# Random Matrix Theory for AI: From Theory to Practice

Mohamed El Amine Seddik

université
PARIS-SACLAY

# Random Matrix Theory for AI: From Theory to Practice

**Thèse de doctorat de l'Université Paris-Saclay**

École doctorale n° 580 Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat : Traitement du signal et des images
Unité de recherche : Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes, 91190, Gif-sur-Yvette, France.
Référent : CentraleSupélec

**Thèse présentée et soutenue en visioconférence totale, le 03/11/2020, par**

**Mohamed El Amine SEDDIK**

**Composition du jury:**

| | |
|---|---|
| **Alexandre Gramfort**<br>Directeur de recherche, INRIA Paris | Président |
| **Julie Delon**<br>Professeure, Université Paris Descartes | Rapportrice & Examinatrice |
| **Jamal Najim**<br>Directeur de recherche, Université Marne la Vallée | Rapporteur & Examinateur |
| **Mérouane Debbah**<br>Professeur, Huawei Labs & CentraleSupélec | Examinateur |
| **Florent Krzakala**<br>Professeur, EPFL | Examinateur |
| **Florent Chatelain**<br>Maître de conférences, Université de Grenoble-Alpes | Examinateur |
| **Romain Couillet**<br>Professeur, CentraleSupélec | Directeur |
| **Mohamed Tamaazousti**<br>Ingénieur-chercheur, CEA Paris-Saclay | Co-encadrant |

Thèse de doctorat

NNT: 2020UPASG010

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

**Titre:** La théorie des matrices aléatoires pour l'IA : de la théorie à la pratique

**Mots clés:** Apprentissage automatique, théorie des matrices aléatoires, concentration de la mesure, réseaux de neurones, réseaux de neurones génératifs adverses

**Résumé:** De nos jours, l'IA repose en grande partie sur l'utilisation de données de grande taille et sur des méthodes d'apprentissage machine améliorées qui consistent à développer des algorithmes de classification et d'inférence en tirant parti de grands ensembles de données de grande taille. Ces grandes dimensions induisent de nombreux phénomènes contre-intuitifs, conduisant généralement à une mauvaise compréhension du comportement de nombreux algorithmes d'apprentissage machine souvent conçus avec des intuitions de petites dimensions de données. En tirant parti du cadre multidimensionnel (plutôt que d'en souffrir), la théorie des matrices aléatoires (RMT) est capable de prédire les performances de nombreux algorithmes non linéaires aussi complexes que certains réseaux de neurones aléatoires, ainsi que de nombreuses méthodes du noyau telles que les SVM, la classification semi-supervisée, l'analyse en composantes principales ou le regroupement spectral. Pour caractériser théoriquement les performances de ces algorithmes, le modèle de données sous-jacent est souvent un modèle de mélange gaussien (MMG) qui semble être une hypothèse forte étant donné la structure complexe des données réelles (par exemple, des images). En outre, la performance des algorithmes d'apprentissage automatique dépend du choix de la représentation des données (ou des caractéristiques) sur lesquelles ils sont appliqués. Encore une fois, considérer les représentations de données comme des vecteurs gaussiens semble être une hypothèse assez restrictive. S'appuyant sur la théorie des matrices aléatoires, cette thèse vise à aller au-delà de la simple hypothèse du MMG, en étudiant les outils classiques d'apprentissage machine sous l'hypothèse de vecteurs aléatoires concentrés qui généralisent les vecteurs Gaussiens. Cette hypothèse est particulièrement motivée par l'observation que l'on peut utiliser des modèles génératifs (par exemple, les GAN) pour concevoir des structures de données complexes et réalistes telles que des images, grâce à des transformations Lipschitzienne de vecteurs gaussiens. Cela suggère notamment que l'hypothèse de concentration sur les données mentionnée ci-dessus est un modèle approprié pour les données réelles et qui est tout aussi mathématiquement accessible que les MMG. Par conséquent, nous démontrons à travers cette thèse, en nous appuyant sur les GANs, l'intérêt de considérer le cadre des vecteurs concentrés comme un modèle pour les données réelles. En particulier, nous étudions le comportement des matrices de Gram aléatoires qui apparaissent au cœur de divers modèles linéaires, des matrices à noyau qui apparaissent dans les méthodes à noyau et également des méthodes de classification qui reposent sur une solution implicite (par exemple, la couche de Softmax dans les réseaux de neurones), avec des données aléatoires supposées concentrées. En particulier, la compréhension du comportement de ces matrices/méthodes, pour des données concentrées, nous permet de caractériser les performances (sur des données réelles si nous les assimilons à des vecteurs concentrés) de nombreux algorithmes d'apprentissage machine, tels que le clustering spectral, les SVM, l'analyse en composantes principales et l'apprentissage par transfert. L'analyse de ces méthodes pour des données concentrées donne le résultat surprenant qu'elles ont asymptotiquement le même comportement que pour les données de MMG. Ce résultat suggère fortement l'aspect d'universalité des grands classificateurs d'apprentissage machine par rapport à la distribution sous-jacente des données.

**Title:** Random Matrix Theory for AI: From Theory to Practice

**Keywords:** Machine learning, random matrix theory, concentration of measure, neural networks, GANs

**Abstract:** AI nowadays relies largely on using large data and enhanced machine learning methods which consist in developing classification and inference algorithms leveraging large datasets of large sizes. These large dimensions induce many counter-intuitive phenomena, leading generally to a misunderstanding of the behavior of many machine learning algorithms often designed with small data dimension intuitions. By taking advantage of (rather than suffering from) the multidimensional setting, random matrix theory (RMT) is able to predict the performance of many non-linear algorithms as complex as some random neural networks as well as many kernel methods such as Support Vector Machines, semi-supervised classification, principal component analysis or spectral clustering. To characterize the performance of these algorithms theoretically, the underlying data model is often a Gaussian mixture model (GMM) which seems to be a strong assumption given the complex structure of real data (e.g., images). Furthermore, the performance of machine learning algorithms depends on the choice of data representation (or features) on which they are applied. Once again, considering data representations as Gaussian vectors seems to be quite a restrictive assumption. Relying on random matrix theory, this thesis aims at going beyond the simple GMM hypothesis, by studying classical machine learning tools under the hypothesis of Lipschitz-ally transformed Gaussian vectors also called concentrated random vectors, and which are more generic than Gaussian vectors. This hypothesis is particularly motivated by the observation that one can use generative models (e.g., GANs) to design complex and realistic data structures such as images, through Lipschitz-ally transformed Gaussian vectors. This notably suggests that making the aforementioned concentration assumption on data is a suitable model for real data and which is just as mathematically accessible as GMM models. Moreover, in terms of data representation, the concentration framework is compatible with one of the most widely used data representations in practice, namely deep neural nets (DNNs) representations, since they consist of a Lipschitz transformation of the input data (e.g., images). Therefore, we demonstrate through this thesis, leveraging on GANs, the interest in considering the framework of concentrated vectors as a model for real data. In particular, we study the behavior of random Gram matrices which appear at the core of various linear models, kernel matrices that appear in kernel methods, and also classification methods that rely on an implicit solution (e.g., Softmax layer in neural networks), with concentrated random inputs. Indeed, these methods are at the heart of many classifications, regression, and clustering machine learning algorithms. In particular, understanding the behavior of these matrices/methods, for concentrated data, allows us to characterize the performances (on real data if we assimilate them to concentrated vectors) of many machine learning algorithms, such as spectral clustering, SVMs, principal component analysis, and transfer learning. Analyzing these methods for concentrated data yields the surprising result that they have asymptotically the same behavior as for GMM data (with the same first and second-order statistics). This result strongly suggests the universality aspect of large machine learning classifiers w.r.t. the underlying data distribution.

**Université Paris-Saclay**
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France

# Random Matrix Theory for AI: From Theory to Practice

Mohamed El Amine SEDDIK

Under the supervision of
Prof. Romain COUILLET & Dr. Mohamed TAMAAZOUSTI

January 21, 2021

# Acknowledgments

First of all, I would like to deeply thank my thesis director Prof. Romain Couillet as well as my co-supervisor Dr. Mohamed Tamaazousti for having constantly supported me throughout my thesis, I also thank them for their confidence, patience, motivation and for having guided me during these three unforgettable years. I had the immense pleasure of working with them both professionally and personally.

I would also like to express my sincere thanks for the two rapporteurs of my thesis: Prof. Julie Delon and Prof. Jamal Najim, so I would like to thank them for the time spent reading my manuscript and their very enriching comments.

I would also like to thank the other members of the jury for my defense: Dr. Alexandre Gramfort, Prof. Mérouane Debbah, Prof. Florent Krzakala and Dr. Florent Chatelain who were present despite the health crisis of Covid-19, I thank them for their questions and for their evaluation of my work.

I would like to express my gratitude to all the institutions (CEA and CentraleSupélec) and laboratories (LVML and L2S) that ensured the smooth running of my thesis. I would also like to thank all the people who were involved in my thesis. I am thinking in particular of Mrs. Odile Caminondo and Mrs. Virginie Valaire for their help on the administrative level within the CEA, and also to Mrs. Anne Batalie and Prof. Gilles Duc within CentraleSupélec. I would also like to thank all my colleagues in the SIALV department at the CEA and also my colleagues at GIPSA in Grenoble, where I spent very memorable moments.

I would like to make a special mention to my teammates during my thesis: Cosme, Youssef, Abdallah, John, Hafiz, Malik, Xiaoyi, Zhenyu, Lorenzo, Cyprien, Souheil, Alex, Alexandre, Tayeb, Charles, Hassane, Vincent, Harold and Vasily. I really enjoyed working with you and I was very pleased to have you as teammates. Finally, I would like to thank my family for their support, for believing in me and for supporting me throughout my years of study and even throughout my life. Special thanks goes out to my parents: Fatima and Mimoun, my grandparents: Malika and Chaib, my brothers: Moussa and Issam, my sister and her husband: Khadija and Mohamed.

# Contents

# List of Figures

# Notations

## Mathematical Symbols

| | |
|---|---|
| $\mathbb{R}$ | Set of real numbers. |
| $\mathbb{C}$ | Set of complex numbers, we denote $\mathbb{C}^+$ the set $\{z \in \mathbb{C}, \Im[z] > 0\}$. |
| $\mathcal{M}_{p,n}$ | Set of matrices of size $p \times n$. |
| $\mathcal{M}_p$ | Set of squared matrices of size $p$. |
| $\mathcal{D}_p$ | Set of diagonal matrices of size $p$. |
| $(\cdot)^\mathsf{T}$ | Transpose operator. |
| $\mathrm{tr}(\cdot)$ | Trace operator. |
| $\mathrm{diag}(\cdot)$ | Diagonal operator, for $\boldsymbol{A} \in \mathcal{M}_n$, $\mathrm{diag}(\boldsymbol{A}) \in \mathbb{R}^n$ is the vector with entries $\{A_{ii}\}_{i=1}^n$; for $\boldsymbol{a} \in \mathbb{R}^n$, $\mathrm{diag}(\boldsymbol{a}) \in \mathcal{D}_n$ is the diagonal matrix taking $\boldsymbol{a}$ as its diagonal. |
| $\|\cdot\|$ | Operator (or spectral) norm of a matrix and Euclidean norm of a vector. |
| $\|\cdot\|_F$ | Frobenius norm of a matrix, $\|\boldsymbol{A}\|_F = \sqrt{\mathrm{tr}(\boldsymbol{A}\boldsymbol{A}^\mathsf{T})}$. |
| $\|\cdot\|_\infty$ | Infinite norm of a matrix, $\|\boldsymbol{A}\|_\infty = \max_{i,j} \lvert A_{ij}\rvert$. |
| $\mathrm{dist}(\cdot)$ | Distance between elements in a metric space. |
| $P(\cdot)$ | Probability of an event with respect to the underlying probability measure space $(\Omega, \mathcal{F}, P)$. |
| $\mathbb{E}[\cdot]$ | Expectation operator, $\mathbb{E}[f] = \int f dP = \int_\Omega f(x) P(dx)$ |
| $\mathrm{Var}[\cdot]$ | Variance operator, $\mathrm{Var}[x] = \mathbb{E}[x^2] - (\mathbb{E}[x])^2$ if the first two moments of $x$ exist |
| $\xrightarrow{a.s.}$ | Almost surely convergence. We say a sequence $x_n \xrightarrow{a.s.} x$ if $P(\lim_{n\to\infty} x_n = x) = 1$ |
| $Q(x)$ | Q-function: the tail distribution of standard Gaussian $Q(x) = \frac{1}{2\pi}\int_x^\infty \exp(-t^2/2)dt$ |
| $\mathcal{O}(1)$, $o(1)$ | A sequence $u_n$ is bounded or converges to zero as $n \to \infty$, respectively. |

## Vectors and Matrices

| | |
|---|---|
| $\boldsymbol{I}_n$ | Identity matrix of size $n$. |
| $\mathbf{1}_n$ | (Column) vector of size $n$ with all entries equal to one. |
| $\mathbf{0}_n$ | (Column) vector of size $n$ with all entries equal to zero. |
| $\boldsymbol{x}_i \in \mathbb{R}^p$ | Input data/feature vector. |
| $\boldsymbol{z}_i \in \mathbb{R}^p$ | Random vector having i.i.d. zero mean and unit variance entries. |
| $\boldsymbol{X}$ | Data/feature matrix having $\boldsymbol{x}_i$ as column vectors. |
| $\boldsymbol{Z}$ | Random matrix having $\boldsymbol{z}_i$ as column vectors such that $\mathbb{E}[\boldsymbol{z}_i] = \mathbf{0}_p$. |
| $\boldsymbol{\mu}$ | Mean vector. |
| $\boldsymbol{C}$ | Covariance matrix. |
| $\boldsymbol{K}$ | Kernel matrix. |
| $\boldsymbol{W}$ | Weight matrix (in a neural network). |
| $\boldsymbol{\beta}$ | Regression vector. |
| $\boldsymbol{Q}$ | Resolvent matrix, see Definition 3. |

# Chapter 1

# Introduction

## Contents



Artificial intelligence (AI) is known as the set of theories and techniques used to create machines capable of simulating human intelligence. One of the most attractive subfields of AI is machine learning (ML) which aims at providing computer algorithms that "learn" automatically through experience in order to make future decisions without being explicitly programmed. Basically, ML algorithms leverage on building mathematical – very often parametric – models which will be optimized based on sample training data and then used afterwards to perform various AI tasks such as classification, regression, clustering etc.

Quite naturally, AI finds applications in various domains and hence one of the most important challenges of ML is to provide algorithms that can be applied to different kind of data (e.g., images, texts, graphs etc.). By construction, these data can be represented in different forms and therefore the performance of ML algorithms will rely largely on the chosen representation. This representation should ideally contain relevant information about the data in order to achieve learning with simple models and small amount of data. Historically, a huge amount of works were focused on the design of hand-crafted representations (or features) and then providing them to simple ML algorithms to resolve the desired tasks. But for most tasks and given the various types of data, these approaches are not easily scaleable to achieve effective AI.

Since the arrival of deep neural networks (DNNs), the ideas of developing hand-crafted features were immediately left out. Indeed, DNNs have surpassed most of the approaches by demonstrating their incredible ability to automatically learning relevant

representations from raw data in a wide range of applications, including computer vision, pattern recognition and natural language processing. Despite their success, a lot of questions are still unanswered regarding the theoretical foundations of DNNs and which are very crucial notably for their explainability. For instance, the full characterization of their learnt representations and/or parameters are still open problems.

One of the main aspects that made DNNs effective in practice is their being over-parametrized models. Indeed, it has been shown that deep architectures of these models surpass shallow ones when dealing with $n$ $p$-dimensional data when both $n$ and $p$ are large, which is often the case in real life scenarios[1]. Moreover, most effective DNNs happen to have a number of parameters $N$ which is at least of order $p$ or even much larger (e.g., LeNet-5 [LeC98] contains $N = 60000$ parameters).

In essence, these large dimensions induce many counter-intuitive phenomena that cause the intuitions of the small dimensions to collapse completely. For better understanding of these phenomena, we will provide subsequently in Section 1.1 some illustrative examples which reveal these counter-intuitive aspects. In the particular case when both $p, n \to \infty$ with $p/n \to 0 \in (0, \infty)$, random matrix theory (RMT) provides powerful tools to assess the performance of various ML algorithms by tacking into account the effect of these dimensions. Indeed, RMT provides access to the internal mechanism of a large wide of ML methods thereby allowing a deeper understanding and systematic improvements of these methods. We shall refer the reader to the thesis of Z. Liao [Lia19] for applications of RMT to kernel methods, random shallow neural networks and neural networks dynamics; the thesis of X. Mai [Mai19] which addresses applications of RMT to semi-supervised learning and support vector machines.

The aforementioned works leverage largely on Gaussian assumptions[2] on the processed data. One of the major outcome of this thesis it to go beyond the Gaussian hypothesis to address the applicability of RMT to real data which are unlikely close to Gaussian vectors. In particular, working under the more generic statistical model of concentrated vectors [LC20], we provide justifications – leveraging on generative adversarial networks (GANs) – about the relevance of such model for realistic data modelling, and we further analyze, under the concentration assumption on the data, the behavior of large kernel matrices (which happen to be at the core of various ML algorithms) as well as some essential components of neural networks such as the last Softmax layer. A major result of the developed works in this thesis is the universality result stated as:

*"Only first and second order statistics of concentrated data matter to describe the behavior of these methods"*

thereby justifying the Gaussianity assumption on data as per the results of [Lia19, Mai19].

Let us now give some illustrative examples of the effect of high-dimensional data which leads to fundamentally different behaviors of ML algorithms comparing to low-dimensional data.

[1] As an example, the MNIST dataset [LeC98] contains $n = 70000$ images of dimension $p = 28 \times 28 = 784$.
[2] Modelling the data as a $k$-class of Gaussian mixture model (GMM).

## 1.1 Illustrative High-dimensional Examples

In this section we highlight the effect of high dimensions on two classical examples: the first one describes the behavior of large sample covariance matrices presented in Subsection 1.1.1; and the second example describes the behavior of large kernel matrices presented in Subsection 1.1.2 which are at the core of a wide range of ML algorithms.

### 1.1.1 Large Sample Covariance Matrices

Covariance matrices are omnipresent in a wide range of signal processing and machine learning approaches, therefore understanding their behavior is of particular interest to achieve consistent estimations and inferences. We will present in this subsection a classical and yet fundamental result from random matrix theory which describes the counter-intuitive behavior of large covariance matrices.

Let $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathbb{R}^p$ some *i.i.d.* random data vectors such that $\mathbb{E}[\boldsymbol{x}_i] = \mathbf{0}$ and $\mathbb{E}[\boldsymbol{x}_i \boldsymbol{x}_i^\intercal] = \boldsymbol{I}_p$. In the case where $\boldsymbol{x}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_p)$, the maximum likelihood estimator for the population covariance matrix (here $\boldsymbol{I}_p$) is the *sample covariance matrix* given by

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^\intercal = \frac{1}{n} \boldsymbol{X} \boldsymbol{X}^\intercal \tag{1.1}$$

where $\boldsymbol{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_n] \in \mathcal{M}_{p,n}$. In the classical setting where $p$ does not scale with the number of samples $n$, we know from the *strong law of large numbers* that, as $n \to \infty$

$$\hat{\boldsymbol{\Sigma}} \xrightarrow{a.s.} \boldsymbol{I}_p$$

In particular, $\|\hat{\boldsymbol{\Sigma}} - \boldsymbol{I}_p\| \xrightarrow{a.s.} 0$ in operator norm or any matrix norm since they are equivalent in finite dimension. However, it turns out that in the *random matrix theory* regime when both $p, n \to \infty$ with $p/n \to c \in (0, \infty)$,

$$\|\hat{\boldsymbol{\Sigma}} - \boldsymbol{I}_p\| \not\to 0$$

Indeed, in particular, in the under-sampling setting when $c > 1$ the sample covariance matrix converges entry-wise to the population matrix since

$$\max_{1 \le i,j \le p} \left| \left[ \hat{\boldsymbol{\Sigma}} - \boldsymbol{I}_p \right]_{ij} \right| \xrightarrow{a.s.} 0$$

while there is a clear *eigenvalue mismatch* between the two quantities which simply unfolds from

$$\begin{aligned} 0 = \lambda_1(\hat{\boldsymbol{\Sigma}}) = \ldots = \lambda_{p-n}(\hat{\boldsymbol{\Sigma}}) \le \lambda_{p-n+1}(\hat{\boldsymbol{\Sigma}}) \le \ldots \le \lambda_p(\hat{\boldsymbol{\Sigma}}) \\ 1 = \lambda_1(\boldsymbol{I}_p) = \ldots = \lambda_p(\boldsymbol{I}_p) \end{aligned}$$

therefore implies no convergence in spectral norm.

Figure 1.1 depicts the histogram of eigenvalues of $\hat{\boldsymbol{\Sigma}}$ for $p = 2000$ and $n = 10000$, as can be seen, instead of having the eigenvalue 1 with multiplicity $p$ one observes a spreading of the eigenvalues of $\hat{\boldsymbol{\Sigma}}$ in the vicinity of 1. This spreading of eigenvalues is particularly described by the so-called *Marčenko-Pastur* Law [MP67] which predicts the limiting eigenvalues distribution of $\hat{\boldsymbol{\Sigma}}$ in terms of the ratio $c = \lim_n p/n$ as follows

$$f_{MP}(x) = \frac{1}{2\pi c} \frac{\sqrt{(\lambda^+ - x)(x - \lambda^-)}}{x} + \max\left(1 - \frac{1}{c}, 0\right) \delta(x), \tag{1.2}$$

[3] $\delta$ denotes here the dirac function.

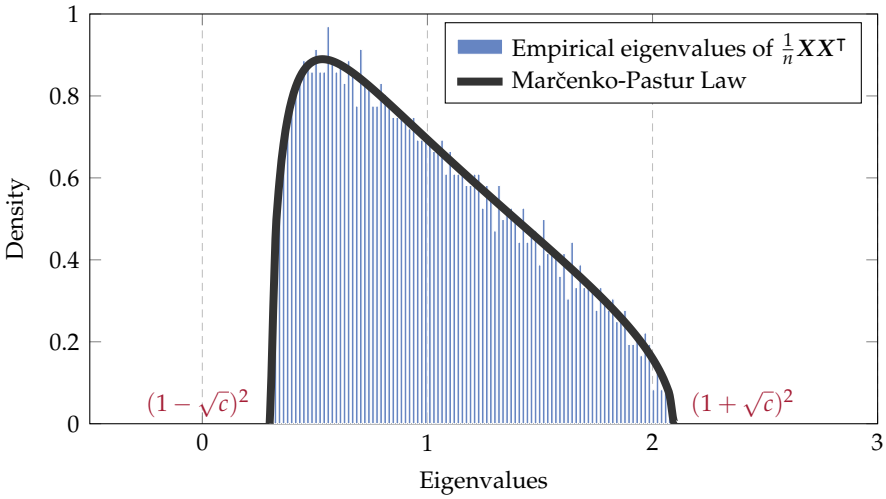



Figure 1.1: Histogram of the eigenvalues of $\hat{\mathbf{\Sigma}} = \frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\intercal$ where $\boldsymbol{X}$ is a random matrix with *i.i.d.* $\mathcal{N}(0,1)$ entries. We considered the dimensions $p = 2000$, $n = 10000$ and we recall $c = p/n$.

where $\lambda^\pm = (1 \pm \sqrt{c})^2$ are respectively the right and left edges of the limiting distribution. Indeed, the theorem for Marčenko & Pastur [MP67] states that the random *empirical spectral distribution (e.s.d.)* $\mu$ of $\hat{\mathbf{\Sigma}}$ defined in terms of its eigenvalues $\lambda_1, \dots, \lambda_p$ as

$$\mu(x) = \frac{1}{p}\sum_{i=1}^{p} \delta_{\lambda_i}(x) \tag{1.3}$$

converges, in the random matrix theory regime $p, n \to \infty$ with $p/n \to c \in (0, \infty)$, to a *deterministic* measure $\mu_{MP}$

$$\mu \xrightarrow{a.s.} \mu_{MP} \tag{1.4}$$

in distribution[4]. Essentially, $\mu_{MP}$ satisfies $\mu_{MP}(\{0\}) = \max(0, 1 - c^{-1})$ and on $(0, \infty)$ it has a continuous density function defined on the compact support $[(1-\sqrt{c})^2, (1+\sqrt{c})^2]$ by $f_{MP}$ defined in equation 1.2. A detailed statement along with a proof of this fundamental result is provided in Subsection 2.2.2 of this manuscript.

**Remark 1.1** (Universality of the Marčenko-Pastur Law)**.** *The Marčenko-Pastur result holds for any random matrix $\boldsymbol{X}$ having random i.i.d. entries with zero mean, unit variance and finite fourth order moment whatever the underlying distribution of its entries.*

We therefore know from this result that the behavior of the sample covariance matrix is fundamentally different in high-dimension leading to counter-intuitive phenomena, which is highlighted here through the behavior of its spectrum. A fundamental question therefore occurs regarding *when data are of high-dimension?* It should be noted that high-dimensionality is relative to the number of observations, which unfolds – through the above example – from the observation that the limiting Marčenko-Pastur density function in equation 1.2 depends only on the ratio $c = \lim_n \frac{p}{n}$. Specifically, we know from this result that the eigenvalues of $\hat{\mathbf{\Sigma}}$ spread from $(1-\sqrt{c})^2$ to $(1+\sqrt{c})^2$ instead of being concentrated around 1. As a result, the eigenvalues deviate on a range of diameter

$$(1+\sqrt{c})^2 - (1-\sqrt{c})^2 = 4\sqrt{c}$$

[4] In the sense that $\int \varphi(t)\mu(dt) \xrightarrow{a.s.} \int \varphi(t)\mu_{MP}(dt)$ for all bounded continuous test function $\varphi$.

For instance, even taking $n = 100\,p$ will result in a spread of eigenvalues around 1 with deviation $4\sqrt{c} = 0.4$ which is relatively large.

**Remark 1.2** (Extensions to structured data models)**.** *The Marčenko-Pastur result has been extended to some structured data models, namely by Bai and Silverstein in [BS10] for data of the form $\boldsymbol{x}_i = \boldsymbol{\Sigma}^{1/2}\boldsymbol{z}_i$ for some positive semi-definite matrix $\boldsymbol{\Sigma}$ and $\boldsymbol{z}_i$ random vectors with i.i.d. entries of zero mean, unit variance and finite fourth order moment. And also to a more convenient data model in the context of machine learning which is the Gaussian mixture model (GMM) by Benaych and Couillet in [BGC16]. The latter result will be presented in more details in Subsection 2.2.3.*

We have highlighted in this subsection the intriguing behavior of the sample covariance matrix when the underlying data are of high-dimension. In particular, we saw that in the random matrix theory regime when both $p, n \to \infty$ with $p/n \to c \in (0, \infty)$, the sample covariance matrix $\hat{\boldsymbol{\Sigma}}$ is no longer a consistent estimation of the population covariance matrix. In the next subsection, we will present another example of the effect of dimensionality on kernel methods, which are an essential component in modern machine learning.

### 1.1.2 Large Kernel Matrices

Kernel methods are a class of machine learning algorithms which were introduced in order to avoid transforming raw data into hand-crafted features and rather simply making the choice of a *kernel function* $\kappa(\boldsymbol{x}, \boldsymbol{y})$ instead, i.e., a *similarity* function between pairs of data samples $\boldsymbol{x}$ and $\boldsymbol{y}$ in their raw representation. One of the nice and very useful properties of (nonnegative definite) kernel methods is that they *implicitly* transform the input data into a high-dimensional feature space without requiring the computation of the data in that space, but rather computing the *inner similarity* between pairs of data, which is known as the *kernel trick*. However, the considered kernel function in practice are very often *non-linear* functions which makes the analysis of the resulting *random kernel matrix* non-trivial to tackle from the random matrix theory perspective.

One of the classical algorithms from kernel methods is the so-called *kernel spectral clustering* which specifically performs unsupervised classification of a sequence of data vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^p$ into $k$ distinct classes. It basically relies on the computation of large pair-wise kernel matrix $\boldsymbol{K} = \{\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)\}_{i,j=1}^n$ and performing a classical $k$-means clustering on the *dominant eigenspace* of the random kernel matrix $\boldsymbol{K}$.

In order to study the behavior of $\boldsymbol{K}$ and so to determine the information encoded in its dominant eigenspace, a crucial step requires the *linearization* of the non-linearity $\kappa(\boldsymbol{x}, \boldsymbol{y})$. Under *asymptotically non-trivial* growth rate assumptions on the data statistics, which basically maintains a feasible yet not too easy clustering problem, it has been shown by Couillet and Benayach in [CBG+16] that for the similarity

$$\kappa(\boldsymbol{x}, \boldsymbol{y}) = f\left(\frac{1}{p}\|\boldsymbol{x} - \boldsymbol{y}\|^2\right)$$

for sufficiently smooth function $f$, the off-diagonal entries of $\boldsymbol{K}$ tend – in the large dimensional regime when $p/n \to c$ as $p \to \infty$ – to a limiting constant independently of the data classes – *the between and within class vectors are "equidistant" in high-dimension*. This

intriguing high-dimensional behavior allows one to study $\boldsymbol{K}$ through a Taylor expansion yielding to a *random matrix equivalent* of $\boldsymbol{K}$, thereby giving access to the characterization of functionals of $\boldsymbol{K}$ and its (informative) eigenspace in the large dimensional setting. The aforementioned *non-trivial assumptions* notably permit an accurate approximation of $\boldsymbol{K}$ by its random matrix equivalent in the large $p, n$ limit.

Couillet and Benaych in [CBG+16] have notably analyzed the behavior of $\boldsymbol{K}$ under the so-called *Gaussian mixture model* assumption on data, which is defined in the following.

**Definition 1** (Gaussian Mixture Model (GMM)). *A sequence of data vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^p$ is said to form a k-class Gaussian mixture model of k classes $\mathcal{C}_1, \ldots, \mathcal{C}_k$ with distinct means and covariances $\{\boldsymbol{\mu}_\ell\}_{\ell=1}^k$ and $\{\boldsymbol{\Sigma}_\ell\}_{\ell=1}^k$ respectively, if for $\boldsymbol{x}_i \in \mathcal{C}_\ell$*

$$\boldsymbol{x}_i = \boldsymbol{\mu}_\ell + \boldsymbol{\Sigma}_\ell^{\frac{1}{2}} \boldsymbol{z}_i \quad \textit{with} \quad \boldsymbol{z}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_p)$$

We further define the following quantities which shall be used subsequently

- (Data) $n_\ell = |\mathcal{C}_\ell|$ the cardinality of class $\mathcal{C}_\ell$ and $c_\ell = \lim_n \frac{n_\ell}{n} \in (0,1)$.
- (Means) $\bar{\boldsymbol{\mu}} = \sum_{\ell=1}^k c_\ell \boldsymbol{\mu}_\ell$ and $\bar{\boldsymbol{\mu}}_\ell = \boldsymbol{\mu}_\ell - \bar{\boldsymbol{\mu}}$.
- (Covariances) $\bar{\boldsymbol{\Sigma}} = \sum_{\ell=1}^k c_\ell \boldsymbol{\Sigma}_\ell$ and $\bar{\boldsymbol{\Sigma}}_\ell = \boldsymbol{\Sigma}_\ell - \bar{\boldsymbol{\Sigma}}$.

Under the GMM model in Definition 1, we recall the growth rate assumptions made in [CBG+16], which are essentially, as $p \to \infty$,

- (Data) $p/n \to c \in (0, \infty)$ and $n_\ell / n \to (0,1)$.
- (Means) $\limsup_p \max_\ell \|\bar{\boldsymbol{\mu}}_\ell\| < \infty$.
- (Covariances) $\limsup_p \max_\ell \|\boldsymbol{\Sigma}_\ell\| < \infty$ and $\limsup_p \max_\ell \frac{1}{\sqrt{p}} \operatorname{tr} \bar{\boldsymbol{\Sigma}}_\ell < \infty$.

**Remark 1.3** (On the non-trivial growth rates). *The above assumptions essentially ensure, as discussed in [CBG+16], that the classification is neither too easy nor too hard as the dimension grows to infinity. For instance, in the case of binary classification (i.e., $k = 2$), these assumptions imply $\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\| = \mathcal{O}(1)$ and $\frac{1}{\sqrt{p}} \operatorname{tr}(\boldsymbol{\Sigma}_1 - \boldsymbol{\Sigma}_2) = \mathcal{O}(1)$ as $p \to \infty$. Thus if $\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|$ and $\frac{1}{\sqrt{p}} \operatorname{tr}(\boldsymbol{\Sigma}_1 - \boldsymbol{\Sigma}_2)$ increase with p, then a simple Bayesian analysis will reveal that the classification will become too easy even with a trivial algorithm, in contrast, if the two quantities vanish as p grows large, the classification becomes theoretically impossible whatever the chosen algorithm. In particular, $\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\| = \mathcal{O}(1)$ unfolds from the following: Consider a Gaussian mixture of two classes such that $\mathcal{C}_\ell : \boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_\ell, \boldsymbol{I}_p)$ for $\ell \in \{1, 2\}$ (i.e., $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{I}_p$). In the case where $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are perfectly known, one has the following optimal decision by the Neyman-Pearson test, specifically for some $\boldsymbol{x} \in \mathcal{C}_1$*

$$(\boldsymbol{x} - \boldsymbol{\mu}_2)^\mathsf{T} (\boldsymbol{x} - \boldsymbol{\mu}_2) - (\boldsymbol{x} - \boldsymbol{\mu}_1)^\mathsf{T} (\boldsymbol{x} - \boldsymbol{\mu}_1) \underset{\mathcal{C}_1}{\overset{\mathcal{C}_2}{\lessgtr}} \log \frac{\det(\boldsymbol{\Sigma}_1)}{\det(\boldsymbol{\Sigma}_2)} = 0$$

*Decomposing $\boldsymbol{x} = \boldsymbol{\mu}_1 + \boldsymbol{z}$ with $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_p)$, the above test can be equivalently written as*

$$g(\boldsymbol{z}) = \frac{1}{p} \|\Delta\boldsymbol{\mu}\|^2 + \frac{2}{p} (\Delta\boldsymbol{\mu})^\mathsf{T} \boldsymbol{z} \underset{\mathcal{C}_1}{\overset{\mathcal{C}_2}{\lessgtr}} 0 \quad \textit{where} \quad \Delta\boldsymbol{\mu} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$$

*Moreover, since $g(\boldsymbol{z})$ is a sum of $p$ independent random variables, we have by Lyapunov's central limit theorem 2.7, as $p \to \infty$*

$$v_g^{-\frac{1}{2}}(g(\boldsymbol{z}) - m_g) \xrightarrow{\mathcal{D}} \mathcal{N}(0,1) \quad \textit{with} \quad m_g = \frac{1}{p}\|\Delta\boldsymbol{\mu}\|^2,\ v_g = \frac{4}{p^2}\|\Delta\boldsymbol{\mu}\|^2$$

*Therefore, the asymptotic (as $p \to \infty$) classification performance of $\boldsymbol{x} \in \mathcal{C}_1$ is non-trivial if and only if the two quantities $m_g$ and $\sqrt{v_g}$ are of the same order, as such $\|\Delta\boldsymbol{\mu}\|$ must be $\mathcal{O}(1)$.*

Under the above assumptions a fundamental high-dimensional behavior occurs which specifically states that the point-wise distance $\frac{1}{p}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2$ *concentrates* around the quantity

$$\tau = \frac{2}{p}\operatorname{tr}\boldsymbol{\Sigma}$$

i.e., the between and within class data vectors are "equidistant" in high dimensions, essentially, one has with probability one that

$$\max_{1\leq i\neq j\leq n}\left|\frac{1}{p}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 - \tau\right| \xrightarrow{a.s.} 0$$

This result unfolds from the decomposition of the pair-wise distance $\frac{1}{p}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2$ into asymptotically controllable terms as follows. For some $\boldsymbol{x}_i \in \mathcal{C}_a$, we denote $\boldsymbol{z}_i = \frac{\boldsymbol{x}_i - \boldsymbol{\mu}_a}{\sqrt{p}}$ and let $\psi_i = \|\boldsymbol{z}_i\|^2 - \frac{1}{p}\operatorname{tr}\boldsymbol{\Sigma}_a$. We therefore have the following decomposition by [CBG+16] for $\boldsymbol{x}_i \in \mathcal{C}_a$ and $\boldsymbol{x}_j \in \mathcal{C}_b$ with $a \neq b \in [k]$

$$\begin{aligned}\frac{1}{p}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 &= \|\boldsymbol{z}_i - \boldsymbol{z}_j\|^2 + \frac{1}{p}\|\boldsymbol{\mu}_a - \boldsymbol{\mu}_b\|^2 + \frac{2}{\sqrt{p}}(\boldsymbol{\mu}_a - \boldsymbol{\mu}_b)^\mathsf{T}(\boldsymbol{z}_i - \boldsymbol{z}_j)\\ &= \tau + \frac{1}{p}\operatorname{tr}\bar{\boldsymbol{\Sigma}}_a + \frac{1}{p}\operatorname{tr}\bar{\boldsymbol{\Sigma}}_b + \psi_i + \psi_j - 2\boldsymbol{z}_i^\mathsf{T}\boldsymbol{z}_j\\ &\quad + \frac{1}{p}\|\bar{\boldsymbol{\mu}}_a - \bar{\boldsymbol{\mu}}_b\|^2 + \frac{2}{\sqrt{p}}(\bar{\boldsymbol{\mu}}_a - \bar{\boldsymbol{\mu}}_b)^\mathsf{T}(\boldsymbol{z}_i - \boldsymbol{z}_j)\end{aligned}$$

From this decomposition, it turns out that, except for $\tau$, all the other terms are at least of order $\mathcal{O}(p^{-\frac{1}{2}})$. Indeed, it is easily seen that $\psi_i = \mathcal{O}(p^{-\frac{1}{2}})$ and $\boldsymbol{z}_i^\mathsf{T}\boldsymbol{z}_j = \mathcal{O}(p^{-\frac{1}{2}})$ while $\frac{2}{\sqrt{p}}(\bar{\boldsymbol{\mu}}_a - \bar{\boldsymbol{\mu}}_b)^\mathsf{T}(\boldsymbol{z}_i - \boldsymbol{z}_j)$ is of order $\mathcal{O}(p^{-1})$.

This observation notably states that the entries of the kernel matrix $\boldsymbol{K}$ will converge to the same value $f(\tau)$ at a first order approximation, specifically with probability one

$$\max_{1\leq i\neq j\leq n}|\mathbf{K}_{ij} - f(\tau)| \xrightarrow{a.s.} 0$$

which suggests that $\boldsymbol{K}$ is approximated at a first order by the rank one matrix $f(\tau)\mathbf{1}_n\mathbf{1}_n^\mathsf{T}$. However, it should be noted that the uniform convergence on the entries of $\boldsymbol{K}$ does not imply at all a convergence in the spectral norm as we saw in the example of the sample covariance matrix from the previous subsection. Still, a finer Taylor expansion analysis and control of the vanishing matrices in spectral norm provides the actual mechanism and information encoded by the kernel matrix. Figure 1.2 visually confirms that the entries of $\boldsymbol{K}$ converge to the same value $f(\tau)$ at a first approximation. We particularly considered in this figure two classes $\mathcal{C}_1$ and $\mathcal{C}_2$ such that $\boldsymbol{\mu}_\ell = (-1)^\ell\boldsymbol{\mu}$ for $\boldsymbol{\mu} = [2, \mathbf{0}_{p-1}]^\mathsf{T} \in \mathbb{R}^p$

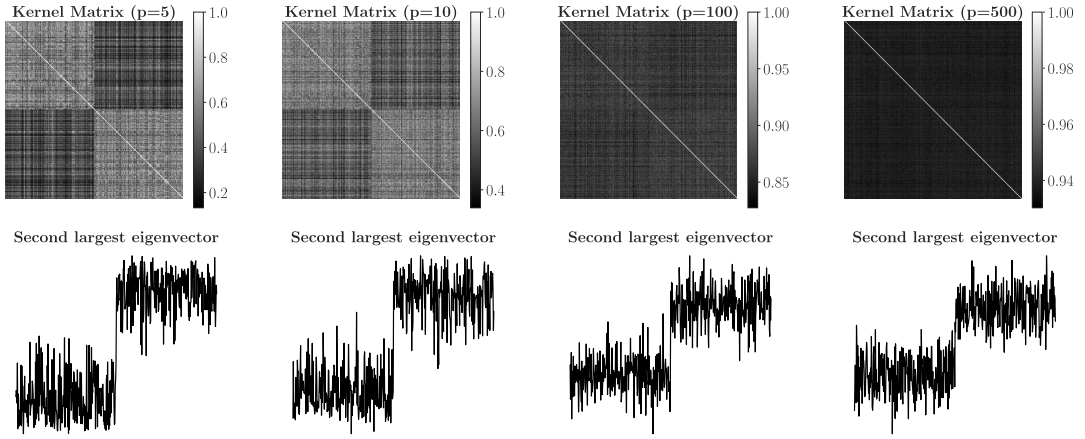



Figure 1.2: **(First line)** Kernel matrix with entries $K_{ij} = f(\frac{1}{p}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2)$ and **(second line)** its corresponding second largest eigenvector which contains clustering information. We considered the kernel function $f(t) = \exp(-t)$ and two classes of means $-\boldsymbol{\mu}$ and $+\boldsymbol{\mu}$ and identity covariance. $\boldsymbol{\mu} = [2, \mathbf{0}_{p-1}]^\intercal \in \mathbb{R}^p$, $n = 500$ and $p \in \{5, 10, 100, 500\}$.

and isotropic covariances. As we see from this figure, as $p$ gets large the behavior of the kernel matrix is fundamentally different from the low-dimension setting, in which case one visualizes a block-structure of the kernel matrix. In contrast, looking at the second largest eigenvector of $\boldsymbol{K}$, we see that the class structure is preserved even in the high-dimensional setting. Indeed, as explained in [CBG+16], the class information appears in the second order approximation of $\boldsymbol{K}$ since the first eigenvector with eigenvalue of order $\mathcal{O}(n)$ is proportional to $\mathbf{1}_n$, therefore not informative about the classes. Essentially, when $p, n \to \infty$, the kernel matrix $\boldsymbol{K}$ is asymptotically well approximated by a so-called spiked random matrix of the form[5]

$$\boldsymbol{K} = \boldsymbol{P}_f + f'(\tau)\boldsymbol{Z}^\intercal\boldsymbol{Z} + f''(\tau)\boldsymbol{W} + o_p(1)$$

where $\boldsymbol{P}_f$ is a low-rank informative matrix which contains information about the classes though the data statistics $\{\boldsymbol{\mu}_\ell\}_{\ell=1}^k$ and $\{\boldsymbol{\Sigma}_\ell\}_{\ell=1}^k$ and also depends on the kernel function $f$ through its local first and second derivatives at $\tau$. The remaining matrix terms are non-informative full-rank noise matrices which exhibit a spreading of eigenvalues in the spectrum of $\boldsymbol{K}$ in the same way as we saw in the previous subsection though the example of the sample covariance matrix.

**Remark 1.4** (On the analysis of the noise random matrices)**.** *The analysis of the noise terms in the approximation of $\boldsymbol{K}$ (e.g., the term $f'(\tau)\boldsymbol{Z}^\intercal\boldsymbol{Z}$) comes through generalizations of the Marčenko-Pastur result to structured data models as we discussed in the previous subsection in Remark 1.2.*

The main consequence of the above approximation is the access to the actual spectral behavior of $\boldsymbol{K}$, in particular, through the exact description of its dominant eigenvectors, thereby accessing the exact theoretical estimation of the performances of kernel spectral clustering as well as a wide range of kernel methods which rely on such kernel matrices

[5]The notation $\boldsymbol{A} = \boldsymbol{B} + o_p(1)$ means that $\|\boldsymbol{A} - \boldsymbol{B}\| = o(1)$ as $p \to \infty$ where $\|\cdot\|$ stands for the spectral norm.

Figure 1.3: Examples of images generated by the BigGAN model [BDS18].

$\boldsymbol{K}$. Indeed, an aftermath of such analysis allows one to "tune" the optimal kenrel function choice for the considered data. Moreover, based upon this fundamental result, Liao and Couillet gave the exact performance estimation of kernel LS-SVM in [LC17], while Mai and Couillet have analyzed and improved semi-supervised learning in [MC17] based on the analysis of $\boldsymbol{K}$.

## 1.2 From GMM to Concentration though GAN

The starting point to obtain the results of the previous section and more fundamentally to analyze the behavior of ML algorithms is to design so-called *deterministic equivalents* which basically *encode* the behavior of large random matrices. We will recall in the next chapter fundamental random matrix theory results along with some applications to simple machine learning models such as spectral clustering with the Gram matrix and classification with a linear classifier.

So far, the considered assumption on data to design such deterministic equivalents is a GMM model (see Definition 1) as developed by Benaych and Couillet [BGC16], where we recalled their main result in Theorem 2.5. However, real data (e.g., images) are *unlikely close* to simple Gaussian vectors and therefore one needs a more realistic model to describe them. Following the well-known quote of R. Feynman: *"What I cannot create, I do not understand"*, it is fundamentally important to be able to create real data in order to fully understand their nature. To this end though, generative models are of particular interest. In particular, since the advent of Generative Adversarial Nets [GPAM+14], it is now possible to create neural network models that can generate complex data structures. Some examples of artificially generated images by the BigGAN model [BDS18] are depicted in Figure 1.3. In particular, GAN generated data are constructed by applying successive *Lipschitz* transformations to high-dimensional Gaussian random vectors. Schematically,

$$\text{Real data} \approx \text{GAN data} = \mathcal{F}_L \circ \cdots \circ \mathcal{F}_1(\boldsymbol{z}) \quad \text{with} \quad \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$$

where the $\mathcal{F}_i$'s are essentially Lipschitz transformations.

On the other hand, the fundamental *concentration of measure* phenomenon [Led05a] states that Lipschitz-ally transformed Gaussian vectors are *concentrated vectors*, i.e., they satisfy the following concentration property: A random vector $\boldsymbol{x} \in E$ is said to be *concentrated* if for any 1-Lipschitz function $f : E \to \mathbb{R}$, for $q \geq 0$ there exists $C, \sigma > 0$ such

that

$$\forall t > 0, \quad \mathbb{P}\left\{|f(\boldsymbol{x}) - \mathbb{E}f(\boldsymbol{x})| \geq t\right\} \leq Ce^{-(t/\sigma)^q}$$

Consequently, GAN data are concentrated random vectors by construction, and therefore this class of random vectors constitute a more appropriate and convincing statistical model for realistic data compared to simple Gaussian vectors. The main objective of this thesis is to exploit the framework of concentrated vectors developed by Louart and Couillet [LC18b] relying on the earlier works of El Karoui [EK+10b], which is particularly motivated by the fact that GAN data belong to this class of random vectors, in order to analyze the behavior of various ML methods. We particularly provide in the following section some key results from this thesis that concern mainly the study of large Gram and kernel matrices under the concentration assumption (see Subsections 1.3.1-1.3.2) and further applications related to neural networks through the study of the Softmax layer in Subsection 1.3.3.

## 1.3 Some ML methods under Concentration

In this section we summarize the main findings of this thesis which basically describe the behavior of standard ML methods under the concentration assumption on data. Precisely, we will assume throughout the following subsections that data are distributed as a *Mixture of Concentrated Vectors* as per the following definition.

**Definition 2** (Mixture of Concentrated Vectors (MCV) [LC18b])**.** *Let $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathcal{M}_{p,n}$ be a data matrix which is constituted of $n$ random vectors distributed in $k$ different classes $\mathcal{C}_1, \ldots, \mathcal{C}_k$, such that the data classes are characterized by the moments, for $\boldsymbol{x}_i \in \mathcal{C}_\ell$*

$$\mathbb{E}\left[\boldsymbol{x}_i\right] = \boldsymbol{\mu}_\ell, \quad \mathbb{E}\left[\boldsymbol{x}_i \boldsymbol{x}_i^\intercal\right] = \boldsymbol{\Sigma}_\ell + \boldsymbol{\mu}_\ell \boldsymbol{\mu}_\ell^\intercal$$

*In particular, the data matrix $\boldsymbol{X}$ satisfy a concentration assumption in the sense that, for any 1-Lipschitz function $f : \mathcal{M}_{p,n} \to \mathbb{R}$ with $\mathcal{M}_{p,n}$ enrolled by the Frobinuous norm $\|\cdot\|_F$, for $q > 0$ there exists $C, \sigma > 0$[6] independent of $p$ and $n$ such that*

$$\forall t > 0, \quad \mathbb{P}\left\{|f(\boldsymbol{X}) - \mathbb{E}f(\boldsymbol{X})| \geq t\right\} \leq Ce^{-(t/\sigma)^q}$$

**Remark 1.5** (On the concentration of GAN random vectors)**.** *We will see in Chapter 3 that GAN random vectors have notably the same concentration as standard Gaussian vectors which unfolds from the fact that GAN's generators networks have controlled Lipschitz norm; i.e., for Gaussian $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$ inputs (as commonly considered) whose observable diameter does not depend on the dimension $d$, the observable diameter of GAN's outputs does not increase with the data dimension. Moreover, the concentration of the data matrix $\boldsymbol{X}$ implies the concentration of its columns vectors $\boldsymbol{x}_i$'s since $\boldsymbol{X} \mapsto \boldsymbol{X}\boldsymbol{j}_i = \boldsymbol{x}_i$ is 1-Lipschitz transformation, where $\boldsymbol{j}_i \in \mathbb{R}^n$ is the canonical vector defined as $(\boldsymbol{j}_i)_j = \delta_{i=j}$.*

In addition to the concentration assumption on data, we will further assume the classical random matrix theory regime as per the following assumption.

**Assumption 1** (Growth rate)**.** *As $p \to \infty$, assume*

1. *$p/n \to c \in (0, \infty)$; $|\mathcal{C}_\ell|/n \to c_\ell \in (0, 1)$.*
2. *The number of classes $k$ is bounded.*

[6] $\sigma$ is called the *observable diameter*.

### 1.3.1 Behavior of Gram Matrices

The first contribution of this thesis concerns the analysis of the behavior of the Gram matrix defined as $\boldsymbol{G} = \frac{1}{p}\boldsymbol{X}^\intercal\boldsymbol{X}$ under the concentration assumption in Definition 2 and the high-dimensional regime in Assumption 1. In particular, this contribution is presented in more details in Chapter 3 and is based on the following works:

- **(C1)** **MEA. Seddik**, C. Louart, M. Tamaazousti, R. Couillet, "*Random Matrix Theory Proves that Deep Learning Representations of GAN-data Behave as Gaussian Mixtures*", International Conference on Machine Learning (ICML'20), Online, 2020.
- **(C1')** **MEA. Seddik**, M. Tamaazousti, R. Couillet, "*Pourquoi les matrices aléatoires expliquent l'apprentissage ? Un argument d'universalité offert par les GANs*", Colloque francophone de traitement du signal et des images (Gretsi'19), Lille, France, 2019.

As will be recalled in Chapter 2, the spectral behavior of $\boldsymbol{G}$ can be analyzed through its resolvent

$$\boldsymbol{R}(z) = (\boldsymbol{G} + z\boldsymbol{I}_n)^{-1}$$

The main result from this contribution is to provide a *deterministic equivalent* (see Definition 6) for $\boldsymbol{R}(z)$ which is given by the following Theorem.

**Theorem 1.1** (Deterministic Equivalent for $\boldsymbol{R}(z)$)**.** *Under the concentration model in Definition 2, the growth rate Assumptions 1 and further assume that for all $\ell \in [k]$, $\|\boldsymbol{\mu}_\ell\| = \mathcal{O}(\sqrt{p})$[7]. Then, $\boldsymbol{R}(z)$ concentrates with an observable diameter of order $\mathcal{O}(p^{-\frac{1}{2}})$. Furthermore,*

$$\left\|\mathbb{E}\boldsymbol{R}(z) - \tilde{\boldsymbol{R}}(z)\right\| = \mathcal{O}\left(\sqrt{\frac{\log(p)}{p}}\right), \quad \tilde{\boldsymbol{R}}(z) = \frac{1}{z}\operatorname{diag}\left\{\frac{\boldsymbol{I}_{n_\ell}}{1+\delta_\ell^*(z)}\right\}_{\ell=1}^{k} + \frac{1}{pz}\boldsymbol{J}\Omega_z\boldsymbol{J}^\intercal$$

*with $\Omega_z = \boldsymbol{M}^\intercal\bar{\boldsymbol{Q}}(z)\boldsymbol{M} \odot \operatorname{diag}\left\{\frac{\delta_\ell^*(z)-1}{\delta_\ell^*(z)+1}\right\}_{\ell=1}^{k}$ where $\boldsymbol{M} = [\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k]$, $\boldsymbol{J} = [\boldsymbol{j}_1, \ldots, \boldsymbol{j}_k]$ and $\bar{\boldsymbol{Q}}(z) = \left(\frac{1}{ck}\sum_{\ell=1}^{k}\frac{\boldsymbol{\Sigma}_\ell + \boldsymbol{\mu}_\ell\boldsymbol{\mu}_\ell^\intercal}{1+\delta_\ell^*(z)} + z\boldsymbol{I}_p\right)^{-1}$ where $\delta^*(z) = [\delta_1^*(z), \ldots, \delta_k^*(z)]^\intercal$ is the unique fixed point of the system of equations for each $\ell \in [k]$*

$$\delta_\ell(z) = \frac{1}{p}\operatorname{tr}\left((\boldsymbol{\Sigma}_\ell + \boldsymbol{\mu}_\ell\boldsymbol{\mu}_\ell^\intercal)\left(\frac{1}{ck}\sum_{j=1}^{k}\frac{\boldsymbol{\Sigma}_j + \boldsymbol{\mu}_j\boldsymbol{\mu}_j^\intercal}{1+\delta_j(z)} + z\boldsymbol{I}_p\right)^{-1}\right).$$

Theorem 1.1 along with the deterministic equivalent of the sample covariance matrix developed in [LC18b] particularly generalize the result of [BGC16] to the class of concentrated vectors. A remarkable outcome from this result is that the behavior of the Gram matrix $\boldsymbol{G}$ depends strictly on the first and second statistical moments $\{\boldsymbol{\mu}_\ell\}_{\ell=1}^k$ and $\{\boldsymbol{\Sigma}_\ell\}_{\ell=1}^k$ of the data thereby providing a *universal behavior* of ML methods which rely on the Gram matrix regardless of the data distribution. This universality behavior has notably been verified (see Chapter 3) using GAN generated images as they satisfy the concentration assumption by design.

