resources, while the sample complexity required for polynomial time algorithms is $n \gtrsim d^2/\varepsilon^2$. They also proposed an initialisation scheme for FastICA

that reaches the optimal performance. While these results establish fundamental statistical and computational limits of ICA, a precise and rigorous analysis of how the two main algorithms used in practice, FastICA and SGD, escape the search phase starting from random initialization is lacking.

In this chapter, we derive sharp algorithmic thresholds on the sample complexity required by ICA to recover non-Gaussian features of high-dimensional inputs, both for FastICA and SGD. Our analysis leverages recent breakthroughs from supervised learning with Gaussian inputs [16, 18, 165] that we generalise to the unsupervised case with non-Gaussian inputs. The difference in the input distribution is the main technical challenge: it adds complexity to our analysis both by introducing additional terms whose statistics need to be controlled, and by impacting qualitatively the intrinsic properties of the loss function.

Contributions (Sample complexities for FastICA & SGD) Our main results on the time scales for learning a non-Gaussian direction with FastICA and SGD are as follows:

- We prove that the popular FastICA algorithm exhibits poor sample complexity, requiring a large-size batch of $n \gtrsim d^4$ to recover a hidden direction in the first step (see section 4.3.3);
- We prove that the sample complexity of online SGD can be reduced down to the computational threshold of $n \gtrsim d^2$, at the cost of fine-tuning the contrast function in a data-dependent way (see section 4.3.5);

In practice, when implemented for D-dimensional images, FastICA is run in the subspace spanned by the leading principal components of the data, with a common prescription suggesting to use a subspace of dimension $d = D/4$ [156]. While this convention by itself does not alleviate the poor sample complexity, improving only the constants, it raises a couple of intriguing question: is it possible to reliably find non-Gaussian directions in high dimensions ($D \to \infty$) by running FastICA in a subspace of fixed dimension $d = O_D(1)$? If so, can we prove that it does, and what does it tell us about the structure of images?

Contributions (Real preprocessed data) Here, we answer all the questions in the affirmative:

- We demonstrate experimentally that FastICA exhibits a search phase at linear sample complexity when trained on ImageNet patches and that running FastICA in a subspace of fixed dimension $d = O_D(1)$ yields non-Gaussian directions in the D-dimensional image ambient space (section 4.4.1);
- We introduce a new "subspace" model for data, for which we prove that strong recovery of a non-Gaussian direction is possible at linear sample complexity when running ICA in a suitable principal subspace (section 4.5);

4.2 Setup

We first describe the most popular algorithm to perform ICA, called FastICA, and we introduce the classic ICA model for synthetic data which will allow us to systematically test the performance of FastICA.

4.2.1 FastICA and contrast functions

Performing ICA on a generic set of inputs $x \in \mathbb{R}^d$ drawn from a data distribution $\mathbb{P}$ with zero mean and identity covariance means finding a unit vector $w \in \mathbb{S}^{d-1}$ which is an extremum of the *population* loss

$$\mathcal{L}(w) := \mathbb{E}_{\mathbb{P}}\, G(w \cdot x), \tag{4.2}$$

where $G : \mathbb{R} \to \mathbb{R}$ is a suitable "contrast function" which measures the non-Gaussianity of the projections $s = w \cdot x$. The two standard choices for $G(s)$ used in practice [120] are

$$G(s) := -e^{-s^2/2} \quad \text{and} \quad G(s) := \frac{\log\cosh(as)}{a} \tag{4.3}$$

for some $a \in [1, 2]$. Another classical choice is the excess kurtosis, $G(s) := s^4 - 3$, even though it is more sensitive to outliers; therefore, the first two contrast functions are preferred in practice since their growth is slower than polynomial. In practice, the expectation is approximated as an average over a set of n centered and whitened inputs $\mathcal{D} = \{x^\mu\}_{\mu=1}^n$ (that we assume to be drawn i.i.d. from $\mathbb{P}$).

As an optimisation algorithm, we focus on the classic FastICA algorithm of Hyvärinen & Oja [71]. While a number of works have recently proposed alternative algorithms with provable guarantees, e.g. Arora *et al.* [166], Anandkumar *et al.* [141], or Voss *et al.* [167], we focus instead on FastICA since it is the most popular ICA algorithm (it is the reference implementation in scikit-learn [120]).

FastICA finds extremal points of the population loss (4.2) via a second-order fixed-point iteration. Here we state the algorithm; see Hyvärinen & Oja [71] for a detailed derivation and discussion. Given a dataset $\mathcal{D}$, we initialise the weight vector randomly on the unit sphere, $w_0 \sim \mathrm{Unif}(\mathbb{S}^{d-1})$, and then iterate the **FastICA updates** for $t \geq 1$ until convergence:

$$\begin{cases} \widetilde{w}_t &= \mathbb{E}_{\mathcal{D}}[x\, G'(w_{t-1} \cdot x)] - \mathbb{E}_{\mathcal{D}}[G''(w_{t-1} \cdot x)] w_{t-1}, \\ w_t &= \widetilde{w}_t / \|\widetilde{w}_t\|. \end{cases} \tag{4.4}$$

The FastICA iteration is made of two contributions: a "gradient" term $\nabla_w \mathcal{L}(w, x)\big|_{w=w_{t-1}} = x\, G'(w_{t-1} \cdot x)$, which drives the algorithm towards a direction where the gradient vanishes, and the "regularisation" $\mathbb{E}_{\mathcal{D}}[G''(w_{t-1} \cdot x)] w_{t-1}$, which ensures quadratic rather than the linear convergence of first-order methods, giving FastICA its name. As we will show in proposition 4.3, the regularisation term is also key for efficient learning.

4.2.2 The ICA data model

To carefully test the sample complexity of FastICA and other ICA algorithms like SGD in a controlled setting, we first study the case of the synthetic *spiked cumulant model* from chapter 2. The projection of the inputs along the spike v yields a non-Gaussian distribution, so this is the direction that ICA should recover. Specifically, we recall that the data points $\mathcal{D} = \{x^\mu\}_{\mu=1}^n$ are drawn according to

$$x^\mu = S\left(\sqrt{\beta}\, \nu^\mu v + z^\mu\right) \in \mathbb{R}^d, \tag{4.5}$$

where $z^\mu \sim \mathcal{N}(0, \mathbb{1}_d)$ is a vector with white noise, while the scalar random variable ν^μ is the latent variable for each input, and follows a non-Gaussian distribution. For concreteness, we choose a Rademacher distribution under which $\nu^\mu = \pm 1$ with equal probability. The signal-to-noise ratio $\beta \geq 0$ sets the relative strength of the non-Gaussian component along v compared to the Gaussian

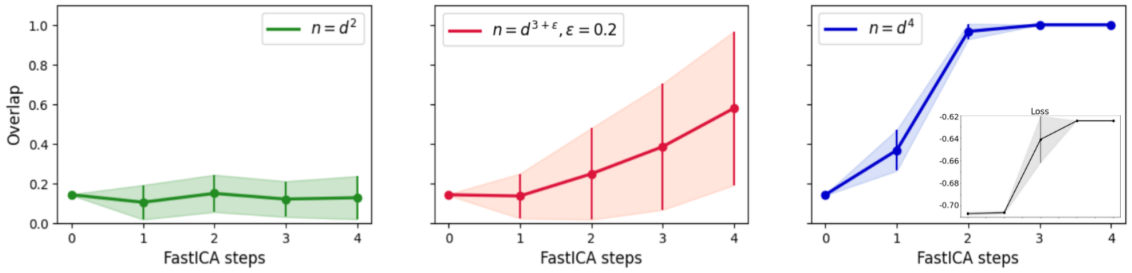

Figure 4.2: **Performance of FastICA on the spiked cumulant model for various sample complexity regimes.** We run the standard FastICA updates (4.4) on data drawn from a noisy ICA model (4.5). At each step of the algorithm, we draw a new batch of size $n = d^2$ (**left**), $n = d^{3+\varepsilon}$ with $\varepsilon = 0.2$ (**middle**) and $n = d^4$ (**right**). In the latter case, we also show the corresponding loss function. We plot the overlap between the planted non-Gaussian direction v and the estimate w produced by FastICA. While recovery of the spike is theoretically possible at $n = d^2$ (see section 4.2.3), FastICA does not recover the spike at that sample complexity, and instead requires $n = d^4$ samples to recover the spike v reliably. Theorem 4.4 confirms this picture for a general data model. Full experimental details in section 4.A.2 and additional cases $\varepsilon = 0.5, 0.8$ in fig. 4.6.

part z^μ of each input. Since whitening the inputs is a standard preprocessing step for FastICA [71], we finally pre-multiply the inputs with the whitening matrix

$$S = \mathbb{1}_d - \frac{\beta}{1 + \beta + \sqrt{1+\beta}} vv^\top \in \mathbb{R}^{d \times d},$$

which ensures that the covariance of the inputs is simply the identity. This whitening step makes it impossible to trivially detect the non-Gaussian direction v by performing PCA or any other method which is based only on the information in the first and second cumulant. An alternative perspective of the ICA model is that it provides inputs where the noise is correlated, and in particular has a negative correlation with the spike to be recovered.

4.2.3 Fundamental limits of recovery

We now briefly recall from chapter 2 the fundamental limits of recovering the spike in the ICA data model (4.5), which will serve as benchmarks to evaluate the performance of specific algorithms like FastICA or SGD. From an information-theoretic point of view, i.e. when given unbounded computational resources, the presence of the non-Gaussian direction v can already be detected using a linear number of samples $n \gtrsim d$. However, detecting the spike at this sample complexity is only possible through exponentially complex algorithms that are based on exhaustive search. Efficient algorithms, that can run in polynomial time in d, require at least a quadratic number of samples $n = \omega(d^2)$.

For details on the algorithmic result, see Dudeja & Hsu [29] and Szekely *et al.* [30] and Auddy & Yuan [162] for a thorough analysis using low-degree methods, and Diakonikolas *et al.* [44] for a Statistical Query (SQ) analysis, which reaches equivalent predictions using the low-degree to SQ equivalence [79]. Auddy & Yuan [162] establishes the information-theoretic limits of reconstructing the spike using a min-max type analysis. A similar statistical-to computational gap appears in the related problem of tensor PCA [57].

4.2.4 Experiment: FastICA requires a lot of data to recover the non-Gaussian direction

We are now in a position to evaluate the FastICA algorithm in the controlled setting of the noisy ICA model (4.5), where the goal is to recover the planted non-Gaussian direction (or "feature") v. Specifically, we can compare the sample complexity of FastICA, i.e. the number of samples FastICA requires to recover v, to the fundamental limits of efficient or polynomial-time recovery, which is $n \gtrsim d^2$.

In fig. 4.2, we plot the absolute value of the overlap between the spike v and the estimate w obtained by running FastICA for a couple of steps with $n = d^\theta$ samples. In the plot, we consider the limit of online learning, in the sense that we draw a new training set at each step of the algorithm; in fig. 4.7, we show that the picture does not change when using the same training set repeatedly; see also our discussion in section 4.3.3.3.

We can see immediately that at quadratic sample complexity, FastICA does not recover the spike at all. Instead, the overlap is stuck at a value of roughly $1/\sqrt{d}$, which is the typical overlap of a randomly direction in $\mathbb{R}^d$ with the spike. This small value for the overlap suggests that the algorithm is stuck in the search phase, and does not recover the spike at quadratic sample complexity. With $n = d^4$ samples instead, FastICA recovers the spike within a few steps. In the intermediate regime of $n = d^{3+\varepsilon}$ samples, $\varepsilon = 0.2$, the algorithm recovers the spike more slowly, and there are large fluctuations.

These experiments suggest that FastICA has a poor sample complexity that is quartic in the input dimension for recovering the spike, with an intermediate regime at $n = d^{3+\varepsilon}$ with $\varepsilon \in (0, 1)$. Our theoretical analysis of FastICA will now confirm that these are indeed the fundamental limits of FastICA.

4.3 Theoretical Results

Before diving into our analysis of FastICA and stochastic gradient descent (SGD) for performing ICA, we highlight how to address the key technical difficulty in analysing learning from non-Gaussian inputs.

4.3.1 ICA, likelihood ratios, and information exponents

The key difficulty in analysing the ICA loss (4.2) or the FastICA update (4.4) is that it entails an average over an explicitly non-Gaussian distribution of the inputs. This is in contrast to the vast majority of works on (supervised) learning algorithms, which usually consider a Gaussian input distribution, an input distribution that is effectively Gaussian via the Gaussian Equivalence Theorem, or inputs that are distributed uniformly on the unit hypercube (as discussed at the end of the section).

Similarly to our study of the mixed cumulant model in chapter 3, the key idea of the theoretical analysis of FastICA and SGD is to rewrite the average over the non-Gaussian inputs as an average over the standard normal distribution $\mathbb{Q} = \mathcal{N}(0, \mathbb{1}_d)$, as

$$\mathcal{L}(w) := \mathbb{E}_{\mathbb{P}}[G(w \cdot x)] = \mathbb{E}_{\mathbb{Q}}[G(w \cdot x)\ell(v \cdot x)] \tag{4.6}$$

by introducing the *likelihood ratio*

$$\ell(s) := \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}}(s), \tag{4.7}$$

akin to how one would proceed in an analysis with the second moment method in hypothesis testing, as seen in chapter 2. Intuitively, the likelihood ratio (or more precisely, its norm) measures how different the distribution of the projection $s = w \cdot x$ is from a standard Gaussian distribution. The likelihood ratio will thus be the key object in our study to determine the number of samples that a given algorithm like FastICA requires to find a projection that is distinctly non-Gaussian and yields a large likelihood ratio.

Under the noisy ICA model (4.5), the likelihood ratio (4.7) depends only on the projection of the inputs along the non-Gaussian direction v to be learnt. We can expand the population loss in terms of a single scalar variable, the overlap $\alpha := w \cdot v$ between weight w and spike v:

$$\mathcal{L}(w) = \mathbb{E}_{\mathbb{Q}}[G(w \cdot x)\ell(v \cdot x)] = \sum_{k \geq 0}^{\infty} \frac{c_k^G c_k^\ell}{k!} \alpha^k, \tag{4.8}$$

where the last equality has been obtained by expanding the contrast function G and the likelihood ratio ℓ in series of Hermite polynomials and by using the orthogonality of Hermite polynomials (lemma A.5 in appendix A). A key quantity for the analysis of ICA is the order k^* of the first term in the expansion (4.8) that has a non-zero coefficient. This quantity was introduced as the "*information exponent*" by Ben Arous *et al.* [16] in the context of supervised learning, when they showed that it governs the sample complexity of online SGD in single-index models, like the noisy ICA model. As we saw in section 1.2, quantities like the *leap index* of Abbé *et al.* [131] and Dandi *et al.* [165] and the *generative exponent* of Damian *et al.* [21] govern the sample complexity for more complex data models with several directions to be learnt. Here, we note that due to the whitening of the inputs, the population loss (4.2) for the noisy ICA model model has information exponent $k^* = 4$, as we show in section 4.C.1.

4.3.2 Main assumptions

For the following theoretical analysis, we assume that:

Assumption 4.1. The contrast function G and the data distribution $\mathbb{P}$ are such that the loss (4.2) is a function of the overlap $\alpha := w \cdot v$, for a given spike v with unit norm. Moreover, we assume that $\ell(\alpha) = \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}}(\alpha) \in \mathrm{L}^2(\mathbb{R}, \mathbb{Q})$.

This assumption is trivially satisfied for the noisy ICA model (4.5) and the standard contrast functions in eq. (4.3), but it allows us to prove our results on the sample complexity of FastICA (theorem 4.4) and smoothed SGD (theorem 4.7) for arbitrary sub-Gaussian latent variables ν (see definition 4.11 in section 4.B). In principle, these hypotheses exclude the case of a fat-tailed distributions on the latent variable, i.e. those that do not belong to $\mathrm{L}^2(\mathbb{R}, \mathbb{Q})$. However, the results of simulations with a Laplace prior on ν (see fig. 4.9) reveal the same picture as in the thin-tailed case, suggesting that this assumption could be further relaxed.

Assumption 4.2 (Contrast function). We assume G to be even. For our analysis of FastICA, we require $G \in C^3(\mathbb{R})$ with bounded derivatives. For smoothed SGD in section 4.3.5, we additionally require

$$\sum_{k \geq 0}^{\infty} (c_k^G)^2 / k! < 1$$

and that there exist some $C_1, C_2 > 0$ such that, for any $s \in \mathbb{R}$, $|G'(s)| \leq C_1(1 + s^2)^{C_2}$ holds.

4.3.3 Sample complexity of FastICA

Our goal is to rigorously quantify the sample complexity of FastICA on the noisy ICA model (4.5). FastICA proceeds by performing a few iterations of the update (4.4) with the full data set. This setting is precisely the large-batch [19, 134] or "giant step" [165] regime that has attracted a lot of theoretical interest recently. Following the approach of these recent works, the key idea of our analysis is to study whether a single step of FastICA with a large number of samples $n \asymp d^\theta$ yields an estimate w_1 that has an overlap with the spike v that does not vanish with the input dimension.

4.3.3.1 The importance of regularisation

It is instructive to first consider briefly the ideal case of infinite data. The following proposition 4.3 shows that FastICA is able to fully recover the spike v as $d \to \infty$ if we replace the empirical average $\mathbb{E}_{\mathcal{D}}$ with the population average $\mathbb{E}_{\mathbb{P}}$. More interestingly, we show that it is the regularisation term, originally introduced to speed up convergence (see section 4.2.1), that gives the essential contribution to escape mediocrity at initialisation.

Proposition 4.3 (Infinite batch size)**.** *Assume that the inputs are distributed according to the noisy ICA model* (4.5) *and that* $c_4^G \neq 0$ *- this holds for the standard contrast functions in eq.* (4.3)*. Set* $\alpha_1 := w_1 \cdot v$*, where* w_1 *is the updated weight vector given by the first iteration of FastICA.*

Then, in the infinite batch-size limit, we have that

$$\alpha_1^2 = 1 - o(1).$$

On the other hand, if the regularisation term is missing and only the "gradient step" is taken, i.e. the iteration reads $\widetilde{w}_t = \mathbb{E}[x\, G'(w_{t-1} \cdot x)]$*, then we obtain* $\alpha_1^2 = O(1/d)$*.*

For the simple proof, see section 4.C.1.

4.3.3.2 Finite samples: hardness of learning

We now provide almost sharp bounds on the sample complexity of FastICA for learning the non-Gaussian feature v. For the analysis, we follow the recent "one-step" analyses [19, 134, 165] and study the overlap of the spike with the weight obtained from performing a single step of FastICA with a large batch of $n = d^\theta$ samples. More precisely, we provide a lower bound (positive results) for the amount of signal that can be weakly recovered given a number of samples on the order of d^{k^*}, where k^* is the information exponent of the loss, and precisely $k^* = 4$ for the noisy ICA model. In the data-scarce regime where $n = o(d^{k^*})$, we provide upper bounds (negative results) on the overlap $\alpha = w \cdot v$ that can be achieved in a single step of FastICA. We distinguish two cases: either there is no improvement with respect to the random initialisation, or the upper bound on the amount of signal learned is dimensionality-dependent, meaning that $n = \Theta(d^{k^*-1})$ are still not sufficient to exit from the search phase in the high-dimensional limit. Yet this regime smoothly bridges the situation of total ignorance, where nothing is learnt at all, and recovery of the spike in one step, thanks to a continuous dependence on the parameter δ.

Theorem 4.4. *(Finite batch size) Let* k^* *be the information exponent of the population loss* (4.2)*. Consider a number of samples* $n = \Theta(d^{k^*-\delta})$*, for* $\delta \in [0, 2]$*. Set* $\alpha_1 := w_1 \cdot v$*, where* w_1 *is the updated weight vector given by the first iteration of FastICA. Then,*

$$\begin{cases} \delta \in (1, 2] & \Rightarrow \quad \alpha_1^2 = O\left(\frac{1}{d}\right), \\ \delta \in (0, 1] & \Rightarrow \quad \alpha_1^2 = O\left(\frac{1}{d^\delta}\right), \\ \delta = 0 & \Rightarrow \quad \alpha_1^2 \geq 1 - o(1). \end{cases}$$

We prove theorem 4.4 in section 4.D. Even if theorem 4.4 is not specific to the noisy ICA model, and instead applies to any input model and contrast function satisfying Assumptions 4.1 and 4.2, it is instructive to look at the special case of Rademacher prior on the latent variable and standard contrast function, eq. (4.3). In this setting, we have proved in section 4.C.1 in the appendix that $k^* = 4$. Therefore, we find that $n = \Theta(d^4)$ is sufficient to perfectly recover the spike with a single step, up to contributions which vanish when d goes to infinity, in line with what we see in the experiments of fig. 4.2. Moreover, if $d^3 \lesssim n \ll d^4$, it turns out that the overlap $\alpha_1 = w_1 \cdot v$ is bounded by a dimension-dependent constant, which explains the slow increase of the overlap in that regime. For quadratic sample complexity, the gradient in the FastICA update does not concentrate, so the algorithm does not recover the signal and remains stuck in the search phase.

Key elements of the proof. The overall ideas of the proof are similar to those employed in the analysis of the teacher-student setup of Theorems 1 and 2 from [165], with the following main differences. First of all, in our unsupervised case, instead of isotropic inputs, the data points are in general non-Gaussian distributed; as already observed in Section 4.2.3, this impacts on the information exponent k^* of the population loss. Second, unlike for SGD, the iteration of FastICA presents both a gradient term and a regularisation term which is data-dependent (the regularisation "constant" depends on $G''(w \cdot x)$, and its analysis is hence just as complex as that of the gradient term). The general strategy is computing the expectations for both the signal $\widetilde{w}_1 \cdot v$ and the noise $\|\widetilde{w}_1\|$, and then identifying how much data n allows for the concentration of the two quantities. To be able to compute the expectations, we exploit the orthogonality properties of Hermite polynomials proved in Corollary A.7. Once concentration of both signal and noise is established, the positive result corresponds to the signal dominating the noise for $\delta = 0$, noise dominating the signal for the regime $\delta \in (0, 1]$. Conversely, if the signal does not concentrate, n is not sufficiently large for the overlap to escape from its scaling at initialisation and first upper bound holds. □

4.3.3.3 Discussion

The almost sharp analysis of the high-dimensional asymptotics of FastICA in theorem 4.4 is our first main result, and explains theoretically why FastICA tends to struggle in high dimensions [161, 162]. While FastICA is often applied after reducing the dimension of the inputs using PCA, it should be noted that the scaling of $n \asymp d^4$ on the noisy ICA model is poor even in moderate to small dimensions. One non-standard way to escape this scaling is to use the ad-hoc initialization scheme proposed recently by Auddy & Yuan [162], which allows to find an initialisation that gives ICA a warm start with a non-trivial overlap given only $n \gtrsim d^2$ samples. However, there remains a large gap between the performance of FastICA as it is commonly used, and the $n \gtrsim d^2$ or, more generally speaking, $n \gtrsim d^{k^*/2}$ bounds we have from low-degree methods, see [21]. One may therefore ask whether other algorithms, and in particular the simple stochastic gradient descent, may be more efficient at performing ICA, not least since neural networks trained with stochastic gradient descent seem to learn ICA-like filters in their first layer rapidly.

4.3.4 Vanilla SGD

In order to take a closer look at feature extraction in the context of deep neural networks, we analyse SGD as an alternative algorithm to perform ICA. Consider a set of n centered and whitened data points $\mathcal{D} = \{x^\mu\}_{\mu=1}^n$. We sample a new data point at each step. For a suitable learning rate $\delta > 0$,

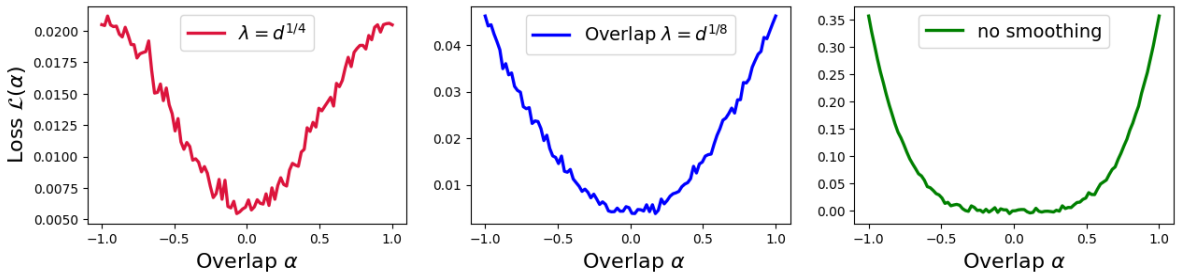

Figure 4.3: **Benefit of smoothing the loss depending on different values of the parameter λ:** We show the smoothed loss $\mathcal{L}_\lambda[G(w \cdot x)]$ for different scalings of the smoothing parameter λ. We can see the loss is most flat around the origin without smoothing, while increasing the smoothing parameter makes the loss more peaked around the origin, making it easier to escape the origin. See section 4.E for details.

each iteration of the spherical online SGD, for $w_0 \sim \mathrm{Unif}(\mathbb{S}^{d-1})$, is defined as

$$\begin{cases} \widetilde{w}_t = w_{t-1} + \frac{\delta}{d} \nabla_{\mathrm{sph}} \mathcal{L}(w, x_t)\big|_{w=w_{t-1}} & t \geq 1, \\ w_t = \widetilde{w}_t / \|\widetilde{w}_t\|, \end{cases}$$

The spherical gradient ∇_{sph} for a function f is given by $\nabla_{\mathrm{sph}} f(w, \cdot) := (\mathbb{1}_d - ww^\top)\nabla_w f(w, \cdot)$. In the case of the noisy ICA model model (4.5), we can directly apply the analysis performed by Ben Arous *et al.* [16], which guarantees that if $n = \Omega(d^3 \log^2 d)$, or more in general $n = \Omega(d^{k^*-1} \log^2 d)$, the spike is "strongly" recovered, meaning that $w_n \cdot v \to 1$ in probability for n going to infinity. Moreover, if $n = O(d^3)$, the spike cannot be recovered, i.e. $\sup_{t \leq n} |v \cdot w_t| \to 0$ in probability.