[7] This hypothesis is less restrictive in the sense that a $p$-dimensional vector with entries of order $\mathcal{O}(1)$ has an $\ell_2$-norm of order $\mathcal{O}(\sqrt{p})$. Furthermore, from a technical standpoint, this assumption also explains the contribution of the class-wise means in the $\delta_\ell(z)$'s subsequently.

### 1.3.2 Behavior of Kernel Matrices

The second contribution of this thesis concerns the analysis of large kernel matrices under the concentration hypothesis. The results of this part are presented in Section 4.1 and are particularly based on the following work:

> **(C2) MEA. Seddik**, M. Tamaazousti, R. Couillet, "*Kernel Random Matrices of Large Concentrated Data: The Example of GAN-generated Images*", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'19), Brighton, United-Kingdom, 2019.

Specifically, we have analyzed in this work the behavior of large kernel matrices of the form

$$\boldsymbol{K} = \left\{ f\left(\frac{1}{p}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right)\right\}_{i,j=1}^{n}$$

where $f : \mathbb{R}_+ \to \mathbb{R}_+$ is supposed to be three times continuously-differentiable in the vicinity of $\tau = \frac{2}{p}\operatorname{tr}\left(\sum_{\ell=1}^{k} c_\ell \boldsymbol{\Sigma}_\ell\right)$, and the data matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathcal{M}_{p,n}$ is assumed to be concentrated in the sense of Definition 2. Under these assumptions and further the non-trivial growth rate assumptions from Subsection 1.1.2, our first result states that *the between and within class data are "equidistant" in high-dimension independently from the classes*, specifically, for some $\delta > 0$ with probability $1 - \delta$

$$\max_{1\leq i\neq j\leq n}\left\{\left|\frac{1}{p}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 - \tau\right|\right\} = \mathcal{O}\left(\frac{\log\left(\frac{p}{\sqrt{\delta}}\right)^{\frac{1}{q}}}{\sqrt{p}}\right)$$

Following the same approach as [EK+10b, CBG+16], the kernel matrix $\boldsymbol{K}$ can therefore be Taylor expanded entry-wise leading to an approximation of it in spectral norm by a spiked random matrix model of the form[8]

$$\boldsymbol{K} = \boldsymbol{P}_f + f'(\tau)\boldsymbol{Z}^\intercal\boldsymbol{Z} + f''(\tau)\boldsymbol{W} + o_p(1)$$

where $\boldsymbol{P}_f$ is a low-rank informative matrix which contains information about the classes though the data statistics $\{\boldsymbol{\mu}_\ell\}_{\ell=1}^k$ and $\{\boldsymbol{\Sigma}_\ell\}_{\ell=1}^k$ and also depends on the kernel function $f$ through its local first and second derivatives at $\tau$. The remaining matrix terms are non-informative full-rank noise matrices which exhibit a spreading of eigenvalues in the spectrum of $\boldsymbol{K}$, in particular, the behavior of the term $f'(\tau)\boldsymbol{Z}^\intercal\boldsymbol{Z}$ is described by the deterministic equivalent from the previous subsection. See Section 4.1 for more details. As for the Gram matrix, $\boldsymbol{K}$ exhibits a *universal behavior* since its random matrix equivalent depends *strictly* on the classes statistics $\{\boldsymbol{\mu}_\ell\}_{\ell=1}^k$ and $\{\boldsymbol{\Sigma}_\ell\}_{\ell=1}^k$.

### 1.3.3 Beyond Kernels to Neural Networks

Kernel matrices appear naturally as a backbone of kernel methods. For instance, it has been shown in [Lia19] that the MSE[9] of random neural networks, the misclassification error rate of kernel ridge regression and the performance of kernel/random feature-based

[8] The notation $\boldsymbol{A} = \boldsymbol{B} + o_p(1)$ means that $\|\boldsymbol{A} - \boldsymbol{B}\| = o(1)$ as $p \to \infty$ where $\|\cdot\|$ stands for the spectral norm.

[9] Mean squared error.

spectral clustering methods, all depend *explicitly* on the eigenspectrum or on a certain functional of a particular random kernel/nonlinear Gram matrix. In this subsection, the aim is to go beyond kernel methods in order to analyze ML methods which have an *implicit* relationship with the input data, i.e., ML methods which are implicitly determined by (convex) optimization problems. In particular, relying on [MLC19] which studies the behavior of logistic regression under the RMT regime and using Gaussian assumptions on data, we push forward this study by assuming a $k$-class concentration model (see Definition 2) and therefore considering the more general Softmax classifier. Specifically, this subsection briefly presents our findings concerning the analysis of the Softmax classifier under the concentration hypothesis and the high-dimensional regime. This contribution is particularly detailed in Section 5.1 and is based on the following work:

**(C4) MEA. Seddik**, C.Louart, R. Couillet, M. Tamaazousti, "*The Unexpected Deterministic and Universal Behavior of Large Softmax Classifiers*", AISTATS 2021.

Specifically, given a set of one-hot-vector[10] labels $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n \in \mathbb{R}^k$ corresponding to each data vector $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^p$, the Softmax classifier consists in minimizing the following objective function

$$\mathcal{L}(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{\ell=1}^{k} y_{i\ell} \log p_{i\ell} + \frac{1}{2}\sum_{\ell=1}^{k} \lambda_\ell \|\boldsymbol{\mu}_\ell\|^2 \quad \text{with} \quad p_{i\ell} = \frac{\varphi(\boldsymbol{w}_\ell^\intercal \boldsymbol{x}_i)}{\sum_{j=1}^{k} \varphi(\boldsymbol{w}_j^\intercal \boldsymbol{x}_i)}$$

where $\boldsymbol{W} = [\boldsymbol{w}_1^\intercal, \ldots, \boldsymbol{w}_k^\intercal]^\intercal \in \mathbb{R}^{pk}$ stands for the class-weights vectors, $\lambda_\ell$'s are class-wise regularization parameters and $\varphi : \mathbb{R} \to \mathbb{R}_+$. Cancelling the loss function gradient with respect to each weight vector $\boldsymbol{w}_\ell$ yields

$$\lambda_\ell \boldsymbol{w}_\ell = -\frac{1}{n}\sum_{i=1}^{n}\left( y_{i\ell}\psi(\boldsymbol{w}_\ell^\intercal \boldsymbol{x}_i) - \frac{\phi(\boldsymbol{w}_\ell^\intercal \boldsymbol{x}_i)}{\sum_{j=1}^{k}\phi(\boldsymbol{w}_j^\intercal \boldsymbol{x}_i)} \sum_{j=1}^{k} y_{ij}\psi(\boldsymbol{w}_j^\intercal \boldsymbol{x}_i)\right)\boldsymbol{x}_i$$

with $\psi = \phi'/\phi$. Our approach consists in writing the above expression as a contracting fixed point equation of $\boldsymbol{W}$. Specifically, for well chosen $\tilde{\boldsymbol{x}}_i$ and $f_i$, we have

$$\boldsymbol{\Lambda}\boldsymbol{W} = \frac{1}{n}\sum_{i=1}^{n} \tilde{\boldsymbol{x}}_i f_i(\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}) \quad \Rightarrow \quad \boldsymbol{W} = \Psi(\boldsymbol{W})$$

where $\tilde{\boldsymbol{X}} = [\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_n] \in \mathcal{M}_{kp,kn}$, $f_i(\tilde{\boldsymbol{X}}^\intercal \boldsymbol{W}) \in \mathbb{R}^{kn}$ and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_k) \otimes \boldsymbol{I}_p \in \mathcal{M}_{kp}$. Therefore, $\Psi$ is requested to be $(1-\varepsilon)$-Lipschitz for some $\varepsilon > 0$ under some assumptions on $\varphi$ and the regularization parameters $\lambda_\ell$. Due to randomness in the data matrix $\boldsymbol{X}$, $\Psi$ is a contraction conditionally on some high-probable event $\mathcal{A}_X$ (See Section 5.1 for more details).

Under the concentration hypothesis from Definiton 2 and additional assumptions on $\varphi$ and $(\lambda_\ell)_{\ell\in[k]}$, our first result states that there exists and event $\mathcal{A}_X$ with $\mathbb{P}(\mathcal{A}_X) > 1 - Ce^{-cn}$ for some constants $C, c > 0$ such that the weights vector $\boldsymbol{W}$ concentrates conditionally on $\mathcal{A}_X$ with an observable diameter of order $\mathcal{O}\left(\sqrt{\log n/n}\right)$, which we compactly write as

$$(\boldsymbol{W} \mid \mathcal{A}_X) \propto \mathcal{E}_q\left(\sqrt{\frac{\log n}{n}}\right)$$

[10] Defined as $y_{i\ell} = 1$ if $\boldsymbol{x}_i \in \mathcal{C}_\ell$ and $y_{i\ell} = 0$ otherwise.

The further characterization of the statistics of the weights vector $\boldsymbol{W}$ requires the study of a resolvent of the form

$$\boldsymbol{Q} = \left( \boldsymbol{\Lambda} - \frac{1}{n} \tilde{\boldsymbol{X}} \boldsymbol{D} \tilde{\boldsymbol{X}}^\intercal \right)^{-1}$$

where $\boldsymbol{D}$ is some block-diagonal matrix whose entries depend on the data matrix $\tilde{\boldsymbol{X}}$. The above resolvent is studied through a deterministic equivalent (see Section 5.1 for more details), which implies the following result.

**Theorem 1.2** (Asymptotic statistics of the Softmax weights)**.** *Under the concentration hypothesis from Definiton 2 and supposing additional assumptions on $\varphi$ and $(\lambda_\ell)_{\ell \in [k]}$. There exists a deterministic mapping*

$$\mathcal{F}_{\boldsymbol{\mu},\boldsymbol{\Sigma}} = \mathcal{F}_{\boldsymbol{\mu},\boldsymbol{\Sigma}} \left( \{\boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell\}_{\ell=1}^k \right) : \mathbb{R}^{pk} \times \mathcal{M}_{pk} \to \mathbb{R}^{pk} \times \mathcal{M}_{pk}$$

*depending only on the data statistics $\{\boldsymbol{\mu}_\ell\}_{\ell=1}^k$ and $\{\boldsymbol{\Sigma}_\ell\}_{\ell=1}^k$, such that the equation*

$$(\boldsymbol{m}, \boldsymbol{C}) = \mathcal{F}_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{m}, \boldsymbol{C}) \quad \textit{with} \quad \boldsymbol{m} \in \mathbb{R}^{pk},\ \boldsymbol{C} \in \mathcal{M}_{pk}$$

*admits a unique solution $(\bar{\boldsymbol{m}}_W, \bar{\boldsymbol{C}}_W)$. Furthermore,*

$$\| \mathbb{E}[\boldsymbol{W}] - \bar{\boldsymbol{m}}_W \| = \mathcal{O}\left( \sqrt{\frac{\log n}{n}} \right) \quad \textit{and} \quad \| \mathbb{E}[\boldsymbol{W}\boldsymbol{W}^\intercal] - \bar{\boldsymbol{C}}_W \|_* = \mathcal{O}\left( \sqrt{\frac{\log n}{n}} \right)$$

Theorem 1.2 is fundamental in the sense that it states that the Softmax classifier retrieves information from the data *only* through the class-wise means and covariances, namely the statistics $\{\boldsymbol{\mu}_\ell\}_{\ell=1}^k$ and $\{\boldsymbol{\Sigma}_\ell\}_{\ell=1}^k$, thereby highlighting *the universal character* of the Softmax classifier. In particular, our result generalizes the study of Mai and Liao [MLC19] of the logistic regression classifier studied in their work under a GMM assumption on data, while we extended their results to a $k$-class MCV (see Definition 2). Moreover, based on Theorem 1.2, the performances of the Softmax classifier become theoretically predictable as per the following corollary.

**Corollary 1.1** (Generalization performance of the Softmax classifier)**.** *For $\ell \in [k]$ let some new test data $\boldsymbol{x} \in \mathcal{C}_\ell$ and define $p_\ell(\boldsymbol{x}) = \varphi(\boldsymbol{w}_\ell^\intercal \boldsymbol{x}) / \sum_{j \in [k]} \varphi(\boldsymbol{w}_j^\intercal \boldsymbol{x})$. There exists $\bar{\kappa}^\ell \in \mathbb{R}^{k-1}$ and $\bar{\boldsymbol{K}}^\ell \in \mathcal{M}_{k-1}$ both depending only on $\{\boldsymbol{\mu}_\ell\}_{\ell=1}^k$ and $\{\boldsymbol{\Sigma}_\ell\}_{\ell=1}^k$ such that the test error*

$$E_t(\boldsymbol{x} \in \mathcal{C}_\ell) = 1 - \mathbb{P}(\forall j \in [k] \setminus \{\ell\} : p_\ell(\boldsymbol{x}) \geq p_j(\boldsymbol{x}))$$

*is asymptotically close to*

$$\bar{E}_t(\boldsymbol{x} \in \mathcal{C}_\ell) = 1 - \mathbb{P}(\boldsymbol{Z}_\ell \in \mathbb{R}_+^{k-1}) \quad \textit{with} \quad \boldsymbol{Z}_\ell \sim \mathcal{N}(\bar{\kappa}^\ell, \bar{\boldsymbol{K}}^\ell)$$

In a nutshell, Corollary 1.1 affirms that the generalization error of the Softmax classifier is nothing but the cumulative distribution of a low-dimensional Gaussian vector, the mean and covariance of which depend strictly on the class wise means and covariances $\{\boldsymbol{\mu}_\ell\}_{\ell=1}^k$ and $\{\boldsymbol{\Sigma}_\ell\}_{\ell=1}^k$ of the input data. This notably demonstrates the *universality* property of the Softmax classifier regardless of the data distribution as long as it satisfies the concentration assumptions in Definition 2.

### 1.3.4 Summary of Section 1.3

The main core of this thesis concerns the analysis of the methods presented in the previous subsections under the concentration hypothesis on data as per Definition 2. A major outcome from these studies is the that the behavior of the studied methods *solely* depend on the class-wise means and covariances $\{\boldsymbol{\mu}_\ell\}_{\ell=1}^k$ and $\{\boldsymbol{\Sigma}_\ell\}_{\ell=1}^k$ of data, thereby highlighting the *universality* aspect of these methods. Our results notably support the validity of the GMM model in the high-dimensional regime as considered in [BGC16, CBG+16, Lia19, Mai19], and support the applicability of random matrix theory to the analysis of AI methods on realistic data as the surprising images generated by GANs. Note that the universality aspect does not suggest that real data are Gaussian vectors but rather that ML algorithms (at least for the studied examples) "see" the data only through its first and second order statistics.

## 1.4 Outline and Contributions

The rest of the manuscript is organized as follows: In Chapter 2 we provide some essential background and notions from random matrix theory along with some toy applications to ML through the so-called spiked models (see Section 2.3). We further provide in Section 2.4 some basic notions from the concentration of measure theory and recall some essential results from [LC18b] which will be used throughout the manuscript. In chapter 3 we present our first contribution concerning the analysis of large Gram matrices under the concentration assumption, from the works:

**(C1)** **MEA. Seddik**, C. Louart, M. Tamaazousti, R. Couillet, "*Random Matrix Theory Proves that Deep Learning Representations of GAN-data Behave as Gaussian Mixtures*", International Conference on Machine Learning (ICML'20), Online, 2020.

**(C1')** **MEA. Seddik**, M. Tamaazousti, R. Couillet, "*Pourquoi les matrices aléatoires expliquent l'apprentissage ? Un argument d'universalité offert par les GANs*", Colloque francophone de traitement du signal et des images (Gretsi'19), Lille, France, 2019.

In Chapter 4, we present our contributions to the analysis of kernel matrices. In particular, Section 4.1 presents our second contribution from the work:

**(C2)** **MEA. Seddik**, M. Tamaazousti, R. Couillet, "*Kernel Random Matrices of Large Concentrated Data: The Example of GAN-generated Images*", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'19), Brighton, United-Kingdom, 2019.

While Section 4.2 presents the analysis of kernel matrices of the type $f(\hat{\boldsymbol{\Sigma}})$ where $\hat{\boldsymbol{\Sigma}}$ stands for the sample covariance matrix, along with an application to the Sparse PCA problem. This work constitutes our third contribution from the work:

**(C3)** **MEA. Seddik**, M. Tamaazousti, R. Couillet, "*A Kernel Random Matrix-Based Approach for Sparse PCA*", International Conference on Learning Representations (ICLR'19), New Orleans, United-States, 2019.

In Chapter 5 we present applications to neural networks. In particular, Section 5.1 treats the study of the Softmax layer in neural networks which constitutes our fourth contribution from the work:

**(C4)** **MEA. Seddik**, C.Louart, R. Couillet, M. Tamaazousti, "*The Unexpected Deterministic and Universal Behavior of Large Softmax Classifiers*", AISTATS 2021.

Moreover, Section 5.2 presents an analysis of $\alpha$-Dropout in a single hidden-layer network which basically consists in dropping out features at random an replacing them with some arbitrary value $\alpha$. Our analysis exhibits a value of $\alpha \neq 0$ which maximizes the generalization accuracy of the studied network. This analysis is based on the following work:

**(C5)** **MEA. Seddik**, R. Couillet, M. Tamaazousti, "*A Random Matrix Analysis of Learning with $\alpha$-Dropout*", The art of learning with missing values ICML workshop (ICML'20), Online, 2020.

In the appendix Chapter B, we present some practical contributions conducted during this thesis. In particular, the work:

**(C6)** **MEA. Seddik**, M. Tamaazousti, J. Lin, "*Generative Collaborative Networks for Single Image SuperResolution*", Neurocomputing'2019.

presents deep learning models for single-image super-resolution. The last contribution presents compression methods for dense neural networks, which is based on the work:

**(C7)** **MEA. Seddik**, H.Essafi, A.Benzine, M.Tamaazousti, "*Lightweight Neural Networks from PCA & LDA Based Distilled Dense Neural Networks*", International Conference on Image Processing (ICIP'20), Online, 2020.

Finally, Chapter 6 concludes the manuscript and discusses perspectives and the appendix Chapter C presents the proofs of the different theoretical results of this thesis.

# Chapter 2

# Random Matrix Theory & Concentration of Measure Theory

**Contents**



Random matrix theory, originally, aims to describe the eigenvalue distribution (also known as the spectral measure) of large random matrices. There exist different approaches in the literature to analyze these random matrices such as the moments method, or approaches involving tools from free probabilities, see [Tao12] and the references therein for an overview. In this manuscript we will consider the Stieltjes transform approach, which is often presented as the central notion of the theory [BS+98a].

Tackling machine learning applications of random matrix theory, the description of random matrices eigenvalues distribution is not the central interest, and the objects of interest are more fundamentally sub-spaces which are described by largest eigenvectors of random matrices (in the case of unsupervised learning) or quadratic forms involving random matrices (in the case of supervised learning). Examples (but not limited to) of machine learning methods which rely on eigenvectors of random matrices are: principal component analysis (PCA) [WEG87], spectral clustering [VLBB08], some semi-supervised learning approaches [AMGS12], Least-squared support vector machines [SV99]

and random neural networks also known as extreme learning machines [HZS06].

Therefore, a more general notion than the Stieltjes transform is of central interest for machine learning application of the theory. This notion, generally referred to as the *resolvent* of large random matrices and will be the central notion of this manuscript. The resolvent of a matrix allows the access and consistent estimation of complex functionals of this matrix (such as the aforementioned quadratic forms), and fundamentally describes its spectral measure, permits the location and description of its isolated eigenvalues and eigenvectors, and provides estimations for bilinear forms, therefore permits the estimation of the performances of various machine learning models. We refer the reader to the works [Mai19, Lia19] where applications of the theory to various machine learning algorithms are presented relying on Gaussian mixture models.

We begin this section by first introducing the fundamental notions from random matrix theory, then give fundamental random matrix theory results and at the end present extensions involving tools from the concentration of measure theory which will allow the theory, as we will present subsequently, to find applications using realistic data.

## 2.1 Fundamental Random Matrix Theory Notions

### 2.1.1 The resolvent matrix notion

We first introduce the notion of the resolvent matrix.

**Definition 3** (Resolvent)**.** *For a given symmetric matrix $\boldsymbol{M} \in \mathcal{M}_p$, the resolvent $\boldsymbol{Q}(z)$ of $\boldsymbol{M}$ is defined, for $z \in \mathbb{C} \setminus \mathbb{R}^+$, as*

$$\boldsymbol{Q}(z) \equiv (\boldsymbol{M} - z\boldsymbol{I}_p)^{-1}. \tag{2.1}$$

We will systematically adopt the following notations for the resolvent matrix throughout this manuscript.

**Notation 1.** *Let $\boldsymbol{X} \in \mathcal{M}_{p,n}$ and $z \in \mathbb{C} \setminus \mathbb{R}^+$, the matrix $\boldsymbol{M}$ in Definition 3, for which we will define a resolvent, will be either of the form $\boldsymbol{X}\boldsymbol{X}^\intercal$ or $\boldsymbol{X}^\intercal\boldsymbol{X}$, we will thus denote*

$$\boldsymbol{Q}(z) \equiv \left(\boldsymbol{X}\boldsymbol{X}^\intercal - z\boldsymbol{I}_p\right)^{-1} \in \mathcal{M}_p, \quad \boldsymbol{R}(z) \equiv (\boldsymbol{X}^\intercal\boldsymbol{X} - z\boldsymbol{I}_n)^{-1} \in \mathcal{M}_n \tag{2.2}$$

Note that the dependence on $z$ will be some times omitted if there is no ambiguity and we will simply write $\boldsymbol{Q}$ or $\boldsymbol{R}$ instead of $\boldsymbol{Q}(z)$ or $\boldsymbol{R}(z)$ respectively.

### 2.1.2 Stieltjes transform and spectral measure

The resolvent $\boldsymbol{Q}$ is naturally related to the *empirical spectral measure* $\mu$ of $\boldsymbol{M}$, through the Stieltjes transform $m_\mu$, which we will be defined subsequently.

**Definition 4** (Empirical spectral measure)**.** *For a given symmetric matrix $\boldsymbol{M} \in \mathcal{M}_p$, the empirical spectral measure $\mu$ associated to $\boldsymbol{M}$ is defined through the normalized counting measure of the eigenvalues $\lambda_1, \ldots, \lambda_p$ of $\boldsymbol{M}$, specifically,*

$$\mu(x) \equiv \frac{1}{p}\sum_{i=1}^{p} \delta_{\lambda_i}(x). \tag{2.3}$$

The spectral measure $\mu$ of $\boldsymbol{M}$ is a (random if $\boldsymbol{M}$ is random) probability measure, since for all $x \in \mathbb{R}$, $\mu(x) \geq 0$ and since $\int_{\mathbb{R}} \mu(x)dx = 1$. Therefore, we can define its associated Stieltjes transform as follows.

**Definition 5.** *(Stieltjes Transform) Given some real probability measure $\mu$ with support $\mathcal{S}(\mu)$, the Stieltjes transform $q(z)$ is defined, for all $z \in \mathbb{C} \setminus \mathcal{S}(\mu)$, as*

$$q(z) \equiv \int_{\mathbb{R}} \frac{d\mu(\lambda)}{\lambda - z}. \tag{2.4}$$

Note that the Stieltjes transform $q(z)$ is closely related to its associated resolvent matrix $\boldsymbol{Q}$ through the following algebraic link.

$$q(z) = \frac{1}{p}\sum_{i=1}^{p}\int_{\mathbb{R}} \frac{\delta_{\lambda_i}(\lambda)}{\lambda - z} = \frac{1}{p}\sum_{i=1}^{p}\frac{1}{\lambda_i - z} = \frac{1}{p}\operatorname{tr}(\boldsymbol{Q}(z)) \tag{2.5}$$

The Stieltjes transform $q$ has various benefits and interesting properties, which are:

1. $q$ is complex analytic on its definition domain $\mathbb{C} \setminus \mathcal{S}(\mu)$.
2. $q(z)$ is bounded for $z \in \mathbb{C} \setminus \mathcal{S}(\mu)$ as
$$|q(z)| \leq \frac{1}{\operatorname{dist}(z, \mathcal{S}(\mu))}.$$
3. if $\Im(z) > 0$ then $\Im[q(z)] > 0$.
4. Since $q'(z) = \int_{\mathbb{R}}(t-z)^{-2}dt > 0$, $m$ is an increasing function with $\lim_{x\to\pm\infty} q(x) = 0$ if $\mathcal{S}(\mu)$ is bounded.

The Stieltjes transform $q$ as its name implies (transform), admits an inverse formula which notably provides the recovery of is underlying spectral measure $\mu$, as per the following theorem.

**Theorem 2.1** (Inverse formula of the Stieltjes transform)**.** *Let $a, b$ be some continuity points of the probability measure $\mu$, therefore the segment $[a, b]$ is measurable with $\mu$ and we precisely have*

$$\mu([a,b]) = \frac{1}{\pi}\lim_{\epsilon\to 0}\int_a^b \Im[q(x+i\epsilon)]dx. \tag{2.6}$$

*Moreover, if $\mu$ admits a density function $f$ at some point $x$, i.e., $\mu(x)$ is differentiable in a neighborhood of $x$ with $\lim_{\epsilon\to 0}\epsilon^{-1}\mu([x-\epsilon/2, x+\epsilon/2]) = f(x)$, then we have the inverse formula*

$$f(x) = \frac{1}{\pi}\lim_{\epsilon\to 0}\Im[q(x+i\epsilon)]. \tag{2.7}$$

*And finally, if $\mu$ has an isolated mass at some point $x$, then*

$$\mu(\{x\}) = -\frac{1}{\pi}\lim_{\epsilon\to 0} i\,\epsilon\, q(x+i\epsilon) \tag{2.8}$$

The resolvent matrix $\boldsymbol{Q}$ can be viewed as a matrix-valued Stieltjes transform. Indeed, relying on [HLN+07], the definition of the Stieltjes transform can be extended to $p \times p$ matrix-valued positive measures $\boldsymbol{M}(d\lambda)$ in the sense that

$$\mu_{\boldsymbol{v}}(d\lambda) \equiv \boldsymbol{v}^{\intercal}\boldsymbol{M}(d\lambda)\boldsymbol{v}, \tag{2.9}$$

is a positive real-valued measure for all $\boldsymbol{v} \in \mathbb{R}^p$. Using the spectral decomposition $\boldsymbol{M} = \boldsymbol{U}\,\mathrm{diag}\{\lambda_i\}_{i=1}^p \boldsymbol{U}^\intercal$ where $\lambda_1, \ldots, \lambda_p$ stand for the eigenvalues of $\boldsymbol{M}$, the resolvent matrix $\boldsymbol{Q}(z)$ of $\boldsymbol{M}$ can be written as

$$\boldsymbol{Q}(z) = \int_{\mathbb{R}} \frac{\boldsymbol{M}(d\lambda)}{\lambda - z} = \boldsymbol{U}\,\mathrm{diag}\left\{\frac{1}{\lambda_i - z}\right\}_{i=1}^p \boldsymbol{U}^\intercal \tag{2.10}$$

And therefore, $\boldsymbol{Q}(z)$ verifies similar properties as the real-valued Stieltjes transform, in particular, $\boldsymbol{Q}(z)$ is complex analytic and satisfies $\|\boldsymbol{Q}\| \leq \mathrm{dist}(z, \mathcal{S}(\mu))^{-1}$.

### 2.1.3 Cauchy's integral and statistical inference

Since $\boldsymbol{Q}$ is complex analytic, it benefits form complex analysis tools which allow us to infer statistical quantities involving complex functionals of $\boldsymbol{M}$. In particular, we have the following theorem which provides an elegant bridge between the resolvent and Cauchy's integral theorem.

**Theorem 2.2** (Cauchy's integral)**.** *Given $\Gamma \subset \mathbb{C}$ some positively oriented contour and a complex function $\zeta(z)$ which is analytic in a region containing the contour $\Gamma$, we have*

- *if $\Gamma$ is surrounding some $z' \in \mathbb{C}$, then $\zeta(z') = -\frac{1}{2\pi i}\oint_\Gamma \frac{\zeta(z)}{z'-z}dz$;*
- *otherwise, $\frac{1}{2\pi i}\oint_\Gamma \frac{\zeta(z)}{z'-z}dz = 0$.*

Using this result, one can immediately access *linear functionals of the eigenvalues $\lambda_1, \ldots, \lambda_p$ of $\boldsymbol{M}$* and using the Stieltjes transform or the resolvent matrix $\boldsymbol{Q}(z)$ by

$$\frac{1}{p}\sum_{i=1}^p \zeta(\lambda_i) = -\frac{1}{2\pi i}\oint_\Gamma \zeta(z)\frac{1}{p}\,\mathrm{tr}(\boldsymbol{Q}(z))dz = -\frac{1}{2\pi i}\oint_\Gamma \zeta(z)q(z)dz \tag{2.11}$$

for all $\zeta$ complex analytic in a compact neighborhood of $\mathcal{S}(\mu)$, the support of $\mu$.

In addition to estimate funcionnals of the eigenvalues values of $\boldsymbol{M}$, one can access its eigenvectors and eigenspaces with similar arguments. Indeed, through the spectral decomposition of $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\intercal$ with $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_p] \in \mathcal{M}_p$ and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_p)$, we have

$$\boldsymbol{Q}(z) = \sum_{i=1}^p \frac{\boldsymbol{u}_i\boldsymbol{u}_i^\intercal}{\lambda_i - z}. \tag{2.12}$$

And if some eigenvalue $\lambda_i$ is of multiplicity one, we can directly access its corresponding eigenvector $\boldsymbol{u}_i$ through

$$\boldsymbol{u}_i\boldsymbol{u}_i^\intercal = -\frac{1}{2\pi i}\oint_{\Gamma_i} \boldsymbol{Q}(z)dz, \tag{2.13}$$

where $\Gamma_{\lambda_i}$ a positive oriented contour surrounding the eigenvalue $\lambda_i$. We will be particularly interested in measuring the alignment between $\boldsymbol{u}_i$ and some deterministic vector $\boldsymbol{u}$, so the evaluation of the quantity

$$|\boldsymbol{u}^\intercal\boldsymbol{u}_i|^2 = -\frac{1}{2\pi i}\oint_{\Gamma_{\lambda_i}} \boldsymbol{u}^\intercal\boldsymbol{Q}(z)\boldsymbol{u}dz \tag{2.14}$$

Note that the resolvent matrix $\boldsymbol{Q}$ provides access to the quantities $\frac{1}{p}\sum_i \zeta(\lambda_i)$ and $|\boldsymbol{u}^\intercal\boldsymbol{u}_i|$ through scalar quantities and quadratic forms involving $\boldsymbol{Q}$, i.e., the quantities $\mathrm{tr}(\boldsymbol{Q}(z))$ and $\boldsymbol{u}^\intercal\boldsymbol{Q}(z)\boldsymbol{u}$. The estimation of these quantities will be notably possible through the notion of *deterministic equivalent* [HLN+07] which will be used as a proxy for statistical inference. The next subsection provides a description for this notion.

### 2.1.4 Deterministic and random equivalents

Throughout this manuscript the matrix $\boldsymbol{M}$ is a *large dimensional random matrix*, which is of either the following forms: 1. a sample covariance matrix $\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\intercal$; 2. a Gram matrix $\frac{1}{p}\boldsymbol{X}^\intercal\boldsymbol{X}$; 3. a kernel covariance matrix $f\left(\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\intercal\right)$ where $f$ is applied entry-wise; 4. or a kernel random of the form $\left\{f\left(\frac{1}{p}\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2\right)\right\}_{i=1}^n$, where $\boldsymbol{X}=[\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n]\in\mathcal{M}_{p,n}$ is a random matrix and $f$ a function with some regularity conditions. In the case of linear models, i.e., matrices of the form $\boldsymbol{M}=\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\intercal$ and under some growth rate conditions, the associated resolvent matrix $\boldsymbol{Q}(z)$ has a deterministic behavior [HLN+07] and therefore equivalent in some sense to a deterministic matrix $\bar{\boldsymbol{Q}}(z)$. Whereas, for kernel models involving some non-linear function $f$, the internal mechanism of the matrix $\boldsymbol{M}$ is accessible through a random matrix equivalent [EK+10b, CBG+16] $\tilde{\boldsymbol{M}}$ which results from a linearization of $\boldsymbol{M}$ and will allow the study of $\boldsymbol{M}$ via a deterministic equivalent of the resolvent of $\tilde{\boldsymbol{M}}$.

The notion of *deterministic equivalent* of a resolvent matrix $\boldsymbol{Q}$ is related to the existence of a *non-asymptotic* deterministic matrix having, in probability or almost surely, the same scalar observations as the random observations through $\boldsymbol{Q}$. This character is notably a manifestation of the concentration of measure phenomenon which we will discuss at the end of this chapter. We therefore have the following definition for the notion of deterministic equivalents [HLN+07].

**Definition 6** (Deterministic Equivalent)**.** *A squared deterministic matrix $\bar{\boldsymbol{Q}}\in\mathcal{M}_p$ is said to be a deterministic equivalent for the symmetric random matrix $\boldsymbol{Q}\in\mathcal{M}_p$ if, for all deterministic matrix $\boldsymbol{A}\in\mathcal{M}_p$ and vectors $\boldsymbol{a},\boldsymbol{b}\in\mathbb{R}^p$ of bounded norms (operator and Euclidean, respectively), we have, as $p\to\infty$, with some probability or almost surely*

$$\frac{1}{p}\operatorname{tr}\boldsymbol{A}(\boldsymbol{Q}-\bar{\boldsymbol{Q}})\to 0,\quad \boldsymbol{a}^\intercal(\boldsymbol{Q}-\bar{\boldsymbol{Q}})\boldsymbol{b}\to 0 \tag{2.15}$$

**Remark 2.1** (On the definition of deterministic equivalents)**.** *Definition 6 can be compactly expressed thanks to the* linear concentration *notion introduced by [LC18b, Definition 2.1] that is particularly adapted with concentration of measure theory presented subsequently in Section 2.4. Specifically, a deterministic matrix $\bar{\boldsymbol{Q}}\in\mathcal{M}_p$ is a deterministic equivalent of a random matrix $\boldsymbol{Q}\in\mathcal{M}_p$ if for any bounded linear form $u:\mathcal{M}_p\to\mathbb{R}$ with a unit operator norm (i.e., for all $\boldsymbol{A}\in\mathcal{M}_p$, $|u(\boldsymbol{A})|\leq\|\boldsymbol{A}\|$) we have*

$$\forall t>0:\ \mathbb{P}\left\{|u(\boldsymbol{Q})-u(\bar{\boldsymbol{Q}})|\geq t\right\}\leq\alpha(t)$$

*where $\alpha:\mathbb{R}_+\to[0,1]$ is any non-increasing and left continuous concentration function. Note that $\mathcal{M}_p$ could be replaced with any normed space $(\mathcal{E},\|\cdot\|)$ but for convenience we restrict the notion to $\mathcal{M}_p$ enrolled with the operator norm $\|\cdot\|$.*

Therefore, Definition 6 allows to access spectral information about the random matrix $\boldsymbol{M}$. For instance, the Stieltjes transform of its (random) spectral measure and bilinear forms involving $\boldsymbol{Q}(z)$, the resolvent of $\boldsymbol{M}$. For instance, if a deterministic equivalent $\bar{\boldsymbol{Q}}(z)$ for $\boldsymbol{Q}(z)$ exists in the sense of Definition 6, we have that almost surely $\frac{1}{p}\operatorname{tr}(\boldsymbol{Q}(z)-\bar{\boldsymbol{Q}}(z))\to 0$ which implies that the Stieltjes transform $m$ of the spectral distribution $\mu$ converges to the deterministic quantity $\frac{1}{p}\operatorname{tr}\bar{\boldsymbol{Q}}(z)$. Similarly, and exploiting Definition 6 using the bilinear form involving $\boldsymbol{Q}(z)$, the (isolated) eigenvectors of $\boldsymbol{M}$ are thus accessible through the deterministic equivalent $\boldsymbol{Q}(z)$.

Note that to exhibit such deterministic equivalent, it suffices to have a control of $\|\mathbb{E}\boldsymbol{Q} - \bar{\boldsymbol{Q}}\|$ which generally unfolds from a control of the variance of the quantities $\frac{1}{p}\operatorname{tr}(\boldsymbol{A}\boldsymbol{Q})$ and $\boldsymbol{a}^\intercal \boldsymbol{Q}\boldsymbol{b}$. $\mathbb{E}\boldsymbol{Q}$ is also a deterministic equivalent for $\boldsymbol{Q}$ but is not tractable in practice, however, we can generally find $\bar{\boldsymbol{Q}}$ such that $\|\mathbb{E}\boldsymbol{Q} - \bar{\boldsymbol{Q}}\| \to 0$ where $\bar{\boldsymbol{Q}}$ is computable through fixed-point equations or in some rare cases in close form, such as the classical examples of Marchenko-Pastur and the semi-circle laws.

We consider the following notations when a deterministic equivalent $\bar{\boldsymbol{Q}}$ exists for a random resolvent matrix $\boldsymbol{Q}$, and when a random matrix equivalent $\tilde{\boldsymbol{M}}$ exists for a random matrix $\boldsymbol{M}$.

**Notation 2** (Deterministic equivalent)**.** *Let $\boldsymbol{Q} \in \mathcal{M}_p$ be a random matrix and $\bar{\boldsymbol{Q}} \in \mathcal{M}_p$ a deterministic matrix, we write*

$$\boldsymbol{Q} \leftrightarrow \bar{\boldsymbol{Q}} \tag{2.16}$$

*if $\bar{\boldsymbol{Q}}$ satisfies Definition 6.*

**Notation 3** (Random equivalent)**.** *Let $\boldsymbol{M}, \tilde{\boldsymbol{M}} \in \mathcal{M}_p$ be two random matrices, we write*

$$\boldsymbol{M} \leftrightsquigarrow \tilde{\boldsymbol{M}} \tag{2.17}$$

*if $\|\mathbb{E}[\boldsymbol{M} - \tilde{\boldsymbol{M}}]\| \to 0$.*

Note that we will denote, throughout the manuscript, deterministic equivalents with an up bar symbol (i.e., $\bar{\boldsymbol{Q}}$) and the random equivalents with an up tilde symbol (i.e., $\tilde{\boldsymbol{M}}$).

## 2.2 Fundamental Random Matrix Theory Results

This section presents fundamental random matrix theory results, which provide explanation about some machine learning algorithms. We particularly provide the historical result of Marchenko-Pastur [MP67] which describes the spectral distribution of large sample covariance matrices. Since machine learning algorithms are generally applied to data described by distinct "clusters", this section will also present the extension of the Marchenko-Pastur result to mixture models data [BGC16]. The nice thing about random matrix theory is that it relies on tools from different branches of mathematics such as complex analysis (as we previously saw), linear algebra and probability theory. We will first briefly present some essential matrix identities and statistical tools and then provide the aforementioned results with short sketches of proofs (for readability).

### 2.2.1 Matrix identities and key lemmas

As we discussed in the previous section, to exhibit a deterministic equivalent $\bar{\boldsymbol{Q}}(z)$ of a given resolvent random matrix $\boldsymbol{Q}(z)$, one needs to control the quantity $\|\mathbb{E}[\boldsymbol{Q}(z) - \bar{\boldsymbol{Q}}(z)]\|$. Since both $\boldsymbol{Q}(z)$ and $\bar{\boldsymbol{Q}}(z)$ are inverse of matrices, the following identity is of central interest.

**Lemma 2.1** (Resolvent identity)**.** *Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be some invertible matrices, we have*

$$\boldsymbol{A}^{-1} - \boldsymbol{B}^{-1} = \boldsymbol{A}^{-1}(\boldsymbol{B} - \boldsymbol{A})\boldsymbol{B}^{-1} \tag{2.18}$$

*Proof.* Simply through $\boldsymbol{A}^{-1}(\boldsymbol{B} - \boldsymbol{A})\boldsymbol{B}^{-1} = \boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{B}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{A}\boldsymbol{B}^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{B}^{-1}$. □

As we have discussed, the random matrix $\boldsymbol{M}$ can be of the form $\boldsymbol{X}\boldsymbol{X}^\intercal$ or $\boldsymbol{X}^\intercal\boldsymbol{X}$ for some random matrix $\boldsymbol{X} \in \mathcal{M}_{p,n}$, thus we have the following lemma which relates the resolvent matrices of $\boldsymbol{X}\boldsymbol{X}^\intercal$ and $\boldsymbol{X}^\intercal\boldsymbol{X}$.

**Lemma 2.2.** *For $\boldsymbol{X} \in \mathcal{M}_{p,n}$, we have*

$$\boldsymbol{X}(\boldsymbol{X}^\intercal\boldsymbol{X} - z\boldsymbol{I}_n)^{-1} = (\boldsymbol{X}\boldsymbol{X}^\intercal - z\boldsymbol{I}_p)^{-1}\boldsymbol{X}, \tag{2.19}$$

*for $z \in \mathbb{C} \setminus \{0\}$ distinct from the eigenvalues of $\boldsymbol{X}\boldsymbol{X}^\intercal$.*

We also have the following lemma, known as Sylvester's identity, which provides a link between the resolvents of $\boldsymbol{X}\boldsymbol{X}^\intercal$ and $\boldsymbol{X}^\intercal\boldsymbol{X}$ through the determinant operator.

**Lemma 2.3** (Sylvester's identity)**.** *For $\boldsymbol{X} \in \mathcal{M}_{p,n}$, we have*

$$\det(\boldsymbol{X}\boldsymbol{X}^\intercal - z\boldsymbol{I}_p) = \det(\boldsymbol{X}^\intercal\boldsymbol{X} - z\boldsymbol{I}_n)(-z)^{p-n}. \tag{2.20}$$

Sylvester's identity notably shows that $\boldsymbol{X}\boldsymbol{X}^\intercal$ and $\boldsymbol{X}^\intercal\boldsymbol{X}$ share the same non-zero eigenvalues except $n - p$ zero eigenvalues of $\boldsymbol{X}^\intercal\boldsymbol{X}$ if $n \geq p$. We thus have the following result which relates the Stieltjes transforms of the spectral measures $\mu_{\boldsymbol{X}\boldsymbol{X}^\intercal}$ and $\mu_{\boldsymbol{X}^\intercal\boldsymbol{X}}$ of the random matrices $\boldsymbol{X}\boldsymbol{X}^\intercal$ and $\boldsymbol{X}^\intercal\boldsymbol{X}$ respectively.

**Lemma 2.4.** *Let $\boldsymbol{X} \in \mathcal{M}_{p,n}$ with $n \geq p$, and $z \in \mathbb{C} \setminus \{0\}$ not an eigenvalue of $\boldsymbol{X}^\intercal\boldsymbol{X}$. Denote $\boldsymbol{Q}(z) \equiv (\boldsymbol{X}\boldsymbol{X}^\intercal - z\boldsymbol{I}_p)^{-1}$ and $\boldsymbol{R}(z) \equiv (\boldsymbol{X}^\intercal\boldsymbol{X} - z\boldsymbol{I}_p)^{-1}$, we have*

$$\operatorname{tr}\boldsymbol{Q}(z) = \operatorname{tr}\boldsymbol{R}(z) + \frac{n-p}{z}. \tag{2.21}$$

*In particular,*

$$\mu_{\boldsymbol{X}\boldsymbol{X}^\intercal} = \frac{n}{p}\mu_{\boldsymbol{X}^\intercal\boldsymbol{X}} + \frac{n-p}{p}\delta_0. \tag{2.22}$$

Generally, the key approach to prove random matrix theory results relies on *leave-one-out* or *perturbation* approach to handle the dependencies. This approach consists in removing some $i$-th contribution from the resolvent $\boldsymbol{Q}$ of $\boldsymbol{M}$ to build a resolvent $\boldsymbol{Q}_{-i}$ of $\boldsymbol{M}_{-i}$ deprived of the $i$-th contribution. The main idea being that $\boldsymbol{Q}$ and $\boldsymbol{Q}_{-i}$ still have the same behavior but $\boldsymbol{Q}_{-i}$ allows one to handle the dependencies. Technically, $\boldsymbol{Q}_{-i}$ is built using matrix identities such as the Woodbury identity given in the following lemma.

**Lemma 2.5** (Woodbury identity)**.** *For $\boldsymbol{M} \in \mathcal{M}_p$ and $\boldsymbol{U}, \boldsymbol{V} \in \mathcal{M}_{p,k}$, such that $\boldsymbol{M}$ and $\boldsymbol{M} + \boldsymbol{U}\boldsymbol{V}^\intercal$ are invertible, we have*

$$(\boldsymbol{M} + \boldsymbol{U}\boldsymbol{V}^\intercal)^{-1} = \boldsymbol{M}^{-1} - \boldsymbol{M}^{-1}\boldsymbol{U}(\boldsymbol{I}_k + \boldsymbol{V}^\intercal\boldsymbol{M}\boldsymbol{U})\boldsymbol{V}^\intercal\boldsymbol{M}^{-1} \tag{2.23}$$

The perturbation $\boldsymbol{U}\boldsymbol{V}^\intercal$ is generally of low rank $k$ and the special case $k = 1$, $\boldsymbol{U}$ and $\boldsymbol{V}$ reduces to vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^p$ respectively. In this case, we simply have the Sherman-Morrison formula given in the following lemma.

**Lemma 2.6** (Sherman-Morrison)**.** *Let $\boldsymbol{M} \in \mathcal{M}_p$ invertible and $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^p$ such that $1 + \boldsymbol{v}^\intercal\boldsymbol{A}^{-1}\boldsymbol{u} \neq 0$, we have*

$$(\boldsymbol{M} + \boldsymbol{u}\boldsymbol{v}^\intercal)^{-1} = \boldsymbol{M}^{-1} - \frac{\boldsymbol{M}^{-1}\boldsymbol{u}\boldsymbol{v}^\intercal\boldsymbol{M}^{-1}}{1 + \boldsymbol{v}^\intercal\boldsymbol{M}^{-1}\boldsymbol{u}}, \quad (\boldsymbol{M} + \boldsymbol{u}\boldsymbol{v}^\intercal)^{-1}\boldsymbol{u} = \frac{\boldsymbol{M}^{-1}\boldsymbol{u}}{1 + \boldsymbol{v}^\intercal\boldsymbol{M}^{-1}\boldsymbol{u}}. \tag{2.24}$$

We give in the following an example where these identities help to handle the dependencies in random matrices.

**Example 1.** *Suppose that* $\boldsymbol{Q}(z) = \left(\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\intercal - z\boldsymbol{I}_p\right)^{-1}$ *where* $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathcal{M}_{p,n}$ *and* $\boldsymbol{x}_i$*'s are independent random vectors. Further denote* $\boldsymbol{Q}_{-i}(z) = \left(\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\intercal - \frac{1}{n}\boldsymbol{x}_i\boldsymbol{x}_i^\intercal - z\boldsymbol{I}_p\right)^{-1}$ *the resolvent* $\boldsymbol{Q}(z)$ *deprived from the i-th vector* $\boldsymbol{x}_i$*. Using Lemma 2.6 we therefore have the identities*

$$\boldsymbol{Q}(z) = \boldsymbol{Q}_{-i}(z) - \frac{\boldsymbol{Q}_{-i}(z)\frac{1}{n}\boldsymbol{x}_i\boldsymbol{x}_i\boldsymbol{Q}_{-i}(z)}{1 + \frac{1}{n}\boldsymbol{x}_i^\intercal\boldsymbol{Q}_{-i}(z)\boldsymbol{x}_i}, \quad \boldsymbol{Q}(z)\boldsymbol{x}_i = \frac{\boldsymbol{Q}_{-i}(z)\boldsymbol{x}_i}{1 + \frac{1}{n}\boldsymbol{x}_i^\intercal\boldsymbol{Q}_{-i}(z)\boldsymbol{x}_i} \tag{2.25}$$

*where* $\boldsymbol{Q}_{-i}(z)$ *is independent of* $\boldsymbol{x}_i$ *by construction.*

Lemma 2.6 will also be of particular interest when handling informative random matrix models as we will present in Section 2.3. A useful application of Lemma 2.6 is when replacing $\boldsymbol{M} \leftarrow \boldsymbol{M} - z\boldsymbol{I}_p$ for $z \in \mathbb{C}$, and letting $\boldsymbol{u} = \gamma\boldsymbol{u}$ where $\gamma \in \mathbb{R}$, which leads to the following rank-1 perturbation lemma for the resolvent of $\boldsymbol{M}$.

**Lemma 2.7** (Perturbation lemma [SB95])**.** *Let* $\boldsymbol{A}, \boldsymbol{M} \in \mathcal{M}_p$ *some symmetric matrices,* $\boldsymbol{u} \in \mathbb{R}^p$*,* $\gamma \in \mathbb{R}$ *and* $z < 0$*, then*

$$\left|\operatorname{tr}\boldsymbol{A}(\boldsymbol{M} + \gamma\boldsymbol{u}\boldsymbol{u}^\intercal - z\boldsymbol{I}_p)^{-1} - \operatorname{tr}\boldsymbol{A}(\boldsymbol{M} - z\boldsymbol{I}_p)^{-1}\right| \leq \frac{\|\boldsymbol{A}\|}{|\Im(z)|} \tag{2.26}$$

Note that the bound in Lemma 2.7 does not depend on $\|\boldsymbol{u}\|$. In particular, denoting the (perturbed) resolvent $\boldsymbol{Q}_\gamma(z) \equiv (\boldsymbol{M} + \gamma\boldsymbol{u}\boldsymbol{u}^\intercal - z\boldsymbol{I}_p)^{-1}$, and letting $\boldsymbol{A} = \frac{1}{p}\boldsymbol{I}_p$, we obtain

$$\frac{1}{p}\operatorname{tr}\boldsymbol{Q}_\gamma(z) = \frac{1}{p}\operatorname{tr}\boldsymbol{Q}(z) + \mathcal{O}(p^{-1}), \tag{2.27}$$

which shows, since $\frac{1}{p}\operatorname{tr}\boldsymbol{Q}(z)$ is the Stieltjes transform of the empirical spectral measure of $\boldsymbol{M}$, that the spectral measure of $\boldsymbol{M} + \gamma\boldsymbol{u}\boldsymbol{u}^\intercal$ is asymptotically close to that of $\boldsymbol{M}$ for any $\boldsymbol{u}$, in the large limit of $p$.

We further have the following fundamental trace Lemma which is at the core of the study of various random matrix models.

**Lemma 2.8** (Trace Lemma [BS08])**.** *Let* $\boldsymbol{x} \in \mathbb{R}^p$ *a random vector with i.i.d. entries with zero mean, unit variance and finite* $2k$ *order moment for some* $k > 1$*. Let* $\boldsymbol{Q} \in \mathcal{M}_p$ *some deterministic or random (independent of* $\boldsymbol{x}$*) matrix, then there exists a constant* $C > 0$ *such that*

$$\mathbb{E}\left[\left|\frac{1}{p}\boldsymbol{x}^\intercal\boldsymbol{Q}\boldsymbol{x} - \frac{1}{p}\operatorname{tr}\boldsymbol{Q}\right|^k\right] \leq C\frac{\|\boldsymbol{Q}\|^k}{p^{\frac{k}{2}}}$$

*In particular, if* $\limsup_p \|\boldsymbol{Q}\| < \infty$*, and* $\boldsymbol{x}$ *has entries with finite fourth-order moment, we have by Markov's inequality and Borel Cantelli Lemma,*

$$\frac{1}{p}\boldsymbol{x}^\intercal\boldsymbol{Q}\boldsymbol{x} - \frac{1}{p}\operatorname{tr}\boldsymbol{Q} \xrightarrow{a.s.} 0$$

Lemma 2.8 basically shows that the quadratic form $\frac{1}{p}\boldsymbol{x}^\intercal\boldsymbol{Q}\boldsymbol{x}$ concentrates around its expectation at large $p$. This result will particularly be exploited in the following Subsection to prove the fundamental Marčenko-Pastur law.

### 2.2.2 The Marčenko-Pastur Law

Having set the principal tools and notions presented previously, we are now in place to state one of the fundamental random matrix theory results, namely the Marčenko-Pastur law.

**Theorem 2.3** (Marčenko-Pastur law [MP67]). *Let $\boldsymbol{X} \in \mathcal{M}_{p,n}$ be a random matrix with i.i.d. entries having zero mean, unit variance and bound fourth order moment. Then, as $p, n \to \infty$ with $p/n \to c \in (0, \infty)$, the empirical spectral distribution $\mu$ of $\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\intercal$ satisfies*

$$\mu \xrightarrow{a.s.} \mu_{MP}$$

*in the weak convergence sense, where $\mu_{MP}$ is a deterministic measure and particularly satisfies $\mu_{MP}(\{0\}) = \max\{0, 1 - c^{-1}\}$, moreover $\mu_{MP}$ has a continuous density function $f_{MP}$ on the compact support $[(1-\sqrt{c})^2, (1+\sqrt{c})^2]$ defined as*

$$f_{MP}(x) = \frac{1}{\sqrt{2\pi c x}} \sqrt{(x - \lambda_-)(\lambda_+ - x)}$$

*where $\lambda_\pm = (1 \pm \sqrt{c})^2$.*

*Proof.* The proof starts by writing the Stieltjes transform $m(z)$ of $\mu$ as

$$m(z) = \frac{1}{p} \operatorname{tr} \left( \frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\intercal - z\boldsymbol{I}_p \right)^{-1} = \frac{1}{p} \sum_{i=1}^{p} \left[ \left( \frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\intercal - z\boldsymbol{I}_p \right)^{-1} \right]_{ii}$$

and writing

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^\intercal \\ \boldsymbol{X}_{-1} \end{bmatrix}$$

therefore, for $\Im(z) > 0$,

$$\begin{aligned} \left[ \left( \frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\intercal - z\boldsymbol{I}_p \right)^{-1} \right]_{11} &= \left[ \begin{pmatrix} \frac{1}{n}\boldsymbol{x}_1^\intercal \boldsymbol{x}_1 - z & \frac{1}{n}\boldsymbol{x}_1^\intercal \boldsymbol{X}_{-1} \\ \frac{1}{n}\boldsymbol{X}_{-1}\boldsymbol{x}_1 & \frac{1}{n}\boldsymbol{X}_{-1}\boldsymbol{X}_{-1}^\intercal - z\boldsymbol{I}_{p-1} \end{pmatrix}^{-1} \right]_{11} \\ &= \frac{1}{-z - z\frac{1}{n}\boldsymbol{x}_1^\intercal \left( \frac{1}{n}\boldsymbol{X}_{-1}^\intercal \boldsymbol{X}_{-1} - z\boldsymbol{I}_n \right)^{-1} \boldsymbol{x}_1} \end{aligned}$$

where the last equality comes from standard block matrix inverse formula from linear algebra. Therefore, by Trace Lemma 2.8, as $p, n \to \infty$

$$\left[ \left( \frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\intercal - z\boldsymbol{I}_p \right)^{-1} \right]_{11} - \frac{1}{-z - z\frac{1}{n} \operatorname{tr} \left( \frac{1}{n}\boldsymbol{X}_{-1}^\intercal \boldsymbol{X}_{-1} - z\boldsymbol{I}_n \right)^{-1}} \xrightarrow{a.s.} 0$$

Moreover, since

$$\boldsymbol{X}^\intercal \boldsymbol{X} = \boldsymbol{X}_{-1}^\intercal \boldsymbol{X}_{-1} + \boldsymbol{x}_1 \boldsymbol{x}_1^\intercal$$

we have by Rank-1 perturbation Lemma 2.7

$$\left[ \left( \frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\intercal - z\boldsymbol{I}_p \right)^{-1} \right]_{11} - \frac{1}{-z - z\frac{1}{n} \operatorname{tr} \left( \frac{1}{n}\boldsymbol{X}^\intercal \boldsymbol{X} - z\boldsymbol{I}_n \right)^{-1}} \xrightarrow{a.s.} 0$$

and recalling Lemma 2.4, we have $\frac{1}{n}\operatorname{tr}\left(\frac{1}{n}\boldsymbol{X}^\intercal\boldsymbol{X}-z\boldsymbol{I}_n\right)^{-1}=\frac{1}{n}\operatorname{tr}\left(\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\intercal-z\boldsymbol{I}_p\right)^{-1}-\frac{n-p}{zn}$ we have

$$\left[\left(\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\intercal-z\boldsymbol{I}_p\right)^{-1}\right]_{11}-\frac{1}{1-\frac{p}{n}-z-z\frac{1}{n}\operatorname{tr}\left(\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\intercal-z\boldsymbol{I}_p\right)^{-1}}\xrightarrow{a.s.}0$$

Repeating the same procedure for the diagonal entries of $\left(\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\intercal-z\boldsymbol{I}_p\right)^{-1}$ up to $p$, and averaging, we have for $\Im(z)>0$

$$m(z)-\frac{1}{1-\frac{p}{n}-z-z\frac{p}{n}m(z)}\xrightarrow{a.s.}0$$

Therefore, $m(z)\xrightarrow{a.s.}m_{MP}(z)$ which is a solution of the *fixed point equation* (with positive branch)

$$m_{MP}(z)=\frac{1}{1-c-z-czm_{MP}(z)}$$

thus

$$m_{MP}(z)=\frac{1-c}{2cz}-\frac{1}{2c}+\frac{\sqrt{(\lambda_+-z)(z-\lambda_-)}}{2cz}$$

Finally, by the inverse Stieltjes Transform in Theorem 2.1, we have for $x\in[\lambda_-,\lambda_+]$

$$\lim_{\varepsilon\to0}\frac{1}{\pi}\Im[m_{MP}(x+i\varepsilon)]=\frac{\sqrt{(\lambda_+-z)(z-\lambda_-)}}{2\pi cx}$$

And for $x=0$,

$$\lim_{\varepsilon\to0}i\varepsilon\Im[m_{MP}(i\varepsilon)]=(1-c^{-1})\mathbf{1}_{c>1}$$

which concludes the proof. □

### 2.2.3 Random matrices of mixture models

From the signal processing or machine learning perspectives, one is interested in *inferring* and *identifying* patterns from the data. Naturally, this supposes that data are made by some *correlation* structures instead of pure noise. Therefore, generalizations of the Marčenko-Pastur result to *structured* data models is more relevant for signal processing and machine learning applications. A first generalization to the sample covariance matrix model has been proposed by Silverstein and Bai [SC95] which we recall in the following theorem.

**Theorem 2.4** (Sample covariance matrix [SC95])**.** *Let $\boldsymbol{\Sigma}\in\mathcal{M}_p$ be some nonnegative definite matrix with e.s.d. $\nu\to\bar{\nu}$ in the weak sense. Let $\boldsymbol{Z}\in\mathcal{M}_{p,n}$ a random matrix with i.i.d. entries having zero mean and unit variance. Consider the data matrix*

$$\boldsymbol{X}=\boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{Z}\in\mathcal{M}_{p,n}$$

*Then, as $p,n\to\infty$ with $p/n\to c\in(0,\infty)$, the e.s.d. $\mu$ of $\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\intercal\in\mathcal{M}_p$ satisfies*

$$\mu\xrightarrow{a.s.}\bar{\mu}$$

*in the weak convergence sense, and the Stieltjes transform $\bar{m}(z)$ of $\bar{\mu}$ is the unique solution with $\Im \bar{m}(z) > 0$ of the fixed point equation*

$$\bar{m}(z) = \left( -z + c \int \frac{t}{1 + t\bar{m}(z)} \bar{\nu}(dt) \right)^{-1}$$

*In particular, the limiting measure $\bar{\mu}$ is continuous on $\mathbb{R}^+$ and real analytic.*

Moreover, in practice, one can face data with hierarchical structure and thus can be distributed in different clusters with different covariance profiles, i.e., a *mixture* model. Benaych and Couillet [BGC16] have notably extended Theorem 2.4 under the Gaussian mixture model (see Definition 1) which is more relevant from the machine learning point of view.

**Theorem 2.5** (Gram matrices of Gaussian mixture models [BGC16]). *Let $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_k \in \mathcal{M}_p$ be some nonnegative definite matrices. Let a data matrix $\boldsymbol{X} = [\boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{(k)}] \in \mathcal{M}_{p,n}$ with $\boldsymbol{X}^{(\ell)} = [\boldsymbol{x}_1^{(\ell)}, \ldots, \boldsymbol{x}_{n_\ell}^{(\ell)}] \in \mathcal{M}_{p,n_\ell}$ and $\boldsymbol{x}_i^{(\ell)} = \boldsymbol{\Sigma}_\ell^{\frac{1}{2}} \boldsymbol{z}_i^{(\ell)}$ where $\boldsymbol{z}_i^{(\ell)}$ is a random Gaussian vector with i.i.d. zero mean and unit variance entries. Let the resolvents*

$$\boldsymbol{Q}(z) = \left( \frac{1}{p} \boldsymbol{X}\boldsymbol{X}^\intercal - z\boldsymbol{I}_p \right)^{-1}, \quad \boldsymbol{R}(z) = \left( \frac{1}{p} \boldsymbol{X}^\intercal \boldsymbol{X} - z\boldsymbol{I}_n \right)^{-1}$$

*Then, as $n_1, \ldots, n_k, p \to \infty$ with $n_\ell / n \to c_a \in (0,1)$, $p/n \to c \in (0, \infty)$ and $k$ being bounded, we have*

$$\begin{aligned} \boldsymbol{Q}(z) \leftrightarrow \bar{\boldsymbol{Q}}(z) &= -\frac{1}{z} \left( \boldsymbol{I}_p + \sum_{\ell=1}^{k} \frac{c_\ell}{c} g_\ell(z) \boldsymbol{\Sigma}_\ell \right)^{-1} \\ \boldsymbol{R}(z) \leftrightarrow \bar{\boldsymbol{R}}(z) &= \operatorname{diag}\left\{ g_\ell(z) \mathbf{1}_{n_\ell} \right\}_{\ell=1}^{k} \end{aligned}$$

*where $g_\ell(z)$ is the unique solution to the fixed point equation*

$$g_\ell(z) = -\frac{1}{z} \left( 1 + \bar{g}_\ell(z) \right)^{-1}, \quad \bar{g}_\ell(z) = -\frac{1}{p\, z} \operatorname{tr} \boldsymbol{\Sigma}_\ell \left( \boldsymbol{I}_p + \sum_{a=1}^{k} \frac{c_a}{c} g_a(z) \boldsymbol{\Sigma}_a \right)^{-1}$$

In essence, Theorem 2.5 provides deterministic equivalents for the resolvents $\boldsymbol{Q}(z)$ and $\boldsymbol{R}(z)$ which is more fundamental, as discussed in Subsection 2.1.3 than just describing the spectrum of the underlying random matrices. Indeed, such deterministic equivalents can then be used as proxies to infer the performances of the studied methods, an example of study concerns kernel methods as performed in [CBG+16]. In the next subsection, we provide two simple examples to illustrate how these deterministic can be exploited to study simple machine learning algorithms. In particular, we first provide a deterministic equivalent for a random matrix model of the form "*Information + Noise*" also referred to as *spiked models* in the random matrix theory community.

## 2.3 Connections with machine learning through spiked models

In this section, we consider a simple toy example model for the data, namely we consider data distributed in two classes $\mathcal{C}_1$ and $\mathcal{C}_2$ with opposite means and isotropic covariances.

Specifically, we consider a data matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathcal{M}_{p,n}$ such that, for $\boldsymbol{x}_i \in \mathcal{C}_\ell$ for $\ell \in \{1, 2\}$

$$\boldsymbol{x}_i = (-1)^\ell \boldsymbol{\mu} + \boldsymbol{z}_i \quad \text{with} \quad \boldsymbol{z}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_p) \tag{2.28}$$

where $\boldsymbol{\mu} \in \mathbb{R}^p$. Without loss of generality, supposing that the data are arranged in classes in the data matrix $\boldsymbol{X}$ and denoting $\boldsymbol{y} = [-1, \ldots, -1, +1, \ldots, +1]^\intercal \in \mathbb{R}^n$ the vector of labels, in matrix from the data matrix is given by

$$\boldsymbol{X} = \boldsymbol{\mu}\boldsymbol{y}^\intercal + \boldsymbol{Z} \quad \text{with} \quad \boldsymbol{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n] \in \mathcal{M}_{p,n} \tag{2.29}$$

The model in equation 2.29 falls within the class of so-called spiked random matrix models which have been largely studied in random matrix theory [BGN11, BAP+05]. In particular, exploiting the result of Hachem *et al.* [HLN+07], one can design deterministic equivalents (in the sense of Definition 6) for the corresponding resolvents of the sample covariance and Gram matrices defined respectively as follows, for $z \in \mathbb{C} \setminus \mathbb{R}_+$

$$\boldsymbol{Q}(z) = \left( \frac{1}{n} \boldsymbol{X}\boldsymbol{X}^\intercal - z\boldsymbol{I}_p \right)^{-1}, \quad \boldsymbol{R}(z) = \left( \frac{1}{n} \boldsymbol{X}^\intercal\boldsymbol{X} - z\boldsymbol{I}_n \right)^{-1} \tag{2.30}$$

We particularly need the following assumptions subsequently for the design of such deterministic equivalents. These assumptions particularly ensure that the above resolvents are of bounded spectral norm asymptotically when $p/n \to (0, \infty)$ as $n \to \infty$.

**Assumption 2** (Growth rate)**.** *As $n \to \infty$,*
*1. $p/n \to c \in (0, \infty)$; 2. $|\mathcal{C}_\ell|/n \to c_a \in (0, 1)$; 3. $\|\boldsymbol{\mu}\| = \mathcal{O}(1)$.*

Exploiting the result of [HLN+07], deterministic equivalents of the above resolvents are respectively given by

$$\boldsymbol{Q}(z) \leftrightarrow \bar{\boldsymbol{Q}}(z) = \left( q^{-1}(z)\boldsymbol{I}_p - \frac{1}{n} z r(z) \|\boldsymbol{y}\|^2 \boldsymbol{\mu}\boldsymbol{\mu}^\intercal \right)^{-1}$$
$$\boldsymbol{R}(z) \leftrightarrow \bar{\boldsymbol{R}}(z) = \left( r^{-1}(z)\boldsymbol{I}_n - \frac{1}{n} z q(z) \|\boldsymbol{\mu}\|^2 \boldsymbol{y}\boldsymbol{y}^\intercal \right)^{-1}$$

where $(q(z), r(z))$ are the unique solution to the fixed point system of equations

$$q(z) = \frac{-1}{z(1 + \frac{1}{n} \operatorname{tr} \bar{\boldsymbol{R}}(z))}, \quad r(z) = \frac{-1}{z(1 + \frac{1}{n} \operatorname{tr} \bar{\boldsymbol{Q}}(z))}$$

Since $\|\boldsymbol{y}\|^2 = n$ and denoting $\bar{\boldsymbol{y}} = \frac{1}{\sqrt{n}} \boldsymbol{y}$, we obtain the following result using the Sherman-Morrison identity from Lemma 2.6.

**Theorem 2.6** (Deterministic equivalents for $\boldsymbol{Q}(z)$ and $\boldsymbol{R}(z)$)**.** *Under Assumption 2, deterministic equivalents for $\boldsymbol{Q}(z)$ and $\boldsymbol{R}(z)$ as defined in equation 2.30 are given by*

$$\boldsymbol{Q}(z) \leftrightarrow \bar{\boldsymbol{Q}}(z) = q(z)\boldsymbol{I}_p - \frac{q^2(z)}{1 + (c + \|\boldsymbol{\mu}\|^2) q(z)} \boldsymbol{\mu}\boldsymbol{\mu}^\intercal$$
$$\boldsymbol{R}(z) \leftrightarrow \bar{\boldsymbol{R}}(z) = r(z)\boldsymbol{I}_n - \frac{r^2(z)\|\boldsymbol{\mu}\|^2}{1 + (1 + \|\boldsymbol{\mu}\|^2) r(z)} \bar{\boldsymbol{y}}\bar{\boldsymbol{y}}^\intercal$$

*where*

$$q(z) = \frac{1 - c - z + \sqrt{(1 - c - z)^2 - 4zc}}{2zc}, \quad r(z) = \frac{c - 1 - z + \sqrt{(c - 1 - z)^2 - 4z}}{2z}$$

We will provide in the following subsections two simple examples of how these deterministic equivalents can be exploited to infer the performances of ML algorithms. In particular, we first consider an example of unsupervised learning involving the Gram matrix $\frac{1}{n}\boldsymbol{X}^\intercal\boldsymbol{X}$ in Subsection 2.3.1 and then an example of supervised linear ridge regression involving the sample covariance matrix $\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\intercal$ in Subsection 2.3.2.

### 2.3.1 Simple example of unsupervised learning

As we discussed in the introduction, kernel spectral clustering is an unsupervised learning algorithm which aims at applying a clustering algorithm on the subspace corresponding to the largest eigenvalues of the kernel matrix. In this subsection, we consider the example of a linear kernel and we will illustrate with this example how Theorem 2.6 can be exploited in order to infer the performances of such algorithm. We therefore consider the kernel matrix defined as

$$\boldsymbol{K} = \frac{1}{n}\boldsymbol{X}^\intercal\boldsymbol{X} \tag{2.31}$$

From Theorem 2.6, the corresponding deterministic equivalent is

$$\boldsymbol{R}(z) \leftrightarrow \bar{\boldsymbol{R}}(z) = r(z)\boldsymbol{I}_n - \frac{r^2(z)\|\boldsymbol{\mu}\|^2}{1+(1+\|\boldsymbol{\mu}\|^2)r(z)}\bar{\boldsymbol{y}}\bar{\boldsymbol{y}}^\intercal$$

which namely shows that the behavior of $\boldsymbol{K}$ involves a noise term $r(z)\boldsymbol{I}_n$ and a rank-1 informative term $\propto \bar{\boldsymbol{y}}\bar{\boldsymbol{y}}^\intercal$, in terms of spectrum $\boldsymbol{K}$ corresponds to a *spiked* Marčenko-Pastur distribution, i.e., the spectrum of $\boldsymbol{K}$ will contain a *bulk* of eigenvalues with limiting law corresponding to the Marčenko-Pastur law and an isolated (above some phase transition) eigenvalue corresponding to the information about the two classes in data. Moreover, the eigenvector of $\boldsymbol{K}$ corresponding to the isolated eigenvalue will be aligned to the labels vector $\bar{\boldsymbol{y}}$ above some phase transition. Let us first determine the isolated eigenvalue in the spectrum of $\boldsymbol{K}$.

**Isolated eigenvalue.** We have the following proposition which provides the asymptotic largest eigenvalue of $\boldsymbol{K}$ under the high-dimensional regime.

**Proposition 2.1** (Isolated spike [BAP+05])**.** *Under Assumption 2, denoting $\lambda_{\max}$ the largest eigenvalue of $\boldsymbol{K}$, then*

$$\lambda_{\max} \xrightarrow{a.s.} \bar{\lambda}_{\max} = \begin{cases} (1+\|\boldsymbol{\mu}\|^2)\left(1+\frac{c}{\|\boldsymbol{\mu}\|^2}\right) > (1+\sqrt{c})^2 & \text{if } \|\boldsymbol{\mu}\|^2 \geq \sqrt{c} \\ (1+\sqrt{c})^2 & \text{otherwise} \end{cases}$$

*Proof.* Given the above form of the deterministic equivalent $\bar{\boldsymbol{R}}(z)$, clearly a spike $\lambda > (1+\sqrt{c})^2$ appears in the spectrum of $\boldsymbol{K}$ when $\bar{\boldsymbol{R}}(z)$ gets singular. Indeed, given the expression of $r(z)$, it is defined for real values in the interval $((1+\sqrt{c})^2, \infty)$, thus to find the isolated spike $\lambda$ it suffices to solve the equation $1+(1+\|\boldsymbol{\mu}\|^2)r(z) = 0$ in $z$ from which we obtain

$$\lambda = (1+\|\boldsymbol{\mu}\|^2)\left(1+\frac{c}{\|\boldsymbol{\mu}\|^2}\right)$$

Furthermore, to obtain the condition on $\|\boldsymbol{\mu}\|^2$ under which such a spike gets out of the right edge $\lambda > (1+\sqrt{c})^2$, we need to express $z$ evaluated at the spike $\lambda$ in terms of $m = \|\boldsymbol{\mu}\|^2$ and then study the resulting function. Indeed, $r(z)$ satisfies

$$zr^2(z) + (1+z-c)r(z) + 1 = 0$$

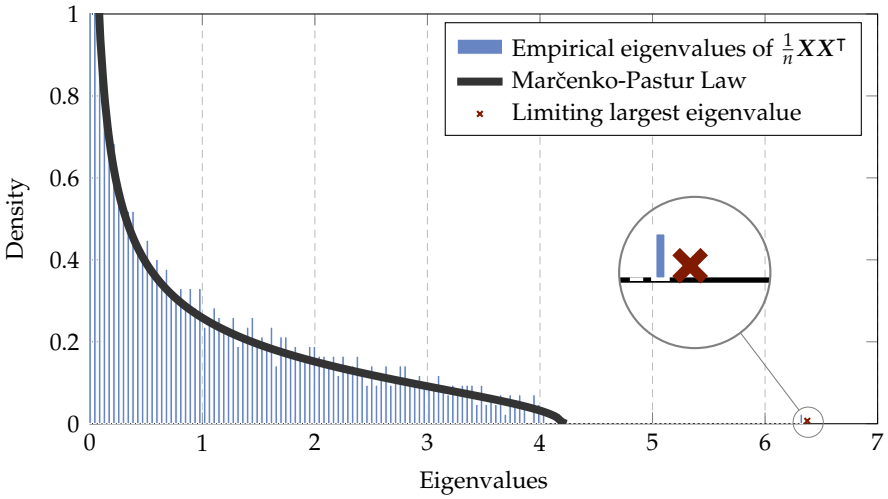



Figure 2.1: Histogram of the eigenvalues of the kernel matrix $\boldsymbol{K} = \frac{1}{n}\boldsymbol{X}^\intercal\boldsymbol{X}$ with $\boldsymbol{X}$ defined in equation 2.29. We considered the dimensions $p = 1100$, $n = 1000$ and $\boldsymbol{\mu} = [2, \mathbf{0}_{p-1}]^\intercal \in \mathbb{R}^p$.

from which we can express the reciprocal function

$$z(r) = \frac{c-1}{1+r} - \frac{1}{(1+r)r}$$

Besides, the spike $\lambda$ satisfies (where we recall $m = \|\boldsymbol{\mu}\|^2$)

$$r(\lambda) = \frac{-1}{1+m}, \quad 1 + r(\lambda) = \frac{m}{1+m}, \quad (1+r(\lambda))r(\lambda) = \frac{-m}{(1+m)^2}$$

Therefore, the spike $\lambda$ gets isolated from the main bulk at the following value

$$z_\lambda(m) = z(r(\lambda)) = \frac{(c-1)(1+m)}{m} + \frac{(1+m)^2}{m}$$

which defines a function *w.r.t.* $m$ that is first decreasing and then increasing for large values of $m$. Consequently, the value of $m$ which minimizes the above function corresponds to the phase transition at which the spike starts to get away from the bulk. Therefore, solving $z'_\lambda(m) = \frac{m^2-c}{m^2} = 0$ yields to $m = \sqrt{c}$ since $m \geq 0$. □

Proposition 2.1 shows that the largest eigenvalue of $\boldsymbol{K}$ converges to a limiting value $\bar{\lambda}_{\max}$ which depend on the *signal strength* only above some condition $\|\boldsymbol{\mu}\|^2 \geq \sqrt{c}$, i.e., only when the classes are theoretically separable (for a large enough value of $\|\boldsymbol{\mu}\|^2$). Figure 2.1 depicts the spectrum of the random kernel matrix $\boldsymbol{K}$, as theoretically predicted, the spectrum is made of a *bulk* of eigenvalues converging to the Marčenko-Pastur Law along with a largest eigenvalue – *spike* – which corresponds to the informative part of the model through the signal $\boldsymbol{\mu}$.

**Remark 2.2** (On the fluctuations of the spike [BAP+05]). *It has been shown in [BAP+05] that the largest eigenvalue of $\boldsymbol{K}$ concentrates around its converging value, and the underlying converges in law has been determined depending on the condition $\|\boldsymbol{\mu}\|^2 \geq \sqrt{c}$. Indeed, when $\|\boldsymbol{\mu}\|^2 < \sqrt{c}$ it has been shown that $p^{\frac{2}{3}}(1+\sqrt{c})^{-\frac{4}{3}}c^{-\frac{1}{2}}(\lambda_{\max} - (1+\sqrt{c})^2) \xrightarrow{\mathcal{D}} \mathcal{T}$, where $\mathcal{T}$ stands for the Tracy-Widom law. In contrast, when $\|\boldsymbol{\mu}\|^2 \geq \sqrt{c}$ then with $\nu = \left(\frac{(1+\|\boldsymbol{\mu}\|^2)^2}{c} - \frac{(1+\|\boldsymbol{\mu}\|^2)^2}{\|\boldsymbol{\mu}\|^4}\right) p^{\frac{1}{2}}$ we have $\nu(\lambda_{\max} - \bar{\lambda}_{\max}) \xrightarrow{\mathcal{D}} \mathcal{N}(0,1)$.*

We will further show in the following that the corresponding eigenvector to the largest eigenvalue of $\boldsymbol{K}$ gets aligned with the true class labels vector $\bar{\boldsymbol{y}}$ for sufficiently separable classes, i.e., when $\|\boldsymbol{\mu}\|$ is large enough.

**Largest eigenvector of $\boldsymbol{K}$.** Let $\hat{\boldsymbol{y}}$ be the largest eigenvector of $\boldsymbol{K}$. In particular, $\hat{y}_i$ stands for the score of the datum $\boldsymbol{x}_i$ to belong to class $\mathcal{C}_1$ if $\hat{y}_i < 0$ or class $\mathcal{C}_2$ otherwise. In order to estimate the performance of spectral clustering on the considered example, one is interested in evaluating the alignment $|\hat{\boldsymbol{y}}^\top \bar{\boldsymbol{y}}|^2$ between the estimated scores $\hat{\boldsymbol{y}}$ and the ground truth ones $\bar{\boldsymbol{y}}$. The deterministic equivalent $\bar{\boldsymbol{R}}(z)$ can therefore be exploited to evaluate the quantity $|\hat{\boldsymbol{y}}^\top \bar{\boldsymbol{y}}|^2$ as discussed in Subsection 2.1.3. Precisely,

$$|\hat{\boldsymbol{y}}^\top \bar{\boldsymbol{y}}|^2 = \frac{-1}{2\pi i} \oint_{\Gamma_{\lambda_{\max}}} \bar{\boldsymbol{y}}^\top \boldsymbol{R}(z) \bar{\boldsymbol{y}} dz = \frac{-1}{2\pi i} \oint_{\Gamma_{\bar{\lambda}_{\max}}} \bar{\boldsymbol{y}}^\top \bar{\boldsymbol{R}}(z) \bar{\boldsymbol{y}} dz + o_p(1) \tag{2.32}$$

where $\Gamma_x$ stands for a sufficiently small positively-oriented complex contour surrounding $x$ and $\bar{\lambda}_{\max} = \left(1 + \|\boldsymbol{\mu}\|^2\right)\left(1 + \frac{c}{\|\boldsymbol{\mu}\|^2}\right)$ is the converging value – as $p, n \to \infty$ – of $\lambda_{\max}$ the largest eigenvalue of $\boldsymbol{K}$. Note however that the above equality holds only when $\bar{\lambda}_{\max}$ is isolated, i.e., when it gets outside the bulk that is only valid if $\|\boldsymbol{\mu}\|^2 > \sqrt{c}$, as we saw in Proposition 2.1. Computing the above complex integral yields to the following result.

**Proposition 2.2** (Largest eigenvector [Pau07])**.** *Under Assumption 2, denoting $\hat{\boldsymbol{y}}$ the eigenvector of $\boldsymbol{K}$ corresponding to the largest eigenvalue, then*

$$|\hat{\boldsymbol{y}}^\top \bar{\boldsymbol{y}}|^2 \xrightarrow{a.s.} \frac{1 - c\|\boldsymbol{\mu}\|^{-4}}{1 + \|\boldsymbol{\mu}\|^{-2}} \cdot 1_{\|\boldsymbol{\mu}\|^2 > \sqrt{c}}$$

*Proof.* The result comes simply through the evaluation of the complex integral

$$\begin{aligned}
|\hat{\boldsymbol{y}}^\top \bar{\boldsymbol{y}}|^2 &= \frac{-1}{2\pi i} \oint_{\Gamma_{\bar{\lambda}_{\max}}} \bar{\boldsymbol{y}}^\top \bar{\boldsymbol{R}}(z) \bar{\boldsymbol{y}} dz + o_p(1) \\
&= \frac{-1}{2\pi i} \oint_{\Gamma_{\bar{\lambda}_{\max}}} r(z) - \frac{r^2(z)\|\boldsymbol{\mu}\|^2}{1 + (1 + \|\boldsymbol{\mu}\|^2) r(z)} dz + o_p(1) \\
&= -\operatorname{Res}_{\bar{\lambda}_{\max}} \left( \frac{r(z)(1 + r(z))}{1 + (1 + \|\boldsymbol{\mu}\|^2) r(z)} \right) + o_p(1) \\
&= -\lim_{z \to \bar{\lambda}_{\max}} \frac{(z - \bar{\lambda}_{\max}) r(z)(1 + r(z))}{1 + (1 + \|\boldsymbol{\mu}\|^2) r(z)} + o_p(1)
\end{aligned}$$

Using the Hospital rule, the above limit can therefore be evaluated as

$$|\hat{\boldsymbol{y}}^\top \bar{\boldsymbol{y}}|^2 = \frac{-r(\bar{\lambda}_{\max})(1 + r(\bar{\lambda}_{\max}))}{(1 + \|\boldsymbol{\mu}\|^2) r'(\bar{\lambda}_{\max})} + o_p(1)$$

It remains thus to evaluate $r$ and $r'$ at the limiting spike $\bar{\lambda}_{\max} = \left(1 + \|\boldsymbol{\mu}\|^2\right)\left(1 + \frac{c}{\|\boldsymbol{\mu}\|^2}\right)$. Indeed, we already know that $\bar{\lambda}_{\max}$ satisfies

$$1 + (1 + \|\boldsymbol{\mu}\|^2) r(\bar{\lambda}_{\max}) = 0 \quad \Rightarrow \quad r(\bar{\lambda}_{\max}) = \frac{-1}{1 + \|\boldsymbol{\mu}\|^2}$$

And since $r(z)$ satisfies

$$z r^2(z) + (1 + z - c) r(z) + 1 = 0$$

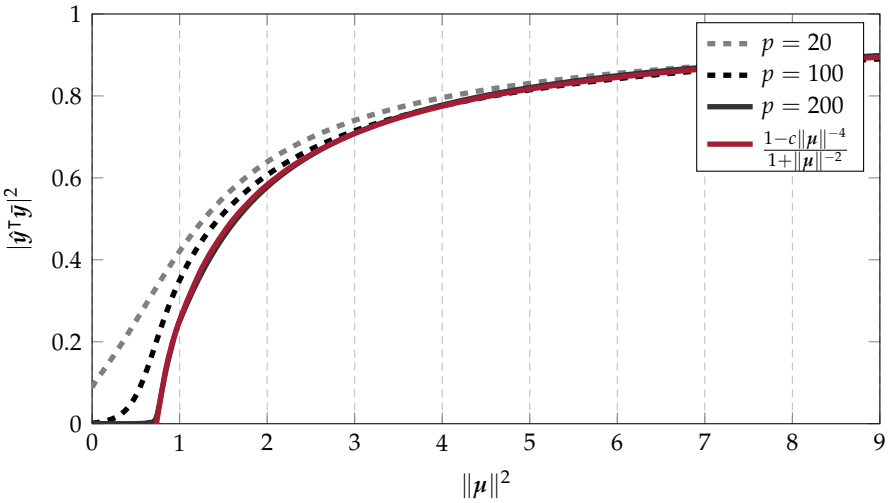



Figure 2.2: Simulated versus limiting values of $|\hat{\boldsymbol{y}}^\intercal \bar{\boldsymbol{y}}|^2$ for $\boldsymbol{K} = \frac{1}{n}\boldsymbol{X}^\intercal \boldsymbol{X}$ with $\boldsymbol{X} = \boldsymbol{Z} + \boldsymbol{\mu}\boldsymbol{y}^\intercal$. We considered $c = p/n = 1/2$, $\boldsymbol{\mu} = [\alpha, \boldsymbol{0}_{p-1}]^\intercal \in \mathbb{R}^p$ varying $\alpha$.

Taking the derivative w.r.t. $z$ and taking $z = \bar{\lambda}_{\max}$, we end up with

$$r'(\bar{\lambda}_{\max}) = \frac{-r(\bar{\lambda}_{\max})(1 + r(\bar{\lambda}_{\max}))}{2\bar{\lambda}_{\max} r(\bar{\lambda}_{\max}) + (1 + \bar{\lambda}_{\max} - c)}$$

□

Proposition 2.2 provides the asymptotic limit of the alignment $|\hat{\boldsymbol{y}}^\intercal \bar{\boldsymbol{y}}|^2$ between the true score labels $\bar{\boldsymbol{y}}$ and the estimated ones $\hat{\boldsymbol{y}}$ as the largest eigenvector of the kernel matrix $\boldsymbol{K}$, thereby providing the asymptotic performances[1] of spectral clustering when both $p$ and $n$ grow large. Figure 2.2 depicts simulated versus limiting values of $|\hat{\boldsymbol{y}}^\intercal \bar{\boldsymbol{y}}|^2$ which shows that as $p$ grow large with the ration $c = \lim_p p/n$ being constant, the simulated alignment gets more and more closer to its asymptotic estimation.

**Remark 2.3** (On other spiked models)**.** *Similar results as Proposition 2.2 hold for a wide range of spiked models, mainly of one of the following forms:*

$$\boldsymbol{K} = \frac{1}{n}\boldsymbol{Z}^\intercal \boldsymbol{Z} + \boldsymbol{P}, \quad \boldsymbol{K} = \frac{1}{n}\boldsymbol{Z}^\intercal (\boldsymbol{I}_p + \boldsymbol{P})\boldsymbol{Z}$$
$$\boldsymbol{K} = \frac{1}{n}(\boldsymbol{Z} + \boldsymbol{P})^\intercal (\boldsymbol{Z} + \boldsymbol{P}), \quad \boldsymbol{K} = \frac{1}{n}\boldsymbol{T}\boldsymbol{Z}^\intercal (\boldsymbol{I}_p + \boldsymbol{P})\boldsymbol{Z}\boldsymbol{T}$$

*where $\boldsymbol{T}$ and $\boldsymbol{P}$ are deterministic matrices with $\boldsymbol{P}$ of low rank, and $\boldsymbol{Z} \in \mathcal{M}_{p,n}$ a random matrix having zero mean and unit variance entries.*

In the following subsection, we will see how to exploit Theorem 2.6 in order to express the performances of linear regression to separate the two classes defined by the data model in equation 2.29.

[1] Let $\hat{\mathcal{C}}_i = \text{sign}(\hat{y}_i)$ be the estimated class $\mathcal{C}_i$ of the datum $\boldsymbol{x}_i$ such that $\bar{y}_i \hat{y}_i > 0$. Then with probability one, the accuracy is given by $\frac{1}{n}\sum_{i=1}^n \delta_{\mathcal{C}_i = \hat{\mathcal{C}}_i} \xrightarrow{a.s.} Q\left(\sqrt{\frac{\xi}{1-\xi}}\right)$ with $\xi = \frac{1-c\|\boldsymbol{\mu}\|^{-4}}{1+\|\boldsymbol{\mu}\|^{-2}}$ and $Q(x) = \frac{1}{2\pi}\int_x^\infty e^{-t^2/2}dt$.

### 2.3.2 Simple example of supervised learning

We consider in this subsection a linear classifier with an $\ell_2$ regularization defined by the following optimization problem

$$\min_{\boldsymbol{w}} \mathcal{E}(\boldsymbol{w}) = \min_{\boldsymbol{w}} \frac{1}{n}\|\boldsymbol{y} - \boldsymbol{X}^\intercal \boldsymbol{w}\|^2 + \gamma\|\boldsymbol{w}\|^2 \tag{2.33}$$

where $\boldsymbol{w}$ stands for the weights vector and $\gamma \in \mathbb{R}^+$ is the regularization parameter. The data matrix $\boldsymbol{X}$ is defined in equation 2.29 and $\boldsymbol{y} \in \mathbb{R}^n$ stands for the vector of labels such that $y_i = (-1)^\ell$ if $\boldsymbol{x}_i \in \mathcal{C}_\ell$ for $\ell \in \{1,2\}$. The solution to the optimization problem in equation 2.33 is explicitly given by

$$\boldsymbol{w} = \frac{1}{n}\boldsymbol{Q}(-\gamma)\boldsymbol{X}\boldsymbol{y} \quad \text{with} \quad \boldsymbol{Q}(z) = \left(\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\intercal - z\boldsymbol{I}_p\right)^{-1} \tag{2.34}$$

where the resolvent $\boldsymbol{Q}(z)$ appears at the core of the classifier. In particular, one is interested on the behavior of the (hard) decision function for a new test datum $\boldsymbol{x} \in \mathcal{C}_\ell$, for $\ell\{1,2\}$, which is defined as

$$g(\boldsymbol{x}) = \boldsymbol{x}^\intercal \boldsymbol{w} = \frac{1}{n}\boldsymbol{x}^\intercal \boldsymbol{Q}(-\gamma)\boldsymbol{X}\boldsymbol{y} \underset{\mathcal{C}_2}{\overset{\mathcal{C}_1}{\gtrless}} 0 \tag{2.35}$$

Given the above form of the decision function, it is a simple sum of independent (not necessarily identically distributed) random variables. Therefore thanks to the Lyapunov's central limit theorem [Bil08], which we recall in the following theorem, $g(\boldsymbol{x})$ has a Gaussian approximation in the large limit when both $p, n \to \infty$ with $p/n \to c \in (0, \infty)$.

**Theorem 2.7** (Lyapunov's CLT [Bil08]). *Let $X_1, \ldots, X_n$ be independent random variables of means $\mathbb{E}X_i = \mu_i$ such that $|X_i|$ have moments of order $2+\varepsilon$ for some $\varepsilon > 0$ and let the Lyapunov condition $\lim_{n\to\infty} \frac{1}{s_n^{2+\varepsilon}} \sum_{i=1}^n \mathbb{E}\left[|X_i - \mu_i|^{2+\varepsilon}\right] = 0$ hold, with $s_n = \mathbb{V}ar^{\frac{1}{2}}\left[\sum_{i=1}^n (X_i - \mu_i)\right]$. Then,*

$$\frac{1}{s_n}\sum_{i=1}^n (X_i - \mu_i) \quad \xrightarrow{\mathcal{D}} \quad \mathcal{N}(0,1)$$

It suffices thus to estimate the first and second moments of the decision function $g(\boldsymbol{x})$ to fully describe its behavior and therefore infer the performances of the linear classifier. We start by estimating $m_\ell = \mathbb{E}g(\boldsymbol{x})$ for $\boldsymbol{x} \in \mathcal{C}_\ell$ independent of the data matrix $\boldsymbol{X}$. In the following we will write $\boldsymbol{Q} = \boldsymbol{Q}(-\gamma)$ and $\bar{\boldsymbol{Q}} = \bar{\boldsymbol{Q}}(-\gamma)$ for simplicity.

$$m_\ell = \mathbb{E}g(\boldsymbol{x}) = \mathbb{E}\left[\frac{1}{n}\boldsymbol{x}^\intercal \boldsymbol{Q}\boldsymbol{X}\boldsymbol{y}\right] = \frac{1}{n}\sum_{i=1}^n \mathbb{E}\left[y_i \boldsymbol{x}^\intercal \boldsymbol{Q}\boldsymbol{x}_i\right]$$

The above expectation requires to manage the statistical dependencies between $\boldsymbol{Q}$ and $\boldsymbol{x}_i$, which can be handled by Example 1, specifically through the identity

$$\boldsymbol{Q}\boldsymbol{x}_i = \frac{\boldsymbol{Q}_{-i}\boldsymbol{x}_i}{1 + \frac{1}{n}\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i} \tag{2.36}$$

Moreover, relying on the trace Lemma 2.8, the quadratic form $\frac{1}{n}\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i$ converges[2] to the quantity $\frac{1}{n}\operatorname{tr}\left(\mathbb{E}[\boldsymbol{x}_i\boldsymbol{x}_i]\bar{\boldsymbol{Q}}\right)$ where $\bar{\boldsymbol{Q}}$ is the deterministic equivalent of $\boldsymbol{Q}$ from Theorem 2.6.

[2]The resolvent $\boldsymbol{Q}_{-i}$ has a bounded spectral norm, this will be highlighted in the next section.

Specifically, we have by the perturbation Lemma 2.7

$$\frac{1}{n}\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i - cq(-\gamma) \xrightarrow{a.s.} 0 \tag{2.37}$$

Consequently, the expectation of the decision function $g(\boldsymbol{x})$ can be estimated as

$$m_\ell - \frac{(-1)^\ell \boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}}(-\gamma)\boldsymbol{\mu}}{1+cq(-\gamma)} \xrightarrow{a.s.} 0 \quad \Rightarrow \quad m_\ell - \frac{(-1)^\ell \|\boldsymbol{\mu}\|^2 q(-\gamma)}{1+(c+\|\boldsymbol{\mu}\|^2)q(-\gamma)} \xrightarrow{a.s.} 0$$

Let us now turn into the estimation of the variance of $g(\boldsymbol{x})$. Let $\boldsymbol{\Sigma} = \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\intercal] = \boldsymbol{I}_p + \boldsymbol{\mu}\boldsymbol{\mu}^\intercal$, we have

$$\begin{aligned}\mathbb{E}\left[(\boldsymbol{x}^\intercal \boldsymbol{w})^2\right] &= \frac{1}{n^2}\mathbb{E}\left[\boldsymbol{y}^\intercal \boldsymbol{X}^\intercal \boldsymbol{Q}\boldsymbol{x}\boldsymbol{x}^\intercal \boldsymbol{Q}\boldsymbol{X}\boldsymbol{y}\right] = \frac{1}{n^2}\mathbb{E}\left[\boldsymbol{y}^\intercal \boldsymbol{X}^\intercal \boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{Q}\boldsymbol{X}\boldsymbol{y}\right] \\ &= \frac{1}{n^2}\sum_{i=1}^n y_i^2 \mathbb{E}\left[\boldsymbol{x}_i^\intercal \boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{Q}\boldsymbol{x}_i\right] + \frac{1}{n^2}\sum_{i\neq j} y_i y_j \mathbb{E}\left[\boldsymbol{x}_i^\intercal \boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{Q}\boldsymbol{x}_j\right]\end{aligned}$$

Using again the identity in equation 2.36 and the above estimate of the quadratic form $\frac{1}{n}\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i$, we end up with

$$\mathbb{E}\left[(\boldsymbol{x}^\intercal \boldsymbol{w})^2\right] - \frac{1}{n}\frac{\operatorname{tr}\left(\boldsymbol{\Sigma}\mathbb{E}[\boldsymbol{Q}_{-i}\boldsymbol{\Sigma}\boldsymbol{Q}_{-i}]\right)}{(1+cq(-\gamma))^2} - \frac{1}{n^2}\sum_{i\neq j} y_i y_j \frac{\mathbb{E}\left[\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}\boldsymbol{\Sigma}\boldsymbol{Q}_{-j}\boldsymbol{x}_j\right]}{(1+cq(-\gamma))^2} \xrightarrow{a.s.} 0$$

where the term $\mathbb{E}[\boldsymbol{Q}_{-i}\boldsymbol{\Sigma}\boldsymbol{Q}_{-i}]$ is handled by the following identities

$$\begin{aligned}\eta(\boldsymbol{A}) &= \frac{1}{n}\operatorname{tr}\left(\boldsymbol{\Sigma}\mathbb{E}[\boldsymbol{Q}\boldsymbol{A}\boldsymbol{Q}]\right) = \frac{(1+cq(-\gamma))\frac{1}{n}\operatorname{tr}\left(\boldsymbol{\Sigma}\bar{\boldsymbol{Q}}\boldsymbol{A}\bar{\boldsymbol{Q}}\right)}{((1+cq(-\gamma)))^2 - \frac{1}{n}\operatorname{tr}\left(\boldsymbol{\Sigma}\bar{\boldsymbol{Q}}\boldsymbol{\Sigma}\bar{\boldsymbol{Q}}\right)} \\ \boldsymbol{\Delta}(\boldsymbol{A}) &= \mathbb{E}\left[\boldsymbol{Q}\boldsymbol{A}\boldsymbol{Q}\right] = \bar{\boldsymbol{Q}}\boldsymbol{A}\bar{\boldsymbol{Q}} + \frac{\eta(\boldsymbol{A})}{1+cq(-\gamma)}\bar{\boldsymbol{Q}}\boldsymbol{\Sigma}\bar{\boldsymbol{Q}}\end{aligned}$$

Which yield to

$$\frac{1}{n}\operatorname{tr}\left(\boldsymbol{\Sigma}\mathbb{E}[\boldsymbol{Q}_{-i}\boldsymbol{\Sigma}\boldsymbol{Q}_{-i}]\right) - cq^2(-\gamma) \xrightarrow{a.s.} 0$$

And the remaining term $\mathbb{E}\left[\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}\boldsymbol{\Sigma}\boldsymbol{Q}_{-j}\boldsymbol{x}_j\right]$ develops thanks to Example 1, through the identity

$$\boldsymbol{Q} = \boldsymbol{Q}_{-i} - \frac{\boldsymbol{Q}_{-i}\frac{1}{n}\boldsymbol{x}_i\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}}{1+\frac{1}{n}\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i}$$

Indeed, we have for $i \neq j$

$$\begin{aligned}\mathbb{E}\left[\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}\boldsymbol{\Sigma}\boldsymbol{Q}_{-j}\boldsymbol{x}_j\right] &= \mathbb{E}\left[\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-ij}\boldsymbol{\Sigma}\boldsymbol{Q}_{-ji}\boldsymbol{x}_j\right] - \frac{1}{n}\mathbb{E}\left[\frac{\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-ij}\boldsymbol{\Sigma}\boldsymbol{Q}_{-ji}\boldsymbol{x}_i\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-ji}\boldsymbol{x}_j}{1+\frac{1}{n}\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-ji}\boldsymbol{x}_i}\right] \\ &\quad - \frac{1}{n}\mathbb{E}\left[\frac{\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-ij}\boldsymbol{x}_j\boldsymbol{x}_j^\intercal \boldsymbol{Q}_{-ij}\boldsymbol{\Sigma}\boldsymbol{Q}_{-ji}\boldsymbol{x}_j}{1+\frac{1}{n}\boldsymbol{x}_j^\intercal \boldsymbol{Q}_{-ij}\boldsymbol{x}_j}\right] + \frac{1}{n^2}\mathbb{E}\left[\frac{\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-ij}\boldsymbol{x}_j\boldsymbol{x}_j^\intercal \boldsymbol{Q}_{-ij}\boldsymbol{\Sigma}\boldsymbol{Q}_{-ji}\boldsymbol{x}_i\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-ij}\boldsymbol{x}_j}{(1+\frac{1}{n}\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-ij}\boldsymbol{x}_i)(1+\frac{1}{n}\boldsymbol{x}_j^\intercal \boldsymbol{Q}_{-ij}\boldsymbol{x}_j)}\right]\end{aligned}$$

And recalling the convergence of the quadratic form in equation 4.16 and putting all the pieces together we end up with the following estimation of $\mathbb{E}\left[(\boldsymbol{x}^\intercal \boldsymbol{w})^2\right]$

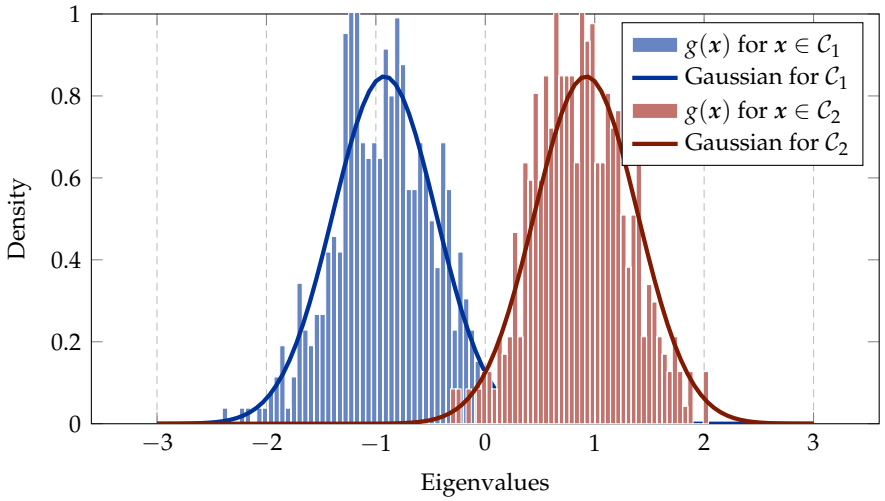



Figure 2.3: Histogram of the decision function $g(\boldsymbol{x})$ for new test data $\boldsymbol{x}$ independent from the training set $\boldsymbol{X}$, versus its Gaussian approximation as per Theorem 2.8. We consider the parameters $p = 1100$, $n_1 = n_2 = 500$, $\gamma = 1 \cdot 10^{-1}$ and $\boldsymbol{\mu} = [3, \mathbf{0}_{p-1}]^\mathsf{T}$.

$$\mathbb{E}\left[(\boldsymbol{x}^\mathsf{T}\boldsymbol{w})^2\right] - \frac{cq^2(-\gamma)}{(1+cq(-\gamma))^2} - \frac{\boldsymbol{\mu}^\mathsf{T}\boldsymbol{\Delta}(\boldsymbol{\Sigma})\boldsymbol{\mu}}{(1+cq(-\gamma))^2} + \frac{2cq^2(-\gamma)\boldsymbol{\mu}^\mathsf{T}\bar{\boldsymbol{Q}}(-\gamma)\boldsymbol{\mu}}{(1+cq(-\gamma))^3} \xrightarrow{a.s.} 0$$

We therefore have the following result which provides a Gaussian approximation of the decision function $g(\boldsymbol{x})$.

**Theorem 2.8** (Gaussian Approximation of $g(\boldsymbol{x})$)**.** *Under Assumption 2, for $\boldsymbol{x} \in \mathcal{C}_\ell$ for $\ell \in \{1,2\}$, we have*

$$(\nu - m_\ell^2)^{-\frac{1}{2}}(g(\boldsymbol{x}) - m_\ell) \xrightarrow{\mathcal{D}} \mathcal{N}(0,1)$$

*where*

$$m_\ell = \frac{(-1)^\ell \|\boldsymbol{\mu}\|^2 q(-\gamma)}{1 + (c + \|\boldsymbol{\mu}\|^2) q(-\gamma)}$$
$$\nu = \frac{cq^2(-\gamma)}{(1+cq(-\gamma))^2} + \frac{\boldsymbol{\mu}^\mathsf{T}\boldsymbol{\Delta}(\boldsymbol{\Sigma})\boldsymbol{\mu}}{(1+cq(-\gamma))^2} - \frac{2cq^2(-\gamma)\boldsymbol{\mu}^\mathsf{T}\bar{\boldsymbol{Q}}(-\gamma)\boldsymbol{\mu}}{(1+cq(-\gamma))^3}$$

Theorem 2.8 states that the considered linear classifier is asymptotically equivalent to the thresholding of two monovariate Gaussian random variables. Figure 2.3 depicts simulated values of $g(\boldsymbol{x})$ along with its asymptotic theoretical Gaussian prediction where a perfect matching is observed. In particular, the generalization performance of the classifier can be expressed thanks to the Gaussian tail function $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-u^2/2} du$ as per the following Corollary to Theorem 2.8.

**Corollary 2.1** (Generalization Performance of the linear classifier)**.** *Under the setting and Assumptions of Theorem 2.8, for $\ell \in \{1,2\}$, with probability one*

$$\mathbb{P}\left\{(-1)^\ell g(\boldsymbol{x}) < 0 \mid \boldsymbol{x} \in \mathcal{C}_\ell\right\} - Q\left(\frac{m_\ell}{\sqrt{\nu - m_\ell^2}}\right) \to 0$$

The same analysis as the one presented in this subsection can be adopted to analyse a wide range of standard classifiers such as LS-SVM [LC17] and even some more sophisticated methods such as the logistic regression algorithm [MLC19]. The core of this analysis leverages the design of a deterministic equivalent for the resolvent $\boldsymbol{Q}$. In the next section, we will recall some recent results from Louart and Couillet [LC18b] which provide a systematic approach to design such deterministic equivalents leveraging on concentration assumptions on data, and which particularly generalizes the Gaussian assumption of the result of Benaych and Couillet [BGC16] recalled in Theorem 2.5. As discussed in the introduction and will be presented in more details in the next chapter, the concentration assumption is of particular interest in practice since it provides a realistic modeling for real data, in particular, for GAN data which satisfy the concentration assumption by construction.

## 2.4 Extensions with concentration of measure theory

The previous results are extensible to a richer class of random vectors, namely to the class of random concentrated vectors [Led05a]. As we discussed in the introduction, this class for random vectors are more appropriate for real data modelling since GANs data fall constructively within this class. In this section, we will briefly recall some essential concentration notions and properties, then we will provide their application to design a deterministic equivalent for the sample covariance matrix following the same approach developed in [LC18b].

### 2.4.1 The notion of concentrated vectors

Being the central tool of this manuscript, we introduce the notion of random concentrated vectors. Note that several and more advanced concentration notions have been recently developed in [LC18b] in order to specifically analyze the behavior of large sample covariance matrices, but for simplicity, we restrict ourselves in this manuscript to the sufficient so-called $q$-exponentially concentrated random vectors.