Hence, vanilla online SGD is already faster that FastICA in the large-batch setting. However, given the information-theoretic results of section 4.2.3, online SGD does not achieve the optimal performance and there remains a statistical-to-computational gap.

4.3.5 Closing the statistical-to-computational gap: smoothing the landscape

We now show that the statistical-to-computational gap of SGD on noisy ICA can be closed by running online spherical SGD on a *smoothed loss* following the approach of Damian *et al.* [18], which was initially introduced in [17]:

Definition 4.5 (Smoothing [18])**.** For $\lambda \geq 0$ and $x \in \mathbb{R}^d$, the smoothed loss function is $L_\lambda(w, x) := \mathcal{L}_\lambda[G(w \cdot x)]$, where the smoothing operator $\mathcal{L}_\lambda$ reads

$$\mathcal{L}_\lambda[G(w \cdot x)] := \mathbb{E}_{z \sim \mu_w} \left[G\left(\frac{w + \lambda z}{\|w + \lambda z\|} \cdot x \right) \right],$$

if μ_w is the uniform distribution over $\mathbb{S}^{d-1}$ conditioned on being orthogonal to w.

The key intuition behind the smoothing operator is that, thanks to large λ (than scales with the input dimension), it allows to evaluate the loss function in regions that are far from the iterate weight w_t, collecting non-local signal that alleviates the flatness of the saddle around $\alpha = 0$ of $\mathcal{L}$, reducing the length of the search phase; we illustrate the effect of smoothing on the loss in fig. 4.3. For implementation purposes, we give an explicit formula for the smoothed gradient of said loss in section 4.E. Note that the "large steps" allowed by a large λ help in escaping the search phase.

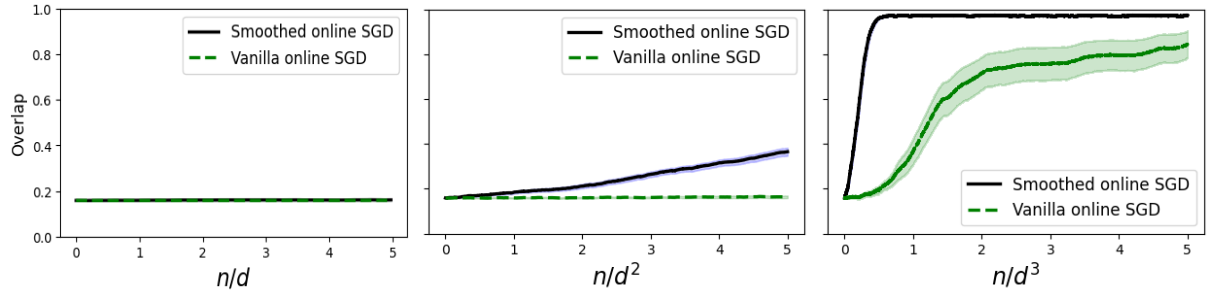

Figure 4.4: **Performances of smoothed online SGD and vanilla SGD on the noisy ICA model, for $G(s) = h_4(s)$ and $G(s) = -\exp(-s^2/2)$ respectively. Left:** In a linear time, nor smoothed and vanilla SGD recover the spike. **Middle:** In the quadratic regime, smoothed SGD recovers, although vanilla SGD does not. **Right:** In the cubic regime, both smoothed and vanilla SGD recover the spike. Experimental details in section 4.A.3.

Damian *et al.* [18] showed that smoothing the loss for a single neuron $\sigma(w \cdot x)$ with activation function $\sigma : \mathbb{R} \to \mathbb{R}$ trained on a supervised task with Gaussian inputs and labels provided by a teacher neuron $\sigma(v \cdot x)$ reduces the sample complexity of weak recovery for online SGD from d^{k^*-1} [16] down to the optimal $d^{k^*/2}$. Their analysis crucially relies on the fact that teacher and student have the same activation function σ. For ICA, one can roughly think of the contrast function as the student activation function, which will in general have a different information exponent from the loss function, which depends also on the likelihood ratio. We therefore generalise the analysis of Damian *et al.* [18] to this mismatched case and show that the dynamics of smoothed SGD on ICA is governed by the interplay of the information exponents of the population loss and the contrast function respectively, called k_1^* and k_2^*, with $k_1^* \geq k_2^*$ by construction and the optimal sample complexity is reached if and only if $k_1^* = k_2^*$. This fact follows immediately from the heuristic analysis performed in section 4.3.5.1, where it is shown that for the optimal $\lambda = d^{1/4}$, the ODE describing the behavior of the overlap α at the beginning of learning is

$$\alpha'(t) = \frac{\alpha(t)}{d^{k_1^* - k_2^*/2}}. \tag{4.9}$$

On the noisy ICA model (4.5), for which $k_1^* = 4$, we thus find the following picture. For the standard contrast functions (eq. (4.3)), $k_1^* \neq k_2^* = 2$, and then smoothed online SGD does not improve the sample complexity of vanilla online SGD: the spike is still recovered in a cubic time. In contrast, if $k_2^* = 4$, i.e. the information exponents of the loss and the contrast function are equal, smoothed online SGD is able to recover the spike in a quadratic time, which means that smoothing the landscape reduces the sample complexity of online SGD down to the optimal threshold suggested by low-degree methods. Even if it would be clearly necessary a fine-tuning of the contrast function in a data-dependent way which cannot be performed in practice, we observe that in our case the fourth-order Hermite polynomial $G(s) = h_4(s)$, which for whitening data is nothing but the excess kurtosis $G(s) = s^4 - 3$, is indeed an optimal contrast functions. Remarkably, the kurtosis is the most classical measure of non-Gaussianity that one can find in the literature (see e.g. Hyvärinen & Oja [71]) and it has been employed to study recent developments of FastICA in [162].

4.3.5.1 Heuristic derivation

We first heuristically derive the sample complexities of Theorem 4.7, following Damian *et al.* [18]. We will use Corollary 4.13 to deal with expectations of multiple products of Hermite polynomials

appearing due to non-Gaussian inputs. A crucial quantity is the signal-to-noise SNR since, as shown in Damian *et al.* [18], at the beginning of leaning the overlap $\alpha := w \cdot v$ is given by the ODE $\alpha'(t) = \mathrm{SNR}/\alpha(t)$ with

$$\mathrm{SNR} := \frac{\mathbb{E}[g \cdot v]^2}{\mathbb{E}[\|g\|^2]},$$

where g is the online spherical gradient $g := \nabla_{\mathrm{sph}} L_\lambda(x, w)$. We compute how the SNR scales with the dimension of the inputs d.

Signal Recall the scaling computed by Damian *et al.* [18] for the signal in the case of an even contrast function:

$$\mathbb{E}[g \cdot v] = \Theta(\alpha\, d^{-(k_1^*-2)/2}/\lambda^2),$$

when $\alpha \le \lambda d^{-1/2}$. Note that it depends on the information exponent k_1^* of the loss.

Noise We look at the scaling of the noise $\mathbb{E}[\|g\|]^2$. By definition, $g = \Theta\Big(\lambda^{-1} x\, \mathcal{L}_\lambda(G'(w \cdot x))\Big)$. Since with high probability $\|x\|^2 = O(d)$, we study the second moment of $s := \mathcal{L}_\lambda(G'(w \cdot x))$. By exploiting the definition of the smoothing operator and writing the expectation in terms of the likelihood ratio, we get that, for two replicas $z, z' \sim \mu_w$,

$$\begin{aligned}\mathbb{E}\left[s^2\right] &= \mathbb{E}\left[\mathop{\mathbb{E}}_{z,z'}\left[G'\left(\frac{w+\lambda z}{\sqrt{1+\lambda^2}}\cdot x\right) G'\left(\frac{w+\lambda z'}{\sqrt{1+\lambda^2}}\cdot x\right)\right]\right] \\ &= \mathop{\mathbb{E}}_{z,z'}\Big(\mathop{\mathbb{E}}_{\mathbb{Q}}\Big[G'\Big(\underbrace{\frac{w+\lambda z}{\sqrt{1+\lambda^2}}}_{w_1}\cdot x\Big) G'\Big(\underbrace{\frac{w+\lambda z'}{\sqrt{1+\lambda^2}}}_{w_2}\cdot x\Big)\ell(v\cdot x)\Big]\Big).\end{aligned}$$

By expanding G' and ℓ, we get a Gaussian expectation of a triple product of Hermite polynomials, whose scaling can be computed thanks to lemma A.6, recalling that G is even and then $c_0^{G'} = 0$. In particular, thanks to Corollary 4.13, we have

$$\mathbb{E}_{\mathbb{Q}}\left[G'(w_1 \cdot x)\, G'(w_2 \cdot x)\, \ell(v \cdot x)\right] = \Theta\Big((w_1 \cdot w_2)^{k_2^*-1}\Big)$$

Since $z \cdot z' = \Theta(1/\sqrt{d})$, we can avoid the dependence on the replicas by simply choosing $\lambda \ll d^{1/4}$. This choice corresponds to eq. (4.17) in Appendix 4.E. Hence, $w_1 \cdot w_2 = \Theta(\lambda^{-2})$. We can conclude that the noise scales as

$$\mathbb{E}[\|g\|^2] = \Theta(d\lambda^{-2k_2^*}).$$

By combining signal and noise, we have that

$$\mathrm{SNR} = \frac{\alpha^2}{\lambda^4} d^{-(k_1^*-1)} \frac{1}{d\lambda^{-2k_2^*}} = \alpha^2 \frac{\lambda^{2(k_2^*-2)}}{d^{k_1^*-1}}.$$

This implies that, in the case of the optimal $\lambda = d^{1/4}$, the ODE describing the behavior of the overlap α for small $t \ge 0$ is

$$\alpha'(t) = \frac{\alpha(t)}{d^{k_1^* - k_2^*/2}},$$

which is eq. (4.9).

Therefore, the number of data points required to reach an order-one overlap is $d^{k^*/2}$, i.e. the optimal one according to the fundamental limits of recovery, if and only if $k_1^* = k_2^*$ and than we have heuristically derived the results of theorem 4.7. The following theorem, which we prove in section 4.E, makes rigorous the heuristics. It holds for any input model satisfying assumption 4.1, assumption 4.2, and the following assumption 4.6:

Assumption 4.6. We assume $\ell \in \mathrm{L}^\infty(\mathbb{R}^d, \mathbb{Q})$ and that the loss (4.2) is strictly monotonically increasing in $\alpha = w \cdot v$.

Theorem 4.7 (Escaping mediocrity)**.** *Define k_1^* and k_2^* as the information exponents of the population loss* (4.2) *and the contrast function, respectively. Assume that $w_0 \sim Unif(\mathbb{S}^{d-1})$ and that the initial overlap $\alpha_0 := w_0 \cdot v$ is such that $\alpha_0 \gtrsim 1/\sqrt{d}$. Consider $\lambda \in [1, d^{1/4}]$. Then, there exist $\eta \in \mathbb{R}$ and $n \in \mathbb{N}$, satisfying*

$$\begin{cases} n = O\left(d^{k_1^*-1}\lambda^{-2(k_2^*-2)}\ polylog(d)\right), \\ \eta = O\left(d^{-k_1^*/2}\lambda^{2(k_2^*-1)}\ polylog(d)\right) \end{cases}$$

such that $\alpha_n \geq 1 - d^{-1/4}$ with high probability. Here, $\alpha_n := w_n \cdot v$ and w_n is the estimator given by smoothed online SGD with $\eta_n = \eta$ and $\lambda_n = \lambda$.

In the following corollary, we derive rigorously the scalings required to recover the non-Gaussian component in the case of the noisy ICA model (4.5). More precisely, up to logarithmic terms we get the optimal sample complexity $n \approx d^2$ when the contrast function is the excess kurtosis ($k_2^* = 4$), and $n \approx d^3$ when the contrast function is one of the standard ones in eq. (4.3) ($k_1^* = 2$). We have therefore proven the scalings obtained in eq. (4.9). The proof of Corollary 4.8 follows immediately from Theorem 4.7 by recalling that for the noisy ICA model (4.5) we get $k_1^* = 4$.

Corollary 4.8 (Escaping mediocrity for "noisy ICA" inputs)**.** *Suppose that the assumptions of Theorem 4.7 hold and that the inputs are distributed according to the noisy ICA model* (4.5)*. Consider $\lambda = d^{1/4}$. If $k_2^* = 4$, smoothed online SGD recovers the spike - in the sense of Theorem 4.7 - within a number of samples of $n = O(d^2 polylog(d))$. On the other hand, given $k_2^* = 2$, the obtained sample complexity is $n = O(d^3 polylog(d))$.*

Proof. Due to online learning, the number of SGD steps equals the number of drawn samples. Then, because of Theorem 4.7 with $\lambda = d^{1/4}$, there exists $n \in \mathbb{N}$ such that $\alpha_n \geq 1 - d^{1/4}$ with high probability and

$$n = O\left(d^{k_1^* - k_2^*/2}\ \mathrm{polylog}(d)\right).$$

Since for the spiked model we have $k_1^* = 4$ and, depending on the value of k_2^*, the thesis follows. □

4.4 The search phase of FastICA on images

We investigate whether FastICA also exhibits an extended search phase on real data. To that end, we ran FastICA on D-dimensional patches sampled at random from the ImageNet data set [151].

In fig. 4.5(b), we show change in the loss $\mathcal{L}(w)$, averaged over a held-out test set, compared to the value of the test loss at initialisation $\mathcal{L}(w_0)$, averaged over different random initial conditions. We find that at linear sample complexity when $n = 2D$ (fig. 4.5(b)), FastICA is clearly stuck in a search phase: the loss does not deviate significantly from its value at initialisation. We can see that not only are the changes in the loss small; as we increase the width of the patches, and hence their dimension D, the curves tend towards zero, suggesting that FastICA does not recover a non-Gaussian direction at linear sample complexity on ImageNet in the asymptotic limit of large inputs $D \to \infty$, in agreement with the previous sections.

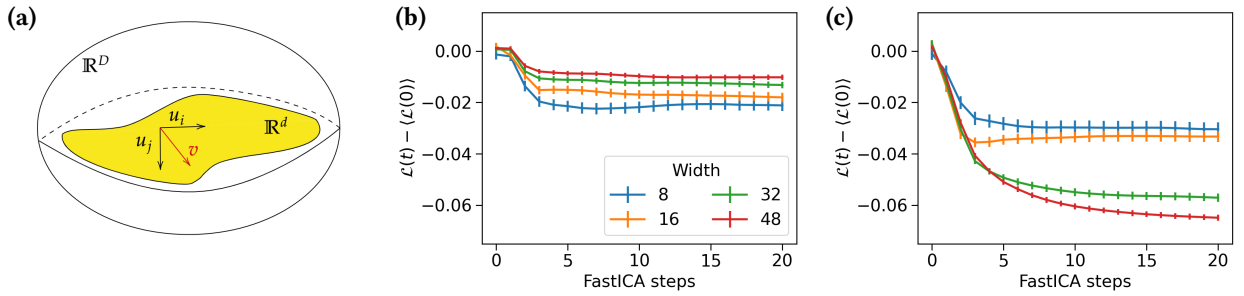

Figure 4.5: **Running ICA in the principal subspace of small dimension $d \sim O_D(1)$ yields non-Gaussian directions that generalise in large dimension D (a)** FastICA can be run in the ambient dimension D of the inputs or on the d-dimensional subspace spanned by the principal components u_i. **(b)** Change in loss with respect to initialisation when running FastICA on ImageNet patches at linear sample complexity $n = 2D$. As we increase the patch width, and hence D, the change in the loss tends towards zero. **(c)** Running FastICA in the subspace spanned by the leading $d = 32$ principal components yields a significantly non-Gaussian direction, as shown by the decay of the loss evaluated in D dimensions. Error bars indicate the error in the mean over runs. Full details in section 4.A.4.

4.4.1 Running FastICA in the principal subspace of ImageNet patches consistently yields non-Gaussian directions

We verify experimentally whether running FastICA in a subspace of *fixed* input dimension $d \sim O_D(1)$ yields a non-Gaussian direction in the ambient D-dimensional space when $D \to \infty$, see fig. 4.5(c).

Recall that FastICA exhibits a search phase at linear sample complexity, as shown in fig. 4.5(b). The picture changes fundamentally if we run FastICA in the subspace spanned by the leading $d = 32$ principal components, keeping the dimension of the subspace *fixed* as we increase the size of the inputs, see fig. 4.5(c). We now find that the FastICA consistently finds a direction that, after projecting back up into the D-dimensional space, yields non-Gaussian projections $s = w \cdot x$.

This experiment therefore suggests that running FastICA in a subspace of *constant*, rather than just smaller dimensions, yields non-Gaussian directions in image space even at linear sample complexity, circumventing the fundamental computational limitations of FastICA.

4.5 FastICA provably recovers a non-Gaussian direction at linear sample complexity on the "subspace model"

We now seek to confirm the picture that emerged from our experiments in section 4.4.1 by proving recovery of a non-Gaussian direction with linear sample complexity by FastICA. In the standard input model for (noisy) ICA, the computational hardness of the task in D dimensions requires $n \gtrsim D^2$ which cannot be reduced by projecting down to lower dimensions [30, 162]. The standard ICA data model therefore cannot account for the experimental results on ImageNet in fig. 4.5.

Definition of the subspace model Here, we therefore introduce the "subspace model", a new data model that accounts for the success of PCA + ICA on ImageNet. The key idea is to spike the higher-order correlations of the inputs with a spike v that is a linear combination of the leading principal components of the inputs, which we denote by $u_r \in \mathbb{S}^{D-1}$. We assume $u_s \perp u_r$ for $s \neq r$

and consider n inputs in D dimensions $x^\mu = (x_i^\mu)$, $\mu = 1, \dots, n$, that are sampled according to

$$x^\mu = \sum_{\rho=1}^{r} \sqrt{\beta_2^\rho} g_\rho^\mu u^\rho + h^\mu v + z^\mu, \qquad v = \sum_{\rho}^{r} \alpha_\rho u^\rho, \tag{4.10}$$

where the Gaussian latent variables $g_\rho^\mu \sim_{\text{i.i.d.}} \mathcal{N}(0,1)$, while the latent variable h^μ is drawn from a non-Gaussian distribution; for example, we could take $h^\mu = \pm 1$ to be Rademacher. All latent variables g_ρ^μ, h^μ are independent of each other. The noise z^μ is drawn from a multivariate normal distribution with mean 0 and covariance $\mathbb{1} - vv^\top$, which ensures that the non-Gaussian direction v can *not* be recovered using principal component analysis (PCA), just like Gabor filters cannot be found by PCA. The coefficients α_ρ are normalised as $\sum_\rho^r \alpha_\rho^2 = 1$. To prove that FastICA can recover the non-Gaussian direction, we proceed in two steps.

Step 1: projecting down We first show that when projecting inputs from the subspace model (4.10) onto the lower dimensional subspace spanned by the top-r leading principal components, we obtain a "reduced" input model that follows the ICA model. More precisely, we find that the reduced inputs $y^\mu = (y_i^\mu)$, $\mu = 1, \dots, n$, follow a noisy ICA model [33] whose "non-Gaussian" spike is simply, up to normalisation, the projection of the original spike v:

$$y^\mu = \sqrt{\gamma}\, h^\mu \widetilde{v} + \hat{z}^\mu \in \mathbb{R}^r, \tag{4.11}$$

where $\widetilde{v} = \text{proj}(Sv)/\|\text{proj}(Sv)\|$, for suitable γ and $\hat{z}^\mu$. This is proved in theorem 4.14. In the proof, we assume that PCA yielded the exact principal subspace $\{u^r\}$. However, in practice, as known from standard results for PCA in high dimensions [168], PCA will return the subspace up to some small error. However, this only reduces the signal-to-noise ratio of the noisy ICA model. More precisely, we prove in Lemma 4.15 that, if n is such that $n, D \to \infty$ and n/D that diverges arbitrarily slowly, with high probability we still recover the reduced inputs (4.11), up to vanishing error terms.

Step 2: Gradient flow We now analyse the performance of ICA on inputs drawn from the reduced model (4.11), at a linear sample complexity. Theorem 4.16, informally stated below, proves that gradient descent (GD) on the ICA loss $\mathcal{L}(w)$ converges to the non-Gaussian spike of the reduced model.

Lemma 4.9. *(Informal) Consider the low dimensional inputs drawn from the reduced model* (4.11)*. Since the population loss for ICA $\mathcal{L}(w)$ exhibits monotonicity and the dynamics of GD can be approximated via gradient flow, with high probability GD converges to the non-Gaussian spike $\widetilde{v}$ in the limit $n \to \infty$.*

4.6 Concluding perspectives

Despite their simplicity, Independent Component Analysis yields similar filters as the early layers of deep convolutional neural networks, and therefore offers an interesting model to study feature learning from non-Gaussian inputs. Here, we considered the problem of recovering a single feature encoded in the non-Gaussian input fluctuations and established almost sharp sample complexity thresholds for FastICA and SGD on a simple synthetic model. Going forward, it will be intriguing to extend our analysis to models with several non-Gaussian directions, along the lines of Dandi *et al.* [165] and the models presented in chapter 3. Moreover, developing a model for synthetic data from which localised, oriented filters can be learned looms as an intriguing challenge [45, 169].

Regarding the subspace model, have shown that ICA can provably recover a non-Gaussian direction v in linear sample complexity if:

i. v lives in the span of the principal components of the data,

ii. we perform PCA as a pre-processing step.

Our experimental observation reflects previous work that showed that neural networks learn distributions of increasing complexity, focusing on pair-wise correlations first before going to higher-order correlations, both on images [28, 31, 45, 170], in natural language processing [126, 171] and on the mixture classification from chapter 3.

Our results also echo earlier work showing that gradient descent occurs in a tiny subspace of input space [172, 173] and that the Hessian of over-parametrised neural networks has a low-rank structure [174–176]. How these two perspectives – on the geometry of the loss landscape and on the data structure – relate to each other is an intriguing avenue for further work.

Appendix

4.A Experimental details

In this appendix, we collect detailed information on how we ran the various experiments of this paper.

4.A.1 Figure 4.1

(a) We plot the filters of the first convolutional layer of an AlexNet trained on ImageNet using mini-batch size of 128, momentum of 0.9, weight decay of 10^{-4}, cosine learning rate schedule with initial learning rate of 10^{-2}. We also trained a DenseNet121 [149] and a ResNet18 [150] using the same recipe, but with larger initial learning rate 10^{-1}.

(b) The Gabor filters were obtained with online SGD performed on 100 000 patches of size 64×64 extracted from ImageNet. We have decided to perform ICA with SGD to use the typical algorithm employed to train deep neural networks. We perform PCA before running ICA by keeping 600 principal components. The contrast function was $G(s) = -\exp(-s^2/2)$.

(c) We applied principal component analysis to the same ImageNet patches to obtain the leading principal components, and plot the eight leading ones from left to right, and top to bottom.

(d) For all three networks trained on ImageNet, we took 50 logarithmically spaced snapshots of the weights during training. For each snapshot, we extracted the weights for each first-layer convolution and computed the dot-products $s_\mu^k = w^k \cdot x^\mu$ between the weight of the kth convolution for randomly sampled patches from ImageNet x^μ. We normalised the s_μ^k for each neuron (fixed k) to have variance 1, and computed the excess kurtosis of the resulting s_μ^k. We then averaged over all the neurons to obtain the curves shown in fig. 4.1(b).

(e) The snapshots were taken from one of the AlexNet models.

4.A.2 Figure 4.2

Here we show the performance of FastICA on the spike cumulant model (4.5) in three different regimes, in the case of $d = 50$ and $\beta = 15$. The spike v has been sampled uniformly in the sphere $\mathbb{S}^{d-1}$. The regimes correspond to the batch-sizes of $n = d^2, n = d^{3+\delta}$ and $n = d^4$, respectively, for $\delta = 0.2$. Each batch contains n fresh data points sampled from the distribution of the inputs $\mathbb{P}$. The contrast function is $G(s) = -\exp(-s^2/2)$. We display on the y-axis the overlap $w_t \cdot v$ for $t = 0, \dots, 4$ (on the x-axis), where w_t is the updated weight vector provided by FastICA in the first four step, plus initialisation $w_0 \sim \mathbb{S}^{d-1}$ under the constraint that $w_0 \cdot v = 1/\sqrt{d}$. An average over 15 runs has been taken.

4.A.3 Figure 4.4

Here we show the velocity for recovering the spike of smoothed online SGD in three different regimes: linear, quadratic and cubic. The noisy ICA model is considered. We use $d = 40$, $\beta = 15$ and a total number of data points of $n = 5d^3$. The parameter λ is the optimal one, i.e. $\lambda = d^{1/4}$. The learning rate is $\eta = 2.9d^{(-k_1^*/2)}\lambda^{(2k_2^*-2)}$, where $k_1^* = 4$ and $k_2^* = 2$, since $G(s) = -\exp(-s^2/2)$. The average is made over 30 runs.