**Definition 7** (Concentrated vector)**.** *Given a set of indices $\mathbb{S}$, a sequence of normed vector spaces $(E_s, \|\cdot\|_s)_{s\in\mathbb{S}}$, a sequence of random vectors $\boldsymbol{Z}_s \in E_s$, a sequence of positive numbers $\sigma_s$, we say that $\boldsymbol{Z}_s$ is* $q$-exponentially concentrated with an observable diameter of order $\mathcal{O}(\sigma_s)$ *if there exists two constants $C, c > 0$ such that for all sequence of* 1*-Lipschitz mappings $f_s : E_s \to \mathbb{R}$:*

$$\forall s \in \mathbb{S},\ \forall t > 0 \ :\ \mathbb{P}\left\{|f_s(\boldsymbol{Z}_s) - \mathbb{E}[f_s(\boldsymbol{Z}_s)]| \geq t\right\} \leq Ce^{-c(t/\sigma_s)^q} \tag{2.38}$$

*We note then $\boldsymbol{Z}_s \propto \mathcal{E}_q(\sigma_s)$; when $\sigma_s = \mathcal{O}(1)$, we write $\boldsymbol{Z}_s \propto \mathcal{E}_q$.*

Therefore, concentrated vectors are defined through the concentration of any 1-Lipschitz real scalar "observation". One of the most important examples of concentrated vectors are standard Gaussian vectors. Precisely, we have the following proposition. Note also that more examples such as uniform and Gamma distribution result in random vectors which satisfy the concentration property as per Definition 7, see [Led05a].

**Proposition 2.3** (Gaussian vectors [Led05a])**.** *Let $d \in \mathbb{N}$ and $\boldsymbol{Z}_d \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$. Then $\boldsymbol{Z}_d$ is a* 2*-exponentially concentrated vector independently on the dimension $d$, i.e. $\boldsymbol{Z}_d \propto \mathcal{E}_2$.*

But the richness of concentrated random vectors lies in their fundamental stability property through Lipschitz operations, which naturally generates wide families of concentrated random vectors.

**Proposition 2.4** (Stability through Lipschitz transformations)**.** *It is easily deduced from Definition 7 that given a sequence of positive numbers $L_s > 0$ and a sequence of $L_s$-Lipschitz transformations $\phi_s : (E_s, \|\cdot\|_s) \to (F_s, \|\cdot\|'_s)$,*

$$\text{if} \quad \mathbf{Z}_s \propto \mathcal{E}_q(\sigma_s), \quad \text{then} \quad \phi_s(\mathbf{Z}_s) \propto \mathcal{E}_q(L_s\sigma_s) \tag{2.39}$$

*Proof.* This comes simply by, for all $f_s : F_s \to \mathbb{R}$, 1-Lipschitz, $\frac{1}{L_s} f_s \circ \phi_s$ is 1-Lipschitz, and one can employ inequality 2.38 to $\frac{t}{L_s}$ which provides the result. □

As we saw previously, we need to establish the concentration of quadratic forms of the form $\frac{1}{p}\boldsymbol{x}^\mathsf{T}\boldsymbol{Q}\boldsymbol{x}$ with $\boldsymbol{Q}$ deterministic or independent from $\boldsymbol{x}$. To this end, we will need the following property.

**Proposition 2.5.** *Let $X_n \in \mathbb{R}$ be a random variable depending on some integer $n \in \mathbb{N}$, satisfying $X_n \propto \mathcal{E}_q(\sigma_n)$ with $\sigma_n \to 0$ as $n \to \infty$ and $\limsup_n |\mathbb{E}X_n| < \infty$. The square of $X_n$ remains concentrated and we specifically have*

$$X_n^2 - \mathbb{E}X_n^2 \propto \mathcal{E}_q(\sigma_n) + \mathcal{E}_{\frac{q}{2}}(\sigma_n^2)$$

*Proof.* From the algebraic identity $X_n^2 - (\mathbb{E}X_n)^2 = (X_n - \mathbb{E}X_n)^2 + 2(X_n - \mathbb{E}X_n)\mathbb{E}X_n$, one has

$$\mathbb{P}\left\{|X_n^2 - (\mathbb{E}X_n)^2| \geq t\right\} \leq \mathbb{P}\left\{|X_n - \mathbb{E}X_n| \geq \sqrt{\frac{t}{2}}\right\} + \mathbb{P}\left\{|X_n - \mathbb{E}X_n| \geq \frac{t}{4\,|\mathbb{E}X_n|}\right\}$$

Applying the identities (where $X'_n$ is an independent copy of $X_n$)

$$\mathbb{P}\left\{|X_n^2 - m_{X_n^2}| \geq t\right\} \leq 2\mathbb{P}\left\{|X_n^2 - (X'_n)^2| \geq t\right\} \leq 4\mathbb{P}\left\{|X_n^2 - (\mathbb{E}X_n)^2| \geq t/2\right\}$$

where $m_{X_n^2}$ is a median of $X_n^2$, and the final result comes from the fact that the expectation and median of $X_n^2$ are asymptotically close to each other as $n \to \infty$. Indeed, we have

$$|\mathbb{E}X_n - m_{X_n}| \leq \mathbb{E}|X_n - m_{X_n}| = \mathcal{O}\left(\int_0^\infty e^{-c(t/\sigma_n)^q} dt\right) = \mathcal{O}(\sigma_n) \to 0$$

The same reasoning holds also for $X_n^2$ which concludes the proof. □

The following Lemma exploits Proposition 2.5 to provide the concentration of the quadratic form $\frac{1}{p}\boldsymbol{x}^\mathsf{T}\boldsymbol{Q}\boldsymbol{x}$ for some concentrated vector $\boldsymbol{x} \in \mathbb{R}^p$ and bounded matrix $\boldsymbol{Q} \in \mathcal{M}_p$ (in spectral norm).

**Lemma 2.9** (Trace Lemma with concentration)**.** *Let $\boldsymbol{x} \in \mathbb{R}^p$ be a random vector such that $\boldsymbol{x} \propto \mathcal{E}_q$ and $\boldsymbol{Q} \in \mathcal{M}_p$ some deterministic (or random independent of $\boldsymbol{x}$) matrix with bounded spectral norm. Then,*

$$\frac{1}{p}\boldsymbol{x}^\mathsf{T}\boldsymbol{Q}\boldsymbol{x} - \frac{1}{p}\operatorname{tr}\left(\mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}\right]\boldsymbol{Q}\right) \propto \mathcal{E}_q\left(\frac{1}{\sqrt{p}}\right) + \mathcal{E}_{\frac{q}{2}}\left(\frac{1}{p}\right)$$

*In particular, there exists a constant $C > 0$ such that*

$$\mathbb{E}\left[\left|\frac{1}{p}\boldsymbol{x}^\mathsf{T}\boldsymbol{Q}\boldsymbol{x} - \frac{1}{p}\operatorname{tr}\left(\mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^\mathsf{T}\right]\boldsymbol{Q}\right)\right|^k\right] \leq Cp^{-\frac{k}{2}}$$

*Proof.* The proof comes simply by rewriting $\frac{1}{p}\boldsymbol{x}^\intercal \boldsymbol{Q}\boldsymbol{x} = \|\frac{1}{\sqrt{p}}\boldsymbol{Q}^{\frac{1}{2}}\boldsymbol{x}\|^2$. Since, $\boldsymbol{x} \to \|\frac{1}{\sqrt{p}}\boldsymbol{Q}^{\frac{1}{2}}\boldsymbol{x}\|$ is $\mathcal{O}\left(\frac{1}{\sqrt{p}}\right)$-Lipschitz we have $\|\frac{1}{\sqrt{p}}\boldsymbol{Q}^{\frac{1}{2}}\boldsymbol{x}\| \propto \mathcal{E}_q\left(\frac{1}{\sqrt{p}}\right)$, and the final concentration comes by Proposition 2.5. Moreover, the control of the $k$-th moment follows from the identity

$$\mathbb{E}|X|^k = \int_0^\infty k\, t^{k-1}\mathbb{P}\left\{|X| \geq t\right\} dt$$

for some random variable $X$. □

We will see in the next subsection how this result is exploited to design a deterministic equivalent for the sample covariance matrix.

### 2.4.2 Resolvent of the sample covariance matrix

In this subsection, we will consider a data matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathcal{M}_{p,n}$ such that $\boldsymbol{X} \propto \mathcal{E}_q$ and we further suppose that the columns of $\boldsymbol{X}$ are independent and have the same second order statistic $\mathbb{E}[\boldsymbol{x}_i\boldsymbol{x}_i^\intercal] = \boldsymbol{\Sigma} \in \mathcal{M}_p$ with bounded spectral norm. As we saw previously, the analysis of the behavior of the sample covariance matrix $\hat{\boldsymbol{\Sigma}} = \frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\intercal$ is related to the analysis of its resolvent, defined for $z \in \mathbb{C} \setminus \mathbb{R}_-$[3] as

$$\boldsymbol{Q}(z) = \left(\hat{\boldsymbol{\Sigma}} + z\boldsymbol{I}_p\right)^{-1} = \left(\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\intercal + z\boldsymbol{I}_p\right)^{-1} \tag{2.40}$$

The resolvent matrix $\boldsymbol{Q}(z)$ notably satisfies several bounds which are of particular interest in the analysis of its behavior as we will see subsequently. Indeed, we recall the following Lemma from [LC18b].

**Lemma 2.10** (The resolvent bounds)**.** *The resolvent $\boldsymbol{Q}(z)$ satisfies the following bounds. For $z \in \mathbb{R}_+$, we have*

$$\textbf{1.}\ \|\boldsymbol{Q}(z)\| \leq \frac{1}{z} \qquad \textbf{2.}\ \|\boldsymbol{Q}(z)\hat{\boldsymbol{\Sigma}}\| \leq 1 \qquad \textbf{3.}\ \|\boldsymbol{Q}(z)\boldsymbol{X}\| \leq \sqrt{\frac{n}{z}}$$

*Proof.* The upper bound for **1.** comes from the smallest eigenvalue of $\hat{\boldsymbol{\Sigma}} + z\boldsymbol{I}_p$ being larger than $z$. **2.** follows from the identity $\boldsymbol{Q}(z)\hat{\boldsymbol{\Sigma}} + z\boldsymbol{Q}(z) = \boldsymbol{I}_p$ and since $\boldsymbol{Q}(z)$ is symmetric positive definite. Finally, **3.** follows from combining **1.** and **2.**, giving the bound

$$\|\boldsymbol{Q}(z)\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\intercal\boldsymbol{Q}(z)\| \leq \frac{1}{z}$$

□

Now we provide the intuition behind the design of a deterministic equivalent for $\boldsymbol{Q}(z)$, we refer the reader to [LC18b] for a detailed proof. From a low-dimensional perspective (i.e., $p$ fixed as $n \to \infty$), one would think that $\boldsymbol{Q}(z)$ would be close to $\left(\boldsymbol{\Sigma} + z\boldsymbol{I}_p\right)^{-1}$. This happens to be completely wrong in the random matrix theory regime, when both $p, n \to \infty$ with $p/n \to c \in (0, \infty)$. However, one can establish that $\boldsymbol{Q}(z)$ is equivalent to some matrix of the form $\bar{\boldsymbol{Q}}(z) = \left(\tilde{\boldsymbol{\Sigma}} + z\boldsymbol{I}_p\right)^{-1}$, in the sense of Definition 6,

[3] For convenience, we shall consider the notation of the resolvent with "$+z$" instead of "$-z$".

for some well-chosen deterministic matrix $\tilde{\boldsymbol{\Sigma}} \in \mathcal{M}_p$. In order to find the explicit expression for $\tilde{\boldsymbol{\Sigma}}$, we start by expressing the difference between $\bar{\boldsymbol{Q}}(z)$ and $\mathbb{E}[\boldsymbol{Q}(z)]$, thanks to the resolvent identity in Lemma 2.1, as follows

$$\tilde{\boldsymbol{Q}}(z) - \mathbb{E}[\boldsymbol{Q}(z)] = \mathbb{E}\left[\boldsymbol{Q}(z)\left(\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\intercal - \tilde{\boldsymbol{\Sigma}}\right)\bar{\boldsymbol{Q}}(z)\right] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\boldsymbol{Q}(z)\left(\boldsymbol{x}_i\boldsymbol{x}_i^\intercal - \tilde{\boldsymbol{\Sigma}}\right)\bar{\boldsymbol{Q}}(z)\right]$$

The next step now consists in handling the statistical dependencies between $\boldsymbol{Q}(z)$ and $\boldsymbol{x}_i$, this is made possible, as we saw in the previous Section, thanks to the Schur identities from Example 1. In particular, first using the identity

$$\boldsymbol{Q}(z)\boldsymbol{x}_i = \frac{\boldsymbol{Q}_{-i}(z)\boldsymbol{x}_i}{1 + \frac{1}{n}\boldsymbol{x}_i^\intercal\boldsymbol{Q}_{-i}(z)\boldsymbol{x}_i}$$

We obtain

$$\begin{aligned}\tilde{\boldsymbol{Q}}(z) - \mathbb{E}[\boldsymbol{Q}(z)] &= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\left(\boldsymbol{Q}(z)\boldsymbol{x}_i\boldsymbol{x}_i^\intercal - \boldsymbol{Q}(z)\tilde{\boldsymbol{\Sigma}}\right)\bar{\boldsymbol{Q}}(z)\right] \\ &= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\left(\frac{\boldsymbol{Q}_{-i}(z)\boldsymbol{x}_i\boldsymbol{x}_i^\intercal}{1 + \frac{1}{n}\boldsymbol{x}_i^\intercal\boldsymbol{Q}_{-i}(z)\boldsymbol{x}_i} - \boldsymbol{Q}(z)\tilde{\boldsymbol{\Sigma}}\right)\bar{\boldsymbol{Q}}(z)\right]\end{aligned}$$

And thanks to the identity

$$\boldsymbol{Q}(z) = \boldsymbol{Q}_{-i}(z) - \frac{1}{n}\frac{\boldsymbol{Q}_{-i}(z)\boldsymbol{x}_i\boldsymbol{x}_i^\intercal\boldsymbol{Q}_{-i}(z)}{1 + \frac{1}{n}\boldsymbol{x}_i^\intercal\boldsymbol{Q}_{-i}(z)\boldsymbol{x}_i}$$

We end up having

$$\begin{aligned}\tilde{\boldsymbol{Q}}(z) - \mathbb{E}[\boldsymbol{Q}(z)] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\boldsymbol{Q}_{-i}(z)\left(\frac{\boldsymbol{x}_i\boldsymbol{x}_i^\intercal}{1 + \frac{1}{n}\boldsymbol{x}_i^\intercal\boldsymbol{Q}_{-i}(z)\boldsymbol{x}_i} - \tilde{\boldsymbol{\Sigma}}\right)\bar{\boldsymbol{Q}}(z)\right] \\ + \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}\left[\boldsymbol{Q}_{-i}(z)\boldsymbol{x}_i\boldsymbol{x}_i^\intercal\boldsymbol{Q}_{-i}(z)\tilde{\boldsymbol{\Sigma}}\bar{\boldsymbol{Q}}(z)\right]\end{aligned}$$

where the second line term happens to be of vanishing spectral norm due to the additional factor $1/n$ and since the matrix $\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\boldsymbol{Q}_{-i}(z)\boldsymbol{x}_i\boldsymbol{x}_i^\intercal\boldsymbol{Q}_{-i}(z)\tilde{\boldsymbol{\Sigma}}\bar{\boldsymbol{Q}}(z)\right]$ has a bounded spectral norm thanks to Lemma 2.10. Moreover, again by Lemma 2.10, since $\boldsymbol{Q}_{-i}(z)$ has a bounded spectral norm and is independent from $\boldsymbol{x}_i$, the quadratic form $\frac{1}{n}\boldsymbol{x}_i^\intercal\boldsymbol{Q}_{-i}(z)\boldsymbol{x}_i$ converges to its expectation $\frac{1}{n}\operatorname{tr}\left(\boldsymbol{\Sigma}\mathbb{E}[\boldsymbol{Q}_{-i}(z)]\right)$ thanks to Lemma 2.9. Quite naturally, one would expect that the deterministic matrix $\tilde{\boldsymbol{\Sigma}}$ should be

$$\tilde{\boldsymbol{\Sigma}} = \frac{\boldsymbol{\Sigma}}{1 + \frac{1}{n}\operatorname{tr}\left(\boldsymbol{\Sigma}\mathbb{E}[\boldsymbol{Q}_{-i}(z)]\right)}$$

so that the difference $\tilde{\boldsymbol{Q}}(z) - \mathbb{E}[\boldsymbol{Q}(z)]$ would be of vanishing spectral norm. Consequently, the term $\frac{1}{n}\operatorname{tr}\left(\boldsymbol{\Sigma}\mathbb{E}[\boldsymbol{Q}_{-i}(z)]\right)$ would naturally be close to $\frac{1}{n}\operatorname{tr}\left(\boldsymbol{\Sigma}\bar{\boldsymbol{Q}}(z)\right)$, therefore defining the deterministic equivalent $\bar{\boldsymbol{Q}}(z)$ of the resolvent $\boldsymbol{Q}(z)$ through an implicit fixed point equation rather than computing the expectation of $\boldsymbol{Q}_{-i}(z)$ which is hard to compute explicitly in general cases. The deterministic equivalent $\bar{\boldsymbol{Q}}(z)$ is therefore given by

$$\boldsymbol{Q}(z) \leftrightarrow \bar{\boldsymbol{Q}}(z) = \left(\frac{\boldsymbol{\Sigma}}{1 + \delta(z)} + z\boldsymbol{I}_p\right)^{-1}$$

where $\delta(z)$ is the unique[4] solution to the fixed point equation

$$\delta(z) = \frac{1}{n} \operatorname{tr}\left( \boldsymbol{\Sigma} \left( \frac{\boldsymbol{\Sigma}}{1+\delta(z)} + z\boldsymbol{I}_p \right)^{-1} \right)$$

Looking carefully to the above expression of $\bar{\boldsymbol{Q}}(z)$, we see that there is an additional term $\delta(z)$ comparing to what one would expect in a low-dimensional regime, i.e., the deterministic equivalent being $\left(\boldsymbol{\Sigma} + z\boldsymbol{I}_p\right)^{-1}$. Indeed, when $p$ is fixed as $n \to \infty$, we particularly have $\delta(z) \approx \frac{1}{n} \operatorname{tr}\left(\boldsymbol{\Sigma}\mathbb{E}[\boldsymbol{Q}_{-i}(z)]\right) \to 0$, which recovers the low-dimensional case.

**Remark 2.4** (On the concentration of $\frac{1}{n}\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}(z)\boldsymbol{x}_i$)**.** *Note that Lemma 2.9 it not good enough to establish the concentration of $\frac{1}{n}\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}(z)\boldsymbol{x}_i$. Indeed, $\boldsymbol{Q}_{-i}(z)$ being random demands to bound a probability involving $\boldsymbol{x}_i$. Specifically, one should decompose the calculus as*

$$\begin{aligned}
&\mathbb{P}\left\{ \left| \frac{1}{n}\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}(z)\boldsymbol{x}_i - \frac{1}{n}\operatorname{tr}\left(\boldsymbol{\Sigma}\mathbb{E}[\boldsymbol{Q}_{-i}(z)]\right) \right| \geq t \right\} \\
&\leq \mathbb{E}\left[ \mathbb{P}\left\{ \left| \frac{1}{n}\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}(z)\boldsymbol{x}_i - \frac{1}{n}\operatorname{tr}\left(\boldsymbol{\Sigma}\boldsymbol{Q}_{-i}(z)\right) \right| \geq \frac{t}{2} \mid \boldsymbol{X}_{-i} \right\} \right] \\
&+ \mathbb{P}\left\{ \left| \frac{1}{n}\operatorname{tr}\boldsymbol{\Sigma}\left(\boldsymbol{Q}_{-i}(z) - \mathbb{E}\left[\boldsymbol{Q}_{-i}(z)\right]\right) \right| \geq \frac{t}{2} \right\}
\end{aligned}$$

*where $\boldsymbol{X}_{-i} \in \mathcal{M}_{p,n-1}$ is the data matrix $\boldsymbol{X}$ prevented from its $i$-th column $\boldsymbol{x}_i$. Therefore, the concentration of $\frac{1}{n}\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}(z)\boldsymbol{x}_i$ follows after handling the above two terms (see [LC18b, Proposition 3.4]).*

Under the more general case of a mixture of $k$-class concentrated vectors (see Definition 2), a deterministic equivalent of the sample covaraince matrix is given by the following result.

**Theorem 2.9** (Deterministic equivalent of the sample covariance matrix [LC18b])**.** *Let a data matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathcal{M}_{p,n}$ be satisfying Definition 2 and let Assumption 1 hold with $\|\boldsymbol{\mu}_\ell\| = \mathcal{O}(\sqrt{p})$ for all $\ell \in [k]$. Then the resolvent of the sample covariance matrix defined as $\boldsymbol{Q}(z) = \left(\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\intercal + z\boldsymbol{I}_p\right)^{-1}$ admits a deterministic equivalent $\bar{\boldsymbol{Q}}(z)$ given by*

$$\boldsymbol{Q}(z) \leftrightarrow \bar{\boldsymbol{Q}}(z) = \left( \sum_{\ell=1}^{k} \frac{c_\ell(\boldsymbol{\Sigma}_\ell + \boldsymbol{\mu}_\ell\boldsymbol{\mu}_\ell^\intercal)}{1+\delta_\ell(z)} + z\boldsymbol{I}_p \right)^{-1}$$

*where $(\delta_1(z), \ldots, \delta_k(z))$ are the unique solution to system of fixed point equations defined for each $j \in [k]$ as*

$$\delta_j(z) = \frac{1}{n}\operatorname{tr}\left( (\boldsymbol{\Sigma}_j + \boldsymbol{\mu}_j\boldsymbol{\mu}_j^\intercal) \left( \sum_{\ell=1}^{k} \frac{c_\ell(\boldsymbol{\Sigma}_\ell + \boldsymbol{\mu}_\ell\boldsymbol{\mu}_\ell^\intercal)}{1+\delta_\ell(z)} + z\boldsymbol{I}_p \right)^{-1} \right)$$

Theorem 2.9 generalizes the result of Benaych and Couillet [BGC16] (recalled in Theorem 2.5) to the more general class of concentrated vectors. However, the above expression of the deterministic equivalent involve strictly the class-wise means and covariances of the data namely the quantities $\{\boldsymbol{\mu}_\ell\}_{\ell=1}^k$ and $\{\boldsymbol{\Sigma}_\ell\}_{\ell=1}^k$, thereby demonstrating the *universality* aspect of the behavior of the sample covariance matrix *w.r.t.* data distributions

[4] See in [LC18b, Proposition 3.8] for more details about the existence and uniqueness of the fixed point equation solution.

satisfying the concentration assumption in Definition 2. We will see in the next chapters, that this universality aspect goes far beyond the sample covariance matrix to more sophisticated non-linear ML methods such as kernel methods and the Softmax layer in neural networks.

# Chapter 3

# Universality of Large Random Matrices

This chapter is based on the following works:

**(C1)** **MEA. Seddik**, C. Louart, M. Tamaazousti, R. Couillet, "*Random Matrix Theory Proves that Deep Learning Representations of GAN-data Behave as Gaussian Mixtures*", International Conference on Machine Learning (ICML'20), Online, 2020.

**(C1')** **MEA. Seddik**, M. Tamaazousti, R. Couillet, "*Pourquoi les matrices aléatoires expliquent l'apprentissage ? Un argument d'universalité offert par les GANs*", Colloque francophone de traitement du signal et des images (Gretsi'19), Lille, France, 2019.

**Contents**



This Chapter contains two main parts. The first part will highlight the importance of modeling real data as concentrated vectors, the second part will consist in studying the behavior of large Gram matrices with concentrated inputs.

## 3.1 GAN Data are Concentrated Data

Concentrated random vectors are particularly interesting from a practical standpoint for real data modeling. In fact, unlike simple Gaussian vectors, the former do not suffer from the constraint of having independent entries which is quite a restrictive assumption when modeling real data such as images or their non-linear features (*e.g.*, DL representations). The other modeling interest of concentrated vectors lies in their being already present in practice as alternatives to real data. Indeed, adversarial neural networks (GANs) have

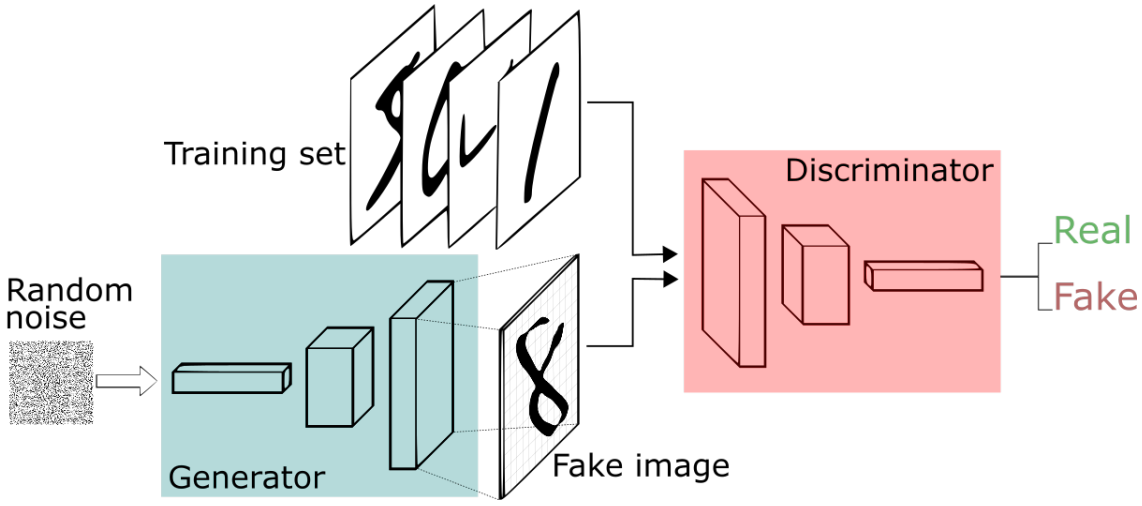



Figure 3.1: A generative adversarial model involves two networks; a Generator which transforms noise vectors to generated images and a Discriminato which seeks to identify real images from fake ones.

the ability nowadays to generate random *realistic* data (for instance realistic images) by applying successive Lipschitz operations to standard Gaussian vectors [GPAM+14].

A GAN architecture involves two networks, a generator model which maps random Gaussian noise to new plausible synthetic data and a discriminator model which classifies real data as real (from the dataset represented by $p_{\text{data}}$) or fake (for the generated data). The discriminator $\mathcal{D}$ is updated directly through a binary classification problem, whereas the generator $\mathcal{G}$ is updated through the discriminator based on the following Min-Max objective

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} [\log \mathcal{D}(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_d)} [\log (1 - \mathcal{D}(\mathcal{G}(z)))]$$

As such, the two models are trained alternatively in an adversarial manner, where the generator seeks to better deceive the discriminator and the former seeks to better identify the fake data [GPAM+14].

In particular, once both models are trained (when they reach a Nash equilibrium), DL representations of GAN-data –and GAN-data themselves– are schematically constructed in practice as follows:

$$\text{Real Data} \approx \text{GAN Data} = \mathcal{F}_N \circ \cdots \circ \mathcal{F}_1(\boldsymbol{z}), \tag{3.1}$$

where $\boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}_d)$, $d$ stands for the input dimension of the generator model, $N$ the number of layers, and the $\mathcal{F}_i$'s either Fully Connected Layers, Convolutional Layers, Pooling Layers, Up-sampling Layers and Activation Functions, Residual Layers or Batch Normalizations. All these operations happen to be *Lipschitz* applications. Precisely,

- **Fully Connected Layers and Convolutional Layers:** These are affine operations which can be expressed as
$$\mathcal{F}_i(\boldsymbol{x}) = \boldsymbol{W}_i\boldsymbol{x} + \boldsymbol{b}_i,$$
for $\boldsymbol{W}_i$ the weight matrix and $\boldsymbol{b}_i$ the bias vector. Here the Lipschitz constant is the operator norm (the largest singular value) of the weight matrix $\boldsymbol{W}_i$, that is $\|\mathcal{F}_i\|_{lip} = \sup_{\boldsymbol{u}\neq 0} \frac{\|\boldsymbol{W}_i\boldsymbol{u}\|_2}{\|\boldsymbol{u}\|_2}$.

- **Pooling Layers and Activation Functions:** Most commonly used activation functions and pooling operations are

$$\begin{aligned}&\mathrm{ReLU}(\boldsymbol{x}) = \max(0, \boldsymbol{x}),\\ &\mathrm{MaxPooling}(\boldsymbol{x}) = [\max(\boldsymbol{x}_{\mathcal{S}_1}), \dots, \max(\boldsymbol{x}_{\mathcal{S}_q})]^\intercal,\end{aligned}$$

  where $\mathcal{S}_i$'s are patches (*i.e.*, subsets of $[\dim(\boldsymbol{x})]$). These are at most 1-Lipschitz operations with respect to the Frobenius norm. Specifically, the maximum absolute sub-gradient of the ReLU activation function is 1, thus the ReLU operation has a Lipschitz constant of 1. Similarly, we can show that the Lipschitz constant of MaxPooling layers is also 1.

- **Residual Connections:** Residual layers act the following way

$$\mathcal{F}_i(\boldsymbol{x}) = \boldsymbol{x} + \mathcal{F}_i^{(1)} \circ \cdots \circ \mathcal{F}_i^{(\ell)}(\boldsymbol{x}),$$

  where the $\mathcal{F}_i^{(j)}$'s are Fully Connected Layers or Convolutional Layers with Activation Functions, and which are Lipschitz operations. Thus $\mathcal{F}_i$ is a Lipschitz operation with Lipschitz constant bounded by $1 + \prod_{j=1}^{\ell} \|\mathcal{F}_i^{(j)}\|_{lip}$.

- **Batch Normalization (BN) Layers:** They consist in statistically standardizing [IS15] the vectors of a small batch $\mathcal{B} = \{\boldsymbol{x}_i\}_{i=1}^b \subset \mathbb{R}^d$ as follows: for each $\boldsymbol{x}_k \in \mathcal{B}$

$$\mathcal{F}_i(\boldsymbol{x}_k) = \mathrm{diag}\left(\frac{\mathbf{a}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}\right)(\boldsymbol{x}_k - \mu_{\mathcal{B}}\mathbf{1}_d) + \mathbf{b}$$

  where $\mu_{\mathcal{B}} = \frac{1}{db}\sum_{k=1}^b \sum_{i=1}^d [\boldsymbol{x}_k]_i$, $\sigma_{\mathcal{B}}^2 = \frac{1}{db}\sum_{k=1}^b \sum_{i=1}^d ([\boldsymbol{x}_k]_i - \mu_{\mathcal{B}})^2$, $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^d$ are parameters to be learned and $\mathrm{diag}(\boldsymbol{v})$ transforms a vector $\boldsymbol{v}$ to a diagonal matrix with its diagonal entries being those of $\boldsymbol{v}$. Thus BN is a Lipschitz transformation with Lipschitz constant $\|\mathcal{F}_i\|_{lip} = \sup_i |\frac{\mathbf{a}_i}{\sqrt{\sigma_{\mathcal{B}}^2+\epsilon}}|$.

Therefore, as illustrated in Figure B.2, since standard Gaussian vectors are concentrated vectors as mentioned in Proposition 2.3 and since the notion of concentrated vectors is stable by Lipschitz transformations thanks to Proposition 2.4, GAN-data (and their DL representations) are concentrated vectors by design given the construction in Equation (3.1). Moreover, in order to generate data belonging to a specific class, Conditional GANs have been introduced [MO14]; once again data generated by these models are concentrated vectors as a consequence of the following Corollary.

**Corollary 3.1.** *Let $\mathcal{G}_1, \dots, \mathcal{G}_n : \mathbb{R}^d \to \mathbb{R}^p$ a set of $n$ Lipschitz applications with Lipschitz constants $\|\mathcal{G}_i\|_{lip}$. Let $\mathcal{G} : \mathbb{R}^{d\times n} \to \mathbb{R}^{p\times n}$ be defined for each $\boldsymbol{X} \in \mathbb{R}^{d\times n}$ as*

$$\mathcal{G}(\boldsymbol{X}) = [\mathcal{G}_1(\boldsymbol{X}_{:,1}), \dots, \mathcal{G}_n(\boldsymbol{X}_{:,n})].$$

*Then, for $\boldsymbol{Z} \in \mathcal{M}_{d,n}$*

$$\boldsymbol{Z} \propto \mathcal{E}_q \quad \Rightarrow \quad \mathcal{G}(\boldsymbol{Z}) \propto \mathcal{E}_q\left(\sup_i \|\mathcal{G}_i\|_{lip}\right). \tag{3.2}$$

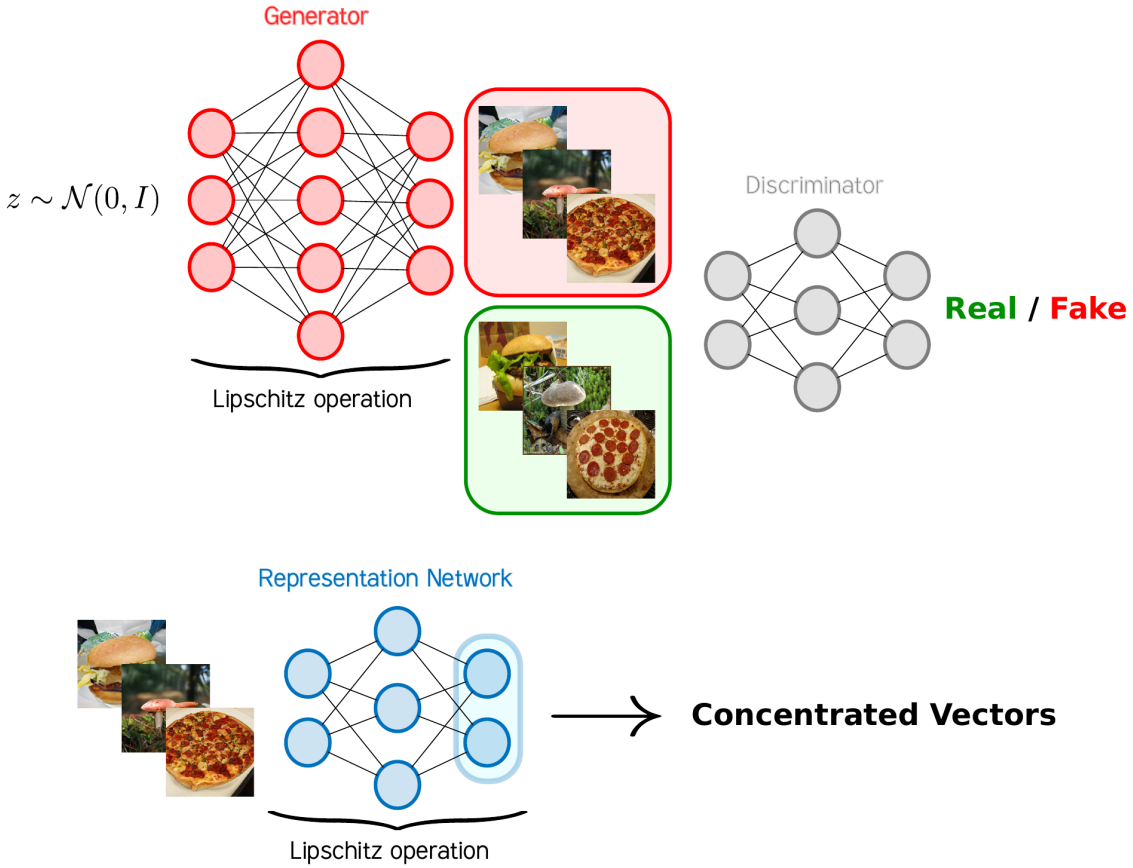



Figure 3.2: Deep learning representations of GAN-data are constructed by applying successive Lipschitz operations to Gaussian vectors, therefore they are *concentrated* vectors by design, since Gaussian vectors are concentrated and thanks to the Lipschitz stability in Proposition 2.4

*Proof.* This is a consequence of Proposition 2.4 since the map $\mathcal{G}$ is $\sup_i \|\mathcal{G}_i\|_{lip}$-Lipschitz with respect to (w.r.t.) the Frobenius norm. Indeed, for $\boldsymbol{X}, \boldsymbol{H} \in \mathbb{R}^{d\times n}$ : $\|\mathcal{G}(\boldsymbol{X}+\boldsymbol{H}) - \mathcal{G}(\boldsymbol{X})\|_F^2 \leq \sum_{i=1}^n \|\mathcal{G}_i\|_{lip}^2 \cdot \|\boldsymbol{H}_{:,i}\|^2 \leq \sup_i \|\mathcal{G}_i\|_{lip}^2 \cdot \|\boldsymbol{H}\|_F^2$. □

Indeed, a generator of a Conditional GAN model can be seen as a set of multiple generators where each generates data of a specific class conditionally on the class label (*e.g.*, BigGAN model [BDS18]).

Yet, in order to ensure that the resulting Lipschitz constant of the combination of the above operations does not scale with the network or data size, so to maintain good concentration behaviors, a careful control of the learned network parameters is needed. This control happens to be already considered in practice in order to ensure the stability of GANs during the learning phase, notably to generate realistic and high-resolution images [RLNH17, BDS18]. The control of the Lipschitz constant of representation networks is also needed in practice in order to make them robust against adversarial examples [SZS+13, GAA+17]. This control is particularly ensured through spectral normalization of the affine layers [BDS18], such as Fully Connected Layers, Convolutional Layers and Batch Normalization. Indeed, spectral normalization [MKKY18] consists in applying the operation $\boldsymbol{W} \leftarrow \boldsymbol{W}/\sigma_1(\boldsymbol{W})$ to the affine layers at each backward iteration of the back-

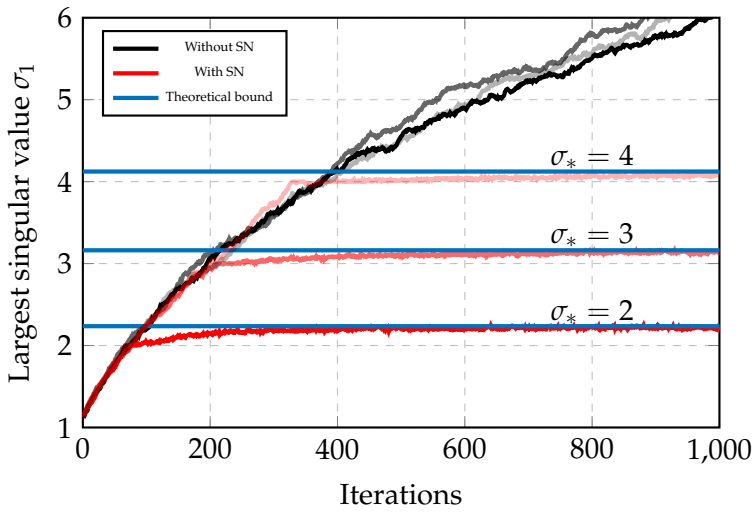



Figure 3.3: Behavior of the largest singular value of a weight matrix in terms of the iterations of a random walk (see proposition 3.1), without spectral normalization in **(black)** and with spectral normalization in (red). The (blue) lines correspond to the theoretical bound $\sqrt{\sigma_*^2 + \eta^2 d_1 d_0}$ for different $\sigma_*$'s. We took $d_0 = d_1 = 100$ and $\eta = 1/d_0$.

propagation algorithm, where $\sigma_1(\boldsymbol{W})$ stands for the largest singular value of the weight matrix $\boldsymbol{W}$. [BDS18], have notably observed that, without spectral constraints, a subset of the generator layers grow throughout their GAN training and explode at collapse. They thus suggested the following spectral normalization –which happens to be less restrictive than the standard spectral normalization $\boldsymbol{W} \leftarrow \boldsymbol{W}/\sigma_1(\boldsymbol{W})$ [MKKY18]– to the affine layers:

$$\boldsymbol{W} \leftarrow \boldsymbol{W} - (\sigma_1(\boldsymbol{W}) - \sigma_*)\, \boldsymbol{u}_1(\boldsymbol{W}) \boldsymbol{v}_1(\boldsymbol{W})^\intercal \tag{3.3}$$

where $\boldsymbol{u}_1(\boldsymbol{W})$ and $\boldsymbol{v}_1(\boldsymbol{W})$ denote respectively the left and right largest singular vectors of $\boldsymbol{W}$, and $\sigma_*$ is an hyper-parameter fixed during training.

To get an insight about the influence of this operation and to ensure that it controls the Lipschitz constant of the generator, the following proposition provides the dynamics of a random walk in the space of parameters along with the spectral normalization in Equation (3.3). Indeed, since stochastic gradient descent (SGD) consists in estimating the gradient of the loss function on randomly selected batches of data, it can be assimilated to a random walk in the space of parameters [ASD18].

**Proposition 3.1** (Lipschitz constant control). *Let $\sigma_* > 0$ and $\mathcal{G}$ be a neural network composed of $N$ affine layers, each one of input dimension $d_{i-1}$ and output dimension $d_i$ for $i \in [N]$, with 1-Lipschitz activation functions. Assume that the weights of $\mathcal{G}$ at layer $i+1$ are initialized as $\mathcal{U}([-\frac{1}{\sqrt{d_i}}, \frac{1}{\sqrt{d_i}}])$, and consider the following dynamics with learning rate $\eta$:*

$$\begin{aligned} &\boldsymbol{W} \leftarrow \boldsymbol{W} - \eta \boldsymbol{E}, \text{ with } \boldsymbol{E}_{i,j} \sim \mathcal{N}(0,1) \\ &\boldsymbol{W} \leftarrow \boldsymbol{W} - \max(0, \sigma_1(\boldsymbol{W}) - \sigma_*)\, \boldsymbol{u}_1(\boldsymbol{W}) \boldsymbol{v}_1(\boldsymbol{W})^\intercal. \end{aligned} \tag{3.4}$$

*Then, $\forall \varepsilon > 0$, the Lipschitz constant of $\mathcal{G}$ is bounded at convergence with high probability as:*

$$\|\mathcal{G}\|_{lip} \leq \prod_{i=1}^{N} \left( \varepsilon + \sqrt{\sigma_*^2 + \eta^2 d_i d_{i-1}} \right). \tag{3.5}$$

*Proof.* Since the Lipschitz constant of a composition of Lipschitz functions is bounded by the product of their Lipschitz constants, we consider the case $N = 1$ and a linear activation function. In this case, the Lipschitz constant corresponds to the largest singular value of the weight matrix. We consider the following notations for the proof

$$\begin{aligned} \bar{\boldsymbol{W}}_t &= \boldsymbol{W}_t - \eta \boldsymbol{E}_t \text{ with } [\boldsymbol{E}_t]_{i,j} \sim \mathcal{N}(0,1) \\ \boldsymbol{W}_{t+1} &= \bar{\boldsymbol{W}}_t - \max(0, \bar{\sigma}_{1,t} - \sigma_*)\, \bar{\boldsymbol{u}}_{1,t} \bar{\boldsymbol{v}}_{1,t}^{\mathsf{T}} \end{aligned}$$

where $\bar{\sigma}_{1,t} = \sigma_1(\bar{\boldsymbol{W}}_t)$, $\bar{\boldsymbol{u}}_{1,t} = \boldsymbol{u}_1(\bar{\boldsymbol{W}}_t)$ and $\bar{\boldsymbol{v}}_{1,t} = \boldsymbol{v}_1(\bar{\boldsymbol{W}}_t)$. The effect of spectral normalization is observed in the case where $\sigma_* > \bar{\sigma}_{1,t}$, otherwise the Lipschitz constant is bounded by $\sigma_*$. We therefore have

$$\|\bar{\boldsymbol{W}}_t\|_F^2 \leq \|\boldsymbol{W}_t\|_F^2 + \eta^2 d_1 d_0 \tag{3.6}$$

$$\|\boldsymbol{W}_{t+1}\|_F^2 = \|\bar{\boldsymbol{W}}_t\|_F^2 + \sigma_*^2 - \bar{\sigma}_{1,t}^2 \tag{3.7}$$

- If $\|\boldsymbol{W}_{t+1}\|_F \geq \|\boldsymbol{W}_t\|_F$, we have by equation 3.6 and equation 3.7

  $$\|\bar{\boldsymbol{W}}_t\|_F^2 \leq \|\bar{\boldsymbol{W}}_t\|_F^2 + \sigma_*^2 - \bar{\sigma}_{1,t}^2 + \eta^2 d_1 d_0 \;\Rightarrow\; \|\bar{\boldsymbol{W}}_t\| = \bar{\sigma}_{1,t} \leq \sqrt{\sigma_*^2 + \eta^2 d_1 d_0} = \delta$$

  And since $\|\boldsymbol{W}_{t+1}\| \leq \|\bar{\boldsymbol{W}}_t\|$, we have $\|\boldsymbol{W}_{t+1}\| \leq \delta$.

- Otherwise, if there exits $\tau$ such that $\|\boldsymbol{W}_{\tau+1}\|_F < \|\boldsymbol{W}_\tau\|_F$, then for all $\varepsilon > 0$ there exists an iteration $\tau' \geq \tau$ such that $\|\boldsymbol{W}_{\tau'}\| \leq \delta + \varepsilon$. Indeed, otherwise we denote $\varepsilon_t = \|\boldsymbol{W}_t\|^2 - \delta^2$ and $\varepsilon_t > 0$ for all $t \geq \tau$. And if for all $t \geq \tau$, $\|\boldsymbol{W}_{t+1}\|_F \leq \|\boldsymbol{W}_t\|_F$, we have by equation 3.6 and equation 3.7

  $$\|\boldsymbol{W}_t\|_F^2 - \|\boldsymbol{W}_{t+1}\|_F^2 \geq \|\bar{\boldsymbol{W}}_t\|^2 - \delta^2 \geq \|\boldsymbol{W}_{t+1}\|^2 - \delta^2 = \varepsilon_{t+1}$$

  Integrating the above expression from $\tau$ to $T - 1 \geq \tau$, we end up with

  $$\|\boldsymbol{W}_\tau\|_F^2 - \|\boldsymbol{W}_T\|_F^2 \geq \sum_{t=\tau}^{T-1} \varepsilon_t \;\Rightarrow\; 0 \leq \|\boldsymbol{W}_T\|_F^2 \leq \|\boldsymbol{W}_\tau\|_F^2 - \sum_{t=\tau}^{T-1} \varepsilon_t,$$

  therefore, when $T \to \infty$, $\varepsilon_t$ has to tend to 0 otherwise the right hand-side of the last inequality will tend to $-\infty$ which is absurd.

□

Proposition 3.1 shows that the Lipschitz constant of a neural network is controlled when trained with the spectral normalization in Equation (3.3). In particular, recalling the notations in Proposition 3.1, in the limit where $d_i \to \infty$ with $\frac{d_i}{d_{i-1}} \to \gamma_i \in (0, \infty)$ for all $i \in [N]$ and choosing the learning rate $\eta = \mathcal{O}(d_0^{-1})$, the Lipschitz constant of $\mathcal{G}$ is of order $\mathcal{O}(1)$ if it has finitely many layers $N$ and $\sigma_*$ is constant. Therefore, with this spectral normalization, it can be assumed that $\|\mathcal{G}\|_{lip} = \mathcal{O}(1)$ when dimensions grow. Figure 3.3 depicts the behavior of the Lipschitz constant of a linear layer with and without spectral normalization in the setting of Proposition 3.1, which confirms the obtained bound.

## 3.2 Random Gram Matrices of Concentrated Data

Now we turn to the analysis of the behavior of large Gram matrices assuming the data being concentrated vectors.

### 3.2.1 Motivation

As it represents the canonical form of similarity, the Gram matrix is at the core of various machine learning algorithms. Moreover, it provides a natural way to quantify the quality of a given representation. Indeed, the performance of machine learning methods depends strongly on the choice of the data representation (or features) on which they are applied. This data representation should ideally contain *relevant information* about the learning task in order to achieve learning with *simple* models and *small* amount of samples. In this sense, the simplest machine learning model is naturally a linear classifier which relies on the Gram matrix, and considering a small amount of data put us naturally in the random matrix theory regime where both $p$ and $n$ are large and comparable. In terms of representation learning, Deep neural networks [RHW+88] have particularly shown impressive results by automatically learning representations from raw data (*e.g.*, images). However, due to the complex structure of deep learning models, the characterization of their hidden representations is still an open problem [B+09].

Specifically, quantifying what makes a given deep learning representation better than another is a fundamental question in the field of *Representation Learning* [BCV13]. Relying on [MBM11] a data representation is said to be *good* when it is possible to build *simple* models on top of it that are *accurate* for the given learning problem. Authors in [MBM11] have notably quantified the layer-wise evolution of the representation in deep networks by computing the principal components of the Gram matrix $\boldsymbol{G}_\ell = \{\phi_\ell(\boldsymbol{x}_i)^\intercal \phi_\ell(\boldsymbol{x}_j)\}_{i,j=1}^n$ at each layer for $n$ input data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, where $\phi_\ell(\boldsymbol{x})$ is the representation of $\boldsymbol{x}$ at layer $\ell$ of the given DL model, and the number of components controls the model simplicity. In their study, the impact of the representation at each layer is quantified through the prediction error of a linear classifier trained on the principal subspace of $\boldsymbol{G}_\ell$.

Pursuing on this idea, given a certain representation model $\boldsymbol{x} \mapsto \phi(\boldsymbol{x})$, we aim in this study at theoretically analyzing the large dimensional behavior, and in particular the spectral information (*i.e.*, eigenvalues and dominant eigenvectors), of the corresponding Gram matrix $\boldsymbol{G} = \{\phi(\boldsymbol{x}_i)^\intercal \phi(\boldsymbol{x}_j)\}_{i,j=1}^n$ in order to determine the information encoded (*i.e.*, the sufficient statistics) by the representation model on a set of real data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. Indeed, standard classification and regression algorithms –along with the last layer of a neural network [YKYR18]– retrieve the data information directly from functionals or the eigenspectrum of $\boldsymbol{G}$[1]. To this end, though, one needs a statistical model for the representations given the distribution of the raw data (*e.g.*, images) which is generally unknown and not analytically tractable. Yet, as we have shown in the previous section, due to recent advances in generative models since the advent of Generative Adversarial Nets [GPAM+14], it is now possible to generate complex data structures by applying successive *Lipschitz* operations to Gaussian random vectors. In particular, GAN-data *are* used in practice as substitutes of real data for data augmentation [ASE17]. On the other hand, the fundamental concentration of measure phenomenon [Led05a] tells us

[1] For instance, spectral clustering uses the dominant eigenvectors of $\boldsymbol{G}$, while support vector machines use functionals (quadratic forms) involving $\boldsymbol{G}$.

that Lipschitz-ally transformed Gaussian vectors satisfy a concentration property. Precisely, defining the class of *concentrated* vectors $\boldsymbol{x} \in E$ through concentration inequalities of $f(\boldsymbol{x})$, for any real Lipschitz observation $f : E \to \mathbb{R}$, implies that deep learning representations of GAN-data fall within this class of random vectors, since the mapping $\boldsymbol{x} \mapsto \phi(\boldsymbol{x})$ *is* Lipschitz. Thus, GAN-data are concentrated random vectors and thus a more appropriate statistical model of realistic data, as we demonstrated in the previous section.

Targeting classification applications by assuming a mixture of concentrated random vectors model (see Definition 2), this study describes the spectral behavior of Gram matrices $\boldsymbol{G}$ in the large $n, p$ regime. Precisely, we show that these matrices have asymptotically (as $n, p \to \infty$ with $p/n \to c < \infty$) the same first-order behavior as for a Gaussian Mixture Model (GMM). As a result, by generating images using the BigGAN model [BDS18] and considering different commonly used deep representation models, we show that the spectral behavior of the Gram matrix computed on these representations is the same as on a GMM model with the same $p$-dimensional means and covariances. A surprising consequence is that, for GAN data, the aforementioned *sufficient statistics* to characterize the quality of a given representation network are only the *first* and *second* order statistics of the representations. This behavior is shown by simulations to extend beyond random GAN-data to real images from the Imagenet dataset [DDS+09].

### 3.2.2 Model and Main Results

#### 3.2.2.1 Mixture of Concentrated Vectors

In this section, we assume data to be a mixture of concentrated random vectors with controlled $\mathcal{O}(1)$ Lipschitz constant (*e.g.*, DL representations of GAN-data as we discussed in the previous section). Precisely, let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ be a set of mutually independent random vectors in $\mathbb{R}^p$. We suppose that these vectors are distributed as one of $k$ classes of distribution laws $\mathcal{L}_1, \ldots, \mathcal{L}_k$ with distinct means $\{\boldsymbol{\mu}_\ell\}_{\ell=1}^k$ and "covariances" $\{\boldsymbol{\Sigma}_\ell\}_{\ell=1}^k$ defined respectively as

$$\boldsymbol{\mu}_\ell = \mathbb{E}_{\boldsymbol{x}_i \sim \mathcal{L}_\ell}[\boldsymbol{x}_i], \quad \boldsymbol{\Sigma}_\ell = \mathbb{E}_{\boldsymbol{x}_i \sim \mathcal{L}_\ell}[\boldsymbol{x}_i \boldsymbol{x}_i^\mathsf{T}]. \tag{3.8}$$

For some $q > 0$, we consider a $q$-exponential concentration property on the laws $\mathcal{L}_\ell$, in the sense that for any family of independent vectors $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_s$ sampled from $\mathcal{L}_\ell$, $[\boldsymbol{y}_1, \ldots, \boldsymbol{y}_s] \in \mathcal{E}_q$ (see Definition 7). Without loss of generality, we arrange the $\boldsymbol{x}_i$'s in a data matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ such that, for each $\ell \in [k]$

$$\boldsymbol{x}_{1+\sum_{j=1}^{\ell-1} n_j}, \ldots, \boldsymbol{x}_{\sum_{j=1}^{\ell} n_j} \sim \mathcal{L}_\ell(\boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell)$$

where $n_\ell$ stands for the number of $\boldsymbol{x}_i$'s sampled from $\mathcal{L}_\ell$. In particular, we have the concentration of the data matrix $\boldsymbol{X}$ as in the following assumption

**Assumption 3** (Concentrated data)**.** *We assume $\boldsymbol{X} \propto \mathcal{E}_q$ for some $q > 1$.*

Such a data matrix $\boldsymbol{X}$ can be constructed through Lipschitz-ally transformed Gaussian vectors ($q = 2$), with controlled Lipschitz constant, thanks to Corollary 3.1. In particular, DL representations of GAN-data are constructed as such, as shown in Section 3.1. We further introduce the following notations that will be used subsequently.

$$\boldsymbol{M} = [\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k] \in \mathbb{R}^{p \times k}, \; \boldsymbol{J} = [\boldsymbol{j}_1, \ldots, \boldsymbol{j}_k] \in \mathbb{R}^{n \times k}, \; \boldsymbol{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n] \in \mathbb{R}^{p \times n},$$

where $\boldsymbol{j}_\ell \in \mathbb{R}^n$ stands for the canonical vector selecting the $\boldsymbol{x}_i$'s of distribution $\mathcal{L}_\ell$, defined by $(\boldsymbol{j}_\ell)_i = \mathbf{1}_{\boldsymbol{x}_i \sim \mathcal{L}_\ell}$, and the $\boldsymbol{z}_i$'s are the centered versions of the $\boldsymbol{x}_i$'s, *i.e.* $\boldsymbol{z}_i = \boldsymbol{x}_i - \boldsymbol{\mu}_\ell$ for $\boldsymbol{x}_i \sim \mathcal{L}_\ell(\boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell)$.