4.A.4 Figure 4.5

(a) We run FastICA on a set of $n = 2D$ patches randomly sampled from ImageNet of width w, with $D = w^2$. We use the logcosh activation function, and use the full dataset to compute the weight update at each step. We preprocess image patches by centering them and whitening them. We compute the ICA loss averaged over a held-out test set of 10 000 patches, and subtract the loss computed for a random direction over the same test set, averaged over 20 random directions.

(b) Same as in (a), except that after whitening, we only keep the leading $d = 32$ principal components of each image and run FastICA in this subspace. We plot the loss in the original space of images of dimension d by projecting the ICA weight vector back into the d-dimensional space using the eigenvectors of the patch-patch covariance matrix.

In these experiments we reused the same batch of inputs at every iteration of FastICA. It is known that this actually simplifies the task of recovering single directions in certain cases [21, 22]; however, in the case of FastICA the task remains hard: the FastICA loss with standard activation functions is only sensitive to even cumulants, and due to whitening, the first non-trivial cumulant is the fourth, yielding a computational lower bound for recovering a non-Gaussian direction of $n \asymp D^2$ [30].

4.A.5 The "3not4" prior on the latent variable, fig. 4.8

For this experiment, we run online FastICA as in fig. 4.2 on a noisy spiked cumulant model, but with a prior over the latent variable that has mean zero, unit variance, a non-trivial third-order cumulant, but zero fourth-order cumulant. More specifically, we consider a latent ν that can take the values $-1, 0$ and 2 with probabilities $1/3, 1/2$ and $1/6$.

4.B Properties and definitions

4.B.1 Probability notions

We recall the properties of the space of squared integrable functions with respect to the standard normal distribution $\mathbb{Q}$:

Definition 4.10 (Gaussian scalar product)**.** Given $f, g : \mathbb{R}^d \to \mathbb{R}$, we define the L^2-scalar product as

$$\langle f, g \rangle_{\mathbb{Q}} := \mathbb{E}_{x\sim\mathbb{Q}} \left[f(x)g(x) \right].$$

The space $\mathrm{L}^2(\mathbb{R}^d, \mathbb{Q})$ contains all the measurable functions $f : \mathbb{R}^d \to \mathbb{R}$ such that $||f||^2 := \langle f, f \rangle_{\mathbb{Q}} < \infty$.

Note that this space with the product defined above is an Hilbert space, and the set of the Hermite polynomials (see appendix A) is an orthogonal basis.

Definition 4.11 (Sub-Gaussianity)**.** Let Z be a real valued random variable, we say that Z is sub-Gaussian if it has finite sub-Gaussian norm, defined as:

$$||Z||_{\psi/2} := \inf \left\{ t > 0 \mid \mathbb{E}[\exp\left(Z^2/t^2\right)] \leq 2 \right\}$$

For more details on sub-Gaussian distributions we refer to [99], Section 2.5.

We also need to introduce the space $\mathrm{L}^\infty(\mathbb{R}^d, \mathbb{Q})$ as the space of essentially bounded functions, with finite $||\cdot||_\infty$ norm, defined as:

$$||f||_\infty := \inf \{ C > 0 \mid \mathbb{Q}(|f(x)| \leq C) = 1 \}$$

Note that the likelihood ratio $\ell = \frac{d\mathbb{P}}{d\mathbb{Q}}$ for model (4.5) belongs to this space as soon as the tails of ν decrease faster than a standard Gaussian. For instance all the distributions with bounded support are included.

We often refer to the notion of events that happen *with high probability*, defined as follows.

Definition 4.12 (Events happening with high probability events)**.** The sequence of events $\{E_d\}_{d\in\mathbb{N}}$ happens with *high probability* if for every $k \geq 0$, there exists d_k such that for $d \geq d_k$

$$1 - P(E_d) \leq \frac{1}{d^k}$$

4.C Measures of non-Gaussianity for ICA

Whether the population loss has to be (globally) maximized or minimized with respect to the weight vector $w \in \mathbb{S}^{d-1}$ depends on the data distribution and the choice of the contrast function G. Therefore, we briefly comment on this, by referring to [71] and [177] for more details. What does maximising *non-Gaussianity* mean? Equivalently: what is a suitable measure for non-Gaussianity? A standard approach is to choose the *negentropy* as a measure of the non-Gaussianity of the random vector $x \sim \mathbb{P}$, i.e.

$$J(x) := H(x_{Gauss}) - H(x),$$

where $H(x) := -\mathbb{E}_{\mathbb{P}}[\log(x)]$ and $x_{Gauss} \sim \mathbb{Q}$. We assume that x has zero mean and that its covariance matrix is the identity. The negentropy, being a Kullback-Leibler divergence, is always non-negative. However, to compute J one needs to access the density of the random vector x, which is usually unknown and not easy to approximate. Hence, the classical approach is to approximate negentropy indirectly. In Section 5 of [177], the authors show that the negentropy can be approximated by

$$\tilde{\mathcal{L}}(w) = \left(\mathbb{E}\left[G(w \cdot x)\right] - \mathbb{E}\left[G(x_{Gauss})\right]\right)^2$$

where G is even and does not grow too fast because of reasons of numerical stability. Typically, G is one of the contrast functions from Equation (4.3). Note that this optimization problem is slightly different from Equation (4.1) (which is the setting considered in practice, as mentioned in [71]). The reason why it is convenient to remove the square is clear as soon as the Hermite expansion

eq. (4.14) is performed, together with the fact $\mathbb{E}\left[G(x_{Gauss})\right] = c_0^G c_0^\ell$:

$$\begin{aligned}\tilde{\mathcal{L}}(w) &= \left(\mathcal{L}(w) - \mathbb{E}\left[G(x_{Gauss})\right]\right)^2 \\ &= \left(c_0^G c_0^\ell + \frac{c_{k^*}^G c_{k^*}^\ell}{k!}\alpha^{k^*} + o(\alpha^{k^*}) - c_0^G c_0^\ell\right)^2 \\ &= \left(\frac{c_{k^*}^G c_{k^*}^\ell}{k!}\right)^2 \alpha^{2k^*} + o(\alpha^{2k^*})\end{aligned}$$

hence the information exponent of this loss function is doubled with respect to $\mathcal{L}$ worsening the sample complexity for the search phase (geometrically, this can be seen as the square "flattening" the loss around the origin).

The removal of the square comes with the side effect that it is not sufficient to just minimize the loss, since the sign becomes relevant: G needs to me maximised in case of platykurtic data since the cumulants of the data are smaller than the gaussian ones, so $\mathbb{E}[G(w \cdot x)] - \mathbb{E}[G(x_{gauss})] < 0$, and conversely needs to be minimized in case of leptokurtic data distributions.

4.C.1 Details on the noisy ICA model model

Here we expand the population loss (4.2) in the specific case where the inputs are distributed according to the spike cumulant model (4.5). We can expand the likelihood ratio ℓ in series of Hermite polynomials in the sense of Definition A.2, and write

$$\ell(v \cdot x) = \sum_{j=0}^{\infty} \frac{c_j^\ell}{j!} h_j(v \cdot x).$$

Note that due to the symmetry of ν, $c_{2j+1}^\ell = 0$ for any $j \in \mathbb{N}$. Moreover, due to the whitening matrix, also $c_2^\ell = 0$; so, apart from $c_0^\ell = 1$, the smallest non-zero coefficient is c_4^ℓ, the one corresponding to first non-trivial cumulant, i.e. the kurtosis. Using the orthogonality identity for Hermite polynomials (Lemma A.5), we get

$$\begin{aligned}\mathcal{L}(w) &= \mathbb{E}_{\mathbb{Q}}[G(w \cdot x)\, \ell(v \cdot x)] \\ &= \sum_{i,j \geq 0}^{\infty} \frac{c_i^G c_j^\ell}{i!j!}\, \mathbb{E}_{\mathbb{Q}}[h_i(w \cdot x) h_j(v \cdot x)] \\ &= c_0^G c_0^\ell + \frac{c_4^G c_4^\ell}{4!}\alpha^4 + o(\alpha^4)\end{aligned}$$

where $\alpha = w \cdot v$ is the only order parameter of the system. For this calculation, we have assumed that $c_4^G \neq 0$, which holds for the contrast functions in Equation (4.3). This implies that the inference problem of detecting the spike v has information exponent $k^* = 4$, in the sense of Definition A.2 in Appendix A

Note also that, for this specific data distribution, ℓ can be computed explicitly, that is

$$\ell(y) = \mathbb{E}_\nu \left[\sqrt{1+\beta}\exp\left(-\frac{1+\beta}{2}\left(\nu - \frac{\beta}{1+\beta}v\right)^2 + \frac{1}{2}\right)\right]. \tag{4.12}$$

For more details regarding this computation, we refer to chapter 2.

4.D FastICA

In fig. 4.6 we show numerically that the large fluctuations seen in fig. 4.2 (middle) hold for various values of ϵ. We run FastICA on the noisy ICA model with size batch $d^{3+\epsilon}$, for $\epsilon = 0.2, 0.5, 0.8$. We plot $\alpha_v = v \cdot w$ in the first four iterations, plus the overlap at initialisation.

We prove now Proposition 4.3, which guarantees that the second order term in the iteration of FastICA does not only speed up the convergence of the algorithm, but helps in escaping the search phase too: indeed, when it is not present, the recovered signal after the first giant step scales as the random weight initialisation, i.e. $\alpha_1^2 = (v \cdot w_1)^2 = O(1/d)$. However, when the iteration of FastICA is complete, the relevant direction is learned in one single step, if the size-batch is infinite and the dimension of the inputs diverge.

Proof of Proposition 4.3 (Infinite batch size). We start by recalling that, up to normalisation, the first iteration of FastICA in the population limit, i.e. for a batch of infinite size, reads

$$\widetilde{w}_1 = \mathbb{E}[x\, G'(w_0 \cdot x)] - \mathbb{E}[G''(w_0 \cdot x)]\, w_0,$$

where the weight vector at initialisation w_0 is uniformly drawn in the unit sphere in $\mathbb{R}^d$. Here, we have considered the complete iteration, with both the first order and the second order terms. Since we assume we are in a high dimensional regime, we have that the overlap between the initialisation and the spike that we want to recover scales as $\alpha_0 := w_0 \cdot v = \Theta(1/\sqrt{d})$. We want to prove that, after the first step, the spike has been totally learned when $d \to +\infty$, namely

$$\alpha_1^2 := (w_1 \cdot v)^2 = 1 - o(1).$$

To do so, we can expand in the series of Hermite polynomials the squared overlap

$$\alpha_1^2 = \frac{\pi_1^2}{\|\widetilde{w}_1\|^2}, \text{ with } \pi_1 := \widetilde{w}_1 \cdot v \tag{4.13}$$

at a sufficiently high order. First of all, we take into account the first and the second order terms $f := x\, G'(w_0 \cdot x)$ and $g := G''(w_0 \cdot x) w_0$ and compute their expectations with respect to the data distribution.
We compute $\mathbb{E}[f]$ by noticing that $f \in \mathbb{R}^d$ can be written as the sum of a spherical gradient and a derivative along the direction of the initialisation, in the following sense: if $L(w, x) := G(w \cdot x)$, we have that

$$f = (\mathbb{1}_d - w_0 w_0^\top)\nabla_w L(w,x)\Big|_{w=w_0} + w_0 w_0^\top \nabla_w L(w,x)\Big|_{w=w_0} = \underbrace{\nabla_{\text{sph}} L(w,x)\Big|_{w=w_0}}_{(\star)_1} + \underbrace{w_0 w_0^\top \nabla_w L(w,x)\Big|_{w=w_0}}_{(\star)_2}.$$

This splitting has to be done since we need unit vectors to be allowed to apply the orthogonality formulas in appendix A. By using the likelihood ratio trick, we get

$$\begin{aligned}
\mathbb{E}[(\star)_1] =& \mathbb{E}\left[\frac{\partial}{\partial w} G(w \cdot x)\Big|_{w=w_0}\right] = \mathbb{E}_{x\sim\mathcal{N}(0,\mathbb{1}_d)}\left[\frac{\partial}{\partial w} G(w \cdot x)\Big|_{w=w_0} \ell(v \cdot x)\right] \\
=& \frac{\partial}{\partial w} \mathbb{E}_{x\sim\mathcal{N}(0,\mathbb{1}_d)}\left[\left(\sum_{i=0}^{\infty} \frac{c_i^G}{i!} h_i(w \cdot x)\right)\left(\sum_{j=0}^{\infty} \frac{c_j^\ell}{j!} h_j(w \cdot x)\right)\Big|_{w=w_0}\right] \\
=& \frac{\partial}{\partial w} \sum_{i,j=0}^{\infty} \frac{c_i^G c_j^\ell}{i!j!} \mathbb{E}_{x\sim\mathcal{N}(0,\mathbb{1}_d)}[h_i(w \cdot x) h_j(w \cdot v)]\Big|_{w=w_0} \\
=& \frac{\partial}{\partial w} \sum_{k=0}^{\infty} \frac{c_k^G c_k^\ell}{k!} \alpha_0^k\Big|_{w=w_0} = v_{\perp w_0}\left(\sum_{k=1}^{\infty} \frac{c_k^G c_k^\ell}{(k-1)!} \alpha_0^{k-1}\right) = v_{\perp w_0}\left(\sum_{k=k^*}^{\infty} \frac{c_k^G c_k^\ell}{(k-1)!} \alpha_0^{k-1}\right)
\end{aligned}$$

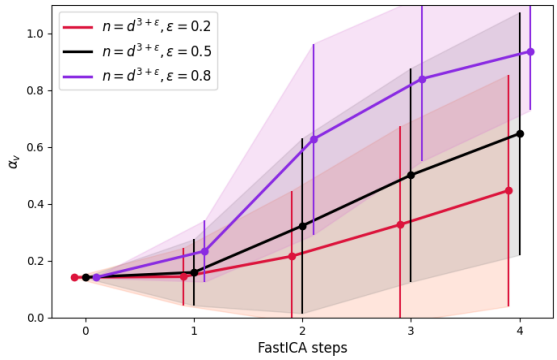

Figure 4.6: **Performance of FastICA on the spiked cumulant model for various sample complexity regimes.** We run the standard FastICA updates (4.4) on data drawn from a noisy ICA model (4.5). At each step of the algorithm, we draw a new batch of size $n = d^{3+\varepsilon}$ with $\varepsilon = 0.2, 0.5, 0.8$. We plot the overlap between the planted non-Gaussian direction v and the estimate w produced by FastICA. Theorem 4.4 confirms this picture for a general data model.

where $v_{\perp w_0} := v - \alpha_0 w_0$ is the projection of the spike in the space orthogonal to w_0. Moreover, recall that k^* is the information exponent of the population loss (4.2), in the sense of Definition A.3. In the case of the spiked model defined in Section 4.2.2, we already know that $k^* = 4$. We keep the terms up to the sixth-order of expansion. Hence,

$$\mathbb{E}\left[(\star)_1\right] = v_{\perp w_0}\left(\frac{c_4^G c_4^\ell}{3!}\alpha_0^3 + \frac{c_6^G c_6^\ell}{5!}\alpha_0^5\right) + o(\alpha_0^6).$$

On the other hand, also have to take into account the component of the gradient in the direction of the initialization. For this second part, we get

$$\begin{aligned}\mathbb{E}[(\star)_2] =& w_0\,\mathbb{E}\left[\frac{d}{d\lambda}L(\lambda w_0, x)\Big|_{\lambda=1}\right] = w_0\,\mathbb{E}\left[G'(w_0\cdot x)(w_0\cdot x)\right]\\ =& w_0\,\mathbb{E}\left[G'(w_0\cdot x)h_1(w_0\cdot x)\ell(v\cdot x)\right]\\ =& w_0\sum_{i,j=0}^{\infty}\frac{c_i^{G'}c_j^\ell}{i!j!}\mathbb{E}_{x\sim\mathcal{N}(0,\mathbb{1}_d)}\left[h_i(w_0\cdot x)h_1(w_0\cdot x)h_j(v\cdot x)\right].\end{aligned}$$

By using Corollary A.7 we can expand up to six order, obtaining

$$\mathbb{E}[(\star)_2] = w_0\left(c_2^G c_0^\ell + \frac{c_4^G c_0^\ell}{3!}\alpha_0^4 + \frac{c_6^G c_4^\ell}{4!}\alpha_0^4 + \frac{c_6^G c_6^\ell}{5!}\alpha_0^6 + \frac{c_8^G c_6^\ell}{6!}\alpha_0^6\right) + o(\alpha_0^6).$$

In conclusion, summing up the two contributions we simply have

$$\mathbb{E}[f] = v_{\perp w_0}\left(\frac{c_4^G c_4^\ell}{3!}\alpha_0^3 + \frac{c_6^G c_6^\ell}{5!}\alpha_0^5\right) + w_0\left(c_2^G c_0^\ell + \frac{c_4^G c_0^\ell}{3!}\alpha_0^4 + \frac{c_6^G c_4^\ell}{4!}\alpha_0^4 + \frac{c_6^G c_6^\ell}{5!}\alpha_0^6 + \frac{c_8^G c_6^\ell}{6!}\alpha_0^6\right) + o(\alpha_0^6).$$

Therefore, it remains to compute $\mathbb{E}[g]$. To do so, it is sufficient to expand G'' in series of Hermite polynomials to obtain

$$\mathbb{E}[g] = w_0\sum_{k=0}^{\infty}\frac{c_k^{G''}c_k^\ell}{k!}\alpha_0^k = w_0\left(c_2^G c_0^\ell + \frac{c_6^G c_4^\ell}{4!}\alpha_0^4 + c_2^G c_0^\ell + \frac{c_8^G c_6^\ell}{6!}\alpha_0^6\right) + o(\alpha_0^6).$$

Now that we have expanded both $\mathbb{E}[f]$ and $\mathbb{E}[g]$ up to sixth order, after a long but simple calculation, we can conclude by writing

$$\pi_1^2 = \|\widetilde{w}_1\|^2 = \frac{(c_4^G c_4^\ell)^2}{3!^2}\alpha_0^6 + o(\alpha_0^6) \quad \Longrightarrow \quad \alpha_1^2 = \frac{\pi_1^2}{||\widetilde{w}_1||^2} = 1 - o(1).$$

This computation has shown that if the first step could possibly be infinitely large, then FastICA would perfectly learn the spike up to contributions which vanish when $d \to \infty$.

We observe now that happens when the iteration of FastICA consists only in the gradient term. The different update would be given by $\widetilde{w}_1 = \mathbb{E}[\, x\, G'(w_0 \cdot x)]$. In this case, it is not needed to expand up to sixth-order, since we can easily see that α_1^2 scales as bad as the random initialisation. In particular,

$$\pi_1^2 = (c_2^G c_0^\ell)^2 \alpha_0^2 + o(\alpha_0^2),\ \|\widetilde{w}_1\|^2 = (c_2^G c_0^\ell)^2 + o(1) \quad \Longrightarrow \quad \alpha_1^2 = \alpha_0^2 + o(\alpha_0^2) = O\Big(\frac{1}{d}\Big).$$

□

We are now going to prove Theorem 4.4, which makes explicit the sample complexity required to learn the spike in a single step with a large, although finite, amount of data points. Recall that, given n samples, the FastICA iteration reads

$$\widetilde{w}_t = \frac{1}{n}\sum_{\nu=1}^n x^\nu\, G'(w_{t-1}\cdot x^\nu) - \frac{1}{n}\sum_{\nu=1}^n G''(w_{t-1}\cdot x^\nu) w_{t-1}.$$

Proof of Theorem 4.4 (Finite batch-size). Define $f := \frac{1}{n}\sum_{\nu=1}^n x^\nu\, G'(w_0 \cdot x^\nu)$ and $g := \frac{1}{n}\sum_{\nu=1}^n x^\nu\, G''(w_0 \cdot x^\nu) w_0$. The key ideas of this strategy are borrowed from the proof of Theorem 1 and 2 in Dandi *et al.* [165], where they consider a teacher-student setup and Gaussian inputs. First of all, we compute the expectations of $\pi_1 = \widetilde{w}_1 \cdot v$ and $\|\widetilde{w}_1\|^2$. After that, we look at how much data we need to obtain concentration for both the quantities, namely what is the scaling for n such that π_1 and $\|\widetilde{w}_1\|^2$ are approximated, up to a sufficiently small error, by their expectations. Then, we get the scaling for α_1^2 by simply computing the ratio between $\mathbb{E}[\pi_1]^2$ and $\mathbb{E}[\|\widetilde{w}_1\|^2]$.

From now on, we consider $n = \Theta(d^{k^*-\delta})$, for $\delta \in [0, 2]$. In order to compute the expectation of π_1, we can use the calculation performed in the proof of Proposition 4.3 to obtain

$$\begin{aligned}\mathbb{E}[\pi_1] = v \cdot \mathbb{E}[f - g] &= v \cdot \frac{1}{n}\sum_{\nu=1}^n \left(\mathbb{E}[x^\nu G'(w_0 \cdot x^\nu)] - \mathbb{E}[G''(w_0 \cdot x^\nu) w_0]\right) \\ &= \frac{c_{k^*}^G c_{k^*}^\ell}{(k^*-1)!}\alpha_0^{k^*-1} + o(\alpha_0^{k^*-1}),\end{aligned}$$

where the last inequality is obtained by following the steps of Proposition 4.3 without explicitly using that $k^* = 4$. Indeed,

$$v \cdot \mathbb{E}[f] = c_0^\ell c_2^G \alpha_0 + \frac{c_{k^*}^G c_{k^*}^\ell}{(k^*-1)!}\alpha_0^{k^*-1} + o(\alpha_0^{k^*-1}) \quad \text{and} \quad v \cdot \mathbb{E}[g] = c_0^\ell c_2^G \alpha_0 + o(\alpha_0^{k^*-1}).$$

We now have to compute $\mathbb{E}[\|w_1\|^2] = \mathbb{E}[\|f\|^2] + \mathbb{E}[\|g\|^2] - 2\,\mathbb{E}[f \cdot g]$. Since

$$\begin{aligned}\mathbb{E}[\|g\|^2] =& \frac{1}{n^2}\mathbb{E}\Big[\sum_{\nu,\nu'=1}^n (x^\nu \cdot x^{\nu'})\, G'(w_0 \cdot x^\nu)\, G'(w_0 \cdot x^{\nu'})\Big] \\ =& \frac{1}{n^2}\mathbb{E}\Big[\sum_{\nu\neq\nu'}^n (x^\nu \cdot x^{\nu'})\, G'(w_0 \cdot x^\nu)\, G'(w_0 \cdot x^{\nu'})\Big] + \frac{1}{n^2}\mathbb{E}\Big[\sum_{\nu=1}^n \|x^\nu\|^2\, G'(w_0 \cdot x^\nu)^2\Big] \\ =& \frac{n(n-1)}{n^2}\|\mathbb{E}[g]\|^2 + \frac{1}{n}\mathbb{E}\Big[\|x\|^2\, G'(w_0 \cdot x)^2\Big]\end{aligned}$$

and it is possible to do the same calculation for $\mathbb{E}[\|f\|^2]$ and $\mathbb{E}[f \cdot g]$, we get that

$$\begin{aligned}\mathbb{E}[\|\widetilde{w}_1\|^2] = \frac{n(n-1)}{n^2}\Big(\|\mathbb{E}[g]\|^2 + \|\mathbb{E}[f]\|^2 - 2\,\mathbb{E}[g]\cdot\mathbb{E}[f]\Big) \\ + \frac{1}{n}\Big(\mathbb{E}[\|x\|^2 G'(w_0\cdot x)^2] + \mathbb{E}[G''(w_0\cdot x)] - 2\,\mathbb{E}[(x\cdot w_0)G'(w_0\cdot x)G''(w_0\cdot x)]\Big)\,.\end{aligned} \tag{4.14}$$

We can compute the scaling for the last three terms by using Lemma A.6. On the other hand, the scaling for each of the first three terms is obtained by expanding $\mathbb{E}[f]$ and $\mathbb{E}[g]$ up to sixth-order and repeating the same calculation of Proposition 4.3 while avoiding the evaluation $k^* = 4$. We get that, for any n, we have

$$\begin{aligned}\mathbb{E}[\|w_1\|^2] &= \frac{n(n-1)}{n^2}\left(\frac{(c^G_{k^*}c^\ell_{k^*})^2}{(k^*-1)!^2}\alpha_0^{2(k^*-1)}\right) \\ &\quad + \frac{1}{n}\Big[d\left(\sum_{k\in 2\mathbb{N}+1}\frac{c^{G'}_k c^\ell_k}{k!}\right) + O(d\alpha_0^{k^*}) + \left(\sum_{k\in 2\mathbb{N}}\frac{c^{G''}_k c^\ell_k}{k!}\right) + O(\alpha_0^{k^*}) - 2c_0^{G''}c_1^{G'}c_0^\ell + O(1)\Big] \\ &= \frac{(c^G_{k^*}c^\ell_{k^*})^2}{(k^*-1)!^2}\alpha_0^{2(k^*-1)} - \frac{1}{n^2}\left(\frac{(c^G_{k^*}c^\ell_{k^*})^2}{(k^*-1)!^2}\alpha_0^{2(k^*-1)}\right) + \frac{1}{n}\Big[d\left(\sum_{k\in 2\mathbb{N}+1}\frac{c^{G'}_k c^\ell_k}{k!}\right) + O(d\alpha_0^{k^*}) \\ &\quad + \left(\sum_{k\in 2\mathbb{N}}\frac{c^{G''}_k c^\ell_k}{k!}\right) + O(\alpha_0^4) - 2c_0^{G''}c_1^{G'}c_0^\ell + O(1)\Big].\end{aligned}$$

Hence, we start to keep into account the different choices of $\delta \in [0,2]$. Recall that $n = \Theta(d^{k^*-\delta})$. We need to know that is the addendum which dominates in the previous formula, depending on k^*. By looking at the scaling of each addendum, one can see that there are two possible regimes, which correspond to the two cases $\alpha_0^{2(k^*-1)} \lessgtr d/n$. In particular, when $\delta = 0$, the signal coming from the contribution of the first line in (4.14) dominates. More precisely, we get that

$$\begin{cases}\delta \in (0,2] \quad \Rightarrow \quad \mathbb{E}[\|w_1\|^2] = \Theta\big(\frac{d}{n}\big), \\ \delta = 0 \quad \Rightarrow \quad \mathbb{E}[\|w_1\|^2] = \frac{(c^G_{k^*}c^\ell_{k^*})^2}{(k^*-1)!^2}\alpha_0^{2(k^*-1)} + o(\alpha_0^{2(k^*-1)}).\end{cases}$$

Hence, we have the scaling of the expectations of π_1 and $\|w_1\|^2$. It is left to see how much data samples are required to approximate well these expectations with high probability, i.e. for which sample complexity π_1 and $\|w_1\|^2$ concentrate. Thanks to Assumptions 4.1, 4.2 and 4.6, the proof is analogous to the one in Section B.4 from [165], leading to:

$$|\pi_1 - \mathbb{E}[\pi_1]| \le |f\cdot v - \mathbb{E}[f\cdot v]| + |g\cdot v - \mathbb{E}[g\cdot v]| = O\big(\frac{\log(n)}{\sqrt{n}}\big).$$

and

$$\begin{aligned}|\|w_1\|^2 - \mathbb{E}[\|w_1\|^2]| &\le |\|f\|^2 - \mathbb{E}[\|f\|^2]| + |\|g\|^2 - \mathbb{E}[\|g\|^2]| + 2\,|f\cdot g - \mathbb{E}[f\cdot g]| \\ &= O\left(\frac{d\,\log(n)^6}{n\sqrt{n}} + \frac{\log(d)^6}{n\sqrt{d}} + \frac{\log(n)^2}{n} + \frac{\log(n)^{k^*}}{d^{(k^*-1)/2}\sqrt{n}}\right).\end{aligned}$$

Negative results: We start by analysing what happens when $\delta \in (0,2]$. We have with high probability that

$$\|w_1\|^2 \ge |\mathbb{E}[\|w_1\|^2]| - |\mathbb{E}[\|w_1\|^2] - \|w_1\|^2| = \Theta\big(\frac{d}{n}\big) - O\left(\frac{d\,\log(n)^6}{n\sqrt{n}} + \frac{\log(d)^6}{n\sqrt{d}} + \frac{\log(n)^2}{n} + \frac{\log(n)^{k^*}}{d^{(k^*-1)/2}\sqrt{n}}\right)$$

which implies that

$$\|w_1\|^2 = \Omega\left(\frac{d}{n}\right).$$

Moreover, with high probability,

$$|\pi_1| \leq |\mathbb{E}[\pi_1]| + |\mathbb{E}[\pi_1] - \pi_1| = O\Big(\frac{1}{d^{(k^*-1)/2}}\Big) + O\left(\frac{\log(n)}{\sqrt{n}}\right) \quad \Rightarrow \quad \pi_1 = O\Big(\frac{1}{d^{(k^*-1)/2}}\Big) + O\left(\frac{\log(n)}{\sqrt{n}}\right).$$

We have to split the reasoning in two cases.