#### 3.2.2.2 Behavior of the Gram matrix of concentrated vectors

Now we study the behavior of the Gram matrix $\boldsymbol{G} = \frac{1}{p}\boldsymbol{X}^\intercal \boldsymbol{X}$ in the large $n, p$ limit and under the model of the previous section. Indeed, $\boldsymbol{G}$ appears as a central component in many classification, regression and clustering methods. Precisely, a finer description of the behavior of $\boldsymbol{G}$ provides access to the internal functioning and performance evaluation of a wide range of machine learning methods such as Least Squares SVMs [A+02], Semi-supervised Learning [CSZ09] and Spectral Clustering [NJW02]. Indeed, the performance evaluation of these methods has already been studied under GMM models in [LC17, MC17, CBG+16] through RMT. On the other hand, analyzing the spectral behavior of $\boldsymbol{G}$ for DL representations quantifies their quality –through its principal subspace [MBM11]– as we have discussed in the introduction. In particular, the Gram matrix decomposes as

$$\boldsymbol{G} = \frac{1}{p}\boldsymbol{J}\boldsymbol{M}^\intercal \boldsymbol{M}\boldsymbol{J}^\intercal + \frac{1}{p}\boldsymbol{Z}^\intercal \boldsymbol{Z} + \frac{1}{p}(\boldsymbol{J}\boldsymbol{M}^\intercal \boldsymbol{Z} + \boldsymbol{Z}^\intercal \boldsymbol{M}\boldsymbol{J}^\intercal). \tag{3.9}$$

Intuitively $\boldsymbol{G}$ decomposes as a low-rank informative matrix containing the class canonical vectors through $\boldsymbol{J}$ and a noise term represented by the other matrices and essentially $\boldsymbol{Z}^\intercal \boldsymbol{Z}$. Given the form of this decomposition, RMT predicts –through an analysis of the spectrum of $\boldsymbol{G}$ and under a GMM model [BGC16]– the existence of a threshold $\xi$ function of the ratio $p/n$ and the data statistics for which the dominant eigenvectors of $\boldsymbol{G}$ contain information about the classes only when $\|\boldsymbol{M}^\intercal \boldsymbol{M}\| \geq \xi$ asymptotically (*i.e.*, only when the means of the different classes are sufficiently distinct). See Subsection 2.3.1 for an illustrative example.

As we saw in Chapter 2, in order to characterize the spectral behavior (*i.e.*, eigenvalues and leading eigenvectors) of $\boldsymbol{G}$ under the concentration assumption in Assumption 3 on $\boldsymbol{X}$, we will be interested in studying the behavior of its resolvent defined as, for $z \in \mathbb{C} \setminus \mathbb{R}_-$

$$\boldsymbol{R}(z) = (\boldsymbol{G} + z\boldsymbol{I}_n)^{-1} \tag{3.10}$$

Practically speaking for spectral clustering with $\boldsymbol{G}$, since $\boldsymbol{R}(z)$ and $\boldsymbol{G}$ share the same eigenvectors with associated eigenvalues $\frac{1}{\lambda_i + z}$ for $\boldsymbol{R}(z)$ with $\{\lambda_i\}_{i=1}^n$ the eigenvalues of $\boldsymbol{G}$, the projector matrix corresponding to the top $m$ eigenvectors $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m]$ of $\boldsymbol{G}$ can be calculated through a Cauchy integral $\boldsymbol{U}\boldsymbol{U}^\intercal = \frac{-1}{2\pi i}\oint_\gamma \boldsymbol{R}(-z)dz$ where $\gamma$ is an oriented complex contour surrounding the top $m$ eigenvalues of $\boldsymbol{G}$. See Subsection 2.1.3 for more details.

To study the behavior of $\boldsymbol{R}(z)$, we look for a so-called *deterministic equivalent* [HLN+07] (see Definition 6) $\tilde{\boldsymbol{R}}(z)$ for $\boldsymbol{R}(z)$. In the following, we present our main result which gives such a deterministic equivalent under the concentration assumption on $\boldsymbol{X}$ in Assumption 3 and under the following assumptions.

**Assumption 4.** *As* $p \to \infty$,

1. $p/n \to c \in (0, \infty)$,

2. *The number of classes $k$ is bounded,*

3. $\|\boldsymbol{\mu}_\ell\| = \mathcal{O}(\sqrt{p})$ *and* $\mathbb{E}\|\boldsymbol{x}_i\| = \mathcal{O}(\sqrt{p})$.

**Theorem 3.1** (Deterministic Equivalent for $\boldsymbol{R}(z)$)**.** *Under the model described in Section 3.2.2.1 and Assumptions 3-4, we have $\boldsymbol{R}(z) \in \mathcal{E}_q(p^{-1/2})$. Furthermore,*

$$\boldsymbol{R}(z) \leftrightarrow \bar{\boldsymbol{R}}(z) = \frac{1}{z}\operatorname{diag}\left\{\frac{\boldsymbol{I}_{n_\ell}}{1+\delta_\ell^*(z)}\right\}_{\ell=1}^{k} + \frac{1}{p\,z}\boldsymbol{J}\Omega_z\boldsymbol{J}^\intercal$$

*Specifically,*

$$\|\mathbb{E}\boldsymbol{R}(z) - \bar{\boldsymbol{R}}(z)\| = \mathcal{O}\left(\sqrt{\frac{\log(p)}{p}}\right)$$

*with* $\Omega_z = \boldsymbol{M}^\intercal\bar{\boldsymbol{Q}}(z)\boldsymbol{M} \odot \operatorname{diag}\left\{\frac{\delta_\ell^*(z)-1}{\delta_\ell^*(z)+1}\right\}_{\ell=1}^{k}$ *and* $\bar{\boldsymbol{Q}}(z) = \left(\frac{1}{ck}\sum_{\ell=1}^{k}\frac{\boldsymbol{\Sigma}_\ell}{1+\delta_\ell^*(z)} + z\boldsymbol{I}_p\right)^{-1}$ *where* $\delta^*(z) = [\delta_1^*(z),\ldots,\delta_k^*(z)]^\intercal$ *is the unique fixed point of the system of equations for each* $\ell \in [k]$

$$\delta_\ell(z) = \frac{1}{p}\operatorname{tr}\left(\boldsymbol{\Sigma}_\ell\left(\frac{1}{ck}\sum_{j=1}^{k}\frac{\boldsymbol{\Sigma}_j}{1+\delta_j(z)} + z\boldsymbol{I}_p\right)^{-1}\right)$$

*Sketch of proof.* The first step of the proof is to show the concentration of $\boldsymbol{R}(z)$. This comes from the fact that the application $\boldsymbol{X} \mapsto \boldsymbol{R}(z)$ is $2z^{-3/2}p^{-1/2}$-Lipschitz w.r.t. the Frobenius norm, thus we have by Proposition 2.4 that $\boldsymbol{R}(z) \in \mathcal{E}_q(p^{-1/2})$.

The second step consists in estimating $\mathbb{E}\boldsymbol{R}(z)$ through a deterministic matrix $\tilde{\boldsymbol{R}}(z)$. Indeed, $\boldsymbol{R}(z)$ can be expressed as a function of $\boldsymbol{Q}(z) = (\boldsymbol{X}\boldsymbol{X}^\intercal/p + z\boldsymbol{I}_p)^{-1}$ as $\boldsymbol{R}(z) = z^{-1}(\boldsymbol{I}_n - \boldsymbol{X}^\intercal\boldsymbol{Q}(z)\boldsymbol{X}/p)$, where the statistical dependency between $\boldsymbol{X}$ and $\boldsymbol{Q}(z)$ is handled through Propositions C.1 and C.2 and finally exploiting the result of [LC19] which shows that $\mathbb{E}\boldsymbol{Q}(z)$ can be estimated through $\bar{\boldsymbol{Q}}(z)$ as per Theorem 2.9, we obtain the estimator $\bar{\boldsymbol{R}}(z)$ for $\mathbb{E}\boldsymbol{R}(z)$.

A more detailed proof is provided the appendix in Section C.1. □

**Remark 3.1** (Equivalence between Theorem 3.1 and Theorem 2.5 )**.** *Note that Theorem 3.1 and Theorem 2.5 are equivalent when $\boldsymbol{M} = \boldsymbol{0}$ with the change of variable $c_j g_j(z) = \frac{1}{1+\delta_j(z)}$. In particular, Theorem 2.5 supposes a Gaussian mixture model with zero means while we generalize this result to non-zero means which appear through the matrix of means $\boldsymbol{M}$.*

This result allows specifically to (i) describe the limiting eigenvalues distribution of $\boldsymbol{G}$, (ii) determine the spectral detectability threshold mentioned above (See Subsection 2.3.1 in the case of binary unsupervised clustering), (iii) evaluate the asymptotic "content" of the leading eigenvectors of $\boldsymbol{G}$ (see again Subsection 2.3.1) and, much more fundamentally, (iv) infer the asymptotic performances of machine learning algorithms (See Subsection 2.3.2) that are based on simple functionals of $\boldsymbol{G}$ (*e.g.*, LS-SVM, spectral clustering etc.). Looking carefully at Theorem 3.1 we see that the spectral behavior of the Gram matrix $\boldsymbol{G}$ computed on concentrated vectors only depends on the *first* and *second* order statistics of the laws $\mathcal{L}_\ell$ (their means $\boldsymbol{\mu}_\ell$ and "covariances" $\boldsymbol{\Sigma}_\ell$). This suggests the surprising result that $\boldsymbol{G}$ has the same behavior as when the data follow a GMM model with the same means and covariances. The asymptotic spectral behavior of $\boldsymbol{G}$ is therefore *universal*


Figure 3.4: **(Top)** GAN generated images using the BigGAN model [BDS18]. **(Bottom)** Real images selected from the Imagenet dataset [DDS+09]. We considered $n = 1500$ images from $k = 3$ classes which are {mushroom, pizza, hamburger}.

with respect to the data distribution laws which satisfy the aforementioned concentration properties (for instance DL representations of GAN-data). We illustrate this universality result in the next section by considering data as CNN representations of GAN generated images.

#### 3.2.2.3 Application to GAN-generated Images

This section presents experiments that confirm the result of Theorem 3.1. In particular, we compare, in the first part, the eigenvalues distribution and the largest eigenvectors of the Gram matrix computed on deep learning representations with those of the Gram matrix computed on Gaussian data with the same first and second order moments. In the second part of this section, we evaluate the performance of a linear SVM model on the principal subspace of the Gram matrix (computed on the representations or on the corresponding Gaussian data) by varying the number of components in the same vein as the work of montavon2011kernel. In the following, all representation networks are standard convolutional neural networks pre-trained on the Imagenet dataset deng2009imagenet, in particular, we used pre-trained models of the Pytorch deep learning framework.

**Spectrum and Dominant Eigenspace of the Gram Matrix.** We consider $n = 1500$ data $x_1, \ldots, x_n \in \mathbb{R}^p$ as CNN representations –across popular CNN architectures of different sizes $p$– of GAN-generated images using the generator of the Big-GAN model [BDS18]. We further use real images from the Imagenet dataset [DDS+09] for comparison. In particular, we empirically compare the spectrum of the Gram matrix of this data with the

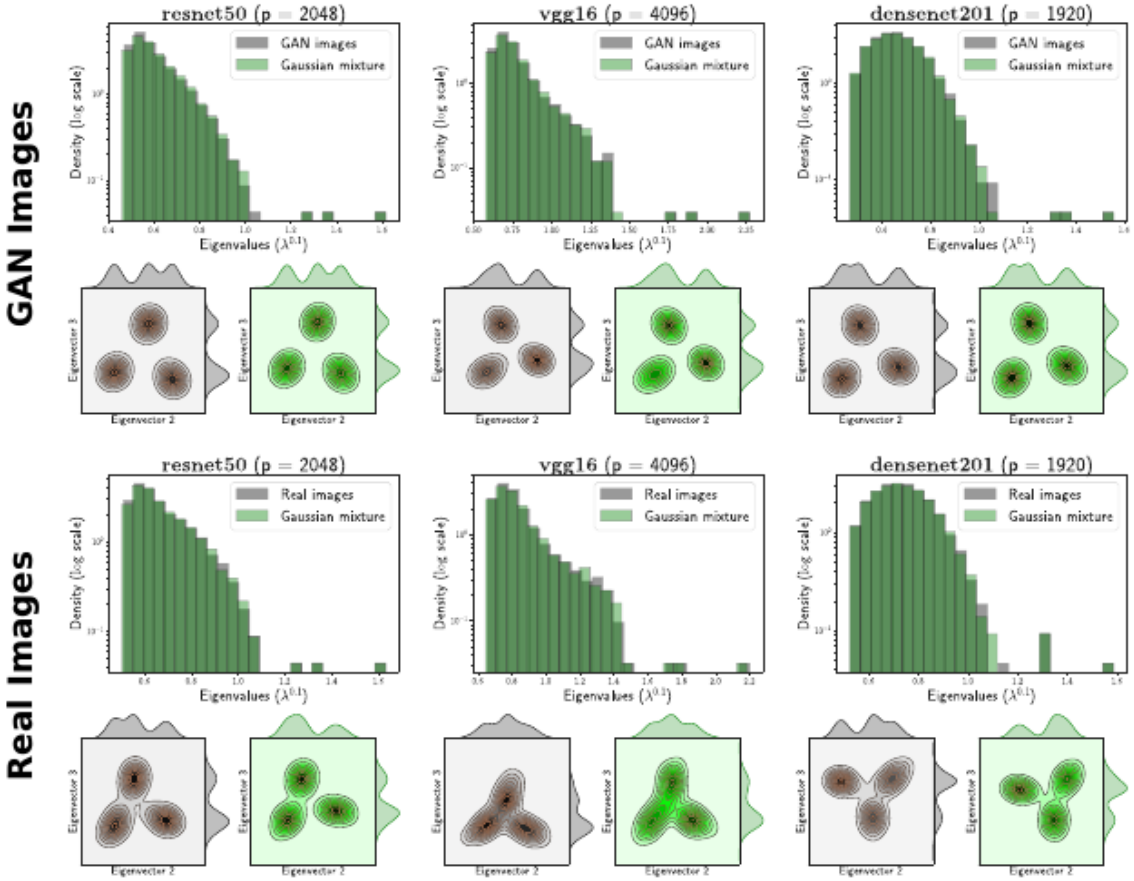



Figure 3.5: **(Top)** Spectrum and leading eigenspace of the Gram matrix for CNN representations of GAN generated images using the BigGAN model [BDS18]. **(Bottom)** Spectrum and leading eigenspace of the Gram matrix for CNN representations of real images selected from the Imagenet dataset [DDS+09]. Columns correspond to the three representation networks which are *resnet50*, *vgg16* and *densenet201*. We used $n = 1500$ images and considered $k = 3$ classes as depicted in Figure 3.4.

Gram matrix of a GMM model with the same means and covariances. We also consider the leading 2-dimensional eigenspace of the Gram matrix which contains clustering information as detailed in the previous section. Figure 3.4 depicts some images generated using the Big-GAN model (Top) and the corresponding real class images from the Imagenet dataset (Bottom). The Big-GAN model is visually able to generate highly realistic images which are by construction concentrated vectors, as discussed in Section 3.1 and therefore satisfy the assumptions of Theorem 3.1.

Figure 3.5 depicts the spectrum and leading 2D eigenspace of the Gram matrix computed on CNN representations of GAN generated and real images **(in gray)**, and the corresponding GMM model with same first and second order statistics (in green). The Gram matrix is seen to follow the same spectral behavior for GAN-data as for the GMM model which is a natural consequence of the universality result of Theorem 3.1 with respect to the data distribution. Besides, and perhaps no longer surprisingly, we further observe that the spectral properties of $\boldsymbol{G}$ for real data (here CNN representations of real images) are conclusively matched by their Gaussian counterpart.

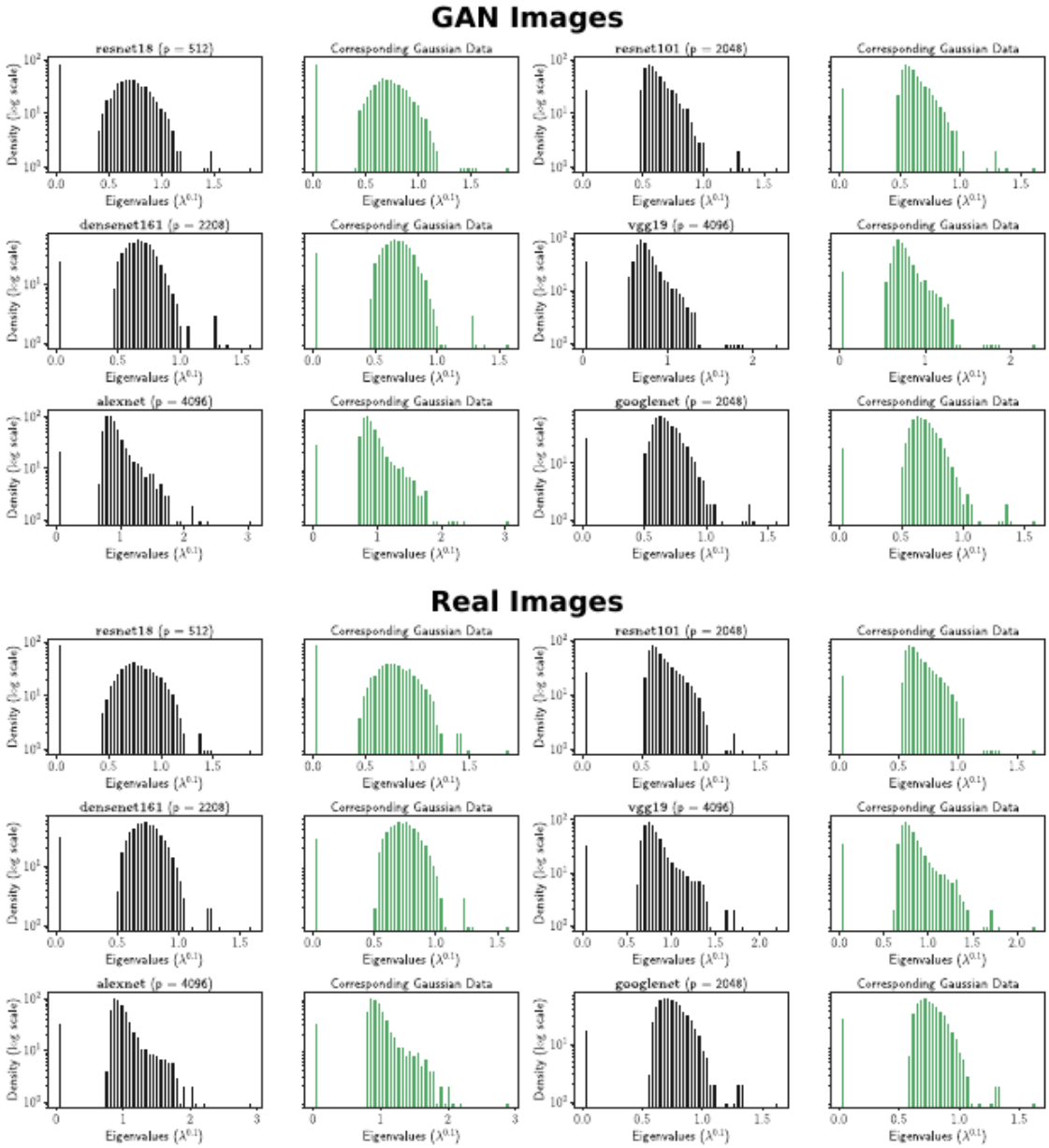



Figure 3.6: Spectrum of the Gram matrix for CNN representations in **(black)** and the corresponding Gaussian data (in green) for GAN generated images using the BigGAN model [BDS18] **(Top)** and for real images randomly selected from the Imagenet dataset [DDS+09] **(Bottom)**. The considered represenation network are *resnet18*, *resnet101*, *densenet161*, *vgg19*, *alexnet* and *googlenet*. We used $n = 600$ images selected among $k = 6$ classes {hamburger, mushroom, pizza, strawberry, coffee, daisy} (100 images per class).

Figure 3.6 shows more results about the Gram matrix spectrum of the representations **(in black)** and the corresponding Gaussian data (in green), by considering more representation networks and using $k = 6$ classes for both GAN images and real images, which confirms the result of Theorem 3.1. This both theoretically and empirically confirms that the proposed random matrix framework is fully compliant with the theoretical analysis

of real machine learning datasets. As a consequence, recalling the work of [MBM11], the *quality* of a given representation is quantified through the prediction accuracy of a linear classifier trained on the principal Gram matrix eigenvectors of the representations computed on a set of samples. Given our result in Theorem 3.1, and the fact that the top $m$ eigenvectors $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m]$ of $\boldsymbol{G}$ are related to the resolvent matrix $\boldsymbol{R}(z)$ through the Cauchy integral $\boldsymbol{U}\boldsymbol{U}^\intercal = \frac{-1}{2\pi i} \oint_\gamma \boldsymbol{R}(-z)dz$ where $\gamma$ is an oriented complex contour surrounding the top $m$ eigenvalues of $\boldsymbol{G}$, we should expect that the prediction accuracy of a linear classifier trained on the principal eigenvectors of $\boldsymbol{G}$ be the same for the representations themselves as for the corresponding Gaussian data with the same first and second order moments. Therefore, the purpose of the following section is to show simulations which confirm this result.

**Linear SVM Performance on the Dominant $k$-dimensional Eigenspace of $\boldsymbol{G}$.** Now we compare the performance of a linear SVM model trained on the dominant Gram matrix's $k$-dimensional eigenspace of the representations versus the corresponding Gaussian data with the same first and second order moments. Experiments were made in the following settings:

- Data types: We do the experiments for both GAN generated images using the BigGAN model [BDS18] and for real images randomly selected for the Imagenet dataset [DDS+09]. In both cases we consider $n = 6000$ images.
- Classes: We consider $k = 6$ classes which are: hamburger, mushroom, pizza, strawberry, coffee and daisy.
- Representation networks: We consider 9 representation networks pre-trained on the Imagenet dataset [DDS+09] which are: *vgg16* ($p = 4096$), *vgg19* ($p = 4096$), *resnet18* ($p = 512$), *resnet50* ($p = 2048$), *resnet101* ($p = 2048$), *densenet161* ($p = 2208$), *densenet201* ($p = 1920$), *alexnet* ($p = 4096$) and *googlenet* ($p = 2048$).

Figure 3.7 depicts the train and test accuracy of a linear SVM trained on the top $k$ eigenvectors of $\boldsymbol{G}$, for the representations (of GAN generated images) and the corresponding Gaussian data, for different values of $k$. As we can notice, the performance of the SVM model on the representations matches its performance on the corresponding Gaussian data with the same first and second order statistics as predicted by Theorem 3.1. This matching seems to extend beyond GAN images (which are concentrated vectors) to real images as depicted in Figure 3.8. As a consequence, our results suggest that the *quality* of a given representation network can be quantified through their first two statistical moments.

### 3.2.3 Central Contribution

The central contributions parts of this work are to highlight the universality aspects of large random matrices on real data, by studying the behavior of the Gram matrix for a large class of random vectors, the so-called *concentrated vectors*, which are much richer than Gaussian vectors. The concentration assumption is particularly motivated by the fact that realistic data can be generated using GANs by Lipschitz transformations of Gaussian vectors, which fall within the class of concentrated vectors. Particularly, by generating images across GANs, we have shown in this work that the spectral behavior

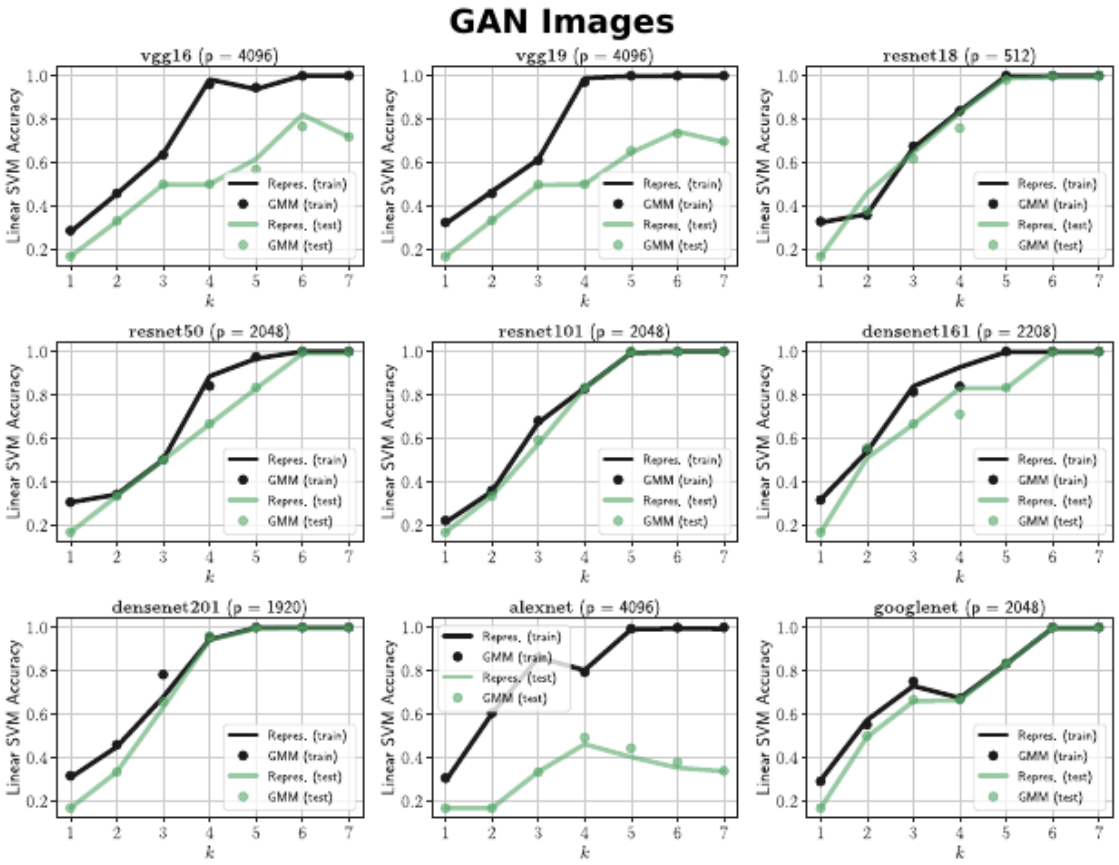



Figure 3.7: **Train** and test accuracy of a linear SVM model trained on the top $k$ eigenvectors of the Gram matrix, computed on the representations of GAN generated images. We generated with the BigGAN model [BDS18] $n = 6000$ images belonging to the 6 classes {hamburger, mushroom, pizza, strawberry, coffee, daisy} (1000 images per class). We considered 9 representation networks which are *vgg16*, *vgg19*, *resnet18*, *resnet50*, *resnet101*, *densenet161*, *densenet201*, *alexnet* and *googlenet*. Lines represent the performance of the SVM model on the representations themselves whereas dotes represent the performance of the SVM model on Gaussian data with the same first and second order moments. We used a train vs test split of $2/3$ and $1/3$ respectively.

of the Gram matrix is the same on the generated data as on a Gaussian mixture model, thereby making the estimation of machine learning algorithms (which are based on the Gram matrix) predictable through random matrix theory for real data if we can assimilate them to GAN generated data. In the next chapter we will go beyond the simple Gram matrix to the more general setting of kernel matrices, which constructively lead to *non-linear* ML methods, and therefore additional notions shall be introduced in order to break the non-linearity.

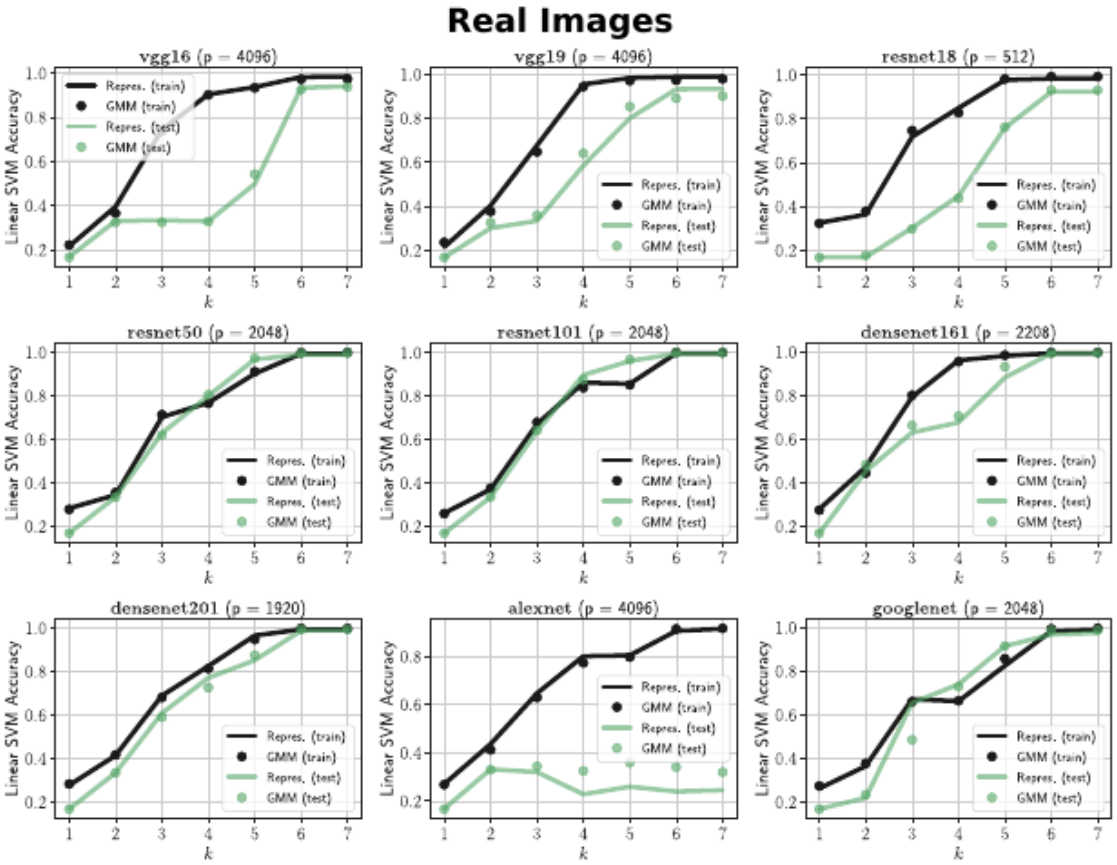



Figure 3.8: **Train** and test accuracy of a linear SVM model trained on the top $k$ eigenvectors of the Gram matrix, computed on the representations of Real images. We randomly sampled $n = 6000$ images belonging to the 6 classes {hamburger, mushroom, pizza, strawberry, coffee, daisy} (1000 images per class) of the Imagenet dataset [DDS+09]. We considered 9 representation networks *vgg16*, *vgg19*, *resnet18*, *resnet50*, *resnet101*, *densenet161*, *densenet201*, *alexnet* and *googlenet*. Lines represent the performance of the SVM model on the representations themselves whereas dotes represent the performance of the SVM model on Gaussian data with the same first and second order moments. We used a train vs test split of 2/3 and 1/3 respectively.

# Chapter 4

# Random Kernel Matrices of Concentrated Data

**Contents**



This chapter is composed of two main parts. The first part analyzes the behavior of kernel matrix of the for $\boldsymbol{K} = \{\frac{1}{p}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\}$ where the $\boldsymbol{x}_i$'s are supposed to be concentrated. The second part analyzes the behavior of kernel matrices of the form $f(\hat{\boldsymbol{\Sigma}})$ where $\boldsymbol{\Sigma}$ stands for the sample covariance matrix.

## 4.1 Kernel Spectral Clustering

This section is based on the following work:

**(C2)** **MEA. Seddik**, M. Tamaazousti, R. Couillet, "*Kernel Random Matrices of Large Concentrated Data: The Example of GAN-generated Images*", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'19), Brighton, United-Kingdom, 2019.

Kernel matrices generalize the inner product similarity of Gram matrices by introducing a non-linear function in order to capture non-linearties from data. the objective of this section is to analyze the behavior of these matrices under the concentration hypothesis and supposing the high-dimensional RMT regime.

### 4.1.1 Motivation

Gram matrices fall within the larger class of kernel matrices since they correspond to the particular case of inner-product similarity. In order to retrieve non-linear structures from data, the used methods for achieving classical classification or regression tasks rely on non-linear approaches including neural networks [KSH12, LC18c] and algorithms that are based on kernel methods, such as kernel-based support vector machines [LC17], semi-supervised classification [MC17], kernel-based PCA [STC19b] and spectral clustering [AKC18, CBG+16].

Due to their non-linear design, these methods are particularly difficult to analyze theoretically. For practical large and numerous data, the study of kernel-based methods relies on the characterization of kernel matrices $\boldsymbol{K} \in \mathbb{R}^{n\times n}$ in the large dimensional regime (*i.e.*, $p/n \to c_0$ as $n \to \infty$). Under *asymptotically non-trivial* growth rate assumptions on the data statistics (*i.e.*, maintaining a feasible get not too easy problem), the entries $K_{ij} = f(\boldsymbol{x}_i^\mathsf{T}\boldsymbol{x}_j/p)$ or $K_{ij} = f(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/p)$ of $\boldsymbol{K}$ tend to a limiting constant independently of the data classes – *the between and within class vectors are "equidistant" in high-dimension*. This observation allows one to study $\boldsymbol{K}$ through a Taylor expansion, thereby giving access to the characterization of functionals of $\boldsymbol{K}$ or its (informative) eigenvalue-eigenvector pairs in the large dimensional regime.

Indeed, such an analysis was initiated in [EK+10b] where it has been shown that $\boldsymbol{K}$ has a linear behavior in the large $p, n$ asymptotics. Under a $k$-class Gaussian mixture model, it has been shown in [CBG+16] that the normalized Laplacian matrix associated with $\boldsymbol{K}$ behaves asymptotically as a so-called spiked random matrix, where some of the isolated eigenvalues and eigenvectors contain clustering information. In particular, the authors in [CBG+16] demonstrated that the obtained theoretical model agrees with empirical results using the popular MNIST dataset [LeC98], thereby suggesting a sort of *universality* of spectral clustering regarding the underlying data distribution.

The aim of this study is to confirm this observation by relaxing the Gaussianity assumption to the class of concentrated vectors leveraging on the observation that GAN data fall within this class of random vectors as presented in Section 3.1. In this study, we analyze the kernel matrix $\boldsymbol{K}$ under a $k$-class *concentration* mixture model [LC19]. Precisely, we prove that $\boldsymbol{K}$ behaves (up to centering) asymptotically as a spiked random matrix in the large $p$ large $n$ regime, thereby generalizing the results of [CBG+16] to a broader class of distributions. We particularly confirm our theoretical findings by considering the input data as popular CNN representations of images generated by the BigGAN model [BDS18], where the latter is trained to fit the manifold distribution of the well-known Imagenet dataset. We further consider real images for comparison.

### 4.1.2 Model and Main Results

We consider the same concentration assumptions as for the analysis of the Gram matrix in the previous chapter, but we use slightly different notations and assumptions. Consider $n$ independent random vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^p$ distributed in $k$ classes represented by $k$ distributions $\mathcal{L}_1, \ldots, \mathcal{L}_k$ supposedly all distinct. We consider the hypothesis of $q$-exponential concentration, meaning that there exists $q \geq 2$ such that for all $s \in \mathbb{N}$, any $\ell \in [k]$ and any family of independent vectors $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_s$ following the distribution $\mathcal{L}_\ell$, we

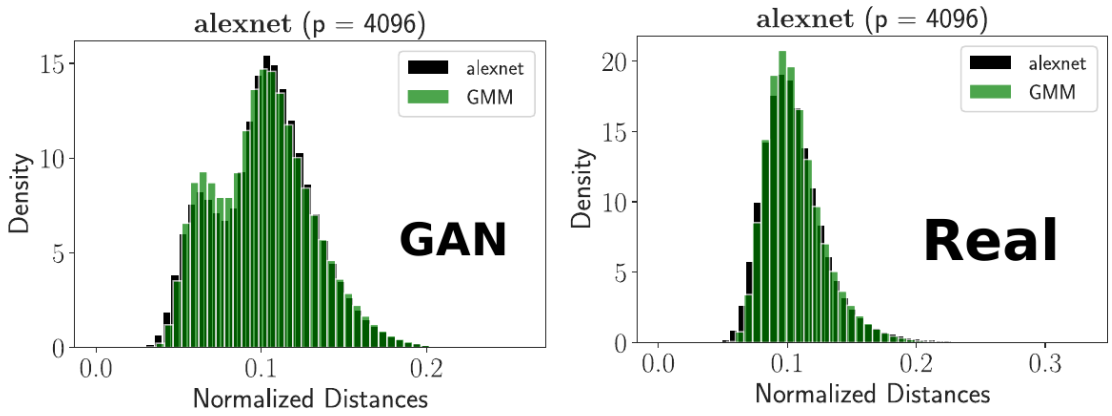



Figure 4.1: Histogram of distances with Alexnet representations for both GAN images **(on the left)** and real images **(on the right)**. We can notice a perfect match between the representations and the corresponding GMM data. Note that, for real images, the pairwise distances *concentrate* around a constant quantity as per Lemma 4.1, this comes from the fact that real images are hardly separable compared to GANs ones.

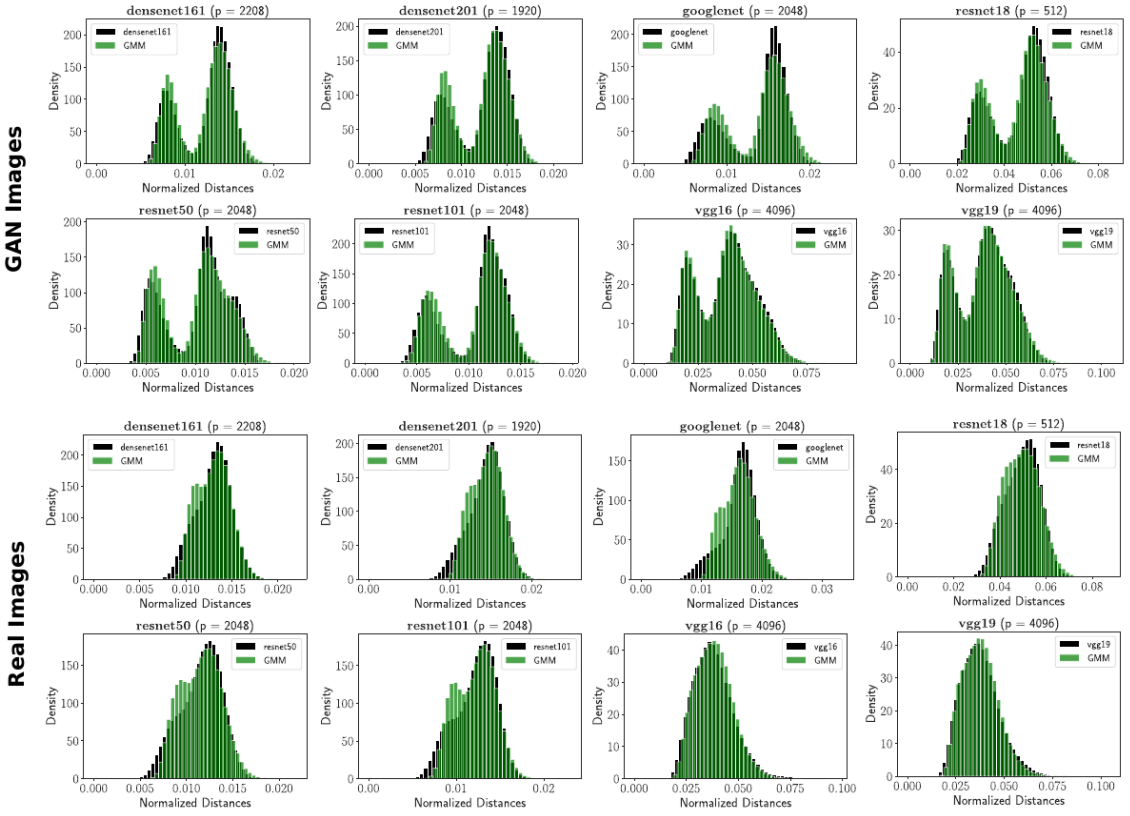



Figure 4.2: Histogram of distances across different representation networks for both GAN images **(on top)** and real images **(on bottom)**. We can notice a perfect match between the representations and the corresponding GMM data. Note that, for real images, the pairwise distances *concentrate* around a constant quantity as per Lemma 4.1, this comes from the fact that real images are hardly separable compared to GANs ones.

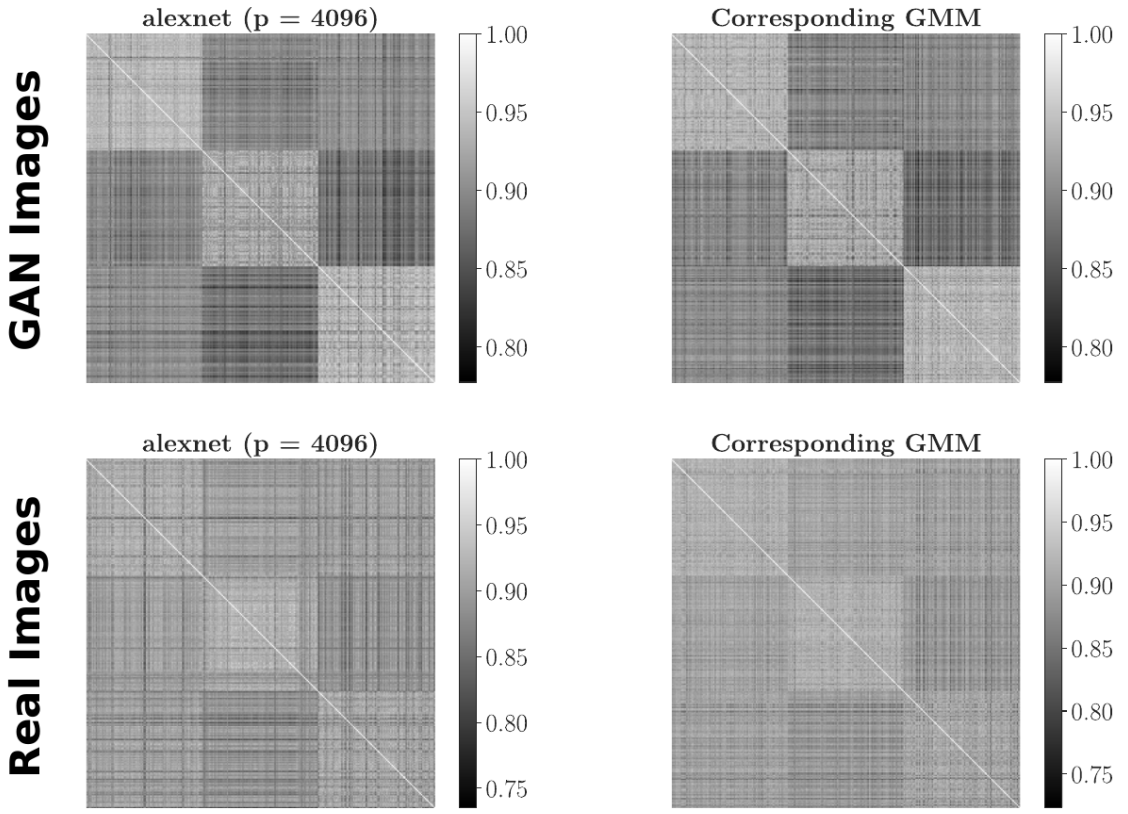



Figure 4.3: **(Left)** Kernel matrix of Alexnet representations for GAN images **(top)** and Real images **(bottom)** with the corresponding GMM data **(right)**. We can notice a perfect matching between the kernel matrix of the representations and the corresponding kernel with GMM data. We further note (at least for Real images) that all the entries of the kernel matrix tend to the same value $f(\tau_p)$ as a first order approximation. The kernel matrix is defined as $K_{ij} = f(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/p)$ where the $\boldsymbol{x}_i$'s stand for the representations (or the corresponding GMM data) and we have chosen $f(t) = exp(-t)$.

have the concentration $[\boldsymbol{y}_1, \ldots, \boldsymbol{y}_s] \propto \mathcal{E}_q$ (see Definition 7). In particular, we consider the concentration of $\boldsymbol{X} \equiv [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathcal{M}_{p,n}$ as per the following assumption

**Assumption 5** (Data concentration)**.** $\boldsymbol{X} \propto \mathcal{E}_q$.

For $\ell \in [k]$, we denote by $\boldsymbol{\mu}_\ell$ the mean of the distribution $\mathcal{L}_\ell$, $\boldsymbol{\Sigma}_\ell$ denotes its covariance matrix, defined respectively as

$$\boldsymbol{\mu}_\ell \equiv \mathbb{E}_{\boldsymbol{x}\sim\mathcal{L}_\ell}[\boldsymbol{x}], \quad \boldsymbol{\Sigma}_\ell \equiv \mathbb{E}_{\boldsymbol{x}\sim\mathcal{L}_\ell}[\boldsymbol{x}\boldsymbol{x}^\intercal] - \boldsymbol{\mu}_\ell\boldsymbol{\mu}_\ell^\intercal, \tag{4.1}$$

and $n_\ell$ stands for the number of vectors among the $x_i$'s following $\mathcal{L}_\ell$. Let $\boldsymbol{\mu} \in \mathbb{R}^p$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{p\times p}$ be respectively defined as

$$\boldsymbol{\mu} \equiv \sum_{\ell=1}^{k} \frac{n_\ell}{n}\boldsymbol{\mu}_\ell, \quad \boldsymbol{\Sigma} \equiv \sum_{\ell=1}^{k} \frac{n_\ell}{n}\boldsymbol{\Sigma}_\ell \tag{4.2}$$

We further denote $\bar{\boldsymbol{\mu}}_\ell \equiv \boldsymbol{\mu}_\ell - \boldsymbol{\mu}$ and $\bar{\boldsymbol{\Sigma}}_\ell \equiv \boldsymbol{\Sigma}_\ell - \boldsymbol{\Sigma}$.

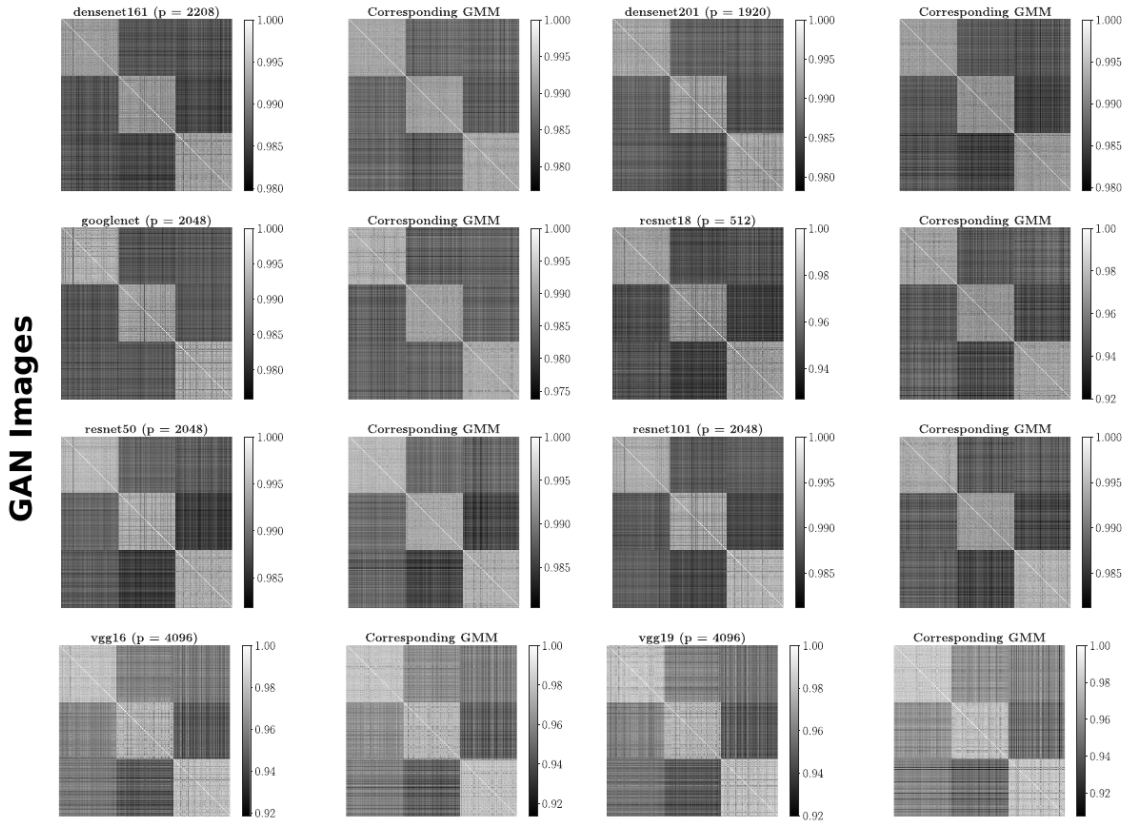



Figure 4.4: Kernel matrix of CNN representations of GAN data using different representations networks. The kernel matrix is defined as $K_{ij} = f(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/p)$ where the $\boldsymbol{x}_i$'s stand for the representations (or the corresponding GMM data) and we have chosen $f(t) = exp(-t)$.

We shall consider the following set of assumptions on the data statistics and the kernel function in the large dimensional regime, meaning that both $p$ and $n$ grow at controlled joint rate. These assumptions notably guarantee the non-triviality of spectral clustering under the considered regime as we have shown in the introduction (see Subsection 1.3.2).

**Assumption 6** (Growth rate). *As $p \to \infty$, consider the following conditions:*

- *(Data)* $\frac{p}{n} \to c_0 \in (0, \infty)$, $\frac{n_\ell}{n} \to c_0 \in (0, 1)$.
- *(Means)* $\limsup_p \max_\ell \|\bar{\boldsymbol{\mu}}_\ell\| < \infty$ *and* $\limsup_p \max_i \frac{1}{\sqrt{p}} \mathbb{E}\|\boldsymbol{x}_i\| < \infty$.
- *(Covariances)* $\limsup_p \max_\ell \|\bar{\boldsymbol{\Sigma}}_\ell\| < \infty$, $\limsup_p \max_{a,b} \frac{1}{\sqrt{p}} \operatorname{tr} \bar{\boldsymbol{\Sigma}}_\ell < \infty$, $\limsup_p \max_{a,b} \frac{1}{p} \operatorname{tr} \bar{\boldsymbol{\Sigma}}_a \bar{\boldsymbol{\Sigma}}_b < \infty$.

**Assumption 7** (Kernel function). *Let $\tau \equiv \frac{2}{p} \operatorname{tr} \boldsymbol{\Sigma}$ and let $f : \mathbb{R}_+ \to \mathbb{R}_+$ be a three-times continuously differentiable function in a neighborhood of the values taken by $\tau$ and such that* $\liminf_n f(\tau) > 0$.