Case 1 : If $\delta \in (0, 1]$, with high probability $\pi_1 = O\Big(\frac{1}{d^{(k^*-1)/2}}\Big)$ and then

$$\alpha_1^2 = \frac{\pi_1^2}{\|w_1\|^2} = \frac{O\left(\frac{1}{d^{k^*-1}}\right)}{\Omega\left(\frac{d}{n}\right)} = \frac{O\left(\frac{1}{d^{k^*-1}}\right)}{\Omega\left(\frac{d^\delta}{d^{k^*-1}}\right)} = O\Big(\frac{1}{\delta^\delta}\Big).$$

It means that the overlap at step one is bounded by above by a quantity which vanishes when the dimension of the inputs d goes to infinity, which is definitely bad news for learning in high dimensions with FastICA. Note however that this case bridges smoothly the cases $\delta = 0$ and the case $\delta = 1$, which correspond to d^{k^*-1}, i.e. the regime in which online SGD is able to learn. It is possible that with more steps, this regime will lead to learning, as it happens for large batch GD (see [15] that we discussed at fig. 1.2 in section 1.2).

Case 2: If $\delta = [1, 2]$, the reason why FastICA does not learn is not due to balancing between two random, although concentrated, quantities. In this case, the number of samples is not enough to reach concentration for π_1, that is $\pi_1 = O\Big(\frac{\log(n)}{\sqrt{n}}\Big)$. Therfore,

$$\alpha_1^2 = \frac{\pi_1^2}{\|w_1\|^2} = \frac{O\left(\frac{\text{polylog}(n)}{n}\right)}{\Omega\left(\frac{d}{n}\right)} = O\left(\frac{\text{polylog}(d)}{d}\right).$$

The latter negative result is *qualitatively* different from the previous one, since it is due to the lack of concentration of π_1. The spike is not recovered at all, meaning that the scaling of α_1 is as bad as the one provided by the random weights initialisation.

Positive result: Let's consider $\delta = 0$, which means that we employee an extensive number of data points, namely $n = \Theta(d^{k^*})$. Then,

$$\|w_1\|^2 \leq |\mathbb{E}[\|w_1\|^2]| + |\mathbb{E}[\|w_1\|^2] - \|w_1\|^2| = \frac{(c_{k^*}^G c_{k^*}^\ell)^2}{(k^*-1)!^2}\alpha_0^{2(k^*-1)} + o(\alpha_0^{2(k*-1)}).$$

On the other hand, we also have the following lower bound:

$$|\pi_1| \geq |\mathbb{E}[\pi_1]| - |\mathbb{E}[\pi_1] - \pi_1| = \left|\frac{c_{k^*}^G c_{k^*}^\ell}{(k^*-1)!}\alpha_0^{k^*-1}\right| - o(\alpha_0^{k^*-1}).$$

In conclusion, we get

$$\alpha_1^2 = \frac{\pi_1^2}{\|w_1\|^2} \geq \frac{\frac{(c_{k^*}^G c_{k^*}^\ell)^2}{(k^*-1)!^2}\alpha_0^{2(k^*-1)} - o(\alpha_0^{2(k^*-1)})}{\frac{(c_{k^*}^G c_{k^*}^\ell)^2}{(k^*-1)!^2}\alpha_0^{2(k^*-1)} + o(\alpha_0^{2(k^*-1)})} = 1 - o(1),$$

meaning that in this regime, when $d \to +\infty$, the relevant direction is perfectly recovered by the first iteration of FastICA. □

4.E Smoothing the landscape

For implementation purposes, we start by writing the formula for the spherical gradient of the smoothed loss, that is

$$\nabla_{\text{sph}} L_\lambda(w,x) = \nabla_{\text{sph}} \mathcal{L}_\lambda(G(w\cdot x)) = x\, \mathbb{E}_{z_1}\left[G'\left(\frac{w\cdot x + \lambda z_1 \|P_w^\perp x\|}{\sqrt{1+\lambda^2}}\right)\left(\frac{1}{\sqrt{1+\lambda^2}} - \frac{\lambda z_1 (w\cdot x)}{\|P_w^\perp x\|\sqrt{1+\lambda^2}}\right)\right],$$

where z_1 is one of the components, for example the first one, of a random vector $z \sim \text{Unif}(\mathbb{S}^{d-2})$. This formula can be obtained by following the proof of Lemma 9 in [18]. Note that we need to compute the spherical gradient in order to write the iteration of *smoothed online SGD*.

We now briefly comment on the benefit of smoothing the loss landscape, which is given by taking into account the expectation of the contrast function evaluated in the dot product between the (normalized) vector $w + \lambda z$ and the vector of the inputs x, instead of evaluating the loss in $s = w \cdot x$. The smoothing operator is defined in definition 4.5. Recall that z is a vector of white noise and note that in theorem 4.7 the parameter λ is large, namely it scales with the dimension d of the inputs. The key intuition behind the smoothing operator is that, thanks to large λ, it allows to evaluate the loss function in regions that are far from the updated weights w_t, collecting non-local signal that alleviates the flatness of the saddle of the loss function at the beginning of learning, reducing the length of the search phase. In fig. 4.3 we plot the smoothed loss function, in the cases of $\lambda = d^{1/4}$, $\lambda = d^{1/8}$ and $\lambda = 0$. The first case corresponds to the optimal value of λ, that is, the one for which the optimal thresholds of theorem 4.7 are obtained. The latter case corresponds to the absence of smoothing. It can be interesting to recall that for $\lambda \gg d^{1/4}$ we have the "noise' which dominates the "signal', in the sense of eq. (4.17), where the admissible values of λ are chosen. For the plots in fig. 4.3 we have used $n = 10^4$ number of samples, $d = 10$ and $G(s) = -h_4(s)$.

In the following section we will go through the rigorous argument that formalizes the heuristic derivation presented in Section 4.3.5.1.

4.E.1 Proof

First we provide the lemma used in the main sketch of the proof.

Corollary 4.13. *Assume that ℓ is the likelihood ratio of a probability distribution and that $G \in C^1(\mathbb{R})$ is even. Then, if $w_1, w_2, w_3 \in \mathbb{S}^{d-1}$ and $w_1 \neq w_2 \neq w_3$, we obtain that*

$$E_{x\sim\mathcal{N}(0,\mathbb{1}_d)}[G'(w_1\cdot x)G'(w_2\cdot x)\ell(w_3\cdot x)] = \Theta((w_1\cdot w_2)^{k_2^*-1}).$$

Proof. By considering the Hermite expansion of G' and ℓ, it clearly follows that

$$E_{x\sim\mathcal{N}(0,\mathbb{1}_d)}[G'(w_1\cdot x)G'(w_2\cdot x)\ell(w_3\cdot x)] = \sum_{i,j,k=0}^{\infty} \frac{c_i^{G'} c_j^{G'} c_k^\ell}{i!j!k!}\mathbb{E}_{x\sim\mathcal{N}(0,\mathbb{1}_d)} \underbrace{[h_i(w_1\cdot x)h_j(w_2\cdot x)h_k(w_3\cdot)]}_{(\star)}, \tag{4.15}$$

where, in view of Lemma A.6, when $\rho := w_1 \cdot w_2$, $\tau := w_1 \cdot w_3$ and $\eta := w_2 \cdot w_3$, it holds that

$$\mathbb{E}[(\star)] = \sum_{t=0}^{j} \frac{i!j!k!}{(j-t)!^2 \Big(\frac{i+j-k}{2}\Big)!\Big(\frac{i+k-j}{2}\Big)!\Big(\frac{k-j+2t-i}{2}\Big)!} \rho^t \tau^{k-j+t} (\eta - \tau\rho)^{j-t} \tag{4.16}$$

and then

$$\mathop{\mathbb{E}}_{x\sim\mathcal{N}(0,\mathbb{1}_d)}[G'(w_1\cdot x)G'(w_2\cdot x)\ell(w_3\cdot x)] = \sum_{i,j,k=0}^{\infty}\sum_{t=0}^{j} \text{coeff}\ \ \rho^t\tau^{k-j+t}(\eta-\tau\rho)^{j-t},$$

where

$$\text{coeff} = \frac{c_i^{G'} c_j^{G'} c_k^{\ell}}{(j-t)!^2 \Big(\frac{i+j-k}{2}\Big)!\Big(\frac{i+k-j}{2}\Big)!\Big(\frac{k-j+2t-i}{2}\Big)!}.$$

We can compute the way this sum scales with the dimension by recalling that since ℓ is a probability distribution, we have $c_0^{\ell} = 1$. However, since G is even, $c_0^{G'} = c_1^{G} = 0$. The smallest indices such that the sum does not vanish are then $k = 0$ and $i = j = k_2^* - 1$. Then, it turns out that the only term which survives in the sum when $k = 0$ is the one with $t = k_2^* - 1$. Therefore, up to checking that the other terms in the sum are dominated by the addendum corresponding to $k = 0$, $i, j = k_2^* - 1$ and $t = k_2^* - 1$, we have

$$\mathop{\mathbb{E}}_{x\sim\mathcal{N}(0,\mathbb{1}_d)}[G'(w_1\cdot x)G'(w_2\cdot x)\ell(w_3\cdot x)] = \Theta(\rho^{k_2^*-1}).$$

□

We prove now Theorem 4.7, which provides the timescales for recovering the relevant direction with smoothed online SGD. Since the Gaussian version of this theorem has already been proved in [18], we will only emphasise the main differences between the Gaussian and non-Gaussian case, which are mainly due to the presence of the (bounded) likelihood ratio and the fact that the analysis cannot take into account only the information exponent k_2^* of the contrast function, but also the information exponent of the population loss, and those in general are not equal. However, we will conclude that smoothed online SGD obtains the optimal sample complexity only in the matched case, which is for $k_1^* = k_2^*$.

Proof of Theorem 4.7 (Escaping mediocrity). The proof is made of two parts. The first part is dedicated to the computation of the scaling for the two main quantities that take part in the iteration of smoothed online SGD, i.e. the signal and the noise. The first part ends when we have collected all the ingredients to prove an analogue of the thesis of Lemma 15 in [18], which for us is given by (4.19). This formula guarantees that the overlap between the spike and the updated weight vector given by smoothed online SGD is precisely the sum of the previous overlap, the signal, the noise and a suitable martingale term. The second part is dedicated to the application of standard arguments in probability theory (for the case of vanilla and smoothed SGD see e.g. [16] and [18] respectively, we already saw a version of this argument for the proof of proposition 3.5), from which it is straight forward to conclude that at some point, i.e. for a certain number of used data points, SGD reaches a regime in which the signal dominates the noise and the spike is recovered.

Signal: We start by studying the scaling of the population (spherical) gradient $\nabla_{\text{sph}} L_\lambda(w)$. This scaling will depend on the information exponent k_1^* of the population loss. To do so, consider

the expansion as series of Hermite polynomials of the population loss

$$\mathcal{L}(w) = \sum_{k \geq k_1^*}^{\infty} \frac{c_k^G c_k^\ell}{k!} \alpha^k,$$

where k_1^* is its information exponent. In Lemma 8 of [18] it is shown that we can apply the smoothing operator to each term and obtain that, for any $k \geq 0$,

$$\mathcal{L}_\lambda(\alpha^k) \approx s_k(\alpha, \lambda), \text{ with } s_k(\alpha, \lambda) := \frac{1}{(1+\lambda^2)^{k/2}} \begin{cases} \alpha^k & \alpha^2 \geq \frac{\lambda^2}{d}, \\ (\frac{\lambda^2}{d})^{k/2} & \alpha^2 \leq \frac{\lambda^2}{d} \text{ and } k \text{ is even}, \\ \alpha(\frac{\lambda^2}{d})^{(k-1)/2} & \alpha^2 \leq \frac{\lambda^2}{d} \text{ and } k \text{ is odd}. \end{cases}$$

By computation, we can obtain that the population (spherical) gradient is $\nabla_{\text{sph}} L_\lambda(w) = (v - \alpha w) c_\lambda(\alpha)$, with

$$c_\lambda(\alpha) := \sum_{k \geq k_1^*} \frac{c_k^G c_k^\ell}{k!} \frac{d}{d\alpha} \mathcal{L}_\lambda(\alpha^k).$$

The scalar product of this population gradient term and the relevant direction, i.e. $v \cdot \nabla_{\text{sph}} L_\lambda(w) = (1 - \alpha^2) c_\lambda(\alpha)$, is called *signal*. It can be proved, like in Lemma 10 of [18], that

$$c_\lambda(\alpha) \approx \frac{s_{k_1^*-1}(\alpha, \lambda)}{\sqrt{1+\lambda^2}}.$$

Noise: Now we take a look at the noise term. Fix $u \in \mathbb{S}^{d-1}$, with $u \perp w$. Recall that we are using the notation $g = \nabla_{\text{sph}} L_\lambda(w, x)$. Since we have assumed that the likelihood ratio ℓ is bounded (e.g. as the case of inputs drawn according to the noisy ICA model (4.5)), we get

$$\mathbb{E}\left[(u \cdot g)^2\right] = \mathbb{E}_{x \sim \mathcal{N}(0, \mathbb{1}_d)}\left[(u \cdot g)^2 \ell(v \cdot x)\right] \leq \|\ell\|_\infty \, \mathbb{E}_{x \sim \mathcal{N}(0, \mathbb{1}_d)}\left[(u \cdot g)^2\right] \tag{4.17}$$

$$\lesssim \frac{\min\left(1+\lambda^2, \sqrt{d}\right)^{-(k_2^*-1)}}{1+\lambda^2} \lesssim (1+\lambda^2)^{-k_2^*}, \tag{4.18}$$

where we have used the fact that $\lambda \in [1, d^{1/4}]$ and Corollary 1 in [18]. In the last inequality, thanks to the choice $\lambda \in [1, d^{1/4}]$, we are using the fact that

$$\min\left(1+\lambda^2, \sqrt{d}\right)^{-(k_2^*-1)} = \left(1+\lambda^2\right)^{-(k_2^*-1)}$$

Note that we have obtained an explicit dependence on the information exponent k_2^* of the contrast function. We can estimate now the second moment of the smoothed spherical gradient, i.e. $\mathbb{E}[\|g\|^2]$, which we call *noise*. To do so, recall that g is a spherical gradient, hence $g \perp w$. To get an upper bound on the scaling of the noise, it is sufficient to fix an orthonormal basis in $\mathbb{R}^d$ such that $e_1 = w$. Then,

$$\mathbb{E}[\|g\|^2] = \sum_{i=2}^{d} \mathbb{E}\left[(e_i \cdot g)^2\right] \lesssim d\, O((1+\lambda^2)^{-k_2^*}).$$

Now that we have the desired upper bound, it is possible to compute the scaling for the expectation of some p- moments of $\|g\|^2$, for p-sufficiently large. This corresponds to Corollary 2 in [18].

By considering both the scaling of the signal and the noise, we can now state the non-Gaussian version of Lemma 15 in [18], which tells us that if $\alpha_{t+1} := v \cdot w_{t+1}$ and $w_{t+1} := \dfrac{w_t + \eta g_t}{\|w_t + \eta g_t\|}$, we obtain that with high probability

$$\alpha_{t+1} = \alpha_t + \eta(1-\alpha_t^2)c_\lambda(\alpha_t) + Z_t + O(\mathbb{E}[\|g_t\|^2]). \tag{4.19}$$

where $\{Z_t\}_{t\in\mathbb{N}}$ is a martingale term with zero mean defined as

$$Z_t := \eta\,(g_t \cdot v) - \mathbb{E}\,[\eta\,(g_t \cdot v)] + r_t - \mathbb{E}\,[r_t]$$

and

$$r_t := [\alpha_t + \eta\,(g_t \cdot v)]\left(\frac{1}{\sqrt{1+\eta^2||g_t||^2}} - 1\right).$$

Hence, at each step of smoothed online SGD, the updated overlap between the new weight vector and the spike is nothing but the previous overlap plus a martingale term, the signal - whose scaling depends on k_1^* - and the noise - whose scaling depends on k_2^*. We have reached the setting of Lemma 16 of [18]. From here on, the proof can be concluded by applying the argument that translates the signal and noise estimated into sample complexities, introduced in the setting of vanilla online SGD by [16]. The proofs of Lemma 16 and Lemma 17 from [18] can be repeated straight forwardly, leading to the required sample complexities for recovering the spike. □

4.F The subspace model turns out to be a spiked cumulant model after preprocessing

Before running ICA, the inputs are usually preprocessed in a canonical way. This standard procedure includes whitening to ensure that the covariance matrix is the identity, together with PCA, to reduce the dimensionality of the original data. It means that, if the inputs are in $\mathbb{R}^D$, they are usually projected in the space generated by the first fixed $r < D$ principal components, and then ICA is performed. We consider the inputs drawn according to the subspace model (4.10). Note that we are assuming that the spikes $\{u^1, \dots, u^r\}$ are the first r - leading principal components, and that the spike v is a linear combination of those. In the following theorem, we prove that the pre-processed inputs follow a spiked cumulant distribution, where the spike which corresponds to the non-Gaussian signal is simply the normalized projection on the suitable low dimensional subspace of the rotated (w.r.t. the whitening matrix) original non-Gaussian spike.

Theorem 4.14 (Reduced ICA model)**.** *Consider the subspace model* (4.10)*. If the common preprocessing steps are performed, i.e. the data are whitened and projected onto the low dimensional subspace spanned by the leading principal components* $\{u^1, \dots, u^r\}$*, the transformed inputs are drawn according to the following model:*

$$y^\mu = \sqrt{\gamma}\, h^\mu \widetilde{v} + \hat{z}^\mu \in \mathbb{R}^r,$$

where $\widetilde{v} = \mathrm{proj}(\mathrm{Sv})/\|\mathrm{proj}(\mathrm{Sv})\|$ *is the normalised projection on the low dimensional subspace of the whitened original spike* v*. More precisely,* γ *is a suitable signal to noise ratio,* $S \in \mathbb{R}^{D\times D}$ *is the whitening matrix for our original inputs and* $\hat{z}^\mu$ *is a zero-mean noise vector with covariance matrix* $\sum_{\rho=1}^r (1-(Sv)_\rho^2)\, e_\rho\, e_\rho^\top$*, with* $\{e_1, \dots, e_r\}$ *vectors of the canonical basis for* $\mathbb{R}^r$*.*

Proof. The subspace model (4.10) is build in such a way that the first r leading components are the spikes $\{u^1, \dots, u^r\}$, being the covariance matrix of the inputs $C = \sum_{\rho=1}^r \beta_2^\rho \, u^\rho (u^\rho)^\top + \mathbb{1}_d$. Importantly, the non-Gaussian spike to be recovered from (4.10) is a linear combination of the already mentioned spikes, which implies that all the signal is contained in the subspace generated by them.

We are going to show that when we whiten the data and project them onto the subspace generated by $\{u^1, \dots, u^r\}$, we still recover a spiked cumulant model in a fixed finite dimension r.

Whitening To whiten the data, if is sufficient to multiply them by the whitening matrix $S := \sqrt{C^{-1}}$, which in the case of the subspace model (4.10) is simply

$$S = U \begin{pmatrix} 1/\sqrt{1+\beta_2^1} & & & & & \\ & \ddots & & & & \\ & & 1/\sqrt{1+\beta_2^r} & & & \\ & & & 1 & & \\ & & & & \ddots & \\ & & & & & 1 \end{pmatrix} U^\top,$$

where $U := [u^1, \dots, u^r, e_{r+1}, \dots, e_D] \in \mathbb{R}^{D \times D}$ is an orthogonal matrix, being $\{e_{r+1}, \dots, e_D\}$ part of the canonical basis for $\mathbb{R}^D$. Moreover, we can from now on set $S_{ii} = 0$ for $i = r+1, \dots, D$, since in the next step we will only retain the first r principal components.

The new - whitened - inputs are now defined as $\widetilde{x}^\mu := Sx^\mu = Uy^\mu$, for any $\mu = 1, \dots, n$. Note that the entries of $y^\mu \in \mathbb{R}^D$ are just the components of the original input x^μ with respect to the basis $\{u^1, \dots, u^r, e_{r+1}, \dots, e_D\}$, after the rotation due to the whitening. More precisely, for $\rho = 1, \dots, r$, the entries of y^μ read

$$y_\rho^\mu := \frac{1}{\sqrt{1+\beta_2^\rho}} (x^\mu \cdot u^\rho) = \frac{1}{\sqrt{1+\beta_2^\rho}} \Big[\sqrt{\beta_2^\rho} g_\rho^\mu + h^\mu \alpha^\rho + (u^\rho \cdot z^\mu) \Big], \tag{4.20}$$

where we have used the fact that the non-Gaussian spike v is a linear combination of the pairwise orthogonal spikes $\{u^1, \dots, u^r\}$ and that α_ρ are the corresponding coefficients. Since the noise vector z^μ has mean zero and covariance matrix $\mathbb{1} - vv^\top$, it holds that $(u^\rho \cdot z^\mu) \sim \mathcal{N}(0, 1 - \alpha_\rho^2)$. Hence,

$$\widetilde{z}_\rho^\mu := \frac{1}{\sqrt{1+\beta_2^\rho}} \Big[\sqrt{\beta}_2^\rho g_\rho^\mu + (u^\rho \cdot z^\mu) \Big] \sim \mathcal{N}\Big(0, \, 1 - \frac{\alpha_\rho^2}{1+\beta_2^\rho}\Big).$$

We conclude that the non trivial entries of each y^μ are given by

$$y_\rho^\mu = \frac{\alpha_\rho}{\sqrt{1+\beta_2^\rho}} h^\mu + \widetilde{z}_\rho^\mu. \tag{4.21}$$

On the other hand, recall that we have $y_i^\mu = 0$ for $i = r+1, \dots, D$.

Projection on the subspace spanned by the principal components We now project each whitened input $\widetilde{x}^\mu$ onto the subspace generated by the first r principal components $\{u^1, \dots, u^r\}$, which simply means that we take track only of the r non trivial components given by (4.21), i.e. the ones for $\rho = 1, \dots, r$. More formally, but with a slight abuse of notation, we keep calling the projected vectors y^μ, so that each projected input is now $y^\mu = [u^1, \dots, u^r]^\top \widetilde{x}^\mu \in \mathbb{R}^r$.

It is left to show that, after whitening and PCA, our data are still spiked and, more precisely, we have recovered nothing but a spiked cumulant model in $\mathbb{R}^r$. Indeed,

$$\begin{aligned} y^\mu &= \sum_{\rho=1}^r y^\mu_\rho \, e_\rho = \Big(\sum_{\rho=1}^r \frac{\alpha_\rho}{\sqrt{1+\beta_2^\rho}} \, e_\rho \Big) \sqrt{\beta_4} h^\mu + \sum_{\rho=1}^r \tilde{z}^\mu_\rho \, e_\rho \\ &= h^\mu \, \mathrm{proj}(Sv) + \hat{z}^\mu = \|\mathrm{proj}(Sv)\| h^\mu \underbrace{\mathrm{proj}(Sv)/\|\mathrm{proj}(Sv)\|}_{\text{new spike } \widetilde{v}} + \hat{z}^\mu \\ &= \|\mathrm{proj}(Sv)\| h^\mu \, \widetilde{v} + \hat{z}^\mu, \end{aligned}$$

where $\mathrm{proj}(Sv)$ is the projection of Sv onto the subspace generated by the spikes $\{u^1, \dots, u^r\}$, $\widetilde{v} := \mathrm{proj}(Sv)/\|\mathrm{proj}(Sv)\|$, and $\hat{z}^\mu$ is a Gaussian random vector of zero mean and covariance matrix $\sum_{\rho=1}^r (1 - (Sv)^2_\rho) \, e_\rho \, e_\rho^\top$.
In conclusion, the new (projected and whitened) inputs are nothing but

$$y^\mu = \|\mathrm{proj}(Sv)\| \, h^\mu \widetilde{v} + \hat{z}^\mu \in \mathbb{R}^r,$$

hence we found that the inputs are drawn according to a spiked cumulant model, where $\mathrm{proj}(Sv)$ is the spike corresponding to the original non-Gaussian signal h^μ. If we set $\gamma = \|\mathrm{proj}(Sv)\|^2$, the thesis follows. □

Thanks to Theorem 4.14, in case of inputs drawn from the subspace model and subjected to standard preprocessing steps, ICA is supposed to recover one single low dimensional spike belonging to the subspace spanned by the leading principal components $\{u^1, \dots, u^r\} \subset \mathbb{R}^r$. If the signal-to-noise ratios $\{\beta_2^1, \dots, \beta_2^r\}$ of the principal components are equal, up to normalisation and rescaling, the said spike is nothing but the vector $\alpha := (\alpha_1, \dots, \alpha_r)^\top$, whose components are the ones of the original spike v corresponding to the non-Gaussian latent variable h^μ in the subspace model. In Section 4.F.1 we discuss what happens to the reduced model when we do not have access to the exact value of the leading principal components, but we estimate it through PCA. Relying on the fact that the population loss for ICA on the reduced model exhibits convenient monotonicity properties, in Section 4.F.2 we prove the convergence, in a linear sample complexity, of gradient descent (GD), therefore concluding that this kind of finite optimisation is indeed effective the context of the subspace model.

4.F.1 Errors in estimation of PCs at linear sample complexity in high dimensions

Recall that we are interested in analysing the performance of ICA at a linear sample complexity, namely when the number of samples scales as the dimension of the ambient space D, up to logarithmic factors. So, we start by using an infinite amount of samples to estimate the first principal components of our D-dimensional inputs. Then, we project down in low dimensions. This means that, in practice, the principal components have to be estimated through PCA, and this may have an impact on the distribution of the reduced model. Indeed, PCA at linear sample complexity only yields an estimate $\hat{u}^\rho$ of the leading principal components, whose overlap with u^ρ can be computed exactly: in case of a finite signal-to-noise ratio and $n = \Theta(D)$ it is strictly smaller than 1, see for example Eq. 3.37 of Bandeira *et al.* [168] for the rank-1 case. However, a number of samples n such that $n/D \to \infty$, even if the divergence is arbitrary slow, is sufficient to perfectly recover with high probability the eigenvectors of the covariance matrix of the inputs. Hence, in practice, one considers the change of basis provided by the matrix U is made by $\widehat{U} := [\hat{u}_1, \dots, \hat{u}_r, e_{r+1}, \dots, e_D] \in \mathbb{R}^{D \times D}$, where $\{\hat{u}^1, \dots, \hat{u}^r\}$ is the set of the estimators provided by PCA at a linear sample complexity of

the first r-leading components. Fix $\varphi \in [0,1]$, which is the overlap between the estimator for each principal component and the principal component itself, and define $\varepsilon = \sqrt{1-\varphi^2}$. When $n/D \to \infty$ arbitrarily slow, e.g. it is sufficient that $n = \Theta(D \log D)$, we have that $\varepsilon = \varepsilon(D) \to 0$, and then clearly $\varphi = \varphi(D) \to 1$. For the derivation of the (perturbed) reduced model, we assume that the estimated PCs are

$$\hat{u}^\rho = \varphi\, u^\rho + \varepsilon \widetilde{u}^\rho, \tag{4.22}$$

for $\rho = 1, \dots, r$ and $\widetilde{u}^\rho \perp u^\rho$, $\|\widetilde{u}^\rho\| = 1$. The following lemma shows that we can obtain the reduced model even if the eigenvectors of the covariance matrix of the inputs are approximated by PCA. The model is simply the one found in Theorem 4.14, up to some errors introduced with the perturbed eigenvectors, which however vanish for $D \to \infty$.

Lemma 4.15. *Consider the subspace model* (4.10) *and assume we perform the usual preprocessing steps for ICA without having access to the exact value of the covariance spikes, but only to $\hat{u}^\rho$ defined in Eq. 4.22. Consider a number of samples n such that n/D diverges arbitrary slow, e.g. $n = \Theta(D \log D)$. Then, with high probability, the transformed inputs are nothing but the reduced model introduced in Theorem 4.14, up to vanishing errors.*

Proof. We need to compute, for $\mu = 1, \dots, D$, the whitened inputs $\widetilde{x}^\mu = S x^\mu = \widehat{U} y^\mu$, where S is the whitening matrix. After that, we will project the inputs onto the principal subspace. Note that for any $\rho, \rho' = 1, \dots, r$, we obtain

$$u^\rho \cdot \hat{u}^{\rho'} = (\varphi u^\rho + \varepsilon \widetilde{u}^\rho) \cdot u^{\rho'} = \varphi\, \delta_{\rho\rho'} + \varepsilon\, (\widetilde{u}^\rho \cdot u^{\rho'}).$$

Hence, with respect to the usual pairwise orthonormality of the spike generating the principal subspace, there is an extra "error term" provided by the fact that we cannot precisely estimate the principal components with a linear number of samples. Taking into account these perturbations, we compute for $\mu = 1, \dots, n$, $\rho = 1, \dots, r$ the components of y^μ, i.e.

$$\begin{aligned}
y^\mu_\rho =& \frac{1}{\sqrt{1+\beta_2^\rho}} \Big[\sum_{\rho'=1}^r \sqrt{\beta_2^{\rho'}}\, g^\mu_{\rho'} (\hat{u}^\rho \cdot u^{\rho'}) + \sqrt{\beta_4} h^\mu \sum_{\rho'}^r \alpha_{\rho'} (\hat{u}^\rho \cdot u^{\rho'}) + z^\mu \cdot \hat{u}^\rho \Big] \\
=& \frac{1}{\sqrt{1+\beta_2^\rho}} \Big[\sqrt{\beta_2^\rho} g^\mu_\rho \varphi + \sum_{\rho'=1}^r \varepsilon \sqrt{\beta_2^{\rho'}}\, g^\mu_{\rho'} (\widetilde{u}^\rho \cdot u^{\rho'}) + \underbrace{h^\mu \alpha_\rho \varphi + \sum_{\rho'}^r \alpha_{\rho'} \varepsilon h^\mu (\widetilde{u}^\rho \cdot u^{\rho'})}_{\text{non-Gaussian signal}} + z^\mu \cdot \hat{u}^\rho \Big] \\
=& \sqrt{\frac{1}{1+\beta_2^\rho}}\, h^\mu \Big(\alpha_\rho \varphi + \sum_{\rho'=1}^r \alpha_{\rho'} (\widetilde{u}^\rho \cdot u^{\rho'}) \varepsilon \Big) + \widetilde{z}^\mu_\rho,
\end{aligned}$$

where the Gaussian noise is

$$\widetilde{z}^\mu_\rho = \frac{1}{\sqrt{1+\beta_2^\rho}} \Big[\sqrt{\beta_2^\rho} g^\mu_\rho \varphi + \sum_{\rho'=1}^r \sqrt{\beta_2^{\rho'}}\, g^\mu_{\rho'} (\widetilde{u}^\rho \cdot u^{\rho'}) \varepsilon + \varphi (z^\mu \cdot u^\rho) + \varepsilon (z^\mu \cdot \widetilde{u}^\rho) \Big].$$

Clearly, for any $\rho = 1, \dots, r$, $\widetilde{z}^\mu_\rho$ is a centred Gaussian random variable with variance

$$c = \frac{1}{1+\beta_2^\rho} \Big(\varphi^2 \beta_2^\rho + \varepsilon^2 \sum_{\rho'=1}^r \beta_2^{\rho'} (\widetilde{u}^\rho \cdot u^{\rho'})^2 + 1 - \alpha_\rho^2 \Big) \in \mathbb{R}.$$

The other components y^μ_ρ for $\rho = r+1, \dots, D$ vanish. In the previous formulas, it has been emphasised the dependence of several terms on the overlap φ between any exact spike and its

estimation provided by PCA. Indeed, we can assume now that all the other terms, depending on ε, vanish with high probability when $D \to \infty$, since e.g. $n = \Theta(D \log D)$. Hence, we get

$$y_\rho^\mu = \varphi \left(\sqrt{\frac{1}{1+\beta_2^\rho}} h^\mu \alpha_\rho + \sqrt{\frac{\beta_2^\rho}{1+\beta_2^\rho}} g_\rho^\mu + (z^\mu \cdot u^\rho) \right) + o_D(1), \tag{4.23}$$

where $\varphi \to 1$ with high probability. We can see that the components of the inputs in (4.23) are the ones previously obtained in Theorem 4.14 multiplied by the overlap φ. Now, the last preprocessing step consists in projecting $y^\mu \in \mathbb{R}^D$ in the right low dimensional subspace. As in Theorem 4.14, with a slight abuse of notation, we adopt the notation y^μ to refer simply to the vector in $\mathbb{R}^r$ whose components are y_ρ^μ, for $\rho = 1, \dots, r$. Hence, the reduced inputs turn out to be $y^\mu + Ry^\mu$, where R is an "error" matrix whose entries are $r_{ij} = \hat{u}^i \cdot \hat{u}^j$ if $i \neq j$ and zero otherwise. Since $\hat{u}^i \cdot \hat{u}^j = \varphi\varepsilon(u^i \cdot \widetilde{u}^j) + \varphi\varepsilon(\widetilde{u}^i \cdot u^j) + \varepsilon^2(\widetilde{u}^i \cdot \widetilde{u}^j)$, these last error terms vanish with high probability when $D \to \infty$. □

4.F.2 Optimization on the subspace

In this section we prove that it is possible to efficiently solve ICA on the subspace model (4.10) with $n = \Theta(D \log(D))$ (or even slower, it is sufficient that $n/D \to \infty$ to be sure that the errors introduced with the estimation of the principal components by PCA vanish for $D \to \infty$) samples after the projection on the principal r-dimensional subspace, with $r = O(1)$. The data model we refer to in this section is the reduced model (4.11), which is a simple spikes cumulant model (see e.g. [30]). In our case, the spike corresponding to the non-Gaussian latent variable, namely the one to be recovered, is $\widetilde{v}$ in (4.11). We choose to directly state and prove theorem 4.16 for inputs distributed according to the reduced model because, when the number of samples is sufficiently large, the errors due to the estimation of the principal components have a limited impact on the distribution of the inputs, in the sense of Lemma 4.15.

On the algorithmic side, we consider a projected gradient method that uses the n samples in K batches of B samples each, with $KB = n$ and both diverging in the high dimensional limit, $K, B = \omega(1)$ (for example $K \approx B \approx \sqrt{n}$). So we denote with $\mathcal{B}_k = (x^\mu)_{\mu \in [B]}^k$ the k-th batch of i.i.d. samples from $\mathbb{P}$, the distribution of the already mentioned inputs in $\mathbb{R}^r$. The algorithm is:

$$\begin{cases} w_0 \sim \mathrm{Unif}\left(\mathbb{S}^{r-1}\right), \\ \widetilde{w}_{k+1} = w_k + \eta \nabla_{\mathrm{sph}} \frac{1}{B} \sum_{x \in \mathcal{B}_k} G(w \cdot x) \quad k > 1, \\ w_{k+1} = \frac{\widetilde{w}_{k+1}}{\|\widetilde{w}_{k+1}\|}, \end{cases} \tag{4.24}$$

where $\nabla_{\mathrm{sph}} := (\mathbb{1} - w_k w_k^\top)\nabla$.

Theorem 4.16 (Convergence of gradient ascent)**.** *Assume that the population loss $\mathcal{L}(w) = \mathbb{E}_{\mathbb{P}} G(w \cdot x)$ depends on w only through the the overlap $\alpha = w \cdot \tilde{v}$, and let G be such that:*

- $\mathcal{L} \in \mathcal{C}^2[-1, 1]$,
- $\mathcal{L}(\alpha)\alpha > 0$ *for all* $\alpha \neq 0$ *in* $[-1, 1]$.

We update w_k for $k \in \{0, \dots, K\}$ following the dynamic dictated by eq. (4.24), where $\mathcal{B}_k = (x^\mu)_{\mu \in [B]}^k$ is the k-th batch of i.i.d. samples from $\mathbb{P}$, the spiked cumulant distribution with spike $\tilde{v} \in \mathbb{R}^r$ of (4.11). Then, w_K converges to the spike $\tilde{v}$ with high probability in the limit $n \to \infty$. More precisely, for any ε there exists n_ε, such that if $n > n_\varepsilon$, it is large enough so that we can choose B and K so that $BK \leq n$ and the event $|\alpha_K| = |w_K \cdot \tilde{v}| \geq 1 - \varepsilon$ happens with probability larger than $1 - \varepsilon$.

Proof. The first part of the proof is devoted to estimate the error of replacing $\nabla_{\text{sph}} \frac{1}{B} \sum_{x \in \mathcal{B}_k} G(w \cdot x)$ with $\nabla_{\text{sph}} \mathcal{L}(\alpha)$. First note that G and $\mathbb{P}$ ($G \in \mathcal{C}^1$ and $\ell \in \mathcal{L}^2$) are regular enough so that we can differentiate under the integral sign and write: $\nabla \mathbb{E}[G(w \cdot x)] = \mathbb{E}[\nabla G(w \cdot x)] = \mathcal{L}'(\alpha) v$, where the last equality can be shown with the orthogonality property of Hermite polynomials (see property 1 in [18]). We can now use the law of large numbers; if B is large enough, we can find an event with probability larger than $1 - \varepsilon$ such that for all k:

$$\left| \mathbb{E}[\nabla G(w \cdot x)] - \frac{1}{B} \sum_{x \in \mathcal{B}_k} \nabla_{\text{sph}} G(w \cdot x) \right| \leq \eta^2.$$

So we can study the following dynamics

$$\begin{cases} \widetilde{w}_{k+1} = w_k + \eta \nabla_{\text{sph}} \mathcal{L}(\alpha_k) + O(\eta^2), \\ w_{k+1} = \frac{\widetilde{w}_{k+1}}{||\widetilde{w}_{k+1}||}, \end{cases}$$

where $\nabla_{\text{sph}} := (\mathbb{1} - w_k w_k^\top) \nabla$ is the spherical gradient, that ensures that $\eta \nabla_{sph} \mathcal{L}(\alpha_k)$ is orthogonal to w_k, hence $||\widetilde{w}_{k+1}|| = \sqrt{1 + \eta^2 ||\nabla_{\text{sph}} \mathcal{L}(\alpha_k)||^2}$. Since the step size $\eta \to 0$, we can apply the Taylor expansion $\frac{1}{\sqrt{1+x^2}} = 1 - x^2/2 + o(x^2)$. Since $\nabla \mathcal{L}$ is bounded thanks to our assumptions on G, this leads to

$$w_{k+1} = w_k + \eta \nabla_{\text{sph}} \mathcal{L}(\alpha_k) + O(\eta^2),$$

where the $O(\eta^2)$ is uniform in k. We take now the scalar product with the target vector $\tilde{v}$ to get the evolution of the overlap:

$$\begin{aligned} \alpha_{k+1} &= \alpha_k + \eta \tilde{v} \cdot (\mathbb{1} - w_k w_k^\top) \mathcal{L}'(\alpha_k) \tilde{v} + O(\eta^2) \\ &= \alpha_k + \eta \mathcal{L}'(\alpha_k)(1 - \alpha_k^2) + O(\eta^2). \end{aligned} \tag{4.25}$$

Now it is helpful to consider the continuous time gradient flow $\alpha(t)$ that satisfies the equation:

$$\begin{cases} \dot{\alpha}(t) = \mathcal{L}'(\alpha(t))(1 - \alpha^2(t)), \\ \alpha(0) = \alpha_0 > 0. \end{cases}$$

Note that we assumed $\alpha_0 > 0$ which happens exactly with probability $1/2$, the case $\alpha_0 < 0$ is exactly analogous, we will only use that $\mathcal{L}'(\alpha) < 0$ for all $\alpha < 0$ to prove convergence to -1 instead of $+1$ (if the initialization is negative the algorithm will converge to $-\tilde{v}$ instead of $\tilde{v}$, identifying the same one dimensional subspace). Let $\varepsilon > 0$ and denote $0 < C_\varepsilon := \inf_{[\alpha_0, 1-\varepsilon]} \mathcal{L}'(\alpha_0)$. We have that for all $t > 0$, such that $\alpha(t) \leq 1 - \varepsilon$, then $\alpha(t) > f(t)$ where f is the solution of the auxiliary Cauchy problem

$$\begin{cases} \dot{f}(t) = C(1 - f^2(t)), \\ f(0) = \alpha_0 > 0, \end{cases}$$

which is solved by

$$f(t) = \frac{(\alpha_0 + 1) e^{2Ct} + \alpha_0 - 1}{(\alpha_0 + 1) e^{2Ct} - \alpha_0 + 1},$$

that converges monotonically to 1 as $t \to \infty$. This proves that $\lim_{t \to \infty} \alpha(t) = 1$. We can now transfer this information to the discrete dynamics using convergence of Euler's method (Theorem

10.2 in [178]). For any $\eta \in (0, \eta_0)$ and $k \in \mathbb{N}$:

$$\begin{aligned} |\alpha_k - \alpha(\eta k)| &\leq k\eta e^{\Lambda k\eta} O(\eta) + \sum_{i=1}^{k} O(\eta^2) \\ &\leq k\eta e^{\Lambda k\eta} O(\eta) + kO(\eta^2), \end{aligned}$$

where the first term comes from the convergence error and the second from the error in (4.25). It is possible to perform the second step since the $O(\eta^2)$ depends only on global property of $\mathcal{L}$, so it is uniform in k. Hence, the discrete dynamics approximate the gradient flow for all k such that $k\eta = O(1)$. To conclude the proof, we just need to pick η small enough and K large enough so that $t_K = K\eta$ is such that $\alpha(t) > 1 - \varepsilon/2$ and $|\alpha_K - \alpha(t_K)| \leq \varepsilon/2$. □

The hypotheses required on G are verified for the most commonly used in practice contrast functions. For example, we can trivially deal with the case of the 4th Hermite polynomial $G(s) = s^4 - 3$ by using the Hermite orthogonality property (property 1 in [18]) and the likelihood ratio trick on the loss for ICA [33] whereas to justify the use of $G(s) = -e^{-s^2/2}$ we can explicitly compute the population loss.

Lemma 4.17 (Monotonicity of the population loss)**.** *Consider $G(s) = -\exp\{-s^2/2\}$ and the reduced model* (4.11)*. Then, the population loss for ICA, i.e. $\mathcal{L}(w) = \mathbb{E}_x[G(w \cdot x)]$, can be explicitly computed and depends only on the overlap $\alpha := w \cdot \tilde{v}$ in the following way:*

$$\mathcal{L}(\alpha) = -\sqrt{\frac{2}{4 - \alpha^2}} \ \exp\left\{-\frac{\alpha^4 + 2\alpha^2}{(2 + \alpha^2)(4 - \alpha^2)}\right\}.$$

Proof. The proof consists in the direct computation of the expectation of $G(x \cdot w)$ with respect to the distribution of the inputs. Precisely, for any $w \in \mathbb{R}^r$, we integrate with respect to the non Gaussian spike h and the noise vector z as follows:

$$\mathcal{L}(w) = \frac{1}{2\sqrt{(2\pi)^r}} \left[\left(\int G(w \cdot x)\, e^{-z^2/2} dz \right) \delta_{\{h=1\}} + \left(\int G(w \cdot x)\, e^{-z^2/2} dz \right) \delta_{\{h=-1\}} \right],$$

where each integrand, for $h = \pm 1$, is simply $e^{-\frac{1}{2}[(w \cdot x)^2 + \|z\|^2]}$. Therefore, we need to compute explicitly $(w \cdot x)^2$, which turns out to be

$$(w \cdot x)^2 = \frac{1}{\sqrt{2}} h\alpha + \frac{\alpha}{\sqrt{2}} + (w_{\perp\tilde{v}} \cdot z),$$

where $\alpha := w \cdot \tilde{v}$ and $w_{\perp\tilde{v}}$ is the portion of the weight vector orthogonal to the spike $\tilde{v}$. Now, we can decompose the noise in the component parallel to the spike, i.e. $z_{//\tilde{v}}$, the component paraller to $w_{\perp\tilde{v}}$, i.e. $z_{//w_{\perp\tilde{v}}}$, and everything else $z_{\perp}$, and integrate with respect these three contributions. □

To summarize, we started from the high dimensional problem eq. (4.10) in $\mathbb{R}^D$ with D large and a number of samples $n = \Theta(D \log(D))$ which would be too small to have recovery of v through a direct application of FastICA. However, the projection on the principal component subspace $\langle u_1, \ldots, u_r \rangle$, which is possible to perform with arbitrarily small error (Lemma 4.15) leads to a spiked cumulant model in $\mathbb{R}^r$ (theorem 4.14). Thanks to suitable properties of the population loss, verified in Lemma 4.17, we are able to recover the spiked direction $\tilde{v}$ with a linear sample complexity, as shown in Theorem 4.16.

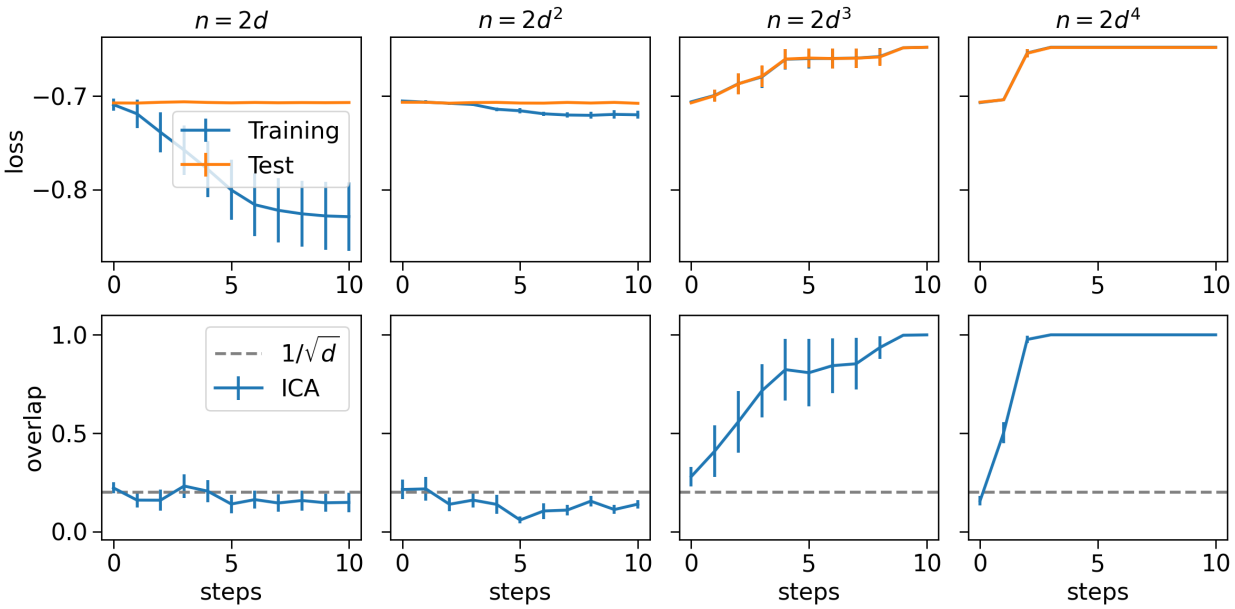

Figure 4.7: **Performance of FastICA on noisy ICA model with Rademacher prior on the latent *without* resampling the training data at each step**. *Parameters*: input dimension $d = 25$, signal-to-noise ratio $\beta = 5$, online learning.