Without loss of generality, for each $\ell \in [k]$, we arrange the $\boldsymbol{x}_i$'s in $\boldsymbol{X}$ as

$$\boldsymbol{x}_{1+\sum_{j=1}^{\ell-1} n_j}, \ldots, \boldsymbol{x}_{\sum_{j=1}^{\ell} n_j} \sim \mathcal{L}_\ell(\boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell)$$

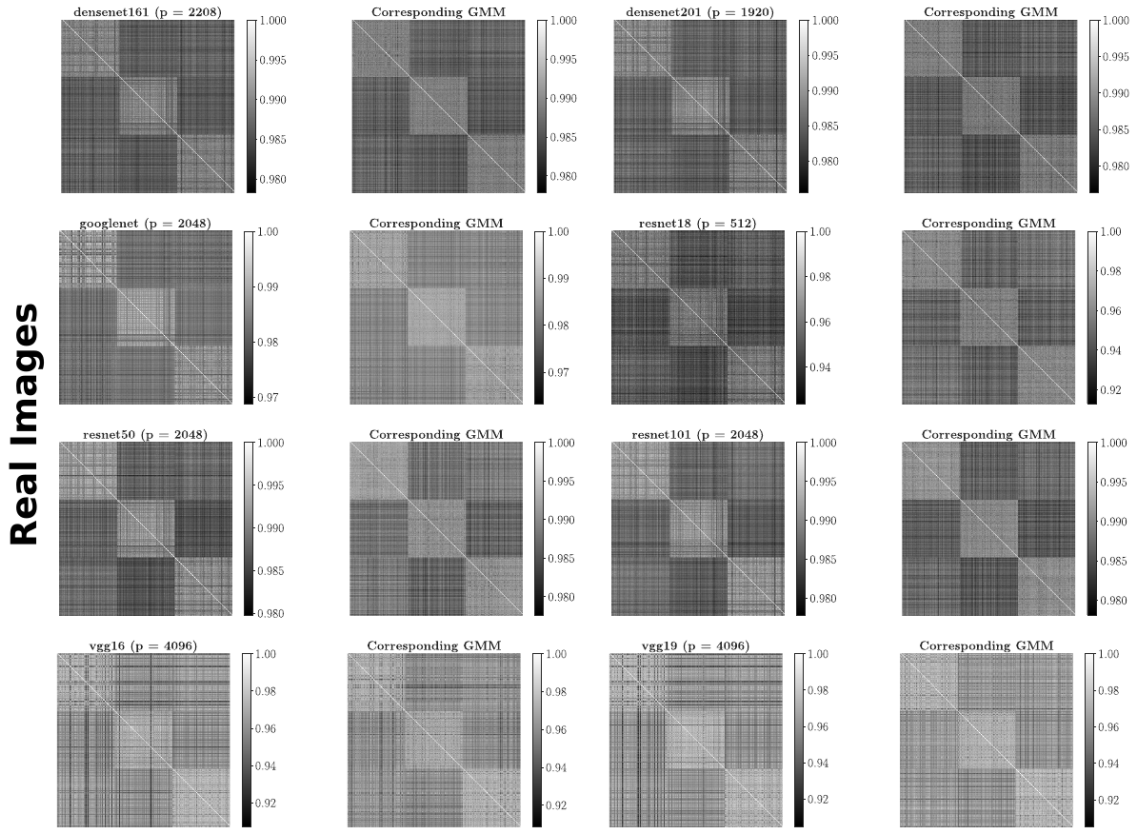



Figure 4.5: Kernel matrix of CNN representations of Real data using different representations networks. The kernel matrix is defined as $K_{ij} = f(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/p)$ where the $\boldsymbol{x}_i$'s stand for the representations (or the corresponding GMM data) and we have chosen $f(t) = exp(-t)$.

and define the kernel matrix $\boldsymbol{K}$ as the translation-invariant random matrix

$$\boldsymbol{K} \equiv \left\{ f\left( \frac{1}{p} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 \right) \right\}_{i,j=1}^{n} \tag{4.3}$$

#### 4.1.2.1 Behavior of Large Kernel Matrices

**Between and Within Class Data are "equidistant" in High-dimension.** The first key and fundamental result states that the between and within class vectors are "equidistant" in the high-dimensional regime. Namely, we have the following lemma under the $q$-exponential concentration hypothesis and Assumption 6.

**Lemma 4.1.** *Denote* $\tau \equiv \frac{2}{p} \operatorname{tr} \boldsymbol{\Sigma}$ *and let Assumption 6 hold. Then for any* $\delta > 0$*, we have with probability at least* $1 - \delta$

$$\max_{1 \le i \ne j \le n} \left\{ \left| \frac{1}{p} \|\mathbf{x}_i - \mathbf{x}_j\|^2 - \tau \right| \right\} = \mathcal{O}\left( \frac{\log(\frac{p}{\sqrt{\delta}})^{1/q}}{\sqrt{p}} \right) \tag{4.4}$$

*Proof.* For $\boldsymbol{x}_i \sim \mathcal{L}_a(\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_a)$, denote

$$\boldsymbol{z}_i \equiv \frac{\boldsymbol{x}_i - \boldsymbol{\mu}_a}{\sqrt{p}} \quad \text{and} \quad \psi_i \equiv \|\boldsymbol{z}_i\|^2 - \mathbb{E}\left[\|\boldsymbol{z}_i\|^2\right] = \|\boldsymbol{z}_i\|^2 - \frac{1}{p} \operatorname{tr} \boldsymbol{\Sigma}_a$$

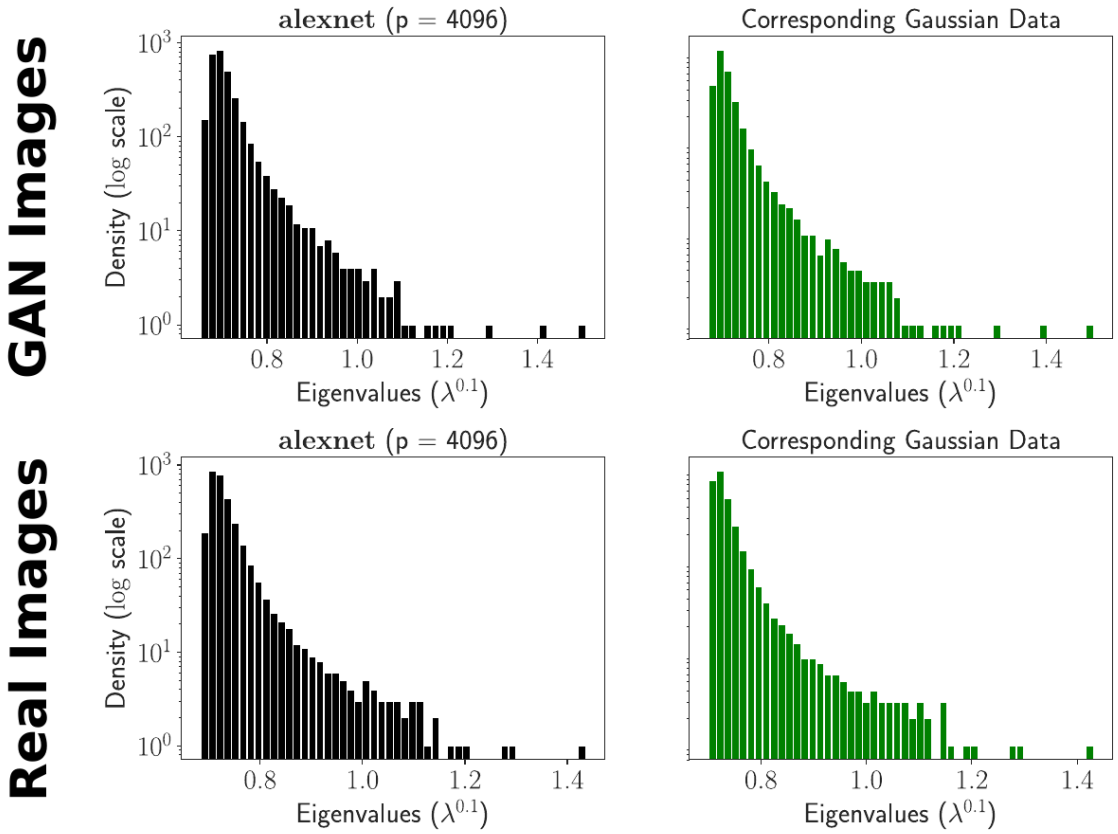



Figure 4.6: **(Left)** Spectrum of the kernel matrix of Alexnet representations for GAN images **(top)** and Real images **(bottom)** with the corresponding GMM data **(right)**. We can notice a perfect matching between the spectrum of the kernel matrix of the representations and the corresponding spectrum with GMM data. The kernel matrix is defined as $K_{ij} = f(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/p)$ where the $\boldsymbol{x}_i$'s stand for the representations (or the corresponding GMM data) and we have chosen $f(t) = exp(-t)$.

Let $a \neq b \in [k]$, $\boldsymbol{x}_i \sim \mathcal{L}_a(\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_a)$ and $x_j \sim \mathcal{L}_b(\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b)$, we decompose the normalized euclidean distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ as

$$\begin{aligned}\frac{1}{p}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 &= \|\boldsymbol{z}_i - \boldsymbol{z}_j\|^2 + \frac{1}{p}\|\boldsymbol{\mu}_a - \boldsymbol{\mu}_b\|^2 + \frac{2}{\sqrt{p}}(\boldsymbol{\mu}_a - \boldsymbol{\mu}_b)'(\boldsymbol{z}_i - \boldsymbol{z}_j) \\ &= \tau + \frac{1}{p}\operatorname{tr}\bar{\boldsymbol{\Sigma}}_a + \frac{1}{p}\operatorname{tr}\bar{\boldsymbol{\Sigma}}_b + \psi_i + \psi_j - 2\boldsymbol{z}_i'\boldsymbol{z}_j \\ &+ \frac{1}{p}\|\bar{\boldsymbol{\mu}}_a - \bar{\boldsymbol{\mu}}_b\|^2 + \frac{2}{\sqrt{p}}(\bar{\boldsymbol{\mu}}_a - \bar{\boldsymbol{\mu}}_b)'(\boldsymbol{z}_i - \boldsymbol{z}_j)\end{aligned}$$

First, we show the asymptotic concentration of each stochastic term in the previous decomposition. Recalling the $q$-exponentially concentration assumption, each $\boldsymbol{x}_i$ is $q$-exponentially concentrated (since it can be seen as a 1-Lipschitz transformation of $\boldsymbol{X}$), precisely $\boldsymbol{x}_i \propto \mathcal{E}_q$. Thus, since $\boldsymbol{x} \mapsto \frac{\boldsymbol{x}-\boldsymbol{\mu}_a}{\sqrt{p}}$ is $\frac{1}{\sqrt{p}}$-Lipschitz, we obtain

$$\boldsymbol{z}_i = \frac{\boldsymbol{x}_i - \boldsymbol{\mu}_a}{\sqrt{p}} \propto \mathcal{E}_q\left(\frac{1}{\sqrt{p}}\right)$$

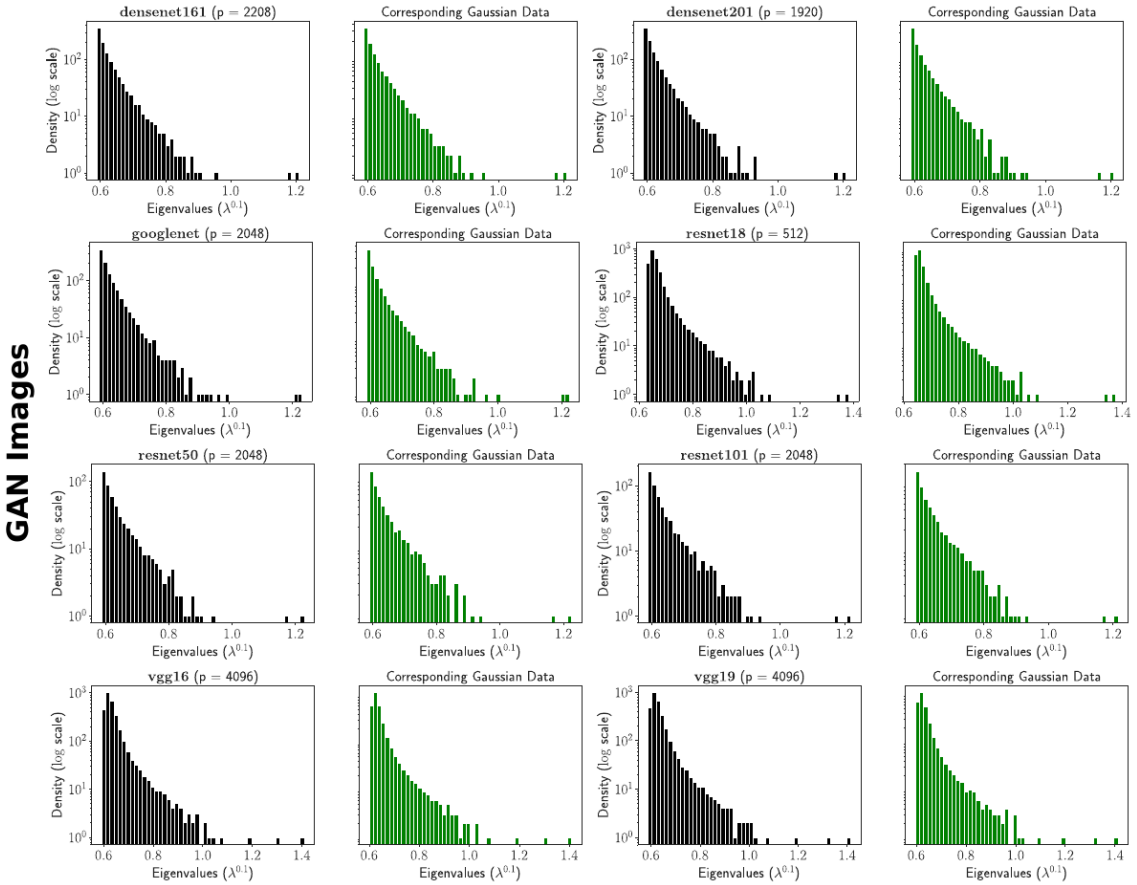



Figure 4.7: Spectrum of the kernel matrix for CNN representations of GAN data using different representations networks. The kernel matrix is defined as $K_{ij} = f(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/p)$ where the $\boldsymbol{x}_i$'s stand for the representations (or the corresponding GMM data) and we have chosen $f(t) = exp(-t)$.

We have by proposition 2.5 in [LC18b] the $q$-exponentially concentration of the concatenation $[z_i, z_j]$ since $z_i$ and $z_j$ are independent, namely

$$[z_i, z_j] \propto \mathcal{E}_q\left(\frac{1}{\sqrt{p}}\right)$$

Moreover, since $[x, y] \mapsto x \pm y$ is 2-Lipschitz, we get

$$z_i \pm z_j \propto \mathcal{E}_q\left(\frac{1}{\sqrt{p}}\right) \tag{4.5}$$

Now, since $z \to \frac{2}{\sqrt{p}}(\bar{\boldsymbol{\mu}}_a - \bar{\boldsymbol{\mu}}_b)'z$ is $(\frac{4}{\sqrt{p}} \limsup_n \max_\ell \|\bar{\boldsymbol{\mu}}_\ell\|)$-Lipschitz and recalling from Assumptions 6 that $\limsup_n \max_\ell \|\bar{\boldsymbol{\mu}}_\ell\| < \infty$, we obtain

$$\frac{2}{\sqrt{p}}(\bar{\boldsymbol{\mu}}_a - \bar{\boldsymbol{\mu}}_b)'(z_i - z_j) \propto \mathcal{E}_q\left(\frac{1}{p}\right) \tag{4.6}$$

By the polarization identity $z_i'z_j = \frac{1}{4}\left[\|z_i + z_j\|^2 - \|z_i - z_j\|^2\right]$, recalling the concentration in equation 4.5 and applying Proposition 2.5 (involving the assumption $\frac{1}{\sqrt{n}}\mathbb{E}\|x_i\| < \infty$)

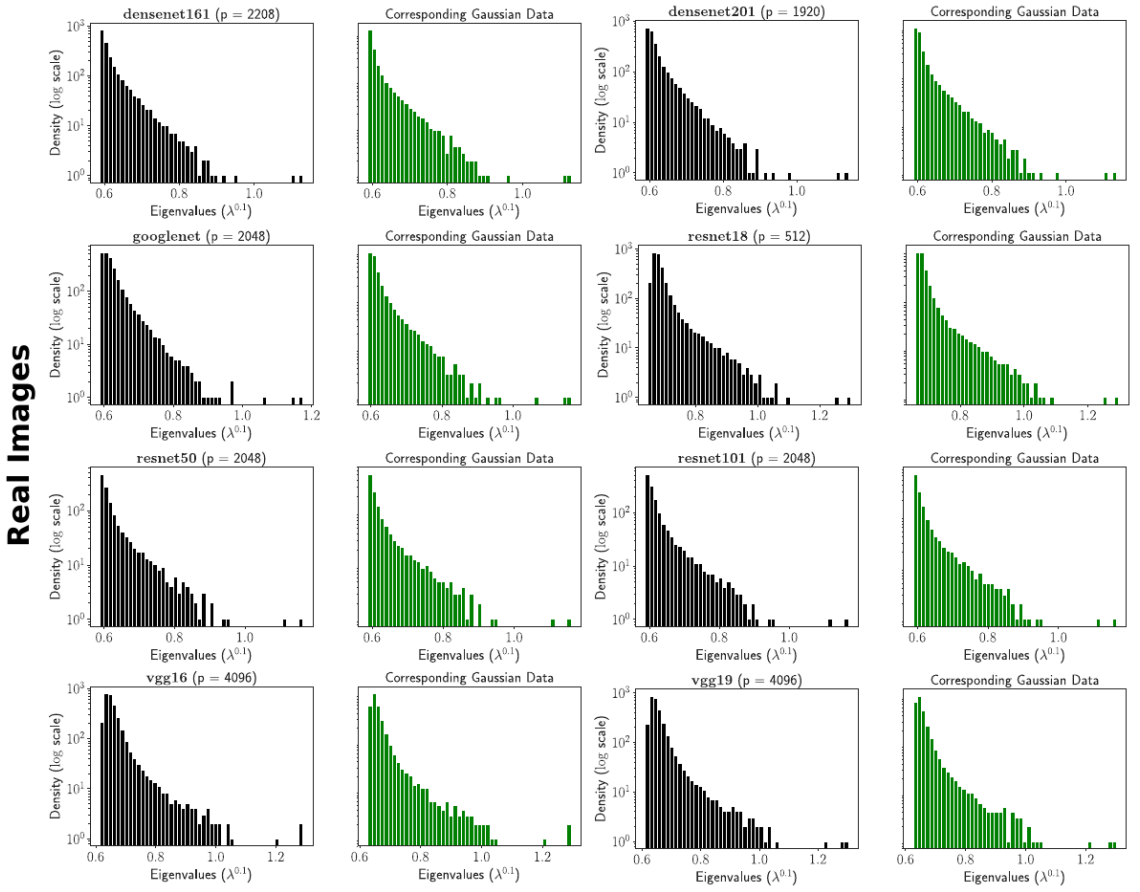



Figure 4.8: Spectrum of the kernel matrix for CNN representations of Real data using different representations networks. The kernel matrix is defined as $K_{ij} = f(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/p)$ where the $\boldsymbol{x}_i$'s stand for the representations (or the corresponding GMM data) and we have chosen $f(t) = exp(-t)$.

and Proposition 2.4, we have

$$\|\boldsymbol{z}_i \pm \boldsymbol{z}_j\|^2 \propto \mathcal{E}_q\left(\frac{1}{\sqrt{p}}\right) + \mathcal{E}_{\frac{q}{2}}\left(\frac{1}{p}\right)$$

Once again, by Proposition 2.4, we obtain

$$\boldsymbol{z}_i'\boldsymbol{z}_j \propto \mathcal{E}_q\left(\frac{1}{\sqrt{p}}\right) + \mathcal{E}_{\frac{q}{2}}\left(\frac{1}{p}\right) \tag{4.7}$$

Similarly, we also have

$$\psi_i \propto \mathcal{E}_q\left(\frac{1}{\sqrt{p}}\right) + \mathcal{E}_{\frac{q}{2}}\left(\frac{1}{p}\right)$$

Now that we have established the concentration of each stochastic term, we determine the order of their maximum values. By equation 4.6, there exists two absolute constants $C$ and $\sigma$ such that

$$\mathbb{P}\left\{|\frac{2}{\sqrt{p}}(\bar{\boldsymbol{\mu}}_a - \bar{\boldsymbol{\mu}}_b)'(\boldsymbol{z}_i - \boldsymbol{z}_j)| \geq t\right\} \leq C\, e^{-(p\,t/\sigma)^q},$$

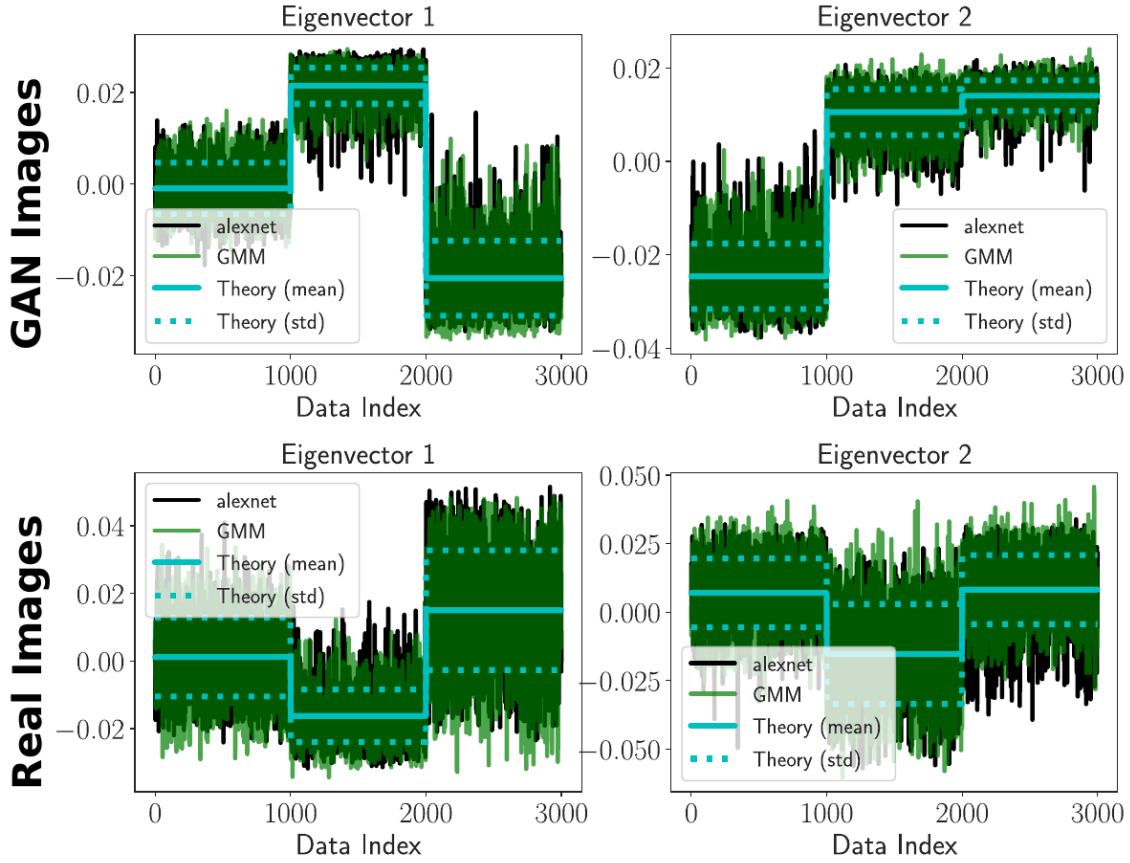



Figure 4.9: Two largest eigenvectors of the kernel matrix of Alexnet representations for GAN images **(top)** and real images **(bottom)** with the corresponding largest eigenvectors with GMM data and the theoretical predictions. The kernel matrix is defined as $K_{ij} = f(\|x_i - x_j\|^2/p)$ where the $x_i$'s stand for the representations (or the corresponding GMM data) and we have chosen $f(t) = exp(-t)$.

by the union bound, we have in particular the concentration of the maximum over $i, j \in [n]$

$$\mathbb{P}\left\{\max_{i,j} |\frac{2}{\sqrt{p}}(\bar{\mu}_a - \bar{\mu}_b)'(z_i - z_j)| \geq t\right\} \leq C\, n^2\, e^{-(p\, t/\sigma)^q},$$

which implies that, for $\delta > 0$, we have with probability at least $1 - \delta$

$$\max_{i,j} |\frac{2}{\sqrt{p}}(\bar{\mu}_a - \bar{\mu}_b)'(z_i - z_j)| = \mathcal{O}\left(\frac{\log(p/\sqrt{\delta})^{1/q}}{p}\right) \tag{4.8}$$

Recalling the concentration of the dot product $z_i' z_j$ in equation 4.7, there exists two absolute constants $C$ and $\sigma$ such that

$$\mathbb{P}\left\{|z_i' z_j| \geq t\right\} \leq C\left(e^{-(p/\sigma)^{q/2} t^{q/2}} + e^{-(p/\sigma)^{q/2} t^q}\right),$$

thus for $t \leq 1$ (only the $q$-concentration domains for small deviations) and by the union bound over $i, j \in [n]$

$$\mathbb{P}\left\{\max_{i,j} |z_i' z_j| \geq t\right\} \leq 2\, C\, n^2\, e^{-(p/\sigma)^{q/2} t^q},$$

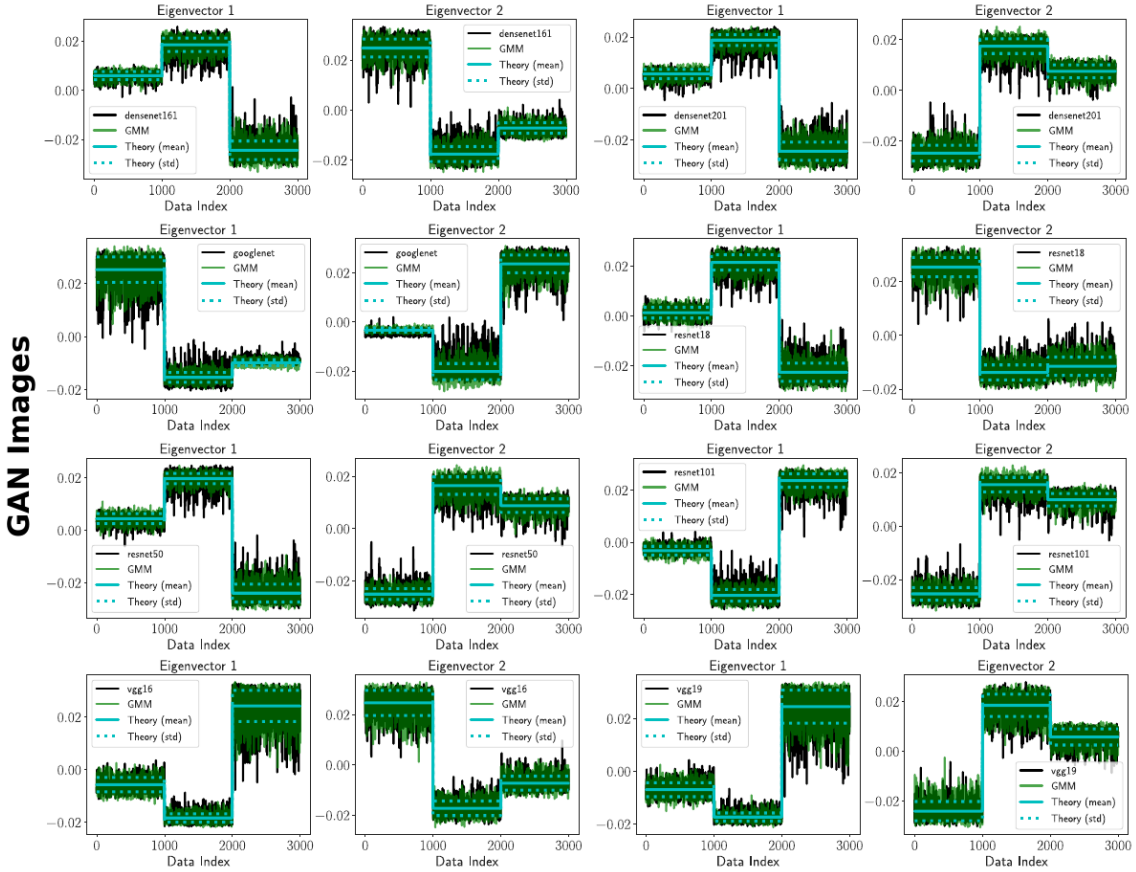



Figure 4.10: Two largest eigenvectors of the kernel matrix for CNN representations for GAN images with the corresponding largest eigenvectors with GMM data and the theoretical predictions. The kernel matrix is defined as $K_{ij} = f(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/p)$ where the $\boldsymbol{x}_i$'s stand for the representations (or the corresponding GMM data) and we have chosen $f(t) = exp(-t)$.

hence, for $\delta > 0$, the following holds with probability at least $1 - \delta$

$$\max_{i,j} |\boldsymbol{z}_i' \boldsymbol{z}_j| = \mathcal{O}\left( \frac{\log(p/\sqrt{\delta})^{1/q}}{\sqrt{p}} \right) \tag{4.9}$$

equivalently we have the concentration of the remaining stochastic term

$$\max_{i,j} |\psi_i + \psi_j| = \mathcal{O}\left( \frac{\log(p/\sqrt{\delta})^{1/q}}{\sqrt{p}} \right) \tag{4.10}$$

Recalling the considered setting, we further have

$$\frac{1}{p} \operatorname{tr} \bar{\boldsymbol{\Sigma}}_\ell = \mathcal{O}(p^{-1/2}), \quad \frac{1}{p} \|\bar{\boldsymbol{\mu}}_a - \bar{\boldsymbol{\mu}}_b\|^2 = \mathcal{O}(p^{-1}).$$

Combining the different orders yields the final result. □

**Random Matrix Equivalent for $\boldsymbol{K}$.** From the observation of Lemma 4.1, all the off-diagonal entries of the kernel matrix $\boldsymbol{K}$ tend to the same quantity $f(\tau)$ asymptotically.

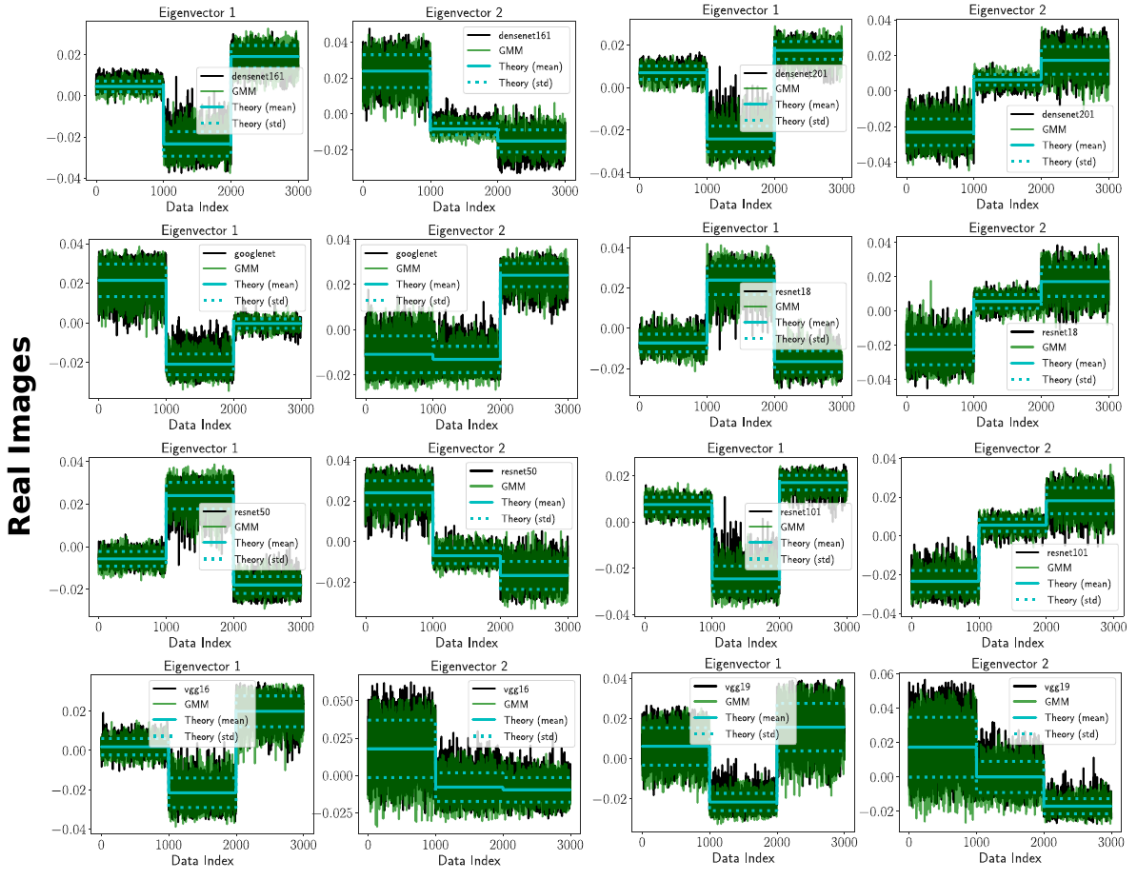



Figure 4.11: Two largest eigenvectors of the kernel matrix for CNN representations for Real images with the corresponding largest eigenvectors with GMM data and the theoretical predictions. The kernel matrix is defined as $K_{ij} = f(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/p)$ where the $\boldsymbol{x}_i$'s stand for the representations (or the corresponding GMM data) and we have chosen $f(t) = exp(-t)$.

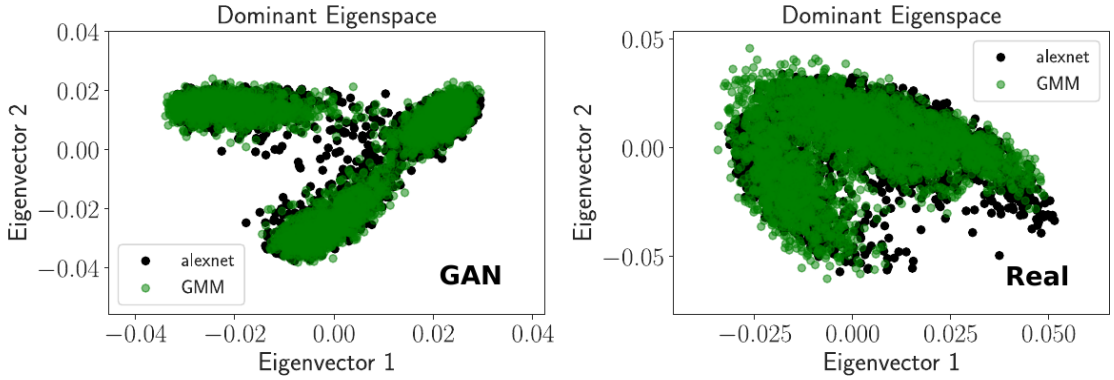



Figure 4.12: Largest eigenspace of the kernel matrix of Alexnet representations for GAN images **(left)** and real images **(right)** and the corresponding GMM data. The kernel matrix is defined as $K_{ij} = f(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/p)$ where the $\boldsymbol{x}_i$'s stand for the representations (or the corresponding GMM data) and we have chosen $f(t) = exp(-t)$.

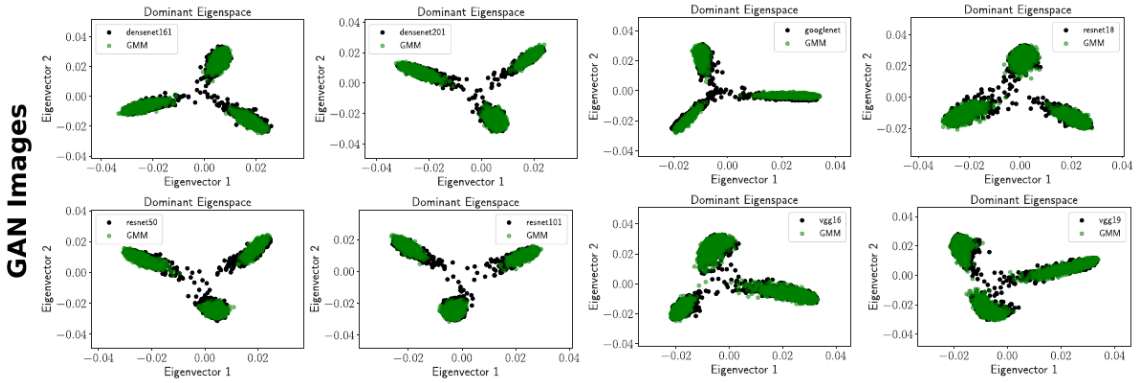



Figure 4.13: Largest eigenspace of the kernel matrix for CNN representations of GAN images and the corresponding GMM data. The kernel matrix is defined as $K_{ij} = f(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/p)$ where the $\boldsymbol{x}_i$'s stand for the representations (or the corresponding GMM data) and we have chosen $f(t) = exp(-t)$.

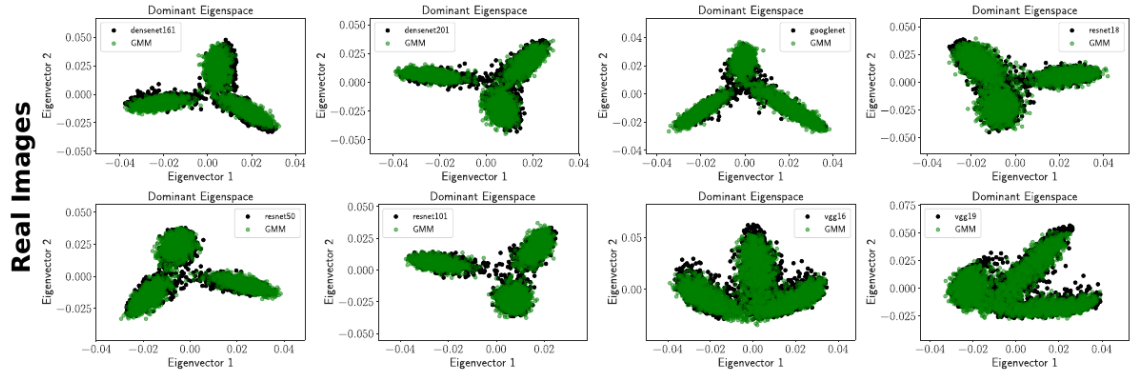



Figure 4.14: Largest eigenspace of the kernel matrix for CNN representations of real images and the corresponding GMM data. The kernel matrix is defined as $K_{ij} = f(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/p)$ where the $\boldsymbol{x}_i$'s stand for the representations (or the corresponding GMM data) and we have chosen $f(t) = exp(-t)$.

Therefore, $\boldsymbol{K}$ can be Taylor expanded entry-wise and we show in the following that it asymptotically has (up to centering) the same behavior as a spiked random matrix.

Before introducing this asymptotic equivalent and for subsequent use, we introduce the following quantities

$$\boldsymbol{M} = [\bar{\boldsymbol{\mu}}_1, \ldots, \bar{\boldsymbol{\mu}}_k] \in \mathcal{M}_{p,k}, \quad \boldsymbol{t} = \left\{ \frac{\operatorname{tr} \bar{\boldsymbol{\Sigma}}_\ell}{\sqrt{p}} \right\}_{\ell=1}^{k} \in \mathbb{R}^k, \quad \boldsymbol{J} = [\boldsymbol{j}_1, \ldots, \boldsymbol{j}_k] \in \mathcal{M}_{n,k}$$

$$\boldsymbol{T} = \left\{ \frac{\operatorname{tr} \bar{\boldsymbol{\Sigma}}_a \bar{\boldsymbol{\Sigma}}_b}{p} \right\}_{a,b=1}^{k} \in \mathcal{M}_k, \quad \boldsymbol{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n] \in \mathcal{M}_{p,n}, \quad \boldsymbol{P} = \boldsymbol{I}_n - \frac{1}{n} \boldsymbol{1}_n \boldsymbol{1}_n^\intercal$$

where $\boldsymbol{j}_\ell \in \mathbb{R}^n$ stands for the canonical vector of the class represented by $\mathcal{L}_\ell$, defined by $(\boldsymbol{j}_\ell)_i = \delta_{\boldsymbol{x}_i \sim \mathcal{L}_\ell}$. The vectors $\boldsymbol{z}_i$ are defined as $\boldsymbol{z}_i \equiv (\boldsymbol{x}_i - \bar{\boldsymbol{\mu}}_\ell)/\sqrt{p}$ for each $\ell \in [k]$. We will further denote the matrix $\boldsymbol{Z}_\ell$, the columns of which are the $\boldsymbol{z}_i'$s in class corresponding to $\mathcal{L}_\ell$.

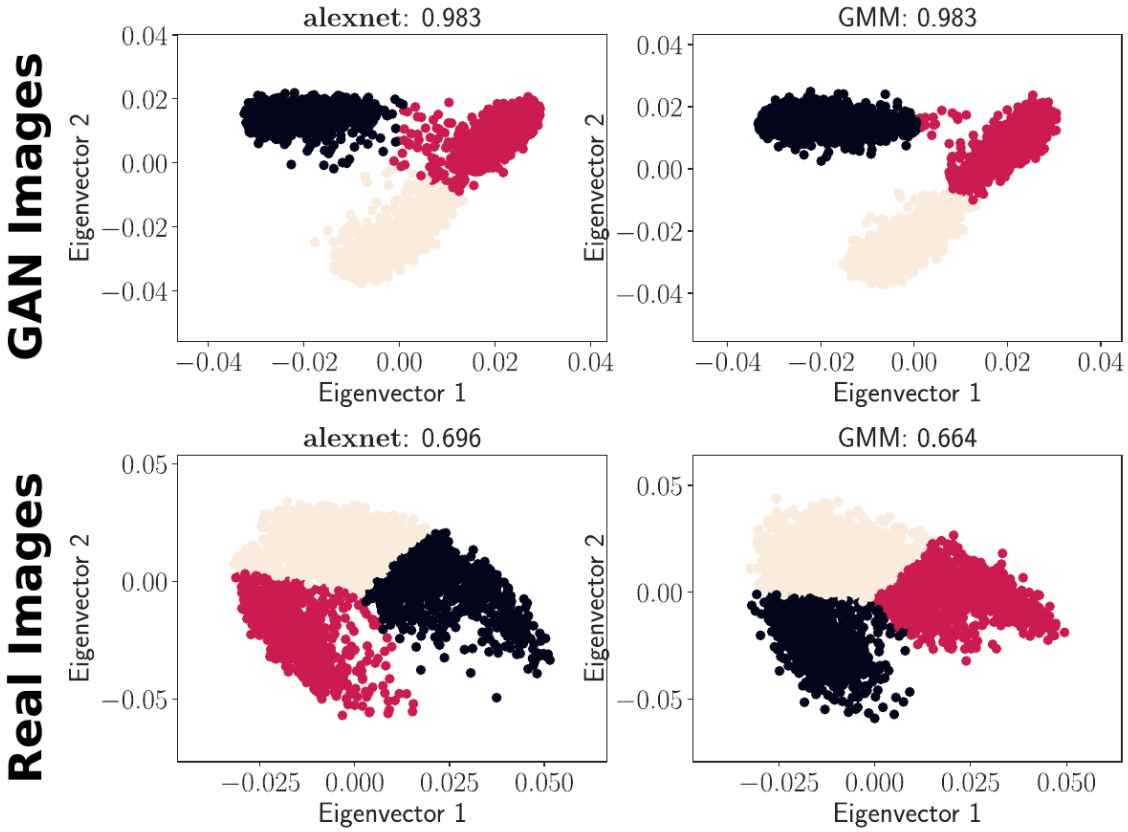



Figure 4.15: Kernel spectral clustering on Alexnet representations using GAN images **(top)** and real images **(bottom)**. The kernel matrix is defined as $K_{ij} = f(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/p)$ where the $\boldsymbol{x}_i$'s stand for the representations (or the corresponding GMM data) and we have chosen $f(t) = exp(-t)$.

The coming result states that there exists a random matrix $\tilde{\boldsymbol{K}}$ such that $\boldsymbol{PKP} \leftrightsquigarrow \hat{\boldsymbol{K}}$, i.e, $\boldsymbol{PKP}$ admits as a random equivalent $\tilde{\boldsymbol{K}}$ asymptotically, where $\tilde{\boldsymbol{K}}$ has a tractable behavior from the random matrix theory standpoint.

**Theorem 4.1** (Asymptotic Random Matrix Equivalent)**.** *Let Assumptions 5, 6 and 7 hold and let $\tilde{\boldsymbol{K}}$ be defined as*

$$\tilde{\boldsymbol{K}} = -2f'(\tau)\left[\boldsymbol{PZ}^\mathsf{T}\boldsymbol{ZP} + \boldsymbol{UAU}^\mathsf{T}\right] + F(\tau)\boldsymbol{P}, \quad F(\tau) = (f(0) - f(\tau) + \tau f'(\tau))$$

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{I}_k & -\frac{f''(\tau)}{2f'(\tau)}\boldsymbol{t} \\ \boldsymbol{I}_k & \boldsymbol{0}_{k\times k} & \boldsymbol{0}_{k\times 1} \\ -\frac{f''(\tau)}{2f'(\tau)}\boldsymbol{t}^\mathsf{T} & \boldsymbol{0}_{1\times k} & -\frac{f''(\tau)}{2f'(\tau)} \end{bmatrix}, \quad \psi_i = \|\boldsymbol{z}_i\|^2 - \mathbb{E}\|\boldsymbol{z}_i\|^2 = \|\boldsymbol{z}_i\|^2 - \frac{1}{p}\operatorname{tr}\boldsymbol{\Sigma}_\ell$$

$$\boldsymbol{A}_{11} = \boldsymbol{M}^\mathsf{T}\boldsymbol{M} - \Xi - \frac{f''(\tau)}{2f'(\tau)}[\boldsymbol{tt}^\mathsf{T} + 2\boldsymbol{T}], \quad \boldsymbol{\Phi} = \boldsymbol{Z}^\mathsf{T}\boldsymbol{M} - \left\{\boldsymbol{Z}_\ell^\mathsf{T}\bar{\boldsymbol{\mu}}_\ell\boldsymbol{1}_k^\mathsf{T}\right\}_{\ell=1}^k$$

$$\boldsymbol{U} = \left[\frac{\boldsymbol{J} - \boldsymbol{1}_n\boldsymbol{c}^\mathsf{T}}{\sqrt{p}}, \boldsymbol{P\Phi}, \boldsymbol{P}\psi\right], \quad \Xi = \left\{\frac{\|\bar{\boldsymbol{\mu}}_a\|^2 + \|\bar{\boldsymbol{\mu}}_b\|^2}{2}\right\}_{a,b=1}^k$$

*Then*

$$\boldsymbol{PKP} \leftrightsquigarrow \tilde{\boldsymbol{K}}$$

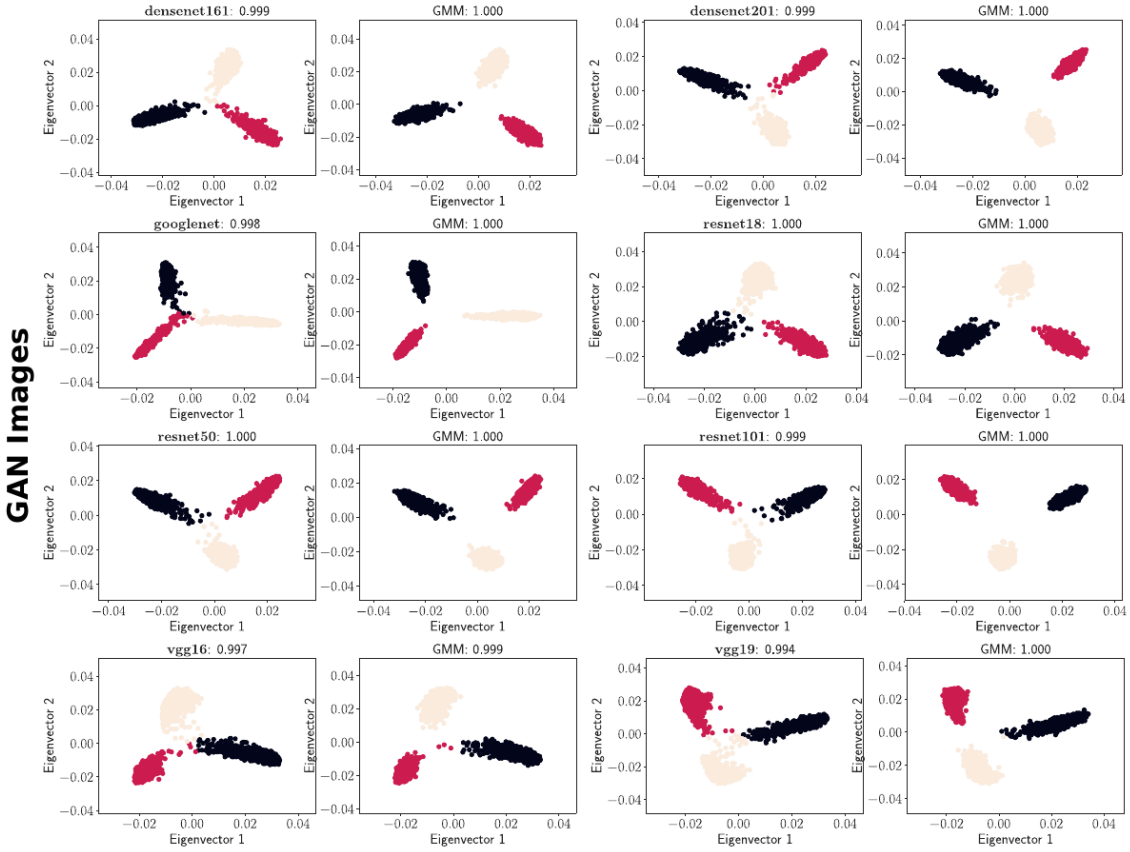



Figure 4.16: Kernel spectral clustering on CNN representations using GAN images. The kernel matrix is defined as $K_{ij} = f(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/p)$ where the $\boldsymbol{x}_i$'s stand for the representations (or the corresponding GMM data) and we have chosen $f(t) = exp(-t)$.

*Specifically, for $\delta > 0$, there exists $C_\delta > 0$ such that for all $\gamma > 0$*

$$\|\boldsymbol{PKP} - \hat{\boldsymbol{K}}\| \leq C_\delta \, p^{-1/2+\gamma} \log(p)^\gamma \text{ with probability at least } 1 - \delta.$$

*Proof.* See Subsection C.2.1 in the Appendix. □

Theorem 4.1 shows that, up to centering by $\boldsymbol{P}$, the kernel matrix $\boldsymbol{K}$ has asymptotically the same behavior as $\tilde{\boldsymbol{K}}$. In particular, the obtained approximation in operator norm implies that $\boldsymbol{PKP}$ and $\tilde{\boldsymbol{K}}$ share the same eigenvalues (by Weyl's inequality [EI98, Thm 4.1]) and same *isolated* eigenvectors asymptotically. Therefore, the asymptotic spectral properties of $\boldsymbol{K}$ (*i.e.*, the classification performance of algorithms involving $\boldsymbol{K}$) may be studied through its equivalent $\tilde{\boldsymbol{K}}$.

Indeed, note that $\tilde{\boldsymbol{K}}$ is made of a sum of a random matrix $\boldsymbol{PZ}^\intercal\boldsymbol{ZP}$ and a maximum $(k-1)$-rank matrix containing linear combinations of the class-wise canonical vectors $\boldsymbol{j}_\ell$ weighted by the inner-products between class means $\boldsymbol{M}^\intercal\boldsymbol{M}$ and class covariance-products and traces (through $\boldsymbol{t}$ and $\boldsymbol{T}$). The matrix $\tilde{\boldsymbol{K}}$ can then be identified as a *spiked random matrix model* [BGN12]. Note however that, unlike the standard spiked random matrices, the low-rank part of $\tilde{\boldsymbol{K}}$ depends statistically on the noise part and the latter is a mixture between random matrices made of concentrated vectors. In particular, the spectrum of $\tilde{\boldsymbol{K}}$ is composed of a *bulk* along with up to $k-1$ isolated eigenvalues, and the associated

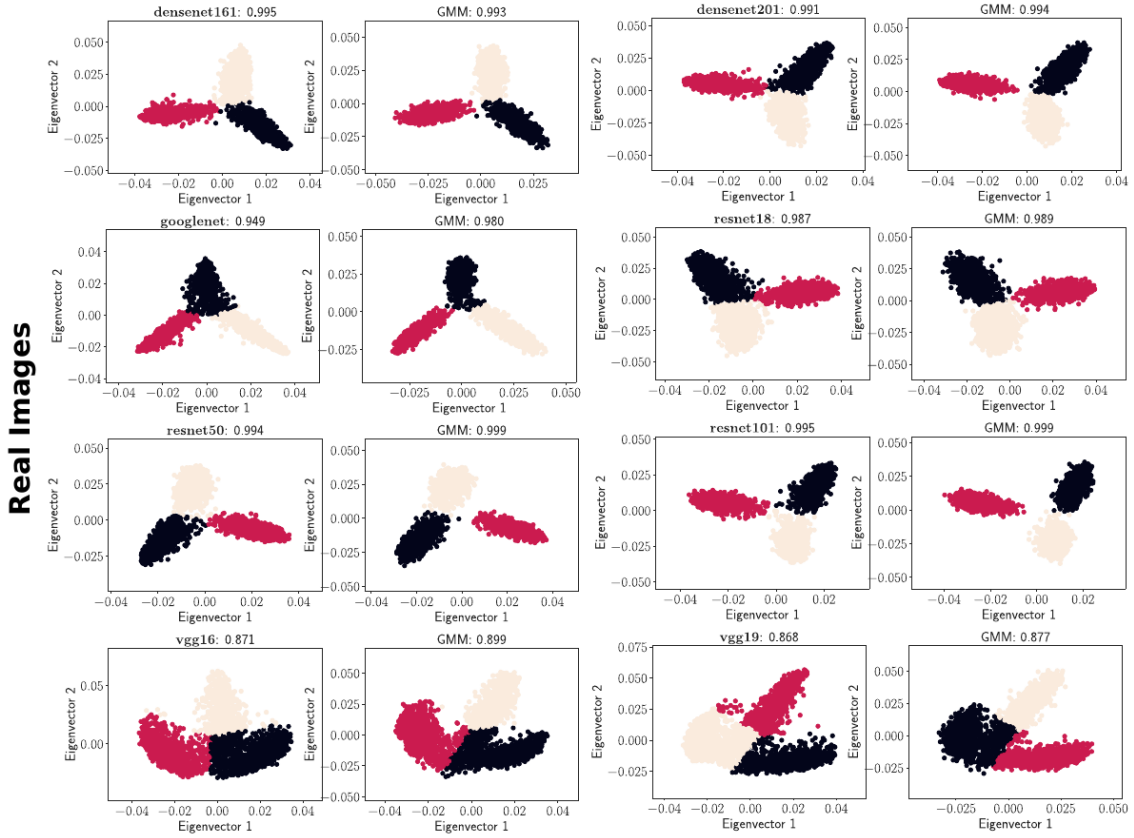



Figure 4.17: Kernel spectral clustering on CNN representations using real images. The kernel matrix is defined as $K_{ij} = f(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/p)$ where the $\boldsymbol{x}_i$'s stand for the representations (or the corresponding GMM data) and we have chosen $f(t) = exp(-t)$.

eigenvectors are aligned with the eigenvectors in $\boldsymbol{U}$, therefore with linear combinations of the class canonical vectors $\boldsymbol{j}_1, \ldots, \boldsymbol{j}_k$. Consequently, characterizing the asymptotic performance of spectral clustering relies on the characterization of the isolated eigenvectors of $\tilde{\boldsymbol{K}}$. In fact, these eigenvectors are *informative* if their associated eigenvalues are far away from the main eigenvalue *bulk*. The latter is due to the presence of the random Gram matrix $\boldsymbol{P}\boldsymbol{Z}^\intercal\boldsymbol{Z}\boldsymbol{P}$ for which the spectrum is described by Theorem 3.1 for concentrated vectors.