4.G Additional numerics

What is the role of non-Gaussian signal coming from non trivial third-order cumulant for FastICA? In general, it is non-zero for real images, in contrast to the spiked cumulant with Rademacher latent ν. A natural idea would be to suggest that it impacts the performances by modifying the information exponent. Indeed, note that an information exponent $k^* = 3$ for the population loss would suggest that $n \asymp d^{k^*-1}$ is the transitionary regime, where recovery is possible. However, since the contrast functions used in FastICA are even, FastICA is blind to information carried by odd cumulants and then $k^* \neq 3$. We demonstrate this explicitly by constructing a latent distribution which has mean zero, unit variance, a non-trivial third-order cumulant, but zero fourth-order cumulant, and find that FastICA does not pick up any signal when $n \approx d^3$, see fig. 4.8. Moreover, since the latent distribution has zero fourth-order cumulant and k^* has to be even, we have $k^* = 6$ and would need $n \approx d^5$ to learn the non-Gaussian contribution.

Another interesting setting is when the latent variables of images are super-gaussian, e.g. following an approximate Laplace distribution [179], since we proved our results only for sub-gaussian distributions. However, we verified experimentally that when the latent variable ν^μ in the noisy ICA model has a Laplace distribution, theorem 4.4 still accurately describes the sample complexity required for recovery, see fig. 4.9.

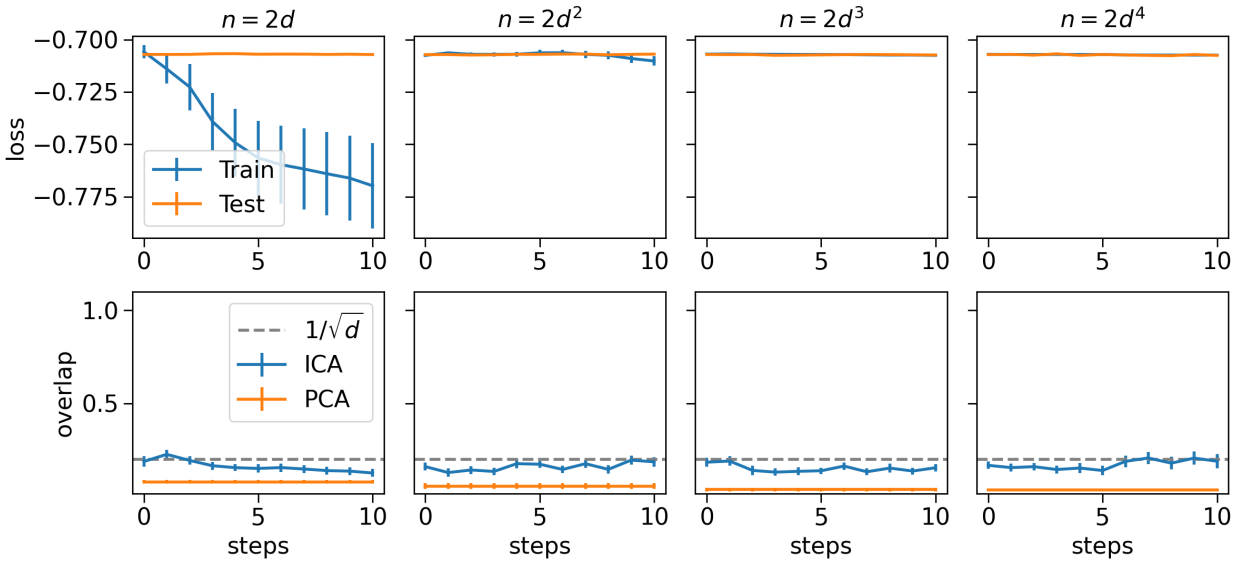

Figure 4.8: **Performance of online FastICA on noisy ICA model with the "3not4" prior on the latent variable**. Under this prior, the latent has mean zero, unit variance, a non-trivial third-order cumulant, but zero fourth-order cumulant; see section 4.A.5 for details. *Parameters*: input dimension $d = 25$, signal-to-noise ratio $\beta = 5$, online learning.

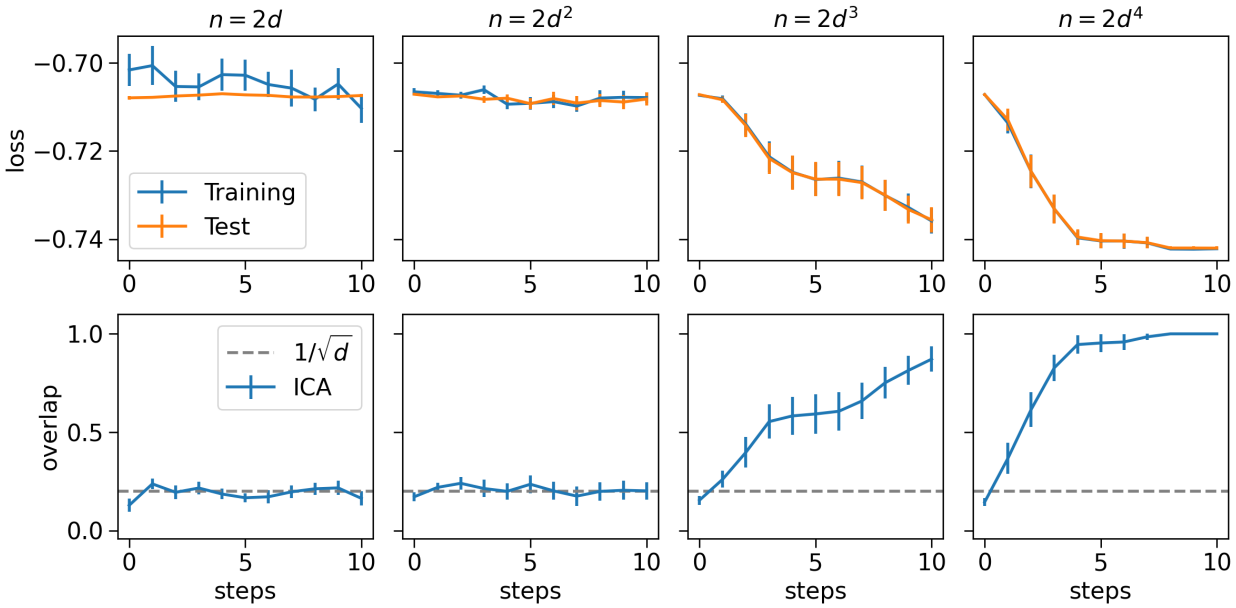

Figure 4.9: **Performance of online FastICA on noisy ICA model with Laplace prior on the latent**. *Parameters*: input dimension $d = 25$, signal-to-noise ratio $\beta = 5$, online learning.

Chapter 5

Concluding perspectives

5.1 Overview of the recurring themes

In this thesis we started from the motivation of understanding what happens *beyond the Gaussian approximation*, due to the initial observation from experiment fig. 1.3, that relying only on Gaussian statistics hurts performance considerably. In chapter 2 we discussed a synthetic model of data that showed how algorithmically hard it can be to distinguish a non-Gaussian distribution from pure Gaussian noise, requiring a quadratic number of samples to reach distinguishability in polynomial time. Then, in chapter 3 we considered neural network learning dynamics, finding a mechanism that allows to exploit the low-order structure in the data to speed up the learning of high-order directions. Finally, in chapter 4 we studied the related *independent component analysis* problem, extending our dynamical analysis to this unsupervised non-Gaussian setting, both in the large batch case of *FastICA* and the online regime of smoothed SGD. We then observed the application on ImageNet and developed the *subspace model* to explain the effectiveness of PCA pre-processing.

From the technical point of view, a transversal theme in the whole manuscript was how in high dimensions, higher-order correlations often correspond to flatter saddles for learning algorithms, which lead to increased hardness. Methods to speed-up the learning can often be interpreted as a way to handle flatness of the saddle point at initialization. In chapter 3 the correlation of latent variables can introduce a signal that changes the steepness of the saddle, and the idea for the *smoothing trick* in chapter 4 is to trad off compute power, adding an additional integral, to reduce the flatness of the saddle.

Another recurrent theme for the analysis of non-Gaussian distributions, was the use of the *likelihood ratio*, as a way to rephrase all the information contained in the input distribution into a function that acts as an implicit teacher, allowing to apply some of the theory from teacher-student models (even though there is often a fundamental hardness barrier that separates the two models as discussed in section 3.4.1.1).

From the modeling point of view, we can try here to formulate some general guidelines on *learning beyond the Gaussian approximation* that emerged in the manuscript.

On one hand, chapter 2 indicates that, in isolation (e.g. in the *spiked cumulant model*), higher-order cumulants are *hard to learn*, requiring a whole order of magnitude more samples to be efficiently learned (quadratic sample complexity) with respect to low-order correlations (e.g. linear sample complexity in spiked Wishart model). This corroborates the hypothesis of sequential learning: when there are no specific interactions, networks first learn low-order, then higher-order cumulants. On the other hand, chapters 3 and 4 indicate that in real data low and high order cumulants non-trivially interact and the sample complexity necessary to learn HOC directions are characterized by such

interdependence, that in our theoretical models took the form of correlations among auxiliary variables (in chapter 3) and overlaps with principal component subspace in chapter 4.

It is important to stress that the picture just described relates to *weak recovery*: once the few important directions have been found, the problem becomes effectively low-dimensional, and there is not a marked difference between low and high-order cumulants anymore. At this point, the dynamics are driven by the peculiar properties of the effective dynamical system in the weakly recovered subspace: the precise values of signal-to-noise ratios affects the dynamics more than the information exponent of the spiked directions (as discussed at the end of chapter 3).

We now overview future directions in the next section.

5.2 Perspectives

There are several questions that this thesis opens up, which can be grouped into three directions:

- on the side of the distributions, the main goal would be to analyze distributions as far from Gaussian as possible. In practice this would mean dropping all sub-Gaussianity requirements on the latent variables. A major difficulty of this problem would be that the *likelihood ratio* would become less and less useful, since it would not enjoy the regularity properties that allowed our analyses in chapter 3 and chapter 4. A possibility would then be to expand with respect to different *null* distributions.

- from the modeling point of view, the direction that naturally continues chapter 4 is to progressively extend the theoretical model to incorporate more and more features of what we observe happening on ImageNet. For instance, investigating what exactly in the data leads to the formation of localised, oriented filters.

- on the side of the architectures, promising directions consist in generalizing the analysis towards different networks, such as self-attention, or as we will briefly sketch in the next section, diffusion models.

5.2.1 Applications to diffusion models

Introduced only ten years ago diffusion models [180, 181] quickly revolutionized generative AI, reaching state-of-the-art performance for many tasks. More recently, a line of work is trying to develop a theoretical understanding of diffusion models, a non-exhaustive list of references is [182–189]. We will consider the setting of [182], modeling the diffusion process for $t \to x(t) \in \mathbb{R}^d$, with $x(0) = a \sim P_0$, the target distribution, that we want to learn how to sample from. The diffusion process has the following form:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = -x\mathrm{d}t + \mathrm{d}\mathcal{W}_t, \tag{5.1}$$

where $\mathcal{W}_t \in \mathbb{R}^d$ is a d dimensional Wiener process. The solution at time t can be written in distribution as:

$$x(t) = x(0)e^{-t} + \sqrt{1 - e^{-2t}}z \tag{5.2}$$

where $z \sim \mathcal{N}(0, \mathbb{1}_d)$. So, defining $\Delta_t = 1 - e^{-2t}$ The density at time t is given by:

$$P_t(x) = \int \mathrm{d}a\, P_0(a) \frac{1}{(2\pi\Delta_t)^{d/2}} \exp\left(-\frac{1}{2}\frac{(x - ae^{-t})^2}{\Delta_t}\right). \tag{5.3}$$

A key quantity that needs to be introduced is the *score* of P_t, that appears in the equation for the backward process. Knowing the score, or being able to approximate it, allows to revert the process and, starting from samples of a standard Gaussian it gives the recipe on how to transform them into samples of P_0. The score is defined as:

$$\mathcal{F}_i(x,t) = \frac{\partial \log P_t(x)}{\partial x_i} = -\frac{x_i - \mathbb{E}\left[a_i | x(t) = x\right] e^{-t}}{\Delta_t}, \tag{5.4}$$

where the last equality, called *Tweedie formula*, is at the core of the feasibility of diffusion models. It gives a recipe on how to approximate the score via empirical averages of the noised process. The objective then becomes learning $\mathcal{F}$. To do this it, one can build MSE for a collection of fixed time intervals, let us denote $\mathcal{S}_t^w(x)$ the approximated score that depends on weight w. The loss function at time t will be

$$\begin{aligned}
\mathcal{L}_t(w) =& \mathbb{E}_x \left[\left\| S_t^w(x) + \frac{x - \mathbb{E}\left[a | x(t) = x\right] e^{-t}}{\Delta_t} \right\|^2 \right] \\
\overset{(5.2)}{=}& \mathbb{E}_{a,z} \left[\left\| S_t^w(x) + \frac{a e^{-t} + \sqrt{\Delta} z - \mathbb{E}\left[a | x(t) = x\right] e^{-t}}{\Delta_t} \right\|^2 \right] \\
=& \mathbb{E}_{a,z} \left[\left\| S_t^w(x) + \frac{z}{\sqrt{\Delta_t}} \right\|^2 + \left\| \frac{a e^{-t} - \mathbb{E}\left[a | x(t) = x\right] e^{-t}}{\Delta_t} \right\|^2 + 2 \frac{z}{\sqrt{\Delta_t}} \cdot \frac{a e^{-t} - \mathbb{E}\left[a | x(t) = x\right] e^{-t}}{\Delta_t} \right],
\end{aligned}$$

where the double product with S_t^w does not appear because $\mathbb{E}_{x,a}[f(x)\left(a - \mathbb{E}\left[a | x(t) = x\right]\right)] = 0$. Note that the second and third addend are independent of w, hence the troublesome term $\mathbb{E}\left[a | x(t) = x\right]$ does not appear in the gradient and the effective loss function is:

$$L_t(w) = \mathbb{E}_{a,z} \left[\left\| S_t^w(a e^{-t} + \sqrt{\Delta_t} z) + \frac{z}{\sqrt{\Delta_t}} \right\|^2 \right] + C. \tag{5.5}$$

This quantity can be well approximated having just samples from P_0, which allow to estimate the integral over a.

One of the most intriguing properties of diffusion models, is that, once the exact average in eq. (5.5) is substituted with the empirical average over the samples of P_0, that becomes the exact expression of the score of a distribution that is a sum of deltas, each centered on one of the points in the dataset. Hence if S^w perfectly learns the task, it will lead to a diffusion process that samples from the initial dataset: only memorization and no generalization.

However, in practice diffusion models work, and in [186] the authors experimentally verify that in before converging to the perfect solution that memorizes, networks tend to learn a regularized version of the score, that generalizes well. The experimental predictions are verified also on a theoretical model involving a random feature student learning the score from a teacher, with Gaussian data. However theoretical study of the learning dynamics of the score, for a feature learning architecture is lacking. We propose to study the learning dynamics of a shallow student, on data models that diffuse between a standard Gaussian into a *spiked and mixed cumulant models*.

Let us first consider the case in which P_0 is the spiked cumulant model. We can choose the case $x(0) = hv + z$ with $h \sim \text{Rademacher}(1/2)$, v is the norm 1 spike and z is a $d-1$-standard Gaussian, in the space orthogonal to v, i.e. $z \sim \mathcal{N}(0, \mathbb{1}_d - vv^\top)$. Then from eq. (5.3), P_t has the following

expression:

$$P_t(x) = \frac{1}{(2\pi)^{(d-1)/2}} \exp\left(-\frac{1}{2} x_{\perp v}^\top x_{\perp v}\right) \frac{1}{2\left(2\pi\Delta_t\right)^{1/2}} \left[\exp\left(-\frac{1}{2}\frac{(x_v - e^{-t})^2}{\Delta_t}\right) + \exp\left(-\frac{1}{2}\frac{(x_v + e^{-t})^2}{\Delta_t}\right)\right]$$
$$= \frac{1}{(2\pi)^{(d-1)/2}} \exp\left(-\frac{1}{2} x_{\perp v}^\top x_{\perp v}\right) \frac{\exp\left(-\frac{1}{2}\frac{x_v^2 + e^{-2t}}{\Delta_t}\right)}{\left(2\pi\Delta_t\right)^{1/2}} \cosh\left(\frac{x_v e^{-t}}{\Delta_t}\right),$$

were $x = x \cdot vv + (x - x \cdot vv) = x_v v + x_{\perp v}$.

Hence the score is:

$$\mathcal{F}(x,t) = -x_{\perp v} - \frac{x_v}{\Delta_t} v + \frac{e^{-t}}{\Delta_t} v \tanh\left(\frac{x_v e^{-t}}{\Delta_t}\right)$$
$$= -x - \frac{e^{-t}}{\Delta_t} v \left(e^{-t} x_v - \tanh\left(\frac{x_v e^{-t}}{\Delta_t}\right)\right). \tag{5.6}$$

Then one can pick a student network of the score, with matched form:

$$S_w(x,t) = -x - w\sigma(w \cdot x) \tag{5.7}$$

and perform the analysis of the information exponent of the network dynamics using the setup from chapter 3. Notice that the first term in eq. (5.4) acts as a *skip connection*, which is relevant to be able to express the full $d-$dimensional score, and also "centers" the dynamics: now $w = 0$ leads to the score of a standard Gaussian distribution.

We can unpack the average over $a \sim P_0$ in eq. (5.5) inserting the likelihood ratio $L(x \cdot v) = 1 + c_4^L h_4(x \cdot v) + \dots$, we get the following expressions.

$$\mathcal{L}_t(w) = \mathbb{E}_{x,z}\left[||\frac{z}{\sqrt{\Delta_t}} - x - w\sigma(w \cdot x)||^2\right] + C$$
$$= \mathbb{E}_{x\sim\mathcal{N}(0,\mathbb{1}_d)}\left[L(v \cdot x)\left(\sigma^2(w \cdot x) + 2x \cdot w\sigma(w \cdot x) - \mathbb{E}_{z|x}\left[\frac{z}{\sqrt{\Delta_t}} \cdot w\sigma(w \cdot x)\right]\right)\right] + \tilde{C}$$

This formula is different from the ones that we saw in chapter 3: Δ_t depends on time and goes to 0 for $t \to 0$, this means that each term's contributions will strongly depend on the diffusion time t. Another difference is that each term depends quadratically on w and there is a potential pushing towards $w = 0$, which would be the optimal solution in the case of $L = 1$: a diffusion between two standard Gaussians. For these reasons this model offers interesting questions to be investigated, some of which are:

- what are the baselines of algorithmic complexity for this model?
- Does the use of Tweedie formula change the sample complexity compared to a direct supervised learning of the score?
- What is the sample complexity SGD needs to recover v, how does this depend on t?
- Are there phase transitions at some values of t, do they depend on the information exponent of L?
- Adding another spike in the covariance, does the phenomenology of learning the *mixed cumulant model* transfer to this case, with correlations of latent variables speeding up learning the score of the spike in the HOCs?

We are confident that the tools developed in this thesis will help with answering these questions.

5.3 Final remarks

In this thesis we had the occasion to examine the complex interplay between architectures, algorithms and data structure. We developed tools to analyse non-Gaussian input distributions and discussed the applications to real data. We contributed in clarifying the hardness of learning a single non-Gaussian direction, then analysed neural network dynamics and ICA in high dimensions, showing when algorithms struggle to learn HOCs and providing theoretical models that can explain how learning happens in realistic settings.

We note that, as in most other areas of theoretical machine learning, there is still a long way to go in order to bridge the gap with practice and we hope that our work will stimulate further exploration of the world of non-Gaussian distributions.

Bibliography

1. Hornik, K., Stinchcombe, M. & White, H. Multilayer feedforward networks are universal approximators. *Neural Networks* **2,** 359–366 (1989) (cit. on p. 4).
2. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems* **2,** 303–314 (1989) (cit. on p. 4).
3. Barron, A. R. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory* **39,** 930–945 (1993) (cit. on p. 4).
4. Kingma, D. & Ba, J. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations* (Dec. 2014) (cit. on p. 4).
5. Neyshabur, B., Tomioka, R. & Srebro, N. In Search of the Real Inductive Bias: On the Role of Implicit Regularization in Deep Learning (Dec. 2014) (cit. on p. 4).
6. Saad, D. & Solla, S. Exact Solution for On-Line Learning in Multilayer Neural Networks. *Phys. Rev. Lett.* **74,** 4337–4340 (1995) (cit. on pp. 4, 57).
7. Goldt, S., Advani, M., Saxe, A. M., Krzakala, F. & Zdeborová, L. Dynamics of stochastic gradient descent for two-layer neural networks in the teacher-student setup. *Advances in neural information processing systems* **32** (2019) (cit. on p. 4).
8. Veiga, R., Stephan, L., Loureiro, B., Krzakala, F. & Zdeborová, L. Phase diagram of Stochastic Gradient Descent in high-dimensional two-layer neural networks. *Advances in Neural Information Processing Systems* **35,** 23244–23255 (2022) (cit. on p. 4).
9. Arnaboldi, L., Stephan, L., Krzakala, F. & Loureiro, B. From high-dimensional & mean-field dynamics to dimensionless ODEs: A unifying approach to SGD in two-layers networks. *arXiv:2302.05882* (2023) (cit. on p. 4).
10. Balcan, M.-F., Blum, A. & Vempala, S. Kernels as features: On kernels, margins, and low-dimensional mappings. *Machine Learning* **65,** 79–94 (2006) (cit. on pp. 5, 23).
11. Cho, Y. & Saul, L. Kernel methods for deep learning. *Advances in neural information processing systems* **22** (2009) (cit. on p. 5).
12. Rahimi, A. & Recht, B. *Random features for large-scale kernel machines* in *Advances in neural information processing systems* (2008), 1177–1184 (cit. on pp. 5, 23, 28).
13. Mei, S. & Montanari, A. The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics* **75,** 667–766 (2022) (cit. on pp. 5, 23, 46, 61).
14. Mei, S., Misiakiewicz, T. & Montanari, A. Generalization error of random feature and kernel methods: Hypercontractivity and kernel matrix concentration. *Applied and Computational Harmonic Analysis* **59,** 3–84 (2022) (cit. on pp. 5, 23, 24).

15. Arnaboldi, L. *et al. Online Learning and Information Exponents: The Importance of Batch size and Time/Complexity Tradeoffs* in *ICML* (2024) (cit. on pp. 5, 6, 99).

16. Ben Arous, G., Gheissari, R. & Jagannath, A. Online Stochastic Gradient Descent on Non-Convex Losses from High-Dimensional Inference. *J. Mach. Learn. Res.* **22** (2021) (cit. on pp. 5, 51–56, 58, 65–71, 77, 81, 84, 85, 101, 103).

17. Biroli, G., Cammarota, C. & Ricci-Tersenghi, F. How to iron out rough landscapes and get optimal performances: averaged gradient descent and its application to tensor PCA. *Journal of Physics A: Mathematical and Theoretical* **53,** 174003 (2020) (cit. on pp. 6, 84).

18. Damian, A., Nichani, E., Ge, R. & Lee, J. D. *Smoothing the Landscape Boosts the Signal for SGD: Optimal Sample Complexity for Learning Single Index Models* in *Thirty-seventh Conference on Neural Information Processing Systems* (2023) (cit. on pp. 6, 10, 51, 77, 84–86, 100–103, 108, 109).

19. Damian, A., Lee, J. & Soltanolkotabi, M. *Neural networks can learn representations with gradient descent* in *Conference on Learning Theory* (2022), 5413–5452 (cit. on pp. 6, 57, 82).

20. Dandi, Y., Krzakala, F., Loureiro, B., Pesce, L. & Stephan, L. Learning Two-Layer Neural Networks, One (Giant) Step at a Time. *arXiv:2305.18270* (2023) (cit. on pp. 6, 53, 57, 59, 61).

21. Damian, A., Pillaud-Vivien, L., Lee, J. D. & Bruna, J. The Computational Complexity of Learning Gaussian Single-Index Models. *arXiv preprint arXiv:2403.05529* (2024) (cit. on pp. 6, 14, 59, 61, 81, 83, 92).

22. Dandi, Y. *et al.* The Benefits of Reusing Batches for Gradient Descent in Two-Layer Networks: Breaking the Curse of Information and Leap Exponents. *arXiv preprint arXiv:2402.03220* (2024) (cit. on pp. 6, 59, 61, 92).

23. Arnaboldi, L., Dandi, Y., Krzakala, F., Pesce, L. & Stephan, L. *Repetita Iuvant: Data Repetition Allows SGD to Learn High-Dimensional Multi-Index Functions* 2025. arXiv: `2405.15459 [stat.ML]` (cit. on pp. 6, 59).

24. Tsiolis, K. C., Mousavi-Hosseini, A. & Erdogdu, M. A. *From Information to Generative Exponent: Learning Rate Induces Phase Transitions in SGD* in *High-dimensional Learning Dynamics 2025* (2025) (cit. on p. 6).

25. Abbé, E., Boix-Adsera, E., Brennan, M. S., Bresler, G. & Nagaraj, D. The staircase property: How hierarchical structure can guide deep learning. *Advances in Neural Information Processing Systems* **34,** 26989–27002 (2021) (cit. on pp. 6, 57).

26. Abbé, E., Adsera, E. B. & Misiakiewicz, T. *The merged-staircase property: a necessary and nearly sufficient condition for sgd learning of sparse functions on two-layer neural networks* in *Conference on Learning Theory* (2022), 4782–4887 (cit. on pp. 6, 57).

27. Bietti, A., Bruna, J. & Pillaud-Vivien, L. On learning gaussian multi-index models with gradient flow. *arXiv preprint arXiv:2310.19793* (2023) (cit. on pp. 6, 57).

28. Refinetti, M., Ingrosso, A. & Goldt, S. *Neural networks trained with SGD learn distributions of increasing complexity* in *International Conference on Machine Learning* (2023), 28843–28863 (cit. on pp. 7, 12, 48, 50, 75, 90).