As for the Gram matrix, we see from Theorem 3.1 and Theorem 4.1 that the spectral behavior of the kernel matrix $\boldsymbol{K}$ depends only on the first and second order statistics of the laws $\mathcal{L}_\ell$, namely their means $\boldsymbol{\mu}_\ell$ and covariances $\boldsymbol{\Sigma}_\ell$. Typically, $\boldsymbol{K}$ have the same behavior when data is described by a GMM model with the same means and covariances. The asymptotic spectral behavior of $\boldsymbol{K}$ is therefore *universal* w.r.t. the data distribution laws which fall within the introduced class of concentrated vectors. This notably explains the observations in [CBG+16] in which the obtained theoretical model, under GMM assumptions, fit with empirical results using the MNIST dataset [LeC98].

In the following, we provide the conditions under which the *informative* eigenvalues become visible in the spectrum of $\tilde{\boldsymbol{K}}$ which provide access to the performances of kernel spectral clustering. Having Theorem 3.1 which describes the noise term $\boldsymbol{P}\boldsymbol{Z}^\intercal\boldsymbol{Z}\boldsymbol{P}$, we can determine the conditions under which the spikes can be visible outside the main *bulk* of $\boldsymbol{P}\boldsymbol{Z}^\intercal\boldsymbol{Z}\boldsymbol{P}$, and the result concerning the isolated eigenvectors. We however need to

introduce the following resolvent from Theorem 2.9

$$\boldsymbol{Q}_\delta \equiv \left( \sum_{\ell=1}^{k} \frac{c_\ell}{k} \frac{\boldsymbol{\Sigma}_\ell}{1+\delta_\ell(z)} + z\boldsymbol{I}_p \right)^{-1} \tag{4.11}$$

where $\delta_\ell(z)$ is the unique solution of the fixed point equation $\delta_\ell(z) = \frac{1}{n}\operatorname{tr}(\boldsymbol{\Sigma}_\ell \boldsymbol{Q}_\delta)$. We further need the following technical assumption on the class-wise covariances to ensure that $\mathbf{PZ}^\intercal\mathbf{ZP}$ does not produce *non-informative* isolated eigenvalues.

**Assumption 8** (Spikes control). *Denote $\lambda_1^\ell, \ldots, \lambda_p^\ell$ the eigenvalues of $\boldsymbol{\Sigma}_\ell$, for each $\ell \in [k]$. As $n \to \infty$, $\frac{1}{p}\sum_{i=1}^{p} \delta_{\lambda_i^\ell} \xrightarrow{\mathcal{D}} \rho_\ell$ with support $\mathcal{S}_\ell$, and $\max_{1\le i\le p} \operatorname{dist}(\lambda_i^\ell, \mathcal{S}_\ell) \to 0$.*

Now we can state the theorem that ensures the presence of *informative* eigenvalues in the spectrum of $\tilde{\boldsymbol{K}}$, and gives the characterization of the corresponding isolated eigenvectors, which results from standard random matrix techniques [BGN12].

**Theorem 4.2** (Isolated eigenvalues). *Suppose that Assumptions 5-6-8 hold and let $\boldsymbol{\Lambda}_z$ be the matrix defined for $z \in \mathbb{C}_+$ as*

$$\boldsymbol{\Lambda}_z = \alpha_\tau(z)\boldsymbol{I}_k - \left[ \alpha_\tau(z) \left( z\boldsymbol{M}^\intercal \boldsymbol{Q}_\delta \boldsymbol{M} + \Xi + \frac{f''(\tau)}{f'(\tau)}\boldsymbol{T} \right) + \frac{f''(\tau)}{2f'(\tau)}\boldsymbol{t}\boldsymbol{t}^\intercal \right] \Gamma_z$$

*where*

$$\begin{aligned}
\alpha_\tau(z) &\equiv 1 + \frac{f''(\tau)}{2zc_0 p f'(\tau)} \sum_{\ell=1}^{k} \eta_\ell(z) \operatorname{tr}\boldsymbol{\Sigma}_\ell^2 \\
\Gamma_z &\equiv \frac{1}{z} \left\{ \frac{c_a\eta_a(z) c_b \eta_b(z)}{\sum_{\ell=1}^{k} c_\ell \eta_\ell(z)} \right\}_{a,b=1}^{k} - \frac{1}{z}\mathcal{D}\{c_\ell\eta_\ell(z)\}_{\ell=1}^{k} + \frac{2}{z}(\boldsymbol{I}_k - \mathcal{D}(\eta))^\intercal \\
\eta_\ell(z) &\equiv \frac{1}{1+\delta_\ell(z)}
\end{aligned}$$

*Let $\lambda^*$ be at an infinitesimal distance from the bulk support $\mathcal{S}$ of $\mathbf{PZ}^\intercal\mathbf{ZP}$, such that $\alpha_\tau(\lambda^*) \neq 0$ and $\boldsymbol{\Lambda}_{\lambda^*}$ has a zero eigenvalue of multiplicity $m^*$. Then $\boldsymbol{PKP}$ produces $m^*$ spikes asymptotically close to*

$$\rho^* \equiv -2f'(\tau)\lambda^* + F(\tau)$$

*Furthermore, the eigenspace projector corresponding to the (asymptotically converging to $\rho^*$) isolated eigenvalues of $\boldsymbol{PKP}$ has a non-vanishing projection onto* $\operatorname{span}(\boldsymbol{j}_1, \ldots, \boldsymbol{j}_k)$.

Theorem 4.2 gives the conditions under which the spikes can be observed in the spectrum of $\tilde{\boldsymbol{K}}$, and states that the corresponding eigenvectors are aligned to some extent to the class canonical vectors $\boldsymbol{j}_1, \ldots, \boldsymbol{j}_k$, which is important for spectral clustering. We refer the reader to [CBG+16] for the performances characterization of kernel spectral clustering in the $k$-class Gaussian mixture model case as the behavior of $\boldsymbol{K}$ is universal.

#### 4.1.2.2 Application to GAN-generated Images

The objective of this subsection it to present simulations to validate our findings concerning the behavior of large kernel matrices. To highlight this aspect, we evaluate the kernel matrix on $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ being CNN representations of GAN-generated images and

we further use real images for comparison. Specifically, we consider the same setting as the previous chapter, i.e., GAN images are generated by the BigGAN model [BDS18] and then represented using several representation models pretrained on the Imagenet dataset [DDS+09]. Moreover, real images are samples from the Imagenet dataset [DDS+09].

**GAN architecture & real images:** We consider as a setting of our experiments the BigGAN model [BDS18] which takes as input a Gaussian noise vector of dimension $d = 140$ and a one-hot-vector to generate a specific class image of dimension $256 \times 256 \times 3$. We particularly consider $k = 3$ classes which are {*pizza*, *mushroom*, *hamburger*}. Examples of the generated images are shown in Figure 3.4. Real images are sampled from the same classes from the Imagenet dataset [DDS+09]. In both cases, we use $n = 3000$ images (i.e., $n_\ell = 1000$ for each class $\mathcal{C}_\ell$) in all our experiments.

**CNN representation:** We consider popular CNN representations pretrained on the Imagenet dataset [DDS+09]. Specifically, we consider 9 representation networks which are: *vgg16* ($p = 4096$), *vgg19* ($p = 4096$), *resnet18* ($p = 512$), *resnet50* ($p = 2048$), *resnet101* ($p = 2048$), *densenet161* ($p = 2208$), *densenet201* ($p = 1920$), *alexnet* ($p = 4096$) and *googlenet* ($p = 2048$).

**GMM data:** For both GAN and real data (i.e., CNN representations of GAN and real images respectively) we build the corresponding GMM data of class $\mathcal{C}_\ell$ as $\boldsymbol{x}_i' = \hat{\boldsymbol{\mu}}_\ell + \hat{\boldsymbol{\Sigma}}_\ell^{1/2} \boldsymbol{z}_i$ with $z_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_p)$ where $\hat{\boldsymbol{\mu}}_\ell$ and $\hat{\boldsymbol{\Sigma}}_\ell$ are respectively the empirical estimates of $\boldsymbol{\mu}_\ell$ and $\boldsymbol{\Sigma}_\ell$.

The key point in the analysis of the kernel matrix $\boldsymbol{K}$ is the behavior of the quantity $\frac{1}{p}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2$. In Figure 4.1, we have depicted the histogram of this quantity for both GAN and real images using the *alexnet* ($p = 4096$) representation. As we can notice, in both cases the histogram of $\frac{1}{p}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2$ matches its Gaussian counterpart $\frac{1}{p}\|\boldsymbol{x}_i' - \boldsymbol{x}_j'\|^2$. Moreover, for real data, this quantity seems to converge to the same value as per Lemma 4.1. We do not observe the same behavior for GAN data which may be the consequence of the fact that GANs have low entropy, i.e., GAN data are easily separable compared to real data. Figure 4.2 depicts the same histogram using other representations from which we obtain the same conclusions.

As a consequence, the entries of the kernel matrix $\boldsymbol{K}$ converge to the same value, as we can visually observe from Figure 4.3 (at least for real images). Still we observe the same behavior of the kernel matrix for GAN data/real data versus their GMM counterparts. Figure 4.4 shows more examples using the different CNN representation for GAN images, and Figure 4.5 depicts other representations of real images (where we observe that the entries $\boldsymbol{K}$ converge to the same quantity as $p$ grows).

More interestingly, we see from Figure 4.6 that the spectrum of $\boldsymbol{K}$ when computed on the representations of both GAN and real images, matches perfectly the spectrum of its GMM counterpart. A result which stays valid across all the representation networks for GAN images in Figure 4.7 as well as real images 4.8. As proved by Theorem 4.1, the kernel matrix $\boldsymbol{K}$ behaves as spiked random matrix model which can be also observed from these figures. In particular, the matching between the representations themselves and the corresponding GAN data demonstrates the *universal behavior* of the kernel matrix $\boldsymbol{K}$. This universal behavior goes beyond the spectrum of $\boldsymbol{K}$ and is also observed through

the largest eigenvectors of $\boldsymbol{K}$ which contain information about the classes as per Theorem 4.2. In particular, we can see from Figure 4.9 that the largest eigenvectors computed on the *alexnet* representations are perfectly aligned with their GMM counterparts, and also for the other representations using GAN images (see Figure 4.10) and real images (see Figure 4.11). The corresponding eigenspaces are also depicted in Figure 4.12 for the *alexnet* representation network and the other representation networks in Figure 4.13 and Figure 4.14.

Since kernel spectral clustering consists in applying the k-means algorithm on the largest subspace of the kernel matrix, we depicted in Figure 4.15 the result of k-means for *alexnet* representations and the corresponding GMM data along with the obtained accuracies. As we can see, the behavior of k-means on the representations is almost the same as on the corresponding GMM data which confirms the *universal aspect* of $\boldsymbol{K}$. This observation holds for the remaining considered representation networks for both GAN images in Figure 4.16 and real images 4.17. Note that the accuracies obtained with GAN data are almost equal to 100% which is a consequence of the fact that GAN data are easily separable, while this is not true for real data as depicted in Figure 4.17.

### 4.1.3 Central Contribution and perspectives

We have studied in this work large kernel matrices for a wide class of random inputs, *i.e.*, concentrated data, which are more generic than Gaussian mixtures as we have shown in the first part of this manuscript. Our study has notably shown the universality aspect of kernel spectral clustering (based on the kernel matrix $\boldsymbol{K}$) by highlighting the fact that the asymptotic behavior of kernel matrices, for concentrated inputs, depend on the first and second moments statistics, thereby match the behavior of a GMM model as empirically observed in [CBG+16]. Moreover, since the performance of spectral clustering is predictable for a GMM model [CBG+16], it is thereby predictable for concentrated data as well and thus for complex data as deep learning representations of the surprising realistic images generated by GANs. In the next section, we will provide the analysis of a different form of kernel matrices which fundamentally differ from this section by the fact that their entries concentrate around different values. Thereby, making the conclusions considerably different, we further apply our result to the problem of Sparse PCA which yield to an effective PCs recovery method.

## 4.2 Sparse Principal Component Analysis

This section is based on the following work:

> **(C3)** **MEA. Seddik**, M. Tamaazousti, R. Couillet, "*A Kernel Random Matrix-Based Approach for Sparse PCA*", International Conference on Learning Representations (ICLR'19), New Orleans, United-States, 2019.

This section presents the analysis of kernel matrices of the form $f(\hat{\boldsymbol{\Sigma}})$ where $\hat{\boldsymbol{\Sigma}}$ is the sample covariance matrix. The behavior of these matrices if fundamentally different from those of the previous section, indeed, the entries of $f(\hat{\boldsymbol{\Sigma}})$ concentrate around different values which makes the analysis a bit different. In particular, we apply our analysis to the Sparse PCA problem which results in competitive method to the existing methods in the state-of-the-art.

### 4.2.1 Motivation

Principal component analysis (PCA) is extensively used in data analysis and machine learning applications. It is a dimension reduction technique that aims to project a given dataset onto principal subspaces spanned by the leading eigenvectors of the sample covariance matrix [WEG87], which represent the principal modes of variance. Basically, the statistical interpretation of PCA lies in the fact that most of the variance in the data is captured by these modes. Consequently, PCA reduces the dimension of the feature space while keeping most of the information in the data. It is well-known [And63] that PCA performs efficiently in the traditional data setting where the number of features is small and the number of samples is large.

Consider a data matrix $\boldsymbol{X} \in \mathcal{M}_{p,n}$ consisting of $n$ centered samples, each sample having $p$ features. The standard PCA method requires the computation of the sample covariance matrix $\hat{\boldsymbol{\Sigma}} = \boldsymbol{X}\boldsymbol{X}^\intercal/n$ and estimates the first principal components $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots$ (*i.e.*, the successive dominant eigenvectors of $\boldsymbol{\Sigma} = \mathbb{E}[\boldsymbol{X}\boldsymbol{X}^\intercal/n]$) by the ordered eigenvectors $\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2, \ldots$ of $\hat{\boldsymbol{\Sigma}}$. Authors in [JL09] demonstrated that, in the high dimensional regime where $n, p \to \infty$ with $p/n \to c > 0$, the principal component $\hat{u}_1$ estimated by standard PCA is inconsistent. Essentially, if $p/n \nrightarrow 0$ then $\|\hat{u}_1 - u_1\|_2 \nrightarrow 0$ in the high-dimensional asymptotic regime (see Subsection 2.3.1 for more details). As discussed in the background Chapter 2 of this manuscript, this phenomenon is well investigated within the field of random matrix theory for covariance models of the form $\hat{\boldsymbol{\Sigma}} = \frac{1}{n}\boldsymbol{\Sigma}^{1/2}\boldsymbol{Z}\boldsymbol{Z}^\intercal\boldsymbol{\Sigma}^{1/2}$, where $\boldsymbol{\Sigma}$ is a positive semi-definite matrix and $\boldsymbol{Z}$ is a $p \times n$ matrix with random *i.i.d.* entries. Essentially relying on the so-called spiked models (see Subsection 2.3) of random matrix theory, where $\boldsymbol{\Sigma}$ is a low-rank perturbation of the identity matrix, namely $\boldsymbol{\Sigma} = \boldsymbol{I}_p + \sum_{i=1}^{k} \omega_i \boldsymbol{u}_i \boldsymbol{u}_i^\intercal$ with $k$ fixed with respect to $p, n$. Authors in [BAP+05] and [Pau07] notably exhibited a phase transition phenomenon: as $p/n \to c$, if $\omega_i < \sqrt{c}$ the estimated principal component $\hat{\boldsymbol{u}}_i$ using standard PCA is (almost surely) asymptotically orthogonal to the true principal component $\boldsymbol{u}_i$ (*i.e.*, $\hat{\boldsymbol{u}}_i^\intercal \boldsymbol{u}_i \to 0$); on the other hand, if $\omega_i > \sqrt{c}$, $\liminf_n |\hat{\boldsymbol{u}}_i^\intercal \boldsymbol{u}_i| > 0$. This phase transition phenomenon has attracted recently much attention within the random matrix community [BGN11, CDMF09, FP07, KY13].

The inconsistency of standard PCA in high dimensions motivated the idea to look for more structural information on the principal components. In particular, considering that the principal components are sparse in an appropriate basis (*e.g.*, in the wavelet domain), a large body of works have emerged and proposed improved PCA approaches that account for sparsity. One of the most consistent sparse PCA methods in the literature is the covariance thresholding (CT) algorithm [KNV15]. Based on the intuition that the small entries of the empirical covariance matrix $\hat{\boldsymbol{\Sigma}}$ induce noise in its principal components, this method consists in applying the popular *soft-thresholding* function (with threshold $\tau > 0$);

$$\mathrm{soft}(\cdot\,;\tau) : t \mapsto \mathrm{sign}(t) \cdot (|t| - \tau)_+, \tag{4.12}$$

entry-wise to the empirical covariance matrix $\hat{\boldsymbol{\Sigma}}$ and performing PCA on the resulting matrix. Authors in [DM14, DM16] have theoretically demonstrated that the covariance thresholding algorithm recovers the sought-for principal components with high probability under controlled growth rates between $p$, $n$ and the sparsity level. In this work, we particularly show that the soft-thresholding method in fact falls within a broader class of kernel-based[1] PCA algorithms that are particularly suited to sparse PCA recov-

[1]Note that *kernel-based* terminology is used to highlight that our work falls within the framework of

ery. This method consists in considering the matrix $f(\hat{\mathbf{\Sigma}})$ instead of $\hat{\mathbf{\Sigma}}$ where $f$ is a function applied entry-wise to $\mathbf{\Sigma}$. By imposing some constraints on $f$, most importantly that $f'(0) = f''(0) = 0$, we show that sparse PCA can be performed with provably high accuracy for sufficiently large $n, p$.

There has been a wide range of approaches to tackle the sparse PCA problem. Mainly, three classes of approaches emerge in the literature. Most popular techniques are optimization based algorithms [dGJL05, MWA06, ZS07, ZHT06, WGR+09], where the idea is to see the problem of sparse PCA through an optimization perspective, and to propose methods to solve the latter by either considering a different formulation – *e.g.* semi-definite programming (SDP) or convex relaxations – or adding penalties to the original optimization problem such as a LASSO regularization. The second class of approaches covers matrix decomposition-based techniques [APD14, PDK13, SH08], where sparse principal components are extracted through solving a low rank matrix approximation problem based on Singular Value Decomposition. Finally, most consistent sparse PCA methods adopt thresholding-based approaches: initial heuristics used factor rotation techniques and thresholding of eigenvectors to obtain sparsity [CJ95]. Based on the well-known power method, [YZ13] introduced an efficient sparse PCA approximation to obtain the exact level of required sparsity, by truncating to zero the principal components iteratively except for their largest entries. A step further, under a spiked covariance model (see Section 4.2.2), [M+13] proposed a very efficient iterative thresholding approach for estimating principal subspaces in the sparse setting. Similarly, assuming a single-spike model, [KNV15] proved that, when the sparsity level $s \geq \Omega(\sqrt{n})$, a standard SDP approach cannot recover consistently the sparse spike; in particular, the authors presented empirical results suggesting that for $s = \mathcal{O}(\sqrt{n})$, recovery is possible by a simple covariance thresholding algorithm. More recently, [DM16] analyzed and theoretically proved, under a spiked model, that indeed the covariance thresholding algorithm [KNV15] succeeds with high probability under controlled growth rates between $p, n$ and $s$.

In this work, while restricting ourselves to a setting where $p$ and $n$ grow at a controlled joint rate, we provide an elementary argument, based on a matrix-wise Taylor expansion controlled through a concentration of measure approach, that generalizes the CT method to a large family of kernel-based methods, by means of a kernel random matrix approach [EK+10b, EK10a]. Concretely, we study kernel random matrices of the form $f(\boldsymbol{X}\boldsymbol{X}^\intercal/n)$ where $\boldsymbol{X} = \mathbf{\Sigma}^{1/2}\boldsymbol{Z}$ and $\boldsymbol{Z}$ is a random matrix with concentrated columns. [EK+10b] studied kernel matrices of the form $f(\boldsymbol{X}^\intercal\boldsymbol{X}/n)$ (*i.e.*, the so-called inner-product kernel matrices), which is equivalent to the case $\mathbf{\Sigma} = \boldsymbol{I}_p$ when considering the form $f(\boldsymbol{X}\boldsymbol{X}^\intercal/n)$. In particular, we elaborate from El Karoui's study by Taylor expanding $f(\boldsymbol{X}\boldsymbol{X}^\intercal/n)$ in the vicinity of $\mathbf{\Sigma}$ entry-wise and controlling the resulting matrices via concentration arguments.

### 4.2.2 Model and Main Results

Consider a data matrix $\boldsymbol{X} \in \mathcal{M}_{p,n}$ defined as

$$\boldsymbol{X} \equiv \mathbf{\Sigma}^{1/2}\boldsymbol{Z} = \left(\boldsymbol{I}_p + \boldsymbol{P}\right)^{1/2}\boldsymbol{Z}, \tag{4.13}$$

kernel random matrices and should not be confused with the standard kernel PCA.

where $\boldsymbol{Z} \in \mathcal{M}_{p,n}$ some random matrix, $\boldsymbol{P} = \sum_{i=1}^{k} \omega_i \boldsymbol{u}_i \boldsymbol{u}_i^\intercal$ and $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k] \in \mathcal{M}_{p,k}$ is isometric. Here, $k$ refers to the number of principal components (or eigenvectors) $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k \in \mathbb{R}^p$ to be evaluated, with $\omega_1 > \ldots > \omega_k > 0$ the corresponding eigenvalues respectively. We define the quantity $\beta_p \equiv \max_i \|[\boldsymbol{\Sigma}^{1/2}]_{\cdot,i}\|$.

We consider the following concentration and growth rate assumptions.

**Assumption 9** (Concentrated data)**.** *$\boldsymbol{Z} \propto \mathcal{E}_2$ with $\mathbb{E}\boldsymbol{Z} = \boldsymbol{0}$ and $\mathbb{E}[\boldsymbol{Z}\boldsymbol{Z}^\intercal/n] = \boldsymbol{I}_p$.*

**Assumption 10** (Growth rate)**.** *There exists $B > 0$ independent of $p, n$ such that $\max_{ij} |[\Sigma_p]_{ij}| < B$. Besides, there exists $\epsilon > 0$ such that $\beta_p \leq B' n^{\frac{1}{4}-\epsilon}$ for all $p, n$ and for some absolute constant $B' > 0$.*

Under these assumptions, our main technical result is as follows:

#### 4.2.2.1 Random Matrix Equivalent

**Theorem 4.3** (Asymptotic Equivalent)**.** *For $f$ a three-times continuously differentiable function, define the matrices $\boldsymbol{F}$ and $\tilde{\boldsymbol{F}}$ respectively by*[2]

$$\boldsymbol{F} \equiv \left\{ f\left( \left[ \frac{1}{n} \boldsymbol{X}\boldsymbol{X}^\intercal \right]_{ij} \right) \right\}_{i,j=1}^{p}, \quad \tilde{\boldsymbol{F}} \equiv f(\boldsymbol{\Sigma}) + \sum_{k=1}^{2} \frac{f^{(k)}(\boldsymbol{\Sigma})}{k!} \odot \left[ \boldsymbol{\Sigma}^{1/2} \left( \frac{1}{n} \boldsymbol{Z}\boldsymbol{Z}^\intercal - \boldsymbol{I}_p \right) \boldsymbol{\Sigma}^{1/2} \right]^{\odot k}.$$

*Then*

$$\boldsymbol{F} \leftrightsquigarrow \tilde{\boldsymbol{F}}$$

*Specifically, for $\eta > 0$, there exists an absolute constant $C_\eta > 0$ such that with probability at least $1 - \eta$*

$$\| \boldsymbol{F} - \tilde{\boldsymbol{F}} \| \leq C_\eta \frac{\beta_p^6 \, p}{n^{3/2} \sqrt{\eta}}. \tag{4.14}$$

For a general smooth function $f$, the kernel random matrix $f(\hat{\boldsymbol{\Sigma}})$ is particularly difficult to analyze through the usual tools of random matrix theory, such as the moment or Stieltjes transform-based methods [Tao12]. Rather than directly analyzing such a kernel random matrix, Theorem 4.3 gives an asymptotic *random* equivalent to it, in operator norm, that has mainly two properties. First, the approximation matrix $\tilde{\boldsymbol{F}}$ contains "simple" objects that have already been analyzed in random matrix theory – in particular, the term $(\boldsymbol{Z}\boldsymbol{Z}^\intercal/n - \boldsymbol{I}_p)$ in the expression of $\tilde{\boldsymbol{F}}$. Second, the approximation in operator norm implies (by Weyl's inequality [EI98, Theorem 4.1]) that, when $\|\boldsymbol{F} - \tilde{\boldsymbol{F}}\|_{op} \to 0$, $\boldsymbol{F}$ and $\tilde{\boldsymbol{F}}$ have the same eigenvalues and same "isolated" eigenvectors asymptotically (see Corollary 4.2 subsequently).

Before sketching a proof for Theorem 4.3, we need the following technical Lemma which establishes the concentration of the entries of the noise term $(\boldsymbol{Z}\boldsymbol{Z}^\intercal/n - \boldsymbol{I}_p)$.

**Lemma 4.2** (A Concentration Result)**.** *For all $i, j \in [p]$, the bilinear form*

$$g_{ij}(\boldsymbol{Z}) \equiv [\boldsymbol{\Sigma}^{1/2}]_{i,\cdot} \left( \frac{1}{n} \boldsymbol{Z}\boldsymbol{Z}^\intercal - \boldsymbol{I}_p \right) [\boldsymbol{\Sigma}^{1/2}]_{\cdot,j} \tag{4.15}$$

[2] $f$ and $f^{(k)}$ are applied entry-wise and $\odot k$ stands for the element-wise $k$-th power.

*satisfies*

$$g_{ij}(\boldsymbol{Z}) \propto \mathcal{E}_1\left(\frac{\beta_p^2}{n}\right) + \mathcal{E}_2\left(\frac{\beta_p^2}{\sqrt{n}}\right). \tag{4.16}$$

*Proof.* Denoting by $\boldsymbol{v}_i$ the $i$-th column vector of the matrix $\boldsymbol{\Sigma}^{1/2}$, we have by the polarization identity, for all $\boldsymbol{M}$ Hermitian,

$$\boldsymbol{v}_i^\intercal \boldsymbol{M} \boldsymbol{v}_j = \frac{1}{4}[(\boldsymbol{v}_i + \boldsymbol{v}_j)^\intercal \boldsymbol{M}(\boldsymbol{v}_i + \boldsymbol{v}_j) - (\boldsymbol{v}_i - \boldsymbol{v}_j)^\intercal \boldsymbol{M}(\boldsymbol{v}_i - \boldsymbol{v}_j)].$$

It thus suffices to prove the result for the quadratic form

$$g(\boldsymbol{Z}) = \boldsymbol{v}^\intercal \left(\frac{1}{n}\boldsymbol{Z}\boldsymbol{Z}^\intercal - \boldsymbol{I}_p\right)\boldsymbol{v}$$

where $\boldsymbol{v} \in \mathbb{R}^p$. Noticing that $\boldsymbol{v}^\intercal \boldsymbol{Z}\boldsymbol{Z}^\intercal \boldsymbol{v} = \|\boldsymbol{v}^\intercal \boldsymbol{Z}\|^2$ and $\mathbb{E}\left[\frac{1}{n}\boldsymbol{v}^\intercal \boldsymbol{Z}\boldsymbol{Z}^\intercal \boldsymbol{v}\right] = \boldsymbol{v}^\intercal \boldsymbol{v}$, we need to prove the concentration of the random variable $\|\boldsymbol{v}^\intercal \boldsymbol{Z}\|^2$. In fact, since $\boldsymbol{v}^\intercal \boldsymbol{Z}$ is a Concentrated vector, by Definition, $\|\boldsymbol{v}^\intercal \boldsymbol{Z}\| \propto \mathcal{E}_2(\|\boldsymbol{v}\|)$ since $M \mapsto \boldsymbol{v}^\intercal M$ and $u \mapsto \|u\|$ are respectively $\|v\|$-Lipschitz and 1-Lipschitz functions. We get the final result by Proposition 2.5. □

Now we turn to the main ingredients to prove Theorem 4.3.

*Sketch of Proof of Theorem 4.3.* The main idea of the proof relies on the following intuition: for large $n$, the entries of $\boldsymbol{Z}\boldsymbol{Z}^\intercal/n - \boldsymbol{I}_p$ and its successive Hadamard products tend to zero at controllable rate. The concentration of measure framework then allows for the control of non-linear functions of the entries of $\boldsymbol{Z}\boldsymbol{Z}^\intercal/n - \boldsymbol{I}_p$ thanks to the previous Lemma. A Taylor expansion of $\boldsymbol{F}$ around $f(\boldsymbol{\Sigma})$ then leads to controlling the operator norm of

$$f^{(3)}(\xi^n) \odot [\boldsymbol{\Sigma}_p^{1/2}(\boldsymbol{Z}\boldsymbol{Z}^\intercal/n - \boldsymbol{I}_p)\boldsymbol{\Sigma}^{1/2}]^{\odot 3}$$

for $\xi^n$ a matrix with entries in the set $[[\boldsymbol{X}\boldsymbol{X}^\intercal/n]_{ij}, [\boldsymbol{\Sigma}]_{ij}]$ (or $[[\boldsymbol{\Sigma}]_{ij}, [\boldsymbol{X}\boldsymbol{X}^\intercal/n]_{ij}]$). This follows precisely from exploiting Lemma 4.2 twice, to control the fluctuations of the entries of both $\xi^n$ (by the conditions on $\max_{ij}|[\boldsymbol{\Sigma}]_{ij}|$ and $\beta_p$) and $[\boldsymbol{\Sigma}^{1/2}(\boldsymbol{Z}\boldsymbol{Z}^\intercal/n - \boldsymbol{I}_p)\boldsymbol{\Sigma}^{1/2}]^{\odot 3}$, with the bound provided in the theorem statement, thereby completing the proof. A detailed proof of Theorem 4.3 is provided in the Appendix. □

In order to simplify our arguments concerning the analysis of Sparse PCA, we make the additional assumption that both $p$ and $n$ are large an comparable. We further suppose that the deterministic quantities $\omega_i$'s and $\beta_p$ are bounded as $n \to \infty$. Specifically, we make the following set of assumptions

**Assumption 11** (RMT Growth rate)**.** *As* $n \to \infty$,

***A1*** $p/n \to c \in (0, \infty)$,

***A2*** $\limsup_n \max_i \omega_i < \infty$; *specifically* $\limsup_n \beta_p < \infty$.

Under this setting, we have the following important corollary to Theorem 4.3.

**Corollary 4.1.** *Define the matrices $\boldsymbol{F}$ and $\tilde{\boldsymbol{F}}$ as in Theorem 4.3 and let Assumptions **A1** and **A2** hold. Then, for $\eta > 0$*

$$\boldsymbol{F} = \tilde{\boldsymbol{F}} + \mathcal{O}_\eta(n^{-\frac{1}{2}}), \tag{4.17}$$

*where $\boldsymbol{X} = \mathcal{O}^m_\eta(n^{-\alpha})$ stands for the fact that $\mathbb{P}\left\{\|\boldsymbol{X}\| \geq C\, n^{-\alpha}\, \eta^{-\frac{1}{2m}}\right\} \leq \eta$ for some absolute constant $C > 0$ and non-negative integer $m$.*

As a consequence of Corollary 4.1, we have, by the $\sin(\Theta)$ theorem of [DK70], the corollary below concerning the eigenvectors of the matrices $F$ and $\tilde{F}$.

**Corollary 4.2.** *Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k$ and $\tilde{\boldsymbol{v}}_1, \ldots, \tilde{\boldsymbol{v}}_k$ denote respectively the $k$ principal eigenvectors of $\boldsymbol{F}$ and $\tilde{\boldsymbol{F}}$. Denote by $\Delta_i = \omega_i - \omega_{i+1}$ for $i \in [k-1]$. Then for $\eta > 0$, we have*

$$\max_{i\in[k]} \min_{y\in\{+1,-1\}} \Delta_i^2 \, \|\boldsymbol{v}_i - y\tilde{\boldsymbol{v}}_i\|^2 = \mathcal{O}_\eta(n^{-1}). \tag{4.18}$$

#### 4.2.2.2 Application to sparse PCA

Before presenting our results concerning the behavior of kernel matrices of the form $f(\hat{\boldsymbol{\Sigma}})$ in the context of sparse PCA, we introduce in following a particular notion of sparsity for large matrices that will be used subsequently.

When considering a large-dimensional random matrix setting, the notion of sparsity for such matrices is particularly attached to the choice of the matrix norm.[3] [EK08] introduced a definition ($\varepsilon$-sparsity) for sparsity of matrices that is compatible with spectral analysis, and specifically adapted to the operator norm. The $\varepsilon$-sparsity definition requires some notions from graph theory that we present in the following:

- To each $p \times p$ symmetric matrix $M$, we define its corresponding adjacency matrix as $\mathcal{A}(M) = \{1_{M_{ij}\neq 0}\}_{i,j=1}^p$, which corresponds to a graph $\mathcal{G}_p$ with $p$ vertices.
- A walk is said to be closed on this graph if it starts and finishes at the same vertex and the number of edges traversed by a walk defines the length of this walk. Denote $\mathcal{C}_p(k)$ the set of closed walks of length $k$ on $\mathcal{G}_p$.

We these notions, we are now in place to introduce the definition of the $\varepsilon$-sparse notion.

**Definition 8** ($\varepsilon$-sparse matrices [EK08, Definition 1])**.** *A sequence of covariance matrices $\{\boldsymbol{\Sigma}_p\}_{p=1}^\infty$ is said to be $\varepsilon$-sparse if the sequence of their associated graphs $\{\mathcal{G}_p\}_{p=1}^\infty$ satisfies, for all $k \in 2\mathbb{N}$,*

$$|\mathcal{C}_p(k)| \leq C_k\, p^{\varepsilon(k-1)+1}$$

*where $\varepsilon \in [0,1]$, $C_k > 0$ independent of $p$ and $|\mathcal{S}|$ denotes the cardinality of the set $\mathcal{S}$.*

The $\varepsilon$-sparsity is both useful and convenient to our study for the following reasons: 1) it is adapted to the analysis of the operator norm of large sparse matrices (as we give concentration results on the operator norm); 2) it is also more general than other sparsity notions such as in [BL08]. In the latter, the authors developed a natural permutation-invariant notion of sparsity which is more specific than Definition 8 as pointed out in the introduction of their article. Furthermore, note that both sparsity notions (Definition 8 and the one in [BL08]) provide equivalent bounds for $\varepsilon < \frac{1}{2}$ and when considering the large dimensional $p \sim n$ setting (see subsection 2.4 in [BL08]); this is precisely the setting considered in Theorem 4.4 introduced subsequently (*cf.* $\mu > 0$).

[3] Considering the identity matrix (which is a sparse matrix), $\|\boldsymbol{I}_p\| = 1$ while $\|\boldsymbol{I}_p\|_F = \sqrt{p} \to \infty$.

**Remark 4.1.** *As Definition 8 is based on a graph defined by its corresponding adjacency matrix, we have the following property: given an $\varepsilon$-sparse matrix $\boldsymbol{M}$ and a function $f$ such that $f(0) = 0$ and $f(x) \neq_{x\neq 0} 0$, the matrix $f(\boldsymbol{M})$, resulting from the application of $f$ entry-wise to $\boldsymbol{M}$, remains $\varepsilon$-sparse; this is simply a consequence of $\mathcal{A}(\boldsymbol{M}) = \mathcal{A}(f(\boldsymbol{M}))$.*

To get an insight on our coming results, consider the scenario where $\boldsymbol{U}$ contains finitely many non-zero entries. In this case, the perturbation matrix $P$ in equation 4.13 contains finitely many non-zero entries (say $s$) on each line and a simple enumeration shows that $|\mathcal{C}_p(k)| \leq p\, s^{k-1}$, thus $\boldsymbol{P}$ is 0-sparse in the sense of Definition 8. Similarly, $\boldsymbol{I}_p$ is 0-sparse and by the additive stability[4] of the $\varepsilon$-sparsity notion, $\boldsymbol{\Sigma}$ remains 0-sparse. More generally, if we assume that it exists $\varepsilon \in [0, \frac{1}{2})$ such that the population covariance matrix $\boldsymbol{\Sigma}$ is $\varepsilon$-sparse, we have the following set of consequences. By Corollary 4.1, choosing $f$ in such a way that $f'(0) = f''(0) = 0$ ensures that the terms $f'(\boldsymbol{\Sigma}) \odot \dots$ and $f''(\boldsymbol{\Sigma}) \odot \dots$ vanish in the expression of $\tilde{\boldsymbol{F}}$. Indeed, for $k \in \{1, 2\}$

(i) Only finitely entries of $f^{(k)}(\boldsymbol{\Sigma})$ do not vanish, precisely by Remark 4.1, since

$$\mathcal{A}(f^{(k)}(\boldsymbol{\Sigma})) = \mathcal{A}(\boldsymbol{\Sigma}),$$

the matrix[5] $f^{(k)}(\boldsymbol{\Sigma})$ is also (almost) $\varepsilon$-sparse.

(ii) The matrix $\boldsymbol{F}^{(k)} = \left[\boldsymbol{\Sigma}^{1/2}\left(\frac{1}{n}\boldsymbol{Z}\boldsymbol{Z}^\intercal - \boldsymbol{I}_p\right)\boldsymbol{\Sigma}^{1/2}\right]^{\odot k}$ has entries of order $\mathcal{O}(n^{-k/2})$. As a result,[6] we have for $\eta > 0$ and for all $m > 0$, $\max_{i,j} |F_{ij}^{(k)}| = \mathcal{O}_\eta^m\left(n^{-\frac{k}{2}+\frac{1}{m}}\right)$.

Since in addition the operator norm of $\boldsymbol{\Sigma}^{1/2}(\boldsymbol{Z}\boldsymbol{Z}^\intercal/n - \boldsymbol{I}_p)\boldsymbol{\Sigma}^{1/2}$ is typically of order $\mathcal{O}(1)$ (see *e.g.*, [BS+98a]), it is then easily seen that, for each $k \geq 1$, the operator norm of the Hadamard product $f^{(k)}(\boldsymbol{\Sigma}) \odot \boldsymbol{F}^{(k)}$ vanishes (see Lemma C.7 in the Appendix). In particular, note that the non-zero entries of $\boldsymbol{\Sigma}$ are controlled through the maximum entry of $\boldsymbol{F}^{(k)}$ which is vanishing asymptotically, as mentioned in item (ii) above. On the opposite $f(\boldsymbol{\Sigma})$ does not vanish since it has entries bounded away from zero (as long of course as $f \neq 0$). We precisely have the following result.

**Theorem 4.4** (Sparse PCA)**.** *Let $\mu > 0$ and suppose $\boldsymbol{\Sigma}$ is a $\frac{1}{2+\mu}$-sparse matrix. For $f$ a three-times continuously differentiable function such that $f'(0) = f''(0) = 0$ and for $\eta > 0$, we have for all $\epsilon \in (0, \frac{\mu}{2(3+2\mu)})$*

$$\boldsymbol{F} = f(\boldsymbol{\Sigma}) + \mathcal{O}_\eta^{\lfloor 1/\epsilon \rfloor}\left(n^{\frac{-\mu}{2(2+\mu)}+\epsilon\left(2-\frac{1}{2+\mu}\right)}\right). \tag{4.19}$$

*Proof.* See Section C.2.2. □

**Remark 4.2.** *Theorem 4.4 gives a general result concerning the estimation of $\varepsilon$-sparse covariance matrices (more precisely, element-wise functionals of sparse covariance matrices). In particular, the spiked model in equation 4.13 with $\boldsymbol{U}$ sparse corresponds to the particular case when $\mu \to \infty$; in this case, for $\eta > 0$ and for all $\epsilon \in (0, 1/4)$, $\boldsymbol{F} = f(\boldsymbol{\Sigma}) + \mathcal{O}_\eta^{\lfloor 1/\epsilon \rfloor}(n^{-\frac{1}{2}+2\epsilon})$.*

One may then perform a PCA on $\boldsymbol{F}$ for some function $f$ with $f'(0) = f''(0) = 0$ (we denote by these two conditions in the following). But, while $\boldsymbol{\Sigma} = \boldsymbol{I}_p + \boldsymbol{P}$ is a low rank perturbation of the identity (therefore having only $k$ eigenvalues strictly greater than 1 with

[4] See Fact .1 in [EK08].

[5] Given $\boldsymbol{M} \in \mathcal{M}_p$, its corresponding adjacency matrix is defined as $\mathcal{A}(\boldsymbol{M}) = \{\mathbf{1}_{M_{ij}\neq 0}\}_{i,j=1}^p$.

[6] See proof of Lemma C.7.

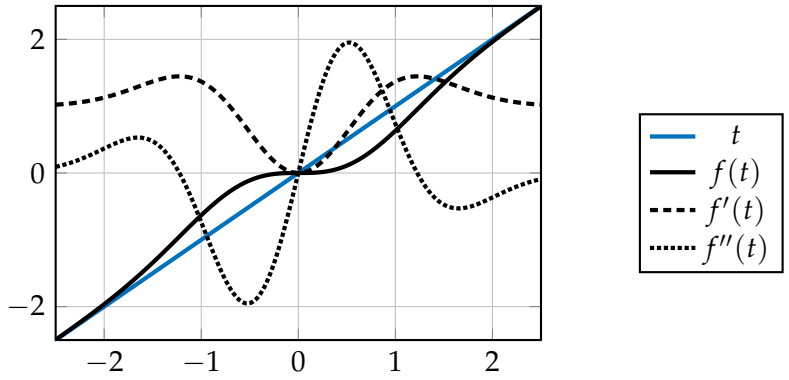



Figure 4.18: The function $f(t) = t(1 - e^{t^2})$ along with its derivatives.

$k$ being fixed w.r.t. $p$), $f(\mathbf{\Sigma})$ is likely more complex and not a mere low rank deformation of the identity. Now, if $\mathbf{\Sigma}$ has all its non-zero entries greater than a certain threshold $\tau$, an appropriate choice for $f$ that avoids the deformation of $\boldsymbol{I}_p + \boldsymbol{P}$ is such that and $f(t) = t$ for all $|t| > \tau$.

Such a convenient choice is

$$f(t) = t(1 - e^{-at^2}), \tag{4.20}$$

for some $a > 0$, which is depicted along with its derivatives in Figure 4.18 for $a = 1$. This function notably satisfies

$$f'(t) = 1 + e^{-at^2}(2at^2 - 1) \Rightarrow f'(0) = 0,$$
$$f''(t) = -2ate^{-at^2}\left(2at^2 - 3\right) \Rightarrow f''(0) = 0.$$

Note that a compromise in the choice of $a$ must be made that both maintains a close approximation of the identity by $f$ on a large range and rather small values of $f''$ in the vicinity of zero. Interestingly, it can be verified that the extrema of $f'$ are independent of $a$ but are found at $\pm\sqrt{\frac{3}{2a}}$ and thus smaller values of $a$ create sharper $f'$ in the vicinity of zero. Similarly, the extrema of $f''$ are found at $\pm\sqrt{\frac{3\pm\sqrt{6}}{2a}} \propto 1/\sqrt{a}$, and precisely given by the four values $2\sqrt{3a(3\pm\sqrt{6})}e^{-\frac{1}{2}(3\pm\sqrt{6})} \propto \sqrt{a}$. Thus smaller $a$ induce larger maxima for $f''$ but no sharper slope.

#### 4.2.2.3 Experimental Validation

In this section, we provide some experiments in the context of sparse PCA. Precise setting given in caption of Figure 4.19. The spectrum of the sample covariance matrix (in gray) is quite different from that of the population covariance $\mathbf{\Sigma}$. One instead observes a "bulk" of eigenvalues spread in the vicinity of 1. The limiting measure of these eigenvalues is given by the Marčenko-Pastur density function

$$f_{MP}(x) = \frac{1}{2\pi c}\frac{\sqrt{(\lambda^+ - x)(x - \lambda^-)}}{x} + \max\left(1 - \frac{1}{c}, 0\right)\delta(x),^{7} \tag{4.21}$$

[7] $\delta$ denotes here the dirac function.

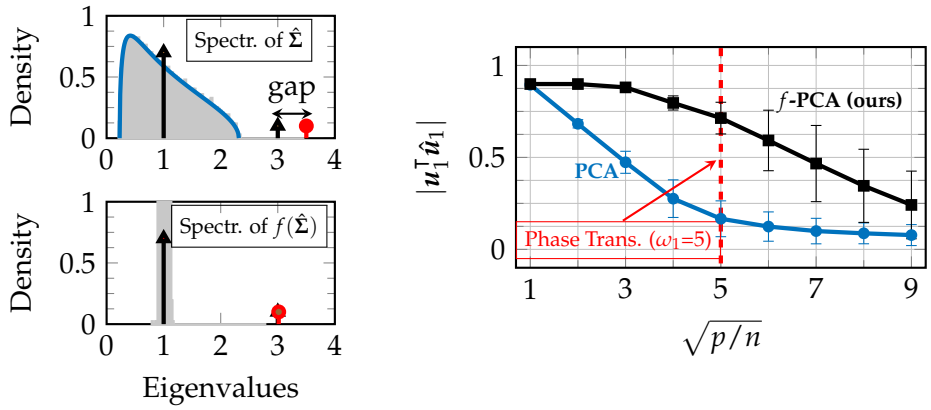



Figure 4.19: **(Left)** Spectrum of $\hat{\mathbf{\Sigma}}$ *(up)* and $f(\hat{\mathbf{\Sigma}})$ *(bottom)* for $p = 2048$ and $n = 7500$. Limiting Marčenko-Pastur density [MP67] in blue versus spectrum of $\mathbf{\Sigma}$ in black, with $\omega_1 = 2$; estimated largest eigenvalue in red. **(Right)** Alignment between estimated PC and GT (the "Three Peak" example of [JL09] in the "Symmlet 8" wavelet basis), in terms of $\sqrt{p/n}$. We considered $\omega_1 = 5$ and thus the phase transition for standard PCA occurs at $\sqrt{p/n} = 5$, thereby suggesting another phase transition for $f$-PCA. Curves obtained from 500 realizations of $\boldsymbol{Z}$ with random *i.i.d.* $\mathcal{N}(0,1)$ entries.

which is represented by the blue curve in Figure 4.19 (left). Furthermore, one observes a gap between the true spike and the estimated spike (in red) through the sample covariance matrix. This phenomenon is well-understood in random matrix theory. In particular, the extreme eigenvalue in our setting converges almost surely to the quantity $(1+\omega_1)\left(1+\frac{c}{\omega_1}\right)$, where we recall that $c = \lim_n p/n$.

However, thanks to sparsity, the spectrum of $F = f(\hat{\mathbf{\Sigma}})$ closely matches that of $\mathbf{\Sigma}$, as suggested by Theorem 4.4. In particular, the extreme eigenvalue, which corresponds to the principal component, is consistently estimated. Figure 4.19 (right) depicts the alignment between the estimated principal component and ground truth, by standard PCA (in blue) and our method (in black), in terms of $\sqrt{p/n}$. Our method retrieves the principal component even when the spike is not visible in the spectrum of $\hat{\mathbf{\Sigma}}$; namely beyond the phase transition $\sqrt{p/n} \geq \omega_1$. In fact, the standard PCA result is too noisy compared with the one when considering $f(\hat{\mathbf{\Sigma}})$, as depicted in Figure 4.20.

**Higher Rank Case.** In this paragraph, we provide further experiments by considering a rank three case and by using the "Three Peak", "Piece Poly" and "Step New" signals of Johnstone et al. [JL09], in the "Symmlet 8" wavelet basis, as principal components. We compare the estimated PCs by our method with the kernel function in equation 4.20 to the estimated ones through standard PCA and the CT method [DM16]. As shown in Figure 4.22, the proposed method retrieves consistently the principal components compared to a standard PCA. In particular, we obtain results that are similar to the ones obtained by the CT method while generalizing it to the class of smooth functions with $f'(0) = f''(0) = 0$.

**Other choices of the kernel function $f$.** In this paragraph, we consider functions of the form $f(t) = \alpha t^3 + \beta t^2 + \gamma t$ where $\alpha, \beta, \gamma \in \mathbb{R}$ are some parameters to fix in order to allow or not the conditions $f'(0) = f''(0) = 0$. In particular, we set different parameters

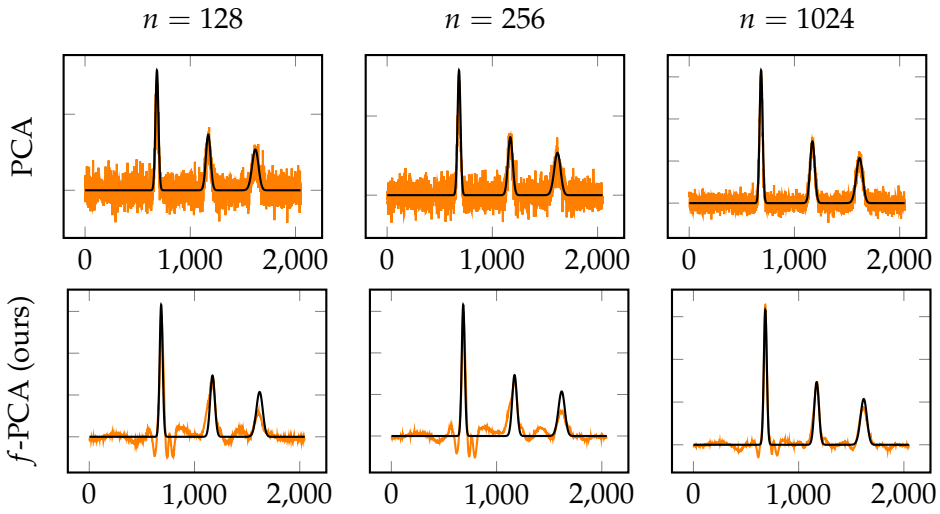



Figure 4.20: Principal component recovery (in orange) by standard PCA **(up)** and our method **(down)** for the "Three Peak" example of johnstone2009consistency. The signal is sparse in the "Symmlet 8" wavelet basis. We use $p = 2048$, $\omega_1 = 5$ for the strength of the spike and different values of $n$.

choices for $\alpha, \beta$ and $\gamma$ in order to validate these conditions. Figure 4.23 depicts different PC recovery using the $f$-PCA method with the considered class of functions. As we can observe from this figure, the "cleanest" signal recovery is obtained when $\alpha \neq 0, \beta = 0, \gamma = 0$ (*i.e.*, when $f'(0) = f''(0) = 0$) thereby validating our theoretical conditions on the kernel function $f$ for a consistent sparse PCA recovery. Note that these conditions are necessary but not sufficient in the sense that $f$ has to be linear for large values of $t$ (In particular, this is the case for the function $f$ given by equation 4.20). In fact, the outcome provided by $f$-PCA for $f(t) = \alpha t^3$ with $\alpha \neq 0$ is not optimal as the obtained signal is deformed (due to the unverified linearity condition), compared to the GT one.

**Complexity and Performance of $f$-PCA.** In terms of complexity, as our method consists in computing the sparse eigenvectors of a $p \times p$ matrix which can be done by power method, the complexity of estimating the principal component is about $\mathcal{O}(ps)$ where $s$ is the sparsity level. And regarding the performance *w.r.t.* state-of-the-art methods, Figure 4.21 depicts the performances of standard PCA, different state-of-the-art sparse PCA methods and our method, in terms of total projections score (left) and total projections error (right), for different values of the amplitudes $\omega_i$'s. We refer, in this figure, to standard PCA as PCA, TpowPCA for the method in [YZ13], ITSPCA for the method in [M+13], CT refers to the method in [DM16] and finally we refer to our method as $f$-PCA. The total projections score $\mathcal{S}$ and error $\mathcal{E}$ are given respectively by

$$\mathcal{S} = \frac{1}{k}\sum_{i=1}^{k}(\boldsymbol{u}_i^\mathsf{T}\hat{\boldsymbol{u}}_i)^2, \quad \mathcal{E} = \|\boldsymbol{U}\boldsymbol{U}^\mathsf{T} - \hat{\boldsymbol{U}}\hat{\boldsymbol{U}}^\mathsf{T}\|_F, \tag{4.22}$$

where $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k]$ are the ground truth principal components and $\hat{\boldsymbol{U}} = [\hat{\boldsymbol{u}}_1, \ldots, \hat{\boldsymbol{u}}_k]$ are the estimated ones.