29. Dudeja, R. & Hsu, D. Statistical-computational trade-offs in tensor PCA and related problems via communication complexity. *The Annals of Statistics* **52,** 131–156 (2024) (cit. on pp. 8, 13, 14, 20, 79).

30. Szekely, E., Bardone, L., Gerace, F. & Goldt, S. *Learning from higher-order correlations, efficiently: hypothesis tests, random features, and neural networks* in *The Thirty-eighth Annual Conference on Neural Information Processing Systems* (2024) (cit. on pp. 9, 79, 88, 92, 107).

31. Bardone, L. & Goldt, S. *Sliding Down the Stairs: How Correlated Latent Variables Accelerate Learning with Neural Networks* in *Proceedings of the 41st International Conference on Machine Learning* (eds Salakhutdinov, R. *et al.*) **235** (PMLR, 2024), 3024–3045 (cit. on pp. 10, 90).

32. Gabor, D. Theory of communication. Part 1: The analysis of information. *Journal of the Institution of Electrical Engineers-part III: radio and communication engineering* **93,** 429–441 (1946) (cit. on pp. 10, 75, 76).

33. Ricci, F., Bardone, L. & Goldt, S. Feature learning from non-Gaussian inputs: the case of Independent Component Analysis in high dimensions. *arXiv preprint arXiv:2503.23896* (2025) (cit. on pp. 10, 89, 109).

34. Ricci, F., Bardone, L. & Goldt, S. *Reduce and Conquer: Independent Component Analysis at linear sample complexity* in *High-dimensional Learning Dynamics 2025* (2025) (cit. on p. 10).

35. Jacot, A., Gabriel, F. & Hongler, C. *Neural Tangent Kernel: Convergence and Generalization in Neural Networks* in *Advances in Neural Information Processing Systems* (eds Bengio, S. *et al.*) **31** (Curran Associates, Inc., 2018) (cit. on pp. 13, 14, 60).

36. Arora, S. *et al.* On exact computation with an infinitely wide neural net. *Advances in neural information processing systems* **32** (2019) (cit. on pp. 13, 14, 60).

37. Geiger, M., Spigler, S., Jacot, A. & Wyart, M. Disentangling feature and lazy training in deep neural networks. *Journal of Statistical Mechanics: Theory and Experiment* **2020,** 113301 (2020) (cit. on pp. 13, 14).

38. Barak, B. *et al.* A nearly tight sum-of-squares lower bound for the planted clique problem. *SIAM Journal on Computing* **48,** 687–735 (2 2019) (cit. on pp. 13, 19, 26, 48).

39. Hopkins, S. B. *et al. The power of sum-of-squares for detecting hidden structures* in *IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)* (2017), 720–731 (cit. on pp. 13, 19, 26, 48).

40. Hopkins, S. B. & Steurer, D. Bayesian estimation from few samples: com- munity detection and related problems. *arXiv:1710.00264* (2017) (cit. on pp. 13, 19, 26, 48).

41. Hopkins, S. *Statistical inference and the sum of squares method* PhD thesis (Cornell University, 2018) (cit. on pp. 13, 19, 26, 48).

42. Baik, J., Ben Arous, G. & Péché, S. Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices. *The Annals of Probability* **33,** 1643–1697 (2005) (cit. on pp. 13, 14, 32).

43. Potters, M. & Bouchaud, J.-P. *A First Course in Random Matrix Theory: for Physicists, Engineers and Data Scientists* (Cambridge University Press, 2020) (cit. on pp. 13, 14).

44. Diakonikolas, I., Kane, D. M. & Stewart, A. *Statistical Query Lower Bounds for Robust Estimation of High-Dimensional Gaussians and Gaussian Mixtures* in *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)* (2017), 73–84 (cit. on pp. 13, 14, 79).

45. Ingrosso, A. & Goldt, S. Data-driven emergence of convolutional structure in neural networks. *Proceedings of the National Academy of Sciences* **119,** e2201854119 (2022) (cit. on pp. 13, 25, 26, 48, 50, 89, 90).

46. Paul, D. Asymptotics of sample eigenstructure for a large dimensional spiked covariance model. *Statistica Sinica,* 1617–1642 (2007) (cit. on p. 13).
47. Berthet, Q. & Rigollet, P. Optimal detection of sparse principal components in high dimension. *The Annals of Statistics* (2012) (cit. on p. 13).
48. Berthet, Q. & Rigollet, P. *Complexity theoretic lower bounds for sparse principal component detection* in *Conference on Learning Theory (COLT)* (2013) (cit. on p. 13).
49. Lesieur, T., Krzakala, F. & Zdeborová, L. *Phase transitions in sparse PCA* in *2015 IEEE International Symposium on Information Theory (ISIT)* (2015), 1635–1639 (cit. on p. 13).
50. Lesieur, T., Krzakala, F. & Zdeborová, L. *MMSE of probabilistic low-rank matrix estimation: Universality with respect to the output channel* in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)* (2015), 680–687 (cit. on p. 13).
51. Perry, A., Wein, A. S., Bandeira, A. S. & Moitra, A. Optimality and sub-optimality of PCA for spiked random matrices and synchronization. *arXiv:1609.05573* (2016) (cit. on pp. 13, 31).
52. Krzakala, F., Xu, J. & Zdeborová, L. *Mutual information in rank-one matrix estimation* in *2016 IEEE Information Theory Workshop (ITW)* (2016), 71–75 (cit. on p. 13).
53. Dia, M., Macris, N., Krzakala, F., Lesieur, T., Zdeborová, L. *et al.* Mutual information for symmetric rank-one matrix estimation: A proof of the replica formula. *Advances in Neural Information Processing Systems* **29** (2016) (cit. on p. 13).
54. Miolane, L. Phase transitions in spiked matrix estimation: information-theoretic analysis. *arXiv:1806.04343* (2018) (cit. on p. 13).
55. Lelarge, M. & Miolane, L. Fundamental limits of symmetric low-rank matrix estimation. *Probability Theory and Related Fields* **173,** 859–929 (2019) (cit. on p. 13).
56. El Alaoui, A., Krzakala, F. & Jordan, M. Fundamental limits of detection in the spiked Wigner model. *The Annals of Statistics* **48,** 863–885 (2020) (cit. on p. 13).
57. Richard, E. & Montanari, A. A statistical model for tensor PCA. *Advances in neural information processing systems* **27** (2014) (cit. on pp. 13, 22, 60, 79).
58. Hopkins, S. B., Shi, J. & Steurer, D. *Tensor principal component analysis via sum-of-square proofs* in *Conference on Learning Theory* (2015), 956–1006 (cit. on p. 13).
59. Montanari, A., Reichman, D. & Zeitouni, O. On the limitation of spectral methods: From the gaussian hidden clique problem to rank-one perturbations of gaussian tensors. *Advances in Neural Information Processing Systems* **28** (2015) (cit. on pp. 13, 31).
60. Perry, A., Wein, A. S. & Bandeira, A. S. Statistical limits of spiked tensor models. *arXiv:1612.07728* (2016) (cit. on pp. 13, 22).
61. Kim, C., Bandeira, A. S. & Goemans, M. X. *Community detection in hypergraphs, spiked tensor models, and sum-of-squares* in *2017 International Conference on Sampling Theory and Applications (SampTA)* (2017), 124–128 (cit. on p. 13).
62. Lesieur, T., Miolane, L., Lelarge, M., Krzakala, F. & Zdeborová, L. *Statistical and computational phase transitions in spiked tensor estimation* in *2017 IEEE International Symposium on Information Theory (ISIT)* (2017), 511–515 (cit. on pp. 13, 22).
63. Arous, G. B., Gheissari, R. & Jagannath, A. Algorithmic thresholds for tensor PCA. *The Annals of Probability* **48,** 2052–2087 (2020) (cit. on p. 13).

64. Jagannath, A., Lopatto, P. & Miolane, L. Statistical thresholds for tensor PCA. *The Annals of Applied Probability* **30,** 1910–1933 (2020) (cit. on p. 13).

65. Niles-Weed, J. & Rigollet, P. Estimation of wasserstein distances in the spiked transport model. *Bernoulli* **28,** 2663–2688 (2022) (cit. on p. 13).

66. Zdeborová, L. & Krzakala, F. Statistical physics of inference: Thresholds and algorithms. *Advances in Physics* **65,** 453–552 (2016) (cit. on p. 13).

67. Lesieur, T., Krzakala, F. & Zdeborová, L. Constrained low-rank matrix estimation: Phase transitions, approximate message passing and applications. *Journal of Statistical Mechanics: Theory and Experiment* **2017,** 073403 (2017) (cit. on p. 13).

68. Bandeira, A. S., Perry, A. & Wein, A. S. Notes on computational-to-statistical gaps: predictions using statistical physics. *Portugaliae mathematica* **75,** 159–186 (2018) (cit. on p. 13).

69. Kunisky, D., Wein, A. S. & Bandeira, A. S. *Notes on computational hardness of hypothesis testing: Predictions using the low-degree likelihood ratio* in *ISAAC Congress (International Society for Analysis, its Applications and Computation)* (2019), 1–50 (cit. on pp. 13, 19, 31, 32, 45, 46, 62).

70. Wang, C. & Lu, Y. M. *The Scaling Limit of High-Dimensional Online Independent Component Analysis* in *Advances in neural information processing systems* **31** (2017) (cit. on pp. 13, 14).

71. Hyvärinen, A. & Oja, E. Independent component analysis: algorithms and applications. *Neural networks* **13,** 411–430 (2000) (cit. on pp. 13, 75, 76, 78, 79, 85, 93).

72. Blanchard, G., Kawanabe, M., Sugiyama, M., Spokoiny, V. & Müller, K.-R. In Search of Non-Gaussian Components of a High-Dimensional Distribution. *Journal of Machine Learning Research* **7,** 247–282 (2006) (cit. on p. 13).

73. Bean, D. M. *Non-Gaussian component analysis* (University of California, Berkeley, 2014) (cit. on p. 14).

74. Vempala, S. S. & Xiao, Y. *Structure from Local Optima: Learning Subspace Juntas via Higher Order PCA* 2012. arXiv: `1108.3329 [cs.CC]` (cit. on p. 14).

75. Tan, Y. S. & Vershynin, R. *Polynomial Time and Sample Complexity for Non-Gaussian Component Analysis: Spectral Methods* in *Proceedings of the 31st Conference On Learning Theory* (eds Bubeck, S., Perchet, V. & Rigollet, P.) **75** (PMLR, July 2018), 498–534 (cit. on p. 14).

76. Goyal, N. & Shetty, A. Non-Gaussian component analysis using entropy methods. *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing* (2018) (cit. on p. 14).

77. Mao, C. & Wein, A. S. *Optimal Spectral Recovery of a Planted Vector in a Subspace* 2022. arXiv: `2105.15081 [math.ST]` (cit. on p. 14).

78. Diakonikolas, I., Kane, D., Ren, L. & Sun, Y. *SQ Lower Bounds for Non-Gaussian Component Analysis with Weaker Assumptions* in *Thirty-seventh Conference on Neural Information Processing Systems* (2023) (cit. on p. 14).

79. Brennan, M. S., Bresler, G., Hopkins, S., Li, J. & Schramm, T. *Statistical Query Algorithms and Low Degree Tests Are Almost Equivalent* in *Proceedings of Thirty Fourth Conference on Learning Theory* (eds Belkin, M. & Kpotufe, S.) **134** (PMLR, Aug. 2021), 774–774 (cit. on pp. 14, 79).

80. Li, Y. & Yuan, Y. Convergence analysis of two-layer neural networks with relu activation. *Advances in neural information processing systems* **30** (2017) (cit. on pp. 14, 60).

81. Du, S., Zhai, X., Poczos, B. & Singh, A. *Gradient Descent Provably Optimizes Over-parameterized Neural Networks* in *International Conference on Learning Representations* (2019) (cit. on p. 14).
82. Li, Y. & Liang, Y. *Learning Overparameterized Neural Networks via Stochastic Gradient Descent on Structured Data* in *Advances in Neural Information Processing Systems 31* (2018) (cit. on pp. 14, 60).
83. Allen-Zhu, Z., Li, Y. & Song, Z. *A convergence theory for deep learning via over-parameterization* in *International Conference on Machine Learning* (2019), 242–252 (cit. on p. 14).
84. Bordelon, B., Canatar, A. & Pehlevan, C. *Spectrum dependent learning curves in kernel regression and wide neural networks* in *International Conference on Machine Learning* (2020), 1024–1034 (cit. on pp. 14, 60).
85. Canatar, A., Bordelon, B. & Pehlevan, C. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature communications* **12,** 2914 (2021) (cit. on p. 14).
86. Nguyen, Q., Mondelli, M. & Montufar, G. F. *Tight bounds on the smallest eigenvalue of the neural tangent kernel for deep relu networks* in *International Conference on Machine Learning* (2021), 8119–8129 (cit. on p. 14).
87. Bach, F. Breaking the curse of dimensionality with convex neural networks. *The Journal of Machine Learning Research* **18,** 629–681 (2017) (cit. on p. 14).
88. Ghorbani, B., Mei, S., Misiakiewicz, T. & Montanari, A. *Limitations of Lazy Training of Two-layers Neural Network* in *Advances in Neural Information Processing Systems* **32** (2019), 9111–9121 (cit. on pp. 14, 60).
89. Ghorbani, B., Mei, S., Misiakiewicz, T. & Montanari, A. *When do neural networks outperform kernel methods?* in *Advances in Neural Information Processing Systems* **33** (2020) (cit. on pp. 14, 60).
90. Chizat, L. & Bach, F. *Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss* in *Conference on Learning Theory* (2020), 1305–1338 (cit. on p. 14).
91. Daniely, A. & Malach, E. Learning parities with neural networks. *Advances in Neural Information Processing Systems* **33,** 20356–20365 (2020) (cit. on p. 14).
92. Paccolat, J., Petrini, L., Geiger, M., Tyloo, K. & Wyart, M. Geometric compression of invariant manifolds in neural networks. *Journal of Statistical Mechanics: Theory and Experiment* **2021,** 044001 (2021) (cit. on p. 14).
93. Yehudai, G. & Shamir, O. *On the Power and Limitations of Random Features for Understanding Neural Networks* in *Advances in Neural Information Processing Systems* **32** (2019), 6598–6608 (cit. on p. 14).
94. Refinetti, M., Goldt, S., Krzakala, F. & Zdeborová, L. *Classifying high-dimensional gaussian mixtures: Where kernel methods fail and neural networks succeed* in *International Conference on Machine Learning* (2021), 8936–8947 (cit. on pp. 14, 56).
95. Baldi, P. & Chauvin, Y. Temporal evolution of generalization during learning in linear networks. *Neural Computation* **3,** 589–603 (1991) (cit. on p. 15).
96. Le Cun, Y., Kanter, I. & Solla, S. A. Eigenvalues of covariance matrices: Application to neural-network learning. *Physical Review Letters* **66,** 2396 (1991) (cit. on p. 15).

97. Saxe, A. M., McClelland, J. L. & Ganguli, S. *Exact solutions to the nonlinear dynamics of learning in deep linear neural networks* in *ICLR* (2014) (cit. on p. 15).

98. Advani, M. S., Saxe, A. M. & Sompolinsky, H. High-dimensional dynamics of generalization error in neural networks. *Neural Networks* **132,** 428–446 (2020) (cit. on p. 15).

99. Vershynin, R. *High-dimensional probability: An introduction with applications in data science* (Cambridge University Press, 2018) (cit. on pp. 18, 93).

100. Kesten, H. & Stigum, B. P. Additional limit theorems for indecomposable multidimensional Galton-Watson processes. *The Annals of Mathematical Statistics* **37,** 1463–1481 (1966) (cit. on p. 19).

101. Decelle, A., Krzakala, F., Moore, C. & Zdeborová, L. Inference and phase transitions in the detection of modules in sparse networks. *Physical Review Letters* **107,** 065701 (2011) (cit. on p. 19).

102. Decelle, A., Krzakala, F., Moore, C. & Zdeborová, L. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical review E* **84,** 066106 (2011) (cit. on p. 19).

103. Rahimi, A. & Recht, B. *Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning* in *Advances in neural information processing systems* (2009), 1313–1320 (cit. on pp. 23, 28).

104. Ghorbani, Mei, S., Misiakiewicz, T. & Montanari, A. "Linearized two-layers neural networks in high dimension." *Annals of Statistics* **49,** 1029–1054 (2021) (cit. on p. 23).

105. Xiao, L., Hu, H., Misiakiewicz, T., Lu, Y. & Pennington, J. Precise Learning Curves and Higher-Order Scalings for Dot-product Kernel Regression. *Advances in Neural Information Processing Systems* **35,** 4558–4570 (2022) (cit. on pp. 23, 24, 60).

106. Misiakiewicz, T. & Montanari, A. Six lectures on linearized neural networks. *arXiv:2308.13431* (2023) (cit. on pp. 23, 24).

107. Loureiro, B. *et al.* Learning gaussian mixtures with generalized linear models: Precise asymptotics in high-dimensions. *Advances in Neural Information Processing Systems* **34,** 10144–10157 (2021) (cit. on pp. 23, 46, 47).

108. Goldt, S., Mézard, M., Krzakala, F. & Zdeborová, L. Modeling the influence of data structure on learning in neural networks: The hidden manifold model. *Physical Review X* **10,** 041044 (2020) (cit. on pp. 23, 24, 46, 61).

109. Gerace, F., Loureiro, B., Krzakala, F., Mézard, M. & Zdeborová, L. *Generalisation error in learning with random features and the hidden manifold model* in *International Conference on Machine Learning* (2020), 3452–3462 (cit. on pp. 23, 46, 47, 61).

110. Hu, H. & Lu, Y. M. Universality laws for high-dimensional learning with random features. *IEEE Transactions on Information Theory* **69,** 1932–1964 (2022) (cit. on pp. 23, 46, 61).

111. Goldt, S. *et al. The gaussian equivalence of generative models for learning with shallow neural networks* in *Mathematical and Scientific Machine Learning* (2022), 426–471 (cit. on pp. 23, 24, 46, 61).

112. Dietrich, R., Opper, M. & Sompolinsky, H. Statistical mechanics of support vector networks. *Physical review letters* **82,** 2975 (1999) (cit. on pp. 24, 47, 60).

113. Gardner, E. & Derrida, B. Three unfinished works on the optimal storage capacity of networks. *Journal of Physics A: Mathematical and General* **22,** 1983 (1989) (cit. on p. 24).

114. Cui, H., Krzakala, F. & Zdeborova, L. *Bayes-optimal learning of deep random networks of extensive-width* in *International Conference on Machine Learning* (2023), 6468–6521 (cit. on p. 24).

115. Camilli, F., Tieplova, D. & Barbier, J. Fundamental limits of overparametrized shallow neural networks for supervised learning. *arXiv preprint arXiv:2307.05635* (2023) (cit. on p. 24).

116. Hu, H. & Lu, Y. M. Sharp asymptotics of kernel ridge regression beyond the linear regime. *arXiv:2205.06798* (2022) (cit. on p. 24).

117. Bell, A. & Sejnowski, T. J. *Edges are the 'Independent Components' of Natural Scenes.* in *Advances in Neural Information Processing Systems* (eds Mozer, M., Jordan, M. & Petsche, T.) **9** (MIT Press, 1996) (cit. on pp. 25, 75).

118. Chizat, L., Oyallon, E. & Bach, F. *On Lazy Training in Differentiable Programming* in *Advances in Neural Information Processing Systems* (eds Wallach, H. *et al.*) **32** (Curran Associates, Inc., 2019) (cit. on pp. 25, 28, 60).

119. Kolda, T. G. & Bader, B. W. Tensor Decompositions and Applications. *SIAM Review* **51,** 455–500 (2009) (cit. on p. 25).

120. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **12,** 2825–2830 (2011) (cit. on pp. 28, 78).

121. Kossaifi, J., Panagakis, Y., Anandkumar, A. & Pantic, M. TensorLy: Tensor Learning in Python. *Journal of Machine Learning Research (JMLR)* **20** (2019) (cit. on p. 29).

122. Banks, J., Moore, C., Neeman, J. & Netrapalli, P. *Information-theoretic thresholds for community detection in sparse networks* in *Conference on Learning Theory* (2016), 383–416 (cit. on p. 31).

123. Bandeira, A. S., Kunisky, D. & Wein, A. S. *Computational Hardness of Certifying Bounds on Constrained PCA Problems* in *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)* (ed Vidick, T.) **151** (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2020), 78:1–78:29 (cit. on pp. 32, 33, 132).

124. McCullagh, P. *Tensor methods in statistics* (Courier Dover Publications, 2018) (cit. on pp. 40, 129, 133).

125. Merger, C. *et al.* Learning Interacting Theories from Data. *arXiv:2304.00599* (2023) (cit. on p. 48).

126. Belrose, N., Pope, Q., Quirke, L., Mallen, A. & Fern, X. Neural Networks Learn Statistics of Increasing Complexity. *arXiv:2402.04362* (2024) (cit. on pp. 48, 50, 90).

127. Nestler, S., Helias, M. & Gilson, M. Statistical temporal pattern extraction by neuronal architecture. *Physical Review Research* **5,** 033177 (2023) (cit. on p. 50).

128. Ben Arous, G., Gheissari, R. & Jagannath, A. *High-dimensional limit theorems for SGD: Effective dynamics and critical scaling* in *Advances in Neural Information Processing Systems* (eds Koyejo, S. *et al.*) **35** (Curran Associates, Inc., 2022), 25349–25362 (cit. on p. 56).

129. Saad, D. & Solla, S. On-line learning in soft committee machines. *Phys. Rev. E* **52,** 4225–4243 (1995) (cit. on p. 57).

130. Kalimeris, D. *et al. SGD on neural networks learns functions of increasing complexity* in *Advances in neural information processing systems* **33** (2019) (cit. on p. 57).

131. Abbé, E., Adserà, E. B. & Misiakiewicz, T. *SGD learning on neural networks: leap complexity and saddle-to-saddle dynamics* in *Proceedings of Thirty Sixth Conference on Learning Theory* (eds Neu, G. & Rosasco, L.) **195** (PMLR, 2023), 2552–2623 (cit. on pp. 57, 81).

132. Jacot, A., Ged, F., Şimşek, B., Hongler, C. & Gabriel, F. Saddle-to-saddle dynamics in deep linear networks: Small initialization training, symmetry, and sparsity. *arXiv preprint arXiv:2106.15933* (2021) (cit. on p. 57).

133. Boursier, E., Pillaud-Vivien, L. & Flammarion, N. Gradient flow dynamics of shallow relu networks for square loss and orthogonal inputs. *Advances in Neural Information Processing Systems* **35,** 20105–20118 (2022) (cit. on p. 57).

134. Ba, J. *et al.* High-dimensional asymptotics of feature learning: How one gradient step improves the representation. *Advances in Neural Information Processing Systems* **35,** 37932–37946 (2022) (cit. on pp. 57, 61, 82).

135. Berthier, R., Montanari, A. & Zhou, K. Learning time-scales in two-layers neural networks. *arXiv:2303.00055* (2023) (cit. on pp. 58, 73).

136. Bietti, A. & Bach, F. *Deep Equals Shallow for ReLU Networks in Kernel Regimes* in *International Conference on Learning Representations* (2021) (cit. on p. 59).

137. Mousavi-Hosseini, A., Wu, D., Suzuki, T. & Erdogdu, M. A. Gradient-based feature learning under structured data. *Advances in Neural Information Processing Systems* **36** (2024) (cit. on p. 59).

138. Spigler, S., Geiger, M. & Wyart, M. Asymptotic learning curves of kernel methods: empirical data versus teacher–student paradigm. *Journal of Statistical Mechanics: Theory and Experiment* **2020,** 124001 (2020) (cit. on p. 60).

139. Cui, H., Krzakala, F. & Zdeborová, L. Optimal learning of deep random networks of extensive-width. *arXiv:2302.00375* (2023) (cit. on p. 60).

140. Baik, J., Arous, G. B. & Péché, S. Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices. *The Annals of Probability* **33,** 1643–1697 (2005) (cit. on p. 60).

141. Anandkumar, A., Ge, R., Hsu, D., Kakade, S. M. & Telgarsky, M. Tensor decompositions for learning latent variable models. *Journal of machine learning research* **15,** 2773–2832 (2014) (cit. on pp. 60, 78).

142. Sarao Mannelli, S. *et al.* Marvels and pitfalls of the langevin algorithm in noisy high-dimensional inference. *Physical Review X* **10,** 011057 (2020) (cit. on p. 60).

143. Sarao Mannelli, S., Krzakala, F., Urbani, P. & Zdeborova, L. *Passed & spurious: Descent algorithms and local minima in spiked matrix-tensor models* in *international conference on machine learning* (2019), 4333–4342 (cit. on p. 60).

144. Sarao Mannelli, S., Biroli, G., Cammarota, C., Krzakala, F. & Zdeborová, L. Who is afraid of big bad minima? analysis of gradient-flow in spiked matrix-tensor models. *Advances in Neural Information Processing Systems* **32** (2019) (cit. on p. 60).

145. Pesce, L., Krzakala, F., Loureiro, B. & Stephan, L. Are Gaussian data all you need? Extents and limits of universality in high-dimensional generalized linear estimation. *arXiv:2302.08923* (2023) (cit. on p. 61).

146. Fischer, K. *et al.* Decomposing neural networks as mappings of correlation functions. *Physical Review Research* **4,** 043143 (2022) (cit. on p. 61).

147. Szekely, E., Bardone, L., Gerace, F. & Goldt, S. Learning from higher-order cumulants: hypothesis tests, random features, and neural networks. *arXiv:2312.xxxxx* (2023) (cit. on pp. 62, 64).

148. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* **25** (2012) (cit. on pp. 72, 75).

149. Huang, G., Liu, Z., van der Maaten, L. & Weinberger, K. Q. *Densely Connected Convolutional Networks* in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017* (IEEE Computer Society, 2017), 2261–2269 (cit. on pp. 72, 91).

150. He, K., Zhang, X., Ren, S. & Sun, J. *Deep Residual Learning for Image Recognition* in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016* (IEEE Computer Society, 2016), 770–778 (cit. on pp. 72, 91).

151. Deng, J. *et al. Imagenet: A large-scale hierarchical image database* in *2009 IEEE conference on computer vision and pattern recognition* (2009), 248–255 (cit. on pp. 72, 75, 87).

152. Riegler, P. & Biehl, M. On-line backpropagation in two-layered neural networks. *Journal of Physics A: Mathematical and General* **28,** L507 (1995) (cit. on p. 73).

153. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521,** 436–444 (2015) (cit. on p. 75).

154. Lindsey, J., Ocko, S. A., Ganguli, S. & Deny, S. *The effects of neural resource constraints on early visual representations* in *International Conference on Learning Representations* (2019) (cit. on p. 75).

155. Guth, F. & Ménard, B. *On the universality of neural encodings in CNNs* in *ICLR 2024 Workshop on Representational Alignment* (2024) (cit. on p. 75).

156. Hyvärinen, A., Hurri, J. & Hoyer, P. O. *Natural image statistics: A probabilistic approach to early computational vision.* (Springer Science & Business Media, 2009) (cit. on pp. 75, 77).

157. Comon, P. Independent component analysis, a new concept? *Signal processing* **36,** 287–314 (1994) (cit. on p. 75).

158. Bell, A. J. & Sejnowski, T. J. An information-maximization approach to blind separation and blind deconvolution. *Neural computation* **7,** 1129–1159 (1995) (cit. on p. 75).

159. Olshausen, B. A. & Field, D. J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **381,** 607–609 (1996) (cit. on p. 75).

160. Hyvarinen, A. Fast and robust fixed-point algorithms for independent component analysis. *IEEE transactions on Neural Networks* **10,** 626–634 (1999) (cit. on p. 76).

161. Zarzoso, V., Comon, P. & Kallel, M. *How fast is FastICA?* in *2006 14th European Signal Processing Conference* (2006), 1–5 (cit. on pp. 76, 83).

162. Auddy, A. & Yuan, M. Large dimensional independent component analysis: Statistical optimality and computational tractability. *to appear in Annals of Statistics* (2024) (cit. on pp. 76, 79, 83, 85, 88).

163. Bottou, L. in *Summer School on Machine Learning* 146–168 (Springer, 2003) (cit. on p. 76).

164. Bottou, L. & Le Cun, Y. Large scale online learning. *Advances in neural information processing systems* **16** (2003) (cit. on p. 76).

165. Dandi, Y., Krzakala, F., Loureiro, B., Pesce, L. & Stephan, L. How Two-Layer Neural Networks Learn, One (Giant) Step at a Time. *Journal of Machine Learning Research* **25,** 1–65 (2024) (cit. on pp. 77, 81–83, 89, 97, 98).

166. Arora, S., Ge, R., Moitra, A. & Sachdeva, S. Provable ICA with unknown Gaussian noise, with implications for Gaussian mixtures and autoencoders. *Advances in Neural Information Processing Systems* **25** (2012) (cit. on p. 78).

167. Voss, J. R., Belkin, M. & Rademacher, L. A pseudo-euclidean iteration for optimal recovery in noisy ICA. *Advances in neural information processing systems* **28** (2015) (cit. on p. 78).

168. Bandeira, A., Singer, A. & Strohmer, T. Mathematics of data science. *Book draft available at https://people. math. ethz. ch/abandeira/BandeiraSingerStrohmer-MDS-draft. pdf* **4** (2020) (cit. on pp. 89, 105).

169. Lufkin, L., Saxe, A. & Grant, E. *Nonlinear dynamics of localization in neural receptive fields* in *Thirty-eighth Conference on Neural Information Processing Systems* (2024) (cit. on p. 89).

170. Kögler, K., Shevchenko, A., Hassani, H. & Mondelli, M. *Compression of Structured Data with Autoencoders: Provable Benefit of Nonlinearities and Depth* in *International Conference on Machine Learning* (2024), 24964–25015 (cit. on p. 90).

171. Rende, R., Gerace, F., Laio, A. & Goldt, S. *A distributional simplicity bias in the learning dynamics of transformers* in *The Thirty-eighth Annual Conference on Neural Information Processing Systems* (2024) (cit. on p. 90).

172. Gur-Ari, G., Roberts, D. A. & Dyer, E. Gradient descent happens in a tiny subspace. *arXiv preprint arXiv:1812.04754* (2018) (cit. on p. 90).

173. Larsen, B. W., Fort, S., Becker, N. & Ganguli, S. How many degrees of freedom do we need to train deep networks: a loss landscape perspective. *arXiv preprint arXiv:2107.05802* (2021) (cit. on p. 90).

174. Sagun, L., Bottou, L. & LeCun, Y. Eigenvalues of the hessian in deep learning: Singularity and beyond. *arXiv preprint arXiv:1611.07476* (2016) (cit. on p. 90).

175. Sagun, L., Evci, U., Guney, V. U., Dauphin, Y. & Bottou, L. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454* (2017) (cit. on p. 90).

176. Papyan, V. Traces of class/cross-class structure pervade deep learning spectra. *Journal of Machine Learning Research* **21,** 1–64 (2020) (cit. on p. 90).

177. Hyvärinen, A., Karhunen, J. & Oja, E. *Independent Component Analysis* eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/0471221317.fmatter_indsub` (John Wiley & Sons, Ltd, 2001) (cit. on p. 93).

178. Quarteroni, A., Sacco, R., Saleri, F. & Gervasio, P. *Matematica Numerica* (Springer Milan, Milano, 2014) (cit. on p. 109).

179. Wainwright, M. J. & Simoncelli, E. Scale mixtures of Gaussians and the statistics of natural images. *Advances in neural information processing systems* **12** (1999) (cit. on p. 110).

180. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N. & Ganguli, S. *Deep Unsupervised Learning using Nonequilibrium Thermodynamics* in *Proceedings of the 32nd International Conference on Machine Learning* (eds Bach, F. & Blei, D.) **37** (PMLR, Lille, France, 2015), 2256–2265 (cit. on p. 113).

181. Song, Y. *et al.* Score-Based Generative Modeling through Stochastic Differential Equations. *ArXiv* **abs/2011.13456** (2020) (cit. on p. 113).

182. Biroli, G. & Mézard, M. Generative Diffusion in Very Large Dimensions. *Journal of Statistical Mechanics: Theory and Experiment* **2023,** 093402 (Oct. 2023) (cit. on p. 113).

183. Biroli, G., Bonnaire, T., De Bortoli, V. & Mézard, M. Dynamical Regimes of Diffusion Models. *Nature Communications* **15,** 9957 (17th Nov. 2024) (cit. on p. 113).

184. Ghio, D., Dandi, Y., Krzakala, F. & Zdeborová, L. Sampling with flows, diffusion, and autoregressive neural networks from a spin-glass perspective. *Proceedings of the National Academy of Sciences* **121,** e2311810121. eprint: `https://www.pnas.org/doi/pdf/10.1073/pnas.2311810121` (2024) (cit. on p. 113).

185. Kamb, M. & Ganguli, S. *An Analytic Theory of Creativity in Convolutional Diffusion Models* arXiv: `2412.20292 [cs]`. `http://arxiv.org/abs/2412.20292` (2025). Pre-published (cit. on p. 113).

186. Bonnaire, T., Urfin, R., Biroli, G. & Mézard, M. *Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training* arXiv: `2505.17638 [cs]`. `http://arxiv.org/abs/2505.17638` (2025). Pre-published (cit. on pp. 113, 114).

187. Merger, C. & Goldt, S. *Generalization Dynamics of Linear Diffusion Models* 2025. arXiv: `2505.24769 [stat.ML]` (cit. on p. 113).

188. George, A. J., Veiga, R. & Macris, N. *Denoising Score Matching with Random Features: Insights on Diffusion Models from Precise Learning Curves* 2025. arXiv: `2502.00336 [cs.LG]` (cit. on p. 113).

189. George, A. J., Veiga, R. & Macris, N. *Analysis of Diffusion Models for Manifold Data* 2025. arXiv: `2502.04339 [math.ST]` (cit. on p. 113).

190. Szegő, G. *Orthogonal Polynomials* (American mathematical society, 1939) (cit. on p. 129).

191. Abramowitz, M. & Stegun, I. A. *Handbook of mathematical functions with formulas, graphs, and mathematical tables* xiv+1046 (National Bureau of Standards, 1964) (cit. on p. 129).

192. Buet-Golfouse, F. A Multinomial Theorem for Hermite Polynomials and Financial Applications. *Applied Mathematics - A Journal of Chinese Universities Series B* **06,** 1017–1030 (2015) (cit. on p. 131).

193. Lukacs, E. A Survey of the Theory of Characteristic Functions. *Advances in Applied Probability* **4,** 1–38 (1972) (cit. on p. 133).

Appendix A

Notions on Hermite polynomials

We recall here the definitions and key properties of the Hermite polynomials.

Definition A.1. The Hermite polynomial of degree m is defined as

$$h_m(x) := (-1)^m e^{\frac{x^2}{2}} \frac{\mathrm{d}^m}{\mathrm{d}x^m} \left(e^{-\frac{x^2}{2}} \right) \tag{A.1}$$

Here is a list of the first 5 Hermite polynomials:

$$\begin{aligned} h_0(x) &= 1 \\ h_1(x) &= x \\ h_2(x) &= x^2 - 1 \\ h_3(x) &= x^3 - 3x \\ h_4(x) &= x^4 - 6x^2 + 3 \end{aligned} \tag{A.2}$$

The general formula is:

$$h_m(x) = m! \sum_{j=0}^{\lfloor m/2 \rfloor} \frac{(-1)^j}{2^j j!(n-2j)!} x^{m-2j} \tag{A.3}$$

The Hermite polynomials enjoy the following properties (for details see section 5.4 in McCullagh [124], Szegő [190] and Abramowitz & Stegun [191]):

- they are orthogonal with respect to the L^2 product weighted with the density of the Normal distribution:
$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} h_n(x) h_m(x) e^{-\frac{x^2}{2}} \mathrm{d}x = n! \delta_{m,n}; \tag{A.4}$$
- h_m is a monic polynomial of degree m, hence $(h_m)_{m \in \{1,\dots,N\}}$ generates the space of polynomials of degree $\leq N$;
- the previous two properties imply that the family of Hermite polynomials is an orthogonal basis for the Hilbert space $L^2(\mathbb{R}, \mathbb{Q})$ where $\mathbb{Q}$ is the normal distribution;
- from eq. (A.3) one can deduce
$$h'_m(x) = m h_{m-1}(x) \tag{A.5}$$

- they enjoy the following recurring relationship

$$h_{m+1}(x) = xh_m(x) - h'_m(x) = xh_m(x) - mh_{m-1}(x), \tag{A.6}$$

which can also be expressed as a relationship between coefficients as follows. If $h_m(x) = \sum_{k=0}^{m} a_{m,k}x^k$, then

$$a_{m+1,k} = \begin{cases} -a_{m,1} & k = 0, \\ a_{m,k-1} - (k+1)a_{m,k+1} & k > 0. \end{cases} \tag{A.7}$$

- From (A.6) it is possible to deduce identities of binomial type, like:

$$h_m(x+y) = \sum_{k=0}^{m} \binom{m}{k} x^{m-k} h_k(y) \tag{A.8}$$

$$h_m(\gamma x) = \sum_{j=0}^{\lfloor m/2 \rfloor} \gamma^{m-2j}(\gamma^2 - 1)^j \binom{m}{2j} (2j-1)!! h_{m-2j}(x) \tag{A.9}$$

Multivariate case In the multivariate m-dimensional case we can consider Hermite tensors $(H_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha}\in\mathbb{N}^m}$ defined as:

$$H_\alpha(x_1, \dots, x_m) = \prod_{i=1}^{m} h_{\alpha_i}(x_i) \tag{A.10}$$

all the properties of the one-dimensional Hermite polynomials generalize to this case, in particular they form an orthogonal basis of $L^2(\mathbb{R}^m, \mathbb{Q})$ where $\mathbb{Q}$ is a multivariate normal distribution. If $\langle \cdot, \cdot \rangle$ is the inner product of that Hilbert space, we have that:

$$\langle H_{\boldsymbol{\alpha}}, H_{\boldsymbol{\beta}} \rangle = \boldsymbol{\alpha}! \delta_{\boldsymbol{\alpha},\boldsymbol{\beta}} \tag{A.11}$$

Of course all this is valid in the case $m = nd$, where we can also use multi-multi-indices to get the following identity:

$$H_{\underline{\boldsymbol{\alpha}}}(x_1^1, \dots, x_d^n) = \prod_{\mu=1}^{n} H_{\boldsymbol{\alpha}^\mu}(x^\mu) = \prod_{\mu=1}^{n}\prod_{i=1}^{d} h_{\boldsymbol{\alpha}_i^\mu}(x_i^\mu) \tag{A.12}$$

Definition A.2 (Hermite expansion)**.** Consider a function $f : \mathbb{R} \to \mathbb{R}$ that is square integrable with respect to the standard normal distribution $p(x) = (1/\sqrt{2\pi})\, e^{-x^2/2}$. Then there exists a unique sequence of real numbers $\{c_k\}_{k\in\mathbb{N}}$ called Hermite coefficients such that

$$f(x) = \sum_{k=0}^{\infty} \frac{c_k}{k!} h_k(x) \quad \text{and} \quad c_k(x) := \mathbb{E}_{x\sim\mathcal{N}(0,1)}[f(x)h_k(x)],$$

where h_i is the i-th probabilist's Hermite polynomial.

Definition A.3 (Information exponent)**.** Given any function f for which the Hermite expansion exists, its information exponent $k^* = k^*(f)$ is the smallest index $k \geq 1$ such that $c_k \neq 0$.

Lemma A.4 (Expectations of Triple and Quadruple products)**.** *For any $i, j, k \in \mathbb{N}$, we have*

$$I_3^*(i,j,k) := \mathbb{E}_{x\sim\mathcal{N}(0,1)}[h_i(x)h_j(x)h_k(x)] = \frac{i!\, j!\, k!}{\left(\frac{i+j-k}{2}\right)!\left(\frac{i+k-j}{2}\right)!\left(\frac{j+k-i}{2}\right)!}$$

if the denominator exists, that is if $i+j+k$ is even and the sum of any two of i,j and k is strictly less than the third, or zero otherwise. What's more, fix the indices $m_1, m_2, m_3, m_4 \in \mathbb{N}$ and order them such that $m_1 \geq m_2 \geq m_3 \geq m_4$.
Define $M := (m_1+m_2+m_3+m_4)/2$.
Then, given $m := \min(m_3, m_4)$ and $I_4^*(m_1,m_2,m_3,m_4) := \mathbb{E}_{x\sim\mathcal{N}(0,\mathbb{1}_d)}[h_{m_1}(x)h_{m_2}(x)h_{m_3}(x)h_{m_4}(x)]$, we have

$$I_4^*(m_1,m_2,m_3,m_4) = \sum_{\nu=0}^{m} \frac{(m_3+m_4-2\nu)!m_1!\,m_2!\,m_3!\,m_4}{(M-m_3-m_4-\nu)!(M-m_1-\nu)!(M-m_2-\nu)!(m_3-\nu)!(m_4-\nu)!\nu!}$$

if $M \in 2\mathbb{N}$ and the denominator exists. Otherwise, $I_4^*(m_1,m_2,m_3,m_4) = 0$.

Lemma A.5. *Consider $w_1, w_2 \in \mathbb{S}^{d-1}$ and set $\alpha = w_1 \cdot w_2$. Then, for any $i,j \in \mathbb{N}$*

$$\mathbb{E}_{x\sim\mathcal{N}(0,\mathbb{1}_d)}[h_i(w_1\cdot x)h_j(w_2\cdot x)] = i!\,\alpha^i\,\delta_{ij}.$$

To extend this orthogonality property to triple an quadruple products of Hermite polynomials, we use the same technique as in [192]

Lemma A.6 (Orthogonality property)**.** *Fix $i,j,k \in \mathbb{N}$. Set $I_3 := \mathbb{E}_{x\sim\mathcal{N}(0,\mathbb{1}_d)}[h_i(w_1\cdot x)h_j(w_2\cdot x)h_k(w_3\cdot x)]$. Consider three different spikes $w_1,w_2,w_3 \in \mathbb{S}^{d-1}$ and set $\rho := w_1\cdot w_2, \tau := w_1\cdot w_3$ and $\eta := w_2\cdot w_3$. Then, we have*

$$I_3 = \sum_{\ell=0}^{j} \frac{i!j!k!}{(j-\ell)!^2\big(\frac{i+j-k}{2}\big)!\big(\frac{i+k-j}{2}\big)!\big(\frac{k-j+2\ell-i}{2}\big)!}\rho^\ell\tau^{k-j+\ell}(\eta-\tau\rho)^{j-\ell}$$

if the denominator exists, and $I_3 = 0$ otherwise.

Proof. We start by writing w_2 as sum of its projection along the first spike w_1 and its part in the orthogonal space, that is $w_2 = \rho w_1 + \sqrt{1-\rho^2}u$, where $u \in \mathbb{S}^{d-1}$ and $u\cdot w_1 = 0$. Then, clearly, $w_2\cdot x = \rho y_1 + \sqrt{1-\rho^2}y_u$, where $y_1 := w_1\cdot x$ and $y_u := u\cdot x$.
We can do the same for the third spike, that is $w_3 = \tau w_1 + \sqrt{1-\tau^2}u'$, where $u' \in \mathbb{S}^{d-1}$ and $u'\cdot w_1 = 0$. Define $y_{u'} = u'\cdot x$. We can now exploit the fact that any Hermite polynomial of a sum can be written as the sum of Hermite polynomials in the following way, that is for any $m\in\mathbb{N}$ we have that $h_m(\alpha z_1 + \sqrt{1-\alpha^2}z_2) = \sum_{n=0}^m \binom{m}{n}\alpha^n(\sqrt{1-\alpha^2})^{m-n}h_\ell(z_1)h_{m-n}(z_2)$, for any $\alpha\in[0,1]$. By applying this formula to w_2 and w_3, we get

$$I_3 = \sum_{\ell,\ell'=0}^{j,k}\binom{j}{\ell}\binom{k}{\ell'}\rho^\ell\tau^{\ell'}(\sqrt{1-\rho^2})^{j-\ell}(\sqrt{1-\tau^2})^{k-\ell'}\underbrace{\mathbb{E}[h_i(y_1)h_\ell(y_1)h_{\ell'}(y_1)]}_{(1)}\underbrace{\mathbb{E}[h_{j-\ell}(y_u)h_{k-\ell'}(y_{u'})]}_{(2)}$$

where we have used the fact that y_1 is independent from y_u and $y_{u'}$ since $w_1\cdot u = w_2\cdot u' = 0$. Then, in view of Lemma A.4 and Lemma A.5, we have

$$(1) = \frac{i!\ell!\ell'!}{\big(\frac{i+\ell-\ell'}{2}\big)!\big(\frac{i+\ell'-\ell}{2}\big)!\big(\frac{\ell+\ell'-i}{2}\big)!} \quad \text{and} \quad (2) = (j-\ell)!(u\cdot u')^{j-\ell}\delta_{j-\ell=k-\ell'}.$$

Hence, imposing that $\ell' = k-j-\ell$, we obtain

$$I_3 = \sum_{\ell=0}^{j} \frac{i!j!k!}{(j-\ell)!\big(\frac{i+j-k}{2}\big)!\big(\frac{i+k-j}{2}\big)!\big(\frac{k-j+2\ell-i}{2}\big)!}\rho^\ell\tau^{k-j+\ell}[\sqrt{1-\rho^2}\,\sqrt{1-\tau^2}\,(u\cdot u')]^{j-\ell}$$

when the denominator exists and it vanishes otherwise. Therefore, the thesis follows from the fact that we have $u \cdot u' = \dfrac{\eta - \tau\rho}{\sqrt{1-\tau^2}\sqrt{1-\rho^2}}$.

□

Corollary A.7 (used formulas with same argument)**.** *Fix $k = 1$ and $i, j \in \mathbb{N}$. Consider I_3 given by A.6 in the case of $w_3 = w_2$. Then,*

$$I_3 = \mathbb{E}_{x\sim\mathcal{N}(0,1)}[h_i(w_1)h_j(w_2)h_1(w_1)] = \frac{i!j!}{\left(\frac{i+j-1}{2}\right)!\left(\frac{i+1-j}{2}\right)!\left(\frac{j+1-i}{2}\right)!}\,\rho^j,$$

when the denominator exists and $I_3 = 0$ otherwise.
Moreover, for any $i, j, k, t \in \mathbb{N}$, consider $I_4 := \mathbb{E}_{x\sim\mathcal{N}(0,1)}[h_i(w_1 \cdot x)h_j(w_1 \cdot x)h_k(w_2 \cdot x)h_t(w_3 \cdot x)]$, with $w_1, w_2, w_3 \in \mathbb{S}^{d-1}$. Then,

$$I_4 = \sum_{\ell=0}^{k}(-1)^{k-\ell}\binom{k}{\ell}\binom{t}{t-k+\ell}\rho^k\tau^t\; I_4^*(i, j, \ell, t-k+\ell),$$

where I_4^* is defined in Lemma A.4 and the indeces i, j, ℓ and $t-k+\ell$ are ordered.

Proof. The first part is obtained by repeating the proof of Lemma A.6 with $w_2 = \rho w_1 + \sqrt{1-\rho^2}u$, for $u \in \mathbb{S}^{d-1}$ and $u \cdot w_1 = 0$, and using the fact that $\mathbb{E}_{x\sim\mathcal{N}(0,\mathbb{1}_d)}[h_{j-\ell}(u \cdot x)] = \delta_{j-\ell}$.
For the second part, it is sufficient to repeat the usual procedure by decomposing both w_2 and w_3 such that $w_2 = \rho w_1 + \sqrt{1-\rho^2}u$ and $w_3 = \tau w_1 + \sqrt{1-\tau^2}u'$ for $u, u' \in \mathbb{S}^{d-1}$ and $w_1 \cdot w_2 = w_1 \cdot w_3 = 0$. Then, the thesis follows from the second part of Lemma A.6. □

Lemma 2.13 translates the problem of computing the norm of the LDLR to the problem of computing the projections $\langle L, H_{\boldsymbol{\alpha}}\rangle$. Note that this quantity is equal to $\mathbb{E}_{\mathbb{P}}[H_{\boldsymbol{\alpha}}(x)]$, which we will call *$\boldsymbol{\alpha}$-th Hermite coefficient* of the distribution $\mathbb{P}$.

The following lemma from [123] provides a version of the integration by parts technique that is tailored for Hermite polynomials.

Lemma A.8. *Let $f : \mathbb{R}^d \to \mathbb{R}^d$ be a function that is continuously differentiable k times. Assume that f and all of its partial derivatives up to order k are bounded by $O(\exp\left(|y|^\lambda\right))$ for some $\lambda \in (0, 2)$, then for any $\boldsymbol{\alpha} \in \mathbb{N}^d$ such that $|\boldsymbol{\alpha}| \le k$*

$$\langle f, H_{\boldsymbol{\alpha}}\rangle = \mathop{\mathbb{E}}_{y\sim\mathcal{N}(0,\mathbb{1})}[H_{\boldsymbol{\alpha}}(y)f(y)] = \mathop{\mathbb{E}}_{y\sim\mathcal{N}(0,\mathbb{1})}[\partial_{\boldsymbol{\alpha}}f(y)] \tag{A.13}$$

Proof. A.13 can be proved by doing induction on k using A.6, see [123] for details. □

A.1 Links with cumulant theory

The cumulants $(\kappa_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha}\in\mathbb{N}^d}$ of a random variable $x \in \mathbb{R}^d$, can be defined as the Taylor coefficients of the expansion in ξ of the *cumulant generating function*:

$$K_x(\xi) := \log\left(\mathbb{E}\left[e^{\xi\cdot x}\right]\right) \tag{A.14}$$

Order-one cumulant is the mean, order-two is the variance, and higher-order cumulants cumulants encode more complex correlations among the variables. The Gaussian distribution is the only non

constant distribution to have a polynomial cumulant generating function (as proved in theorem 7.3.5 in [193]), if $z \sim \mathcal{N}(\mu, \Sigma)$, then:

$$K_z(\xi) = \mu\xi + \frac{1}{2}\xi^\top \Sigma \xi.$$

Hence cumulants with order higher than three can also be seen as a measure of how much a distribution deviates from Gaussianity (i.e. how much it deviates from its best Gaussian approximation).

On this point the similarity with Hermite coefficients is evident. Indeed, up to order-five, on whitened distributions, cumulants and Hermite coefficients coincide. But form sixth-order onward, they start diverging. They are still linked by deterministic relationship, but its combinatorial complexity increases swiftly and it is not easy to translate formulas involving Hermite coefficients into cumulants and vice versa. For this reason, our low-degree analysis of the likelihood ratio leading to 2.6 keeps the formalism that naturally arises from the computations, which is based on Hermite coefficients. A detailed discussion of the relation between Hermite polynomials and cumulants in the context of asymptotic expansions of distributions like the Gram–Charlier and Edgeworth series can be found in chapter 5 of McCullagh [124].