As suggested theoretically and verified experimentally, our proposed method strongly attenuates the "noise component" of the sample covariance matrix and thus consistently

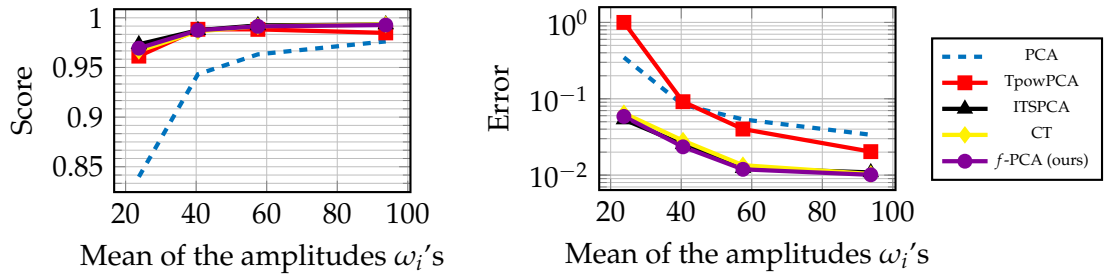



Figure 4.21: Performances of standard PCA, different state-of-the-art sparse PCA methods and our method in term of total projections score **(left)** and total projections error **(right)** for different values of the amplitudes $\omega_i$. The PCs $u_i$, for $i \in [4]$ are the "Three Peak", "Piece Poly", "Step New" and "Sing" signals of johnstone2009consistency. We use $p = 2048$ and $n = 1024$. The soft-parameters $a$ and $\tau$ (respectively for our method and CT) are selected by cross-validation using a validation set of size $n$. The selected parameters are $a = 20$ and $\tau = 0.1$.

estimates the principal components. In particular, in term of total projections score, PCA is the most inconsistent. In general, ITSPCA, CT and our method give equivalent results. The same holds when considering the total projections error as a metric, except that TpowPCA performs inconsistently, compared to PCA, for small values of amplitudes due to the initialization step from the PCA eigenvectors.

**Discussion.** The mostly used concurrent methods to PCA in a sparse context are iterative truncated power methods (such as the TPower [YZ13] algorithm or the ITSPCA [M+13] approach). These algorithms, despite great observed performances, as compared to standard PCA, suffer from two limitations. First, they are usually initialized from the PCA eigenvectors themselves and may not converge to good estimates. For weak signals, PCA is so impacted by noise that the mentioned initialization limitation may lead to non convergent or dramatically erroneous outcomes of the method. The proposed approach deals precisely with this limitation by strongly attenuating the "noise component" of the sample covariance matrix. In particular, our approach gives equivalent results to the CT method while generalizing it to the class of smooth functions $f$ such that $f'(0) = f''(0) = 0$, in the considered regime. The second limitation concerns the choice of the hyper-parameters; in fact, TPower and ITSPCA need to set up an arbitrary deterministic threshold value that maintains at each iteration step only most powerful components. The proposed method as well as CT need also to set up a "soft" parameter ($a$ and $\tau$ respectively). But, on the basis of [CS13, KC17], we believe that our present investigation can be extended to the *asymptotically non-trivial* setting where $\omega_i = \mathcal{O}(1/\sqrt{p})$ (in which case the dominant eigenmodes scale at a similar rate with residual noise); this setting may likely allow to exhibit and estimate optimal hyper-parameter choices. Notably, this setting has already been used in [TAKC18] in a different context, for hyper-parameters estimation.

### 4.2.3 Central Contribution and Perspectives

We tackled in this work the problem of sparse PCA in the large dimensional setting through a random matrix perspective, thereby generalizing recent ideas to a broader

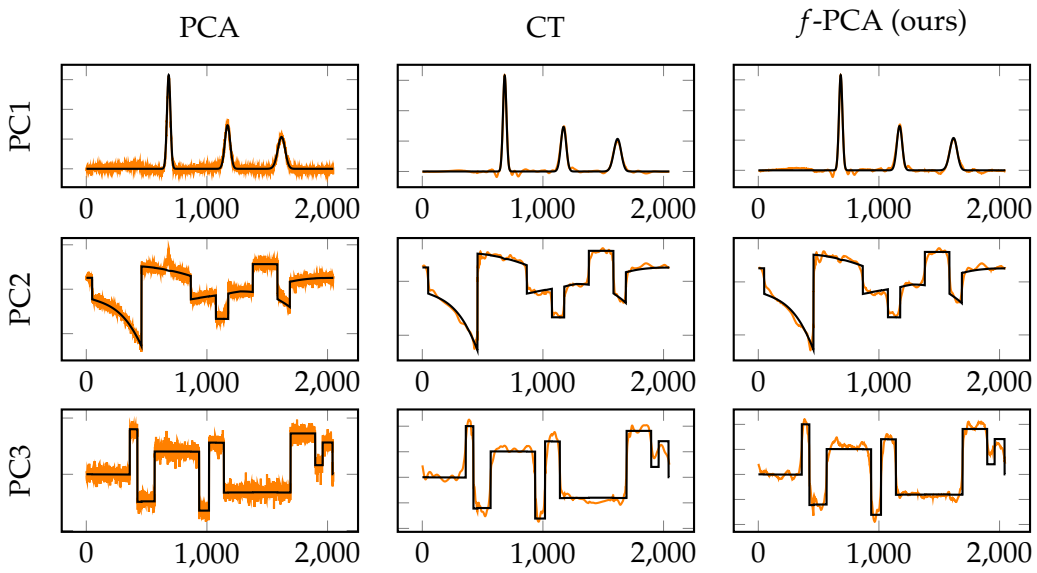



Figure 4.22: Multiple principal components ($k = 3$) recovery (in orange) with standard PCA **(left)**, the CT method **(middle)** and our method **(right)** where the PCs are considered to be the "Three Peak", "Piece Poly" and "Step New" signals of johnstone2009consistency, in the "Symmlet 8" wavelet basis. We use $p = 2048$, $n = 1024$ and the spikes strengths are set respectively as $\omega_1 = 100$, $\omega_2 = 75$ and $\omega_3 = 50$. In particular, we note the similarity between the results obtained by our method and CT.

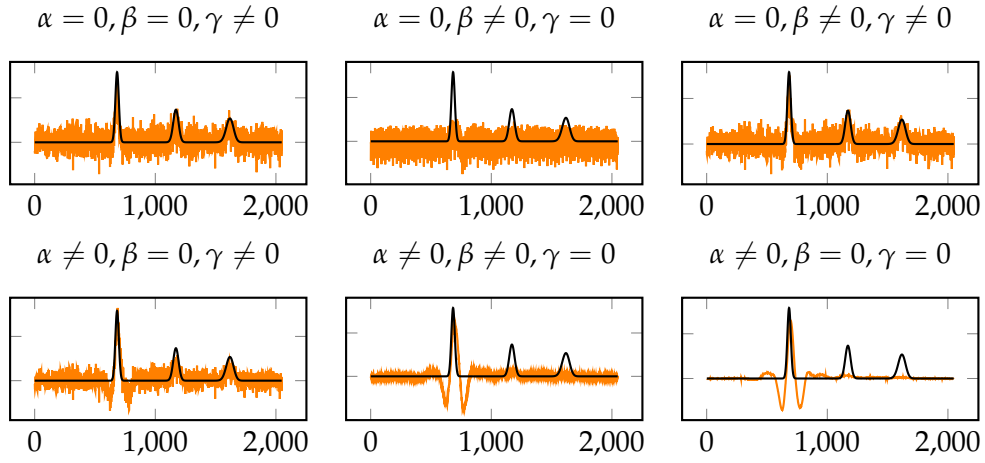



Figure 4.23: PC recovery (in orange) by $f$-PCA with the function $f(t) = \alpha t^3 + \beta t^2 + \gamma t$ for different values of the parameters $(\alpha, \beta, \gamma) \in \mathbb{R}^3$. We consider the "Three Peak" example of johnstone2009consistency which is sparse in the "Symmlet 8" wavelet basis. We use $p = 2048$, $n = 256$ and $\omega_1 = 5$. In particular, we notice that the "cleanest" signal is obtained when $\alpha \neq 0, \beta = 0, \gamma = 0$ which validate our theoretical conditions $f'(0) = f''(0) = 0$.

kernel-based method. Our analysis of this problem has yielded insights into how the principal components can be consistently estimated. Namely, given a spiked covariance model $\hat{\mathbf{\Sigma}}$ and a smooth function $f$, we gave in this work sufficient conditions on $f$ to consistently estimate the principal components through the matrix $f(\hat{\mathbf{\Sigma}})$. Our result is based on the concentration assumption which generalizes the Gaussian case. However, the proposed method needs to set up a "soft" parameter ($a$). Based on [CS13, KC17], the present work can be extended to the *asymptotically non-trivial* setting where $\omega_i = \mathcal{O}(1/\sqrt{p})$ (in which case the dominant eigenmodes scale at a similar rate with residual noise); which may likely allow to exhibit and estimate optimal hyper-parameter choices.

# Chapter 5

# Beyond Kernel Matrices, to Neural Networks

**Contents**



In the previous chapter, we have mainly presented kernel methods which explicitly depend on the input data, even when coupled with a classifier such as LS-SVM [LC17]. This Chapter will present methods that go beyond kernel matrices and which appear as sub-components of neural networks. Specifically, this chapter is composed of two main sections. The first section provides an analysis of the Softmax layer in neural networks, while the second section addresses questions about the Dropout operation in neural networks.

## 5.1 A random matrix analysis of Softmax layers

This section is based on the following work:

**(C4)** **MEA. Seddik**, C.Louart, R. Couillet, M. Tamaazousti, "*The Unexpected Deterministic and Universal Behavior of Large Softmax Classifiers*", AISTATS 2021.

### 5.1.1 Motivation

The intricate nature of deep neural network training leaves little insight on the specific information encoded into the inter-layer connectivity weights of a fully trained network, thereby so far not allowing for particularly useful interpretation and control of their performances [YKYR18].

At the very source of these difficulties are the multiple non-linearities and the implicit optimization scheme involved in the network design: the activation functions in the intermediate layers as well as the soft or hard final decision layer [LWL+17]. For lack of a tractable comprehensive analysis, literature studies have mostly focused on individual components which, when isolated, become tractable. For instance, the effect of non-linearities in a single-hidden layer network was analysed in [PW17, LC18c], the learning dynamics in elementary network designs in [SMG13, dCPS+18] and the overall understanding of the geometry of the loss surface in a largely approximated version of a deep neural net in [PB17, CHM+15].

These works are however restricted to the analysis either of the intermediate layers of practical neural nets, or oversimplify the network to an extent that makes the results rather impractical. The present study instead focuses on the training of the weights of the last decision layer, by specifically studying the widely used Softmax component in neural networks classifiers. The Softmax classifier has the property, which we will see to be of importance here, to be optimal for Gaussian mixture inputs with equal covariance [YW19]. Specifically, assuming the feature representations of the data fixed at the penultimate layer of the network, and modelling these features as *concentrated random vectors* [Led05b] (which is a natural assumption as concentrated random vectors enjoy the property to be stable through Lipschitz maps, and thus through the action of intermediate neural network layers [SLTC20]), in this work we have studied the statistical behavior of this last layer once trained (see Figure 5.1).

From a technical standpoint, as the Softmax classifier training corresponds to a (possibly non-convex) optimization problem, our analysis of the Softmax weights is performed by first expressing the optimization problem as a contracting fixed point equation, and then showing that the assumed *concentration properties* of the data features naturally transfer to the solution of the fixed-point equation, and thus to the Softmax weights. This has the major consequence that, as $n, p \to \infty$, the Softmax weights tend to have a deterministic behavior which we express explicitly as a function of the data statistics and the Softmax parameters.

Our most fundamental findings may be summarized as follows: **1.** the above deterministic behavior exhibits a surprising *universality* of the Softmax classifier, in the sense that the large dimensional statistics of the weights solely depend on the statistical means and covariances of the input data features; **2.** this suggests in turn that, quite counter-intuitively, at least as far as the last Softmax classification layer is concerned, no further discriminative feature of the data is extracted and, possibly most outstandingly, *the Softmax layer treats the input data as if they were Gaussian random vectors*; this, in passing, supports the Gaussianity assumption on the data representations commonly considered in the literature [HRU+17, PRU+18]; **3.** combined to the aforementioned optimality of the Softmax classifier on Gaussian mixture models with strongly discriminative class-wise means, this compellingly supports an overall classification optimality of the Softmax

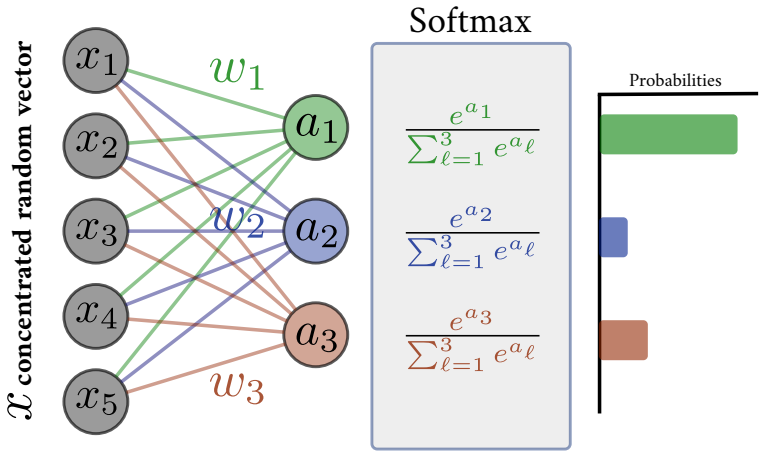



Figure 5.1: Illustration of the Softmax classifier with *concentrated random vectors* [Led05b, LC18b] (belonging to some space $\mathcal{X}$) as input data, i.e., satisfying the property $\mathbb{P}\left(|\varphi(\boldsymbol{x}) - \mathbb{E}\varphi(\boldsymbol{x})| > t\right) \leq C e^{-(t/\sigma)^q}$, for all 1-Lipschitz $\varphi : \mathcal{X} \to \mathbb{R}$ (see Definition 7). GAN data as well as their deep network-based representations are practical examples of such random vectors [SLTC20].

classifier on large dimensional representations of real data. A similar behavior was already pointed out, yet not well understood, by the authors in [MVPC13, GCM18]; **4.** Our findings are supported both theoretically and practically by considering the input data features as CNN-representations of images generated by the BigGAN model [BDS18].

### 5.1.2 Model setting: the Softmax classifier

Let $(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_n, \boldsymbol{y}_n)$ be a set of $n$ labeled data associated to one of $k$ classes $\mathcal{C}_1, \ldots, \mathcal{C}_k$, where $\boldsymbol{x}_i \in \mathbb{R}^p$ and $\boldsymbol{y}_i \in \mathbb{R}^k$ is one-hot encoded vectors such that $y_{i\ell} = 1$ if $\boldsymbol{x}_i \in \mathcal{C}_\ell$. The $\boldsymbol{x}_i$'s are assumed to be the input of an $\ell_2$-regularized Softmax classifier with regularization parameters $(\lambda_\ell)_{\ell \in [k]} \in \mathbb{R}^+$, which aims to determine the class-wise weight vectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k \in \mathbb{R}^p$ minimizing the loss[1], for some real-valued function $\phi : \mathbb{R} \to \mathbb{R}$:

$$\mathcal{L}(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k) = -\frac{1}{n} \sum_{i=1}^n \sum_{\ell=1}^k y_{i\ell} \log p_{i\ell} + \frac{1}{2} \sum_{\ell=1}^k \lambda_\ell \|\boldsymbol{w}_\ell\|^2 \quad \text{with} \quad p_{i\ell} = \frac{\phi(\boldsymbol{w}_\ell^\mathsf{T} \boldsymbol{x}_i)}{\sum_{j=1}^k \phi(\boldsymbol{w}_j^\mathsf{T} \boldsymbol{x}_i)}$$

In particular, the classical Softmax classifier corresponds to the case where $\phi(t) = e^t$ [GP17]. Cancelling the loss function gradient with respect to each weight vector $\boldsymbol{w}_\ell$ yields

$$\lambda_\ell \boldsymbol{w}_\ell = -\frac{1}{n} \sum_{i=1}^n \left( y_{i\ell} \psi(\boldsymbol{w}_\ell^\mathsf{T} \boldsymbol{x}_i) - \frac{\phi(\boldsymbol{w}_\ell^\mathsf{T} \boldsymbol{x}_i)}{\sum_{j=1}^k \phi(\boldsymbol{w}_j^\mathsf{T} \boldsymbol{x}_i)} \sum_{j=1}^k y_{ij} \psi(\boldsymbol{w}_j^\mathsf{T} \boldsymbol{x}_i) \right) \boldsymbol{x}_i, \quad \ell \in [k], \tag{5.1}$$

where $\psi \equiv \phi'/\phi$. Denoting $a_\ell = \boldsymbol{w}_\ell^\mathsf{T} \boldsymbol{x}$ for some data vector $\boldsymbol{x}$ with corresponding one-hot label vector $\boldsymbol{y}$. The expression of equation 5.1 can be obtained using the chain rule

[1] Biases are not introduced in the present formulation as their effect is known to be negligible in practice [KXR+19] and would only decrease the readability and accessibility of our results.

through the following derivations.

$$\frac{\partial \mathcal{L}}{\partial a_j} = -\sum_\ell y_\ell \frac{1}{p_\ell}\frac{\partial p_\ell}{\partial a_j} = -y_j\psi(a_j) + p_j\sum_\ell y_\ell\psi(a_\ell)$$

where we used the fact that $\frac{\partial p_i}{\partial a_i} = \frac{\phi'(a_i)}{\phi(a_i)}p_i(1-p_i)$ if $i=j$ and $\frac{\partial p_i}{\partial a_j} = -\frac{\phi'(a_j)}{\phi(a_j)}p_ip_j$ otherwise.

Under the concentration assumptions on the data matrix $\boldsymbol{X} \equiv [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathcal{M}_{p,n}$ and on $\psi$, and assuming $p, n$ large, we subsequently show that the vector $\boldsymbol{W} \equiv [\boldsymbol{w}_1^\intercal, \ldots, \boldsymbol{w}_k^\intercal]^\intercal \in \mathbb{R}^{pk}$ has a well defined behavior, which in turn allows us to accurately predict the performances of the Softmax classifier.

### 5.1.3 Assumptions & Main Results

We first characterize the data classes: if $y_{i\ell} = 1$, then $\boldsymbol{x}_i \in \mathbb{R}^p$ is a random vector with

$$\mathbb{E}[\boldsymbol{x}_i] \equiv \boldsymbol{\mu}_\ell, \quad \mathbb{E}_{\boldsymbol{x}_i}[\boldsymbol{x}_i\boldsymbol{x}_i^\intercal] - \boldsymbol{\mu}_\ell\boldsymbol{\mu}_\ell^\intercal \equiv \boldsymbol{\Sigma}_\ell.$$

The vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are further assumed to be independent and are such that $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathcal{M}_{p,n}$ satisfies a *concentration* property as stated by the following assumption.

**Assumption 12** (Concentrated data)**.** *Letting $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$, $\boldsymbol{X} \propto \mathcal{E}_2$.*

In the terms of Definition 7, Assumption 12 holds here for $s = (p, n)$ with $\mathbb{S} = \mathbb{N}^2$. In order to be able to transfer the concentration of $\boldsymbol{X}$ to the Softmax weights $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k$, a further condition is needed: the number of data $n$ must scale with the data dimension $p$, i.e., $\mathbb{S} = \{(p, n) \in \mathbb{N}^2, \kappa p \leq n \leq Kp\}$ for some $K > \kappa > 0$.[2] This is summarized by the request:

**Assumption 13** (Growth rate)**.** *$n = O(p)$ and $p = O(n)$.*

Concentrated vectors satisfy a host of interesting properties (the reader being referred to [Led05b] for a detailed account and to [LC18c] for their application to random matrix asymptotics, closer to the present work). We merely stress here one of these properties, of central importance to the present work, and which fundamentally justifies the appearance of Gaussian-like behaviors in large neural networks, even when the neural network input is far from Gaussian [KGC18, NBA+18].

**Theorem 5.1** (CLT for concentrated vectors [Kla07, FGP07])**.** *Let $\boldsymbol{X} \in \mathbb{R}^p$ be a random vector with $\mathbb{E}[\boldsymbol{X}] = 0$ and $\mathbb{E}[\boldsymbol{X}\boldsymbol{X}^\intercal] = I_p$, and $\sigma$ be the uniform measure on the sphere $\mathcal{S}^{p-1} \subset \mathbb{R}^p$ of radius 1. Then, if $\boldsymbol{X} \propto \mathcal{E}_2$, there exists two constants $C, c > 0$ and a set $\Theta \subset \mathcal{S}^{p-1}$ such that $\sigma(\Theta) \geq 1 - \sqrt{p}Ce^{-c\sqrt{p}}$ and $\forall \boldsymbol{\theta} \in \Theta$:*

$$\sup_{t\in\mathbb{R}} |\mathbb{P}(\boldsymbol{\theta}^\intercal\boldsymbol{X} \geq t) - G(t)| \leq p^{-1/4}$$

*for $G$ the cumulative distribution function of an $\mathcal{N}(0,1)$ random variable.*

[2] Formally, in the present setting, it is sufficient that $p \leq \frac{1}{\kappa}n$. However, to obtain simpler expressions, it is convenient to assume, in addition, that $n \leq Kp$.

#### 5.1.3.1 Concentration of the weights vector of the Softmax classifier

Having set the assumptions and main technical tools, we turn now to the characterization of the the statistical behavior of the Softmax classifier weights $\boldsymbol{W} = [\boldsymbol{w}_1^\intercal, \ldots, \boldsymbol{w}_k^\intercal]^\intercal \in \mathbb{R}^{pk}$, as a result, accesses the asymptotic performances of the classifier. To this end, our approach is to first write the implicit defining equation 5.1 of $\boldsymbol{W}$ under the formal form $\boldsymbol{W} = \Psi(\boldsymbol{W})$, for $\Psi : \mathbb{R}^{pk} \to \mathbb{R}^{pk}$ to characterize, and then to *transfer* the concentration of $\boldsymbol{X}$ (Assumption 12) to a concentration of $\boldsymbol{W}$.

For the concentration of $\boldsymbol{X}$ to propagate into $\boldsymbol{W}$ defined through the formal form $\boldsymbol{W} = \Psi(\boldsymbol{W})$, $\Psi$ is required to have contracting properties, which in turn will enforce structural conditions on the operator $\phi$ and on the regularizers $(\lambda_\ell)_{\ell \in [k]}$. Specifically $\Psi$ is requested to be $(1-\varepsilon)$-Lipschitz (for some $\varepsilon > 0$) so to ensure, thanks to the Banach fixed point theorem, the existence and uniqueness of $\boldsymbol{W} \in \mathbb{R}^{pk}$. However, being a *random map* depending on $\boldsymbol{X}$, $\Psi$ is only contracting under the (asymptotically highly probable) event $\mathcal{A}_X$ that the norm of $\boldsymbol{X}$ is not too large. Indeed, we can rewrite equation 5.1 as

$$\boldsymbol{\Lambda} \boldsymbol{W} = \frac{1}{n} \tilde{\boldsymbol{X}} f(\tilde{\boldsymbol{X}}^\intercal \boldsymbol{W}) \tag{5.2}$$

where $\boldsymbol{\Lambda} = \operatorname{diag}(\boldsymbol{\Lambda}_1, \ldots, \boldsymbol{\Lambda}_k) \otimes \boldsymbol{I}_p$, $f(\tilde{\boldsymbol{X}}^\intercal \boldsymbol{W}) \in \mathbb{R}^{kn}$ concatenates the elements $f_i(\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}) \in \mathbb{R}^k$, $\tilde{\boldsymbol{X}} = (\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_n) \in \mathcal{M}_{kp,kn}$, where we introduced $\tilde{\boldsymbol{x}}_i = \begin{pmatrix} \boldsymbol{x}_i & & \\ & \ddots & \\ & & \boldsymbol{x}_i \end{pmatrix} \in \mathcal{M}_{pk,k}$ $\forall i \in [n]$, and the functions:

$$\begin{array}{rccc} f_i : \mathbb{R}^k & \longrightarrow & \mathbb{R}^k & \\ \boldsymbol{v} & \longmapsto & \left[ \dfrac{\phi(\boldsymbol{v}_\ell)}{\sum_{j=1}^k \phi(\boldsymbol{v}_j)} \displaystyle\sum_{j=1}^k y_{ij} \psi(\boldsymbol{v}_j) - y_{i\ell} \psi(\boldsymbol{v}_\ell) \right]_{1 \le \ell \le k} & . \end{array}$$

Rewriting (5.5) as $\boldsymbol{W} = \Psi(\boldsymbol{W})$, one sees that $\Psi$ is contracting if $\|d\Psi|_{\boldsymbol{w}}\| \le 1 - \varepsilon$ for some $\varepsilon > 0$. Now, since $\|d\Psi|_{\boldsymbol{w}}\| \le \frac{1}{n} \|\boldsymbol{\Lambda}^{-1}\| \|\tilde{\boldsymbol{X}}\|^2 \|df|_{\tilde{\boldsymbol{X}}^\intercal \boldsymbol{w}}\|$ we first need $\|df\|_\infty = \sup_{\boldsymbol{v} \in \mathbb{R}^k} \|df|_{\boldsymbol{v}}\|$ to be bounded, that is ensured by:

**Assumption 14** (Regularity). $\|\frac{\phi'}{\phi}\|_\infty \le \infty$ *and* $\|\frac{\phi''}{\phi}\|_\infty \le \infty$.

Besides, we also need to be able to bound $\frac{1}{n}\|\tilde{\boldsymbol{X}}\|^2 = \frac{1}{n}\|\boldsymbol{X}\|^2$. The spectral norm being lower than the Frobenius norm (involved in Assumption 1 giving the concentration of $\tilde{\boldsymbol{X}}$), it is a 1-Lipschitz observation of $\tilde{\boldsymbol{X}}$ thus there exists two constants $C, c > 0$ (independent of $p, n$), such that:

$$\mathbb{P}\left( \left| \frac{1}{\sqrt{n}} \|\tilde{\boldsymbol{X}}\| - \frac{1}{\sqrt{n}} \mathbb{E}[\|\tilde{\boldsymbol{X}}\|] \right| \ge t \right) \le C e^{-cnt^2}. \tag{5.3}$$

This concentration inequality proves that the random variable $\frac{1}{\sqrt{n}}\|\tilde{\boldsymbol{X}}\|$ is almost deterministic and equal to $\frac{1}{\sqrt{n}}\mathbb{E}[\|\tilde{\boldsymbol{X}}\|]$. Bounding this last quantity necessities the following result:

$$\left| \mathbb{E}[\|\tilde{\boldsymbol{X}}\|] - \|\mathbb{E}[\tilde{\boldsymbol{X}}]\| \right| = O(\sqrt{p+n}), \tag{5.4}$$

It suffices to have a bound on $\|\mathbb{E}[\tilde{\boldsymbol{X}}]\|$ which comes from the following assumption:

**Assumption 15.** $\sup_{1\leq \ell \leq k} \|\boldsymbol{\mu}_\ell\| \leq O(1)$

This assumption implies that $\|\mathbb{E}[\tilde{\boldsymbol{X}}]\| = O(\sqrt{n})$, and we can deduce from equation 5.4 that $\mathbb{E}[\|\tilde{\boldsymbol{X}}\|] = O(\sqrt{n})$ (recall from Assumption 2 that $p = O(n)$).

**Remark 5.1.** *For classification problems, if $\forall a, b \in [k], a \neq b$, $\|\boldsymbol{\mu}_a - \boldsymbol{\mu}_b\| \gg 1$ then the classification becomes trivial in the large dimensional regime, a reasonable assumption then is $\|\boldsymbol{\mu}_a - \boldsymbol{\mu}_b\| \leq O(1)$. However, in cases where $\sup_{1\leq \ell \leq k} \|\boldsymbol{\mu}_\ell\|$ is of order $O(\sqrt{p})$, it is still possible to work with the data matrix $\tilde{\boldsymbol{X}} - \frac{1}{n}\tilde{\boldsymbol{X}}\mathbf{1}_n\mathbf{1}_n^\intercal$ that satisfies Assumption 15 but has the drawback that the columns are then dependent. However, this is not a big issue since this dependence is very small and it can be managed thanks to some technical consideration that we want to avoid here.*

Now, assuming

**Assumption 16.** $\frac{1}{n}\mathbb{E}[\|\boldsymbol{X}\|]^2 \|df\|_\infty \|\Lambda^{-1}\| < 1$.

and noting $\varepsilon = \frac{1}{2} - \frac{1}{2}\|\Lambda^{-1}\|\|f'\|_\infty \mathbb{E}[\|\tilde{\boldsymbol{X}}\|/\sqrt{n}]^2$, it can be deduced from equation 5.3 that the event:

$$\mathcal{A}_X = \left\{ \frac{1}{n} \left| \|\tilde{\boldsymbol{X}}\|^2 - \mathbb{E}[\|\tilde{\boldsymbol{X}}\|]^2 \right| \leq \frac{\varepsilon}{2\|\Lambda^{-1}\|\|df\|_\infty} \right\}$$

has a very high probability to happen (bigger than $1 - Ce^{-cn}$ for two constant $C, c > 0$) and it satisfies $\mathcal{A}_X \subset \{\|d\Psi\|_\infty \leq 1 - \varepsilon\}$. As a consequence, our fixed point $\boldsymbol{W}$ is uniquely determined under the event $\mathcal{A}_X$ that appears in the following theorem which provides the concentration of the weights vector $\boldsymbol{W}$ as stated by the following Theorem.

**Theorem 5.2** (Concentration of $\boldsymbol{W}$ **[?]**)**.** *Under the previous Assumptions, there exist two constants $C, c > 0$ and an event $\mathcal{A}_X$ with $\mathbb{P}(\mathcal{A}_X) > 1 - Ce^{-cn}$ such that*[3]

$$(\boldsymbol{W} \mid \mathcal{A}_X) \propto \mathcal{E}_2\left(\sqrt{\log n / n}\right).$$

**Remark 5.2.** *The concentration of the random vector $\boldsymbol{W}$ is far from being trivial because $\boldsymbol{W}$ is not explicitly written as a Lipschitz transformation of $\boldsymbol{X}$ and additional tools are necessary to prove the concentration of $\boldsymbol{W}$. The complete proof will be provided in an extended version of this paper.*

Since their observable diameter ($\sqrt{\log n / n}$) vanishes at large $n$, it therefore entails from Theorem 5.2 that *the random weights vector $\boldsymbol{W}$ tend to be deterministic as $p, n$ grow large*. In particular, $\boldsymbol{W}$ concentrates around its expectation which can be estimated in terms of the data statistics as we will see subsequently. Indeed, the subsequent result further characterizes the first and second order statistics of $\boldsymbol{W}$ at large $p, n$.

**Theorem 5.3** (Asymptotic statistics of $\boldsymbol{W}$)**.** *Define the statistics*

$$\boldsymbol{m}_W \equiv \mathbb{E}[\boldsymbol{W}], \quad \boldsymbol{C}_W \equiv [\boldsymbol{W}\boldsymbol{W}^\intercal] - \boldsymbol{m}_W\boldsymbol{m}_W^\intercal.$$

*Then, under Assumptions 12 and 13 and additional assumptions on $\phi$ and $(\lambda_\ell)_{\ell\in[k]}$ (Assumptions 14, 15 and 16), there exists a deterministic mapping $\mathcal{F}_{\boldsymbol{\mu},\boldsymbol{\Sigma}} = \mathcal{F}_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_k)$ :*

[3]Formally, the random vector $\boldsymbol{W}$ is a measurable mapping $\Omega \to \mathbb{R}^{pk}$, where $(\Omega, \mathcal{F}, \mathbb{P})$ is the underlying probability space. If $\mathbb{P}(\mathcal{A}) > 0$, for $\mathcal{A} \in \mathcal{F}$, the random vector $(\boldsymbol{W} \mid \mathcal{A})$ is the measurable mapping $\mathcal{A} \to \mathbb{R}^{pk}$ such that, $\forall \omega \in \mathcal{A}$, $(\boldsymbol{W} \mid \mathcal{A})(\omega) = \boldsymbol{W}(\omega)$. The statistics of $(\boldsymbol{W} \mid \mathcal{A})$ are then computed in the probability space $(\mathcal{A}, \mathcal{F} \wedge \mathcal{A}, \mathbb{P}_\mathcal{A})$, where $\mathcal{F} \wedge \mathcal{A} = \{B \cap \mathcal{A}, B \in \mathcal{F}\}$ and $\forall B \in \mathcal{F}$, $\mathbb{P}_\mathcal{A}(B) = \mathbb{P}(B)/\mathbb{P}(\mathcal{A})$.

$\mathbb{R}^{pk} \times \mathcal{M}_{pk} \longrightarrow \mathbb{R}^{pk} \times \mathcal{M}_{pk}$ *depending only on the statistics* $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$ *and* $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_k$ *of* $\boldsymbol{X}$*, such that the equation*

$$(\boldsymbol{m}, \boldsymbol{C}) = \mathcal{F}_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{m}, \boldsymbol{C}) \quad \textit{with} \quad \boldsymbol{m} \in \mathbb{R}^{pk},\ \boldsymbol{C} \in \mathcal{M}_{pk}$$

*admits a unique solution* $(\bar{\boldsymbol{m}}_W, \bar{\boldsymbol{C}}_W)$*. Besides,*

$$\|\bar{\boldsymbol{m}}_W - \boldsymbol{m}_W\| \leq O\left(\sqrt{\log n / n}\right) \quad \textit{and} \quad \|\bar{\boldsymbol{C}}_W - \boldsymbol{C}_W\|_* \leq O\left(\sqrt{\log n / n}\right).$$

The central outcome of Theorem 5.3 is that, under the data concentration Assumption 12, the behavior of the Softmax classifier *only depends on the class-wise means and covariances of the input data*. This arises as a direct consequence of the Lipschitz character of the Softmax classifier which preserves concentration (by the stability result of Remark **??**), and of the presence of a *projection of the parameter vectors* $\boldsymbol{w}_\ell$ *onto the concentrated data* $\boldsymbol{x}_i$ at the core of the optimization formulation: according to Theorem C.1, these projections induce an asymptotic Gaussian behavior with mean and variance depending *only* on the first statistics of the data and the weights vector $\boldsymbol{W}$.

We now provide the main ingredients to obtain the result of Theorem 5.3, which mainly unfolds from two essential steps: (i) the control of the statistical dependencies between $\boldsymbol{W}$ and $\boldsymbol{X}$, and (ii) the estimation of the statistics $\boldsymbol{m}_W$ and $\boldsymbol{C}_W$ of the weight vector $\boldsymbol{W}$. We start by reformulating (5.1) in the compact and convenient form as we saw previously

$$\boldsymbol{\Lambda} \boldsymbol{W} = \frac{1}{n} \sum_{i=1}^{n} \tilde{\boldsymbol{x}}_i f_i(\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}) \tag{5.5}$$

**(i) Control of the statistical dependencies.** Applying the expectation operator both sides to (5.5), the main technical difficulty arises from the evaluation of $\mathbb{E}[\tilde{\boldsymbol{x}}_i f_i(\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W})]$ due to the elaborate dependencies between the weight vector $\boldsymbol{W}$ and the data $\tilde{\boldsymbol{x}}_i$. Note that $\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}$ *a priori* has no reason of being Gaussian (even in the limit) and the performances of the Softmax classifier may depend on high order statistics of $\boldsymbol{X}$. These statistical dependencies are dealt with by introducing a mapping $\boldsymbol{W}_{-i} : [0,1] \to \mathbb{R}^{pk}$, defined for $i \in [n]$, as the unique solution to:

$$\forall t \in [0,1] \ :\ \boldsymbol{\Lambda} \boldsymbol{W}_{-i}(t) = \frac{1}{n} \sum_{j \neq i} \tilde{\boldsymbol{x}}_j f_j(\tilde{\boldsymbol{x}}_j^\intercal \boldsymbol{W}_{-i}(t)) + \frac{1}{n} t \tilde{\boldsymbol{x}}_i f_i\left(\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}_{-i}(t)\right). \tag{5.6}$$

This mapping can be seen as a path between the weights vector $\boldsymbol{W} = \boldsymbol{W}_{-i}(1)$ of the Softmax classifier and $\boldsymbol{W}_{-i}(0)$ which is completely independent of $\tilde{\boldsymbol{x}}_i$ and which will be simply denoted $\boldsymbol{W}_{-i}$.

Using the inverse function theorem, the mapping $t \mapsto \boldsymbol{W}_{-i}(t)$ is differentiable, we then deduce the following central close form formula:

$$\boldsymbol{W}_{-i}'(t) = \frac{1}{n} \boldsymbol{Q}_{-i}(t) \tilde{\boldsymbol{x}}_i \chi_i'(t) \quad \text{with} \quad \boldsymbol{Q}_{-i}(t) \equiv \left( \boldsymbol{\Lambda} - \frac{1}{n} \tilde{\boldsymbol{X}}_{-i} \boldsymbol{D}^{(i)}(t) \tilde{\boldsymbol{X}}_{-i}^\intercal \right)^{-1} \in \mathcal{M}_{kp}, \tag{5.7}$$

where $\chi_i(t) \equiv t f_i\left(\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}_{-i}(t)\right))$, $\boldsymbol{D}_j^{(i)}(t) \equiv d f_j \tilde{\boldsymbol{x}}_j^\intercal \boldsymbol{W}_{-i}(t) \in \mathcal{M}_k$, $\boldsymbol{D}^{(i)}(t) \in \mathcal{M}_{kn}$ is a block-diagonal matrix with block-diagonal matrices $\boldsymbol{D}_j^{(i)}(t) \in \mathcal{M}_k$ for $j \in [n]$, and finally

$\tilde{\boldsymbol{X}}_{-i} \equiv (\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_{i-1}, 0, , \tilde{\boldsymbol{x}}_{i+1}, \ldots, \tilde{\boldsymbol{x}}_n) \in \mathcal{M}_{pk,kn}$ ($\tilde{\boldsymbol{X}}_{-i}$ is independent of $\tilde{\boldsymbol{x}}_i$).

Relying on concentration of measure arguments [LC20], the random vector $\boldsymbol{Q}_{-i}(t)\tilde{\boldsymbol{x}}_i$ is almost constant in terms of $t$ and thus almost equal to $\boldsymbol{Q}_{-i}(0)\tilde{\boldsymbol{x}}_i$. Moreover, the fact that $\boldsymbol{Q}_{-i}(0)$ (also simply denoted $\boldsymbol{Q}_{-i}$) is independent of $\tilde{\boldsymbol{x}}_i$ allows us to integrate the identity (5.7) to obtain the core result of the analysis of the Softmax weights relating $\boldsymbol{W}$ and $\boldsymbol{W}_{-i}$.

**Theorem 5.4.** *Under the event $\mathcal{A}_X$, we have $\boldsymbol{W} - \boldsymbol{W}_{-i} + \frac{1}{n}\boldsymbol{Q}_{-i}\tilde{\boldsymbol{x}}_i f_i(\tilde{\boldsymbol{x}}_i^\mathsf{T}\boldsymbol{W}) \propto \mathcal{E}_2\left(\frac{1}{n}\right)$. In particular, we have the bound:*

$$\mathbb{E}\left[\left\|\boldsymbol{W} - \boldsymbol{W}_{-i} + \frac{1}{n}\boldsymbol{Q}_{-i}\tilde{\boldsymbol{x}}_i f_i(\tilde{\boldsymbol{x}}_i^\mathsf{T}\boldsymbol{W})\right\|\right] \leq O\left(\frac{\sqrt{\log n}}{n}\right). \tag{5.8}$$

Specifically, we have the following concentration inequality, for some constants $C, c > 0$:

$$\forall t > 0 \,:\, \mathbb{P}\left(\left\|\tilde{\boldsymbol{x}}_i^\mathsf{T}\boldsymbol{W} - \tilde{\boldsymbol{x}}_i^\mathsf{T}\boldsymbol{W}_{-i} + \frac{1}{n}\tilde{\boldsymbol{x}}_i^\mathsf{T}\boldsymbol{Q}_{-i}\tilde{\boldsymbol{x}}_i f_i(\tilde{\boldsymbol{x}}_i^\mathsf{T}\boldsymbol{W})\right\| \geq t \mid \mathcal{A}_X\right) \leq Ce^{-cnt^2/\log n}. \tag{5.9}$$

**(ii) Estimation of the mean and covariance of the Softmax weights.** By breaking the statistical dependencies of the problem through $\boldsymbol{W}_{-i}$, we may now access and estimate the statistics $\boldsymbol{m}_W$ and $\boldsymbol{C}_W$. This precisely comes from a deterministic approximation of the quadratic form $\frac{1}{n}\tilde{\boldsymbol{x}}_i^\mathsf{T}\boldsymbol{Q}_{-i}\tilde{\boldsymbol{x}}_i$ in (5.9) in the large $n, p$ limit, a result inspired by [LC20]:

**Proposition 5.1.** *For any $\ell \in [k]$, let $n_\ell$ be the number of columns of $\boldsymbol{X}$ in class $\mathcal{C}_\ell$ and, for any block diagonal matrix $\boldsymbol{\Delta} = \mathrm{diag}(\boldsymbol{\Delta}_\ell)_{1\leq\ell\leq k} \in \mathcal{M}_{k^2}$ ($\forall \ell \in [k], \boldsymbol{\Delta}_\ell \in \mathcal{M}_k$), then*

$$\boldsymbol{Q}(t) \leftrightarrow \bar{\boldsymbol{Q}}(\boldsymbol{\Delta}) \equiv \left(\boldsymbol{\Lambda} - \sum_{a=1}^{k}\frac{n_a}{n}\Gamma_a(\boldsymbol{\Delta}_a)\otimes\boldsymbol{S}_a\right)^{-1} = \begin{pmatrix} \bar{\boldsymbol{Q}}_{1,1}(\boldsymbol{\Delta}) & \ldots & \bar{\boldsymbol{Q}}_{1,k}(\boldsymbol{\Delta}) \\ \vdots & \ddots & \vdots \\ \bar{\boldsymbol{Q}}_{k,1}(\boldsymbol{\Delta}) & \ldots & \bar{\boldsymbol{Q}}_{k,k}(\boldsymbol{\Delta}) \end{pmatrix} \in \mathcal{M}_{kp}$$

*where $\Gamma_\ell(\boldsymbol{\Delta}_a) = \mathbf{E}\left[(\boldsymbol{I}_k - \boldsymbol{D}_j^{-i}(0)\boldsymbol{\Delta}_a)^{-1}\boldsymbol{D}_j^{-i}(0)\right]$ for $\boldsymbol{x}_j$ in class $\mathcal{C}_\ell$ and $\boldsymbol{S}_\ell = \mathbb{E}[\boldsymbol{x}_j\boldsymbol{x}_j^\mathsf{T}]$. Specifically, the fixed point equation*

$$\boldsymbol{\Delta}_\ell = \left[\frac{1}{n}\mathrm{Tr}\left(\boldsymbol{S}_\ell\bar{\boldsymbol{Q}}_{a,b}(\boldsymbol{\Delta})\right)\right]_{1\leq a,b\leq k}$$

*admits a unique solution $\boldsymbol{\Delta} \in \mathcal{M}_{k^2}$ that satisfies, for any $\boldsymbol{x}_i$ is in class $\ell \in [k]$,*

$$\forall t > 0 \,:\, \mathbb{P}\left(\left\|\frac{1}{n}\tilde{\boldsymbol{x}}_i^\mathsf{T}\boldsymbol{Q}_{-i}\tilde{\boldsymbol{x}}_i - \boldsymbol{\Delta}_\ell\right\| \geq t \mid \mathcal{A}_X\right) \leq Ce^{-cnt^2/\log n} \quad \textit{for some constants } C, c > 0.$$

From this result, using the identity (5.9), we then obtain an estimation for $\tilde{\boldsymbol{x}}_i^\mathsf{T}\boldsymbol{W}$:

**Proposition 5.2.** *For any $\boldsymbol{v} \in \mathbb{R}^k$, there exists a unique point $g_i(\boldsymbol{v}) \in \mathbb{R}^k$ satisfying:*

$$g_i(\boldsymbol{v}) = \boldsymbol{v} - \boldsymbol{\Delta}_i f_i(g_i(\boldsymbol{v})),$$

*and, for some constants $C, c > 0$,*

$$\mathbb{P}\left(\left\|\tilde{\boldsymbol{x}}_i^\mathsf{T}\boldsymbol{W} - g_i(\tilde{\boldsymbol{x}}_i^\mathsf{T}\boldsymbol{W}_{-i})\right\| \geq t \mid \mathcal{A}_X\right) \leq Ce^{-cnt^2/\log n}.$$

Therefor, letting $h_i = f_i \circ g_i$, by Hölder's inequality [Fin92], since $\tilde{\boldsymbol{x}}_i$ is concentrated,

$$\left\| \boldsymbol{m}_W - \frac{1}{n}\sum_{i=1}^n \boldsymbol{\Lambda}^{-1}\mathbb{E}[\tilde{\boldsymbol{x}}_i h_i(\tilde{\boldsymbol{x}}_i^\mathsf{T}\boldsymbol{W}_{-i})] \right\| = O\left(\sqrt{\frac{\log n}{n}}\right)$$
$$\left\| \boldsymbol{C}_W - \frac{1}{n^2}\sum_{i=1}^n \boldsymbol{\Lambda}^{-1}\mathbb{E}[\tilde{\boldsymbol{x}}_i h_i(\tilde{\boldsymbol{x}}_i^\mathsf{T}\boldsymbol{W}_{-i}) h_i(\tilde{\boldsymbol{x}}_i^\mathsf{T}\boldsymbol{W}_{-i})^\mathsf{T}\tilde{\boldsymbol{x}}_i^\mathsf{T}]\boldsymbol{\Lambda}^{-1} \right\|_* = O\left(\sqrt{\frac{\log n}{n}}\right)$$

Knowing from Theorem C.1 that $\tilde{\boldsymbol{x}}_i^\mathsf{T}\boldsymbol{W}_{-i}$ is asymptotically Gaussian, $\mathbb{E}[\tilde{\boldsymbol{x}}_i h_i(\tilde{\boldsymbol{x}}_i^\mathsf{T}\boldsymbol{W}_{-i})]$ and $\mathbb{E}[\tilde{\boldsymbol{x}}_i h_i(\tilde{\boldsymbol{x}}_i^\mathsf{T}\boldsymbol{W}_{-i}) h_i(\tilde{\boldsymbol{x}}_i^\mathsf{T}\boldsymbol{W}_{-i})^\mathsf{T}\tilde{\boldsymbol{x}}_i^\mathsf{T}]$ can be explicitly evaluated (for instance using Stein's Lemma [LN08]), and only depend on the statistical means and covariances of $\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_n$ and of $\boldsymbol{W}_{-i}$ (which has the same statistics as $\boldsymbol{W}$). Their exact expressions are provided in the appendix in Section C.3.1. Finally, let us introduce the $2k$ functions $m_1, \ldots, m_k : \mathbb{R}^{kp} \times \mathcal{M}_{kp} \to \mathbb{R}^{kp}$ and $c_1, \ldots, c_k : \mathbb{R}^{kp} \times \mathcal{M}_{kp} \to \mathcal{M}_{kp}$ defined, $\forall i \in [n]$, by

$$m_{k(i)}(\boldsymbol{m}_W, \boldsymbol{C}_W) = \boldsymbol{\Lambda}^{-1}\mathbb{E}[\tilde{\boldsymbol{x}}_i h_i(\tilde{\boldsymbol{x}}_i^\mathsf{T}\boldsymbol{W}_{-i})]$$
$$c_{k(i)}(\boldsymbol{m}_W, \boldsymbol{C}_W) = \frac{1}{n}\boldsymbol{\Lambda}^{-1}\mathbb{E}[\tilde{\boldsymbol{x}}_i h_i(\tilde{\boldsymbol{x}}_i^\mathsf{T}\boldsymbol{W}_{-i}) h_i(\tilde{\boldsymbol{x}}_i^\mathsf{T}\boldsymbol{W}_{-i})^\mathsf{T}\tilde{\boldsymbol{x}}_i^\mathsf{T}]\boldsymbol{\Lambda}^{-1},$$

where $k(i)$ denotes the class of $\tilde{\boldsymbol{x}}_i$ and $\boldsymbol{m}_W$ and $\boldsymbol{C}_W$ are respectively the mean and covariance of $\boldsymbol{W}$. The mappings $(m_\ell)_{1\le\ell\le k}$ and $(\sigma_\ell)_{1\le\ell\le k}$ are uniquely determined by the means $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_n$ and the covariances $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_n$. In particular, the deterministic pair $(\bar{\boldsymbol{m}}_W, \bar{\boldsymbol{C}}_W)$, defined as the unique solution of

$$\bar{\boldsymbol{m}}_W = \sum_{\ell=1}^k \frac{n_\ell}{n} m_\ell(\bar{\boldsymbol{m}}_W, \bar{\boldsymbol{C}}_W) \quad \text{and} \quad \bar{\boldsymbol{C}}_W = \sum_{\ell=1}^k \frac{n_\ell}{n} c_\ell(\bar{\boldsymbol{m}}_W, \bar{\boldsymbol{C}}_W), \tag{5.10}$$

is a good approximation for $(\boldsymbol{m}_W, \boldsymbol{C}_W)$ as stated in Theorem 5.3.

Once the Softmax classifier is trained, the probability for a new datum $\boldsymbol{x}$ to belong to class $\ell \in [k]$ is explicitly given by $p_\ell(\boldsymbol{x}) = \phi(\boldsymbol{w}_\ell^\mathsf{T}\boldsymbol{x}) / \sum_{j\in[k]} \phi(\boldsymbol{w}_j^\mathsf{T}\boldsymbol{x})$. As a consequence of Theorem C.1, $\boldsymbol{w}_\ell^\mathsf{T}\boldsymbol{x}$ has a high probability to be Gaussian (since $\boldsymbol{x}$ is concentrated and $\boldsymbol{w}_\ell$ has a deterministic behavior). The performances of the Softmax classifier are therefore theoretically tractable.

**Corollary 5.1** (Generalization performance of the Softmax classifier)**.** *For $\ell \in [k]$, there exists $\bar{\kappa}^\ell \in \mathbb{R}^{k-1}$ and $\bar{\boldsymbol{K}}^\ell \in \mathcal{M}_{k-1}$ both depending only on $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_k$ such that the misclassification error $E_t(\boldsymbol{x} \in \mathcal{C}_\ell)$ of a new datum $\boldsymbol{x}$ belonging to class $\ell \in [k]$ defined as $E_t(\boldsymbol{x} \in \mathcal{C}_\ell) \equiv 1 - \mathbb{P}(\forall j \in [k] \setminus \{\ell\} : p_\ell(\boldsymbol{x}) \ge p_j(\boldsymbol{x}))$ is asymptotically approximated as*

$$E_t(\boldsymbol{x} \in \mathcal{C}_\ell) - 1 + \mathbb{P}(\boldsymbol{Z}_\ell \in \mathbb{R}_+^{k-1}) \xrightarrow{a.s.} 0 \quad \textit{with} \quad \boldsymbol{Z}_\ell \sim \mathcal{N}(\bar{\kappa}^\ell, \bar{\boldsymbol{K}}^\ell). \tag{5.11}$$

In essence, Corollary 5.1 states that the generalization performance of the Softmax classifier reduces to the cumulative distribution of a low-dimensional Gaussian vector, the mean and covariance of which only depend on the class-wise means and covariances of the input data. This demonstrates the remarkable *universality* property of the Softmax classifier with respect to the data distribution, which we recall is only requested to satisfy a very loose concentration behavior (Assumption 12). The exact expressions of $\bar{\kappa}^\ell$ and $\bar{\boldsymbol{K}}^\ell$, along with a justification of the corollary, are given subsequently.

Let us decompose the matrix $S_W = \mathbb{E}[WW^T]$ followingly:

$$S_W = \left(\begin{array}{c|c|c} S_W^{1,1} & \dots & S_W^{1,k} \\ \hline \vdots & & \vdots \\ \hline S_W^{k,1} & \dots & S_W^{k,k} \end{array}\right) \in \mathcal{M}_{kp},$$

where for all $a, b \in [k]$, $S_W^{a,b} \in \mathcal{M}_p$, so that we can introduce the low-dimensional random vector $z \sim \mathcal{N}(^\ell, K^\ell)$ with:

$$^\ell \equiv \mu_\ell^\intercal m_W \qquad \text{and} \qquad K^\ell \equiv (\mathrm{Tr}((\Sigma_\ell + \mu_\ell \mu_\ell^\intercal) S_W^{a,b}))_{1 \le a,b \le k} - \mu_\ell^\intercal m_W m_W^\intercal \mu_\ell. \tag{5.12}$$

The expected misclassification error $E_t(x \in \mathcal{C}_\ell)$ on a test data $x$ belonging to class $\ell \in [k]$ expresses followingly:

$$E_t(x \in \mathcal{C}_\ell) = 1 - \mathbb{P}(\forall j \in [k] \setminus \{\ell\} : p_\ell(x) \ge p_j(x)) \quad \text{where} \quad p_j(x) = \frac{\phi(w_j^\intercal x)}{\sum_{h=1}^k \phi(w_h^\intercal x)}.$$

Since $x$ and $w$ are both concentrated and independent Theorem C.1 allows us to assume that for all $j \in [k]$, $\tilde{x}^\intercal W = (w_j^\intercal x)_{1 \le j \le k} \sim \mathcal{N}(^\ell, K^\ell)$ (the objects $^\ell$ and $K^\ell$ where introduced in equation C.10). To simplify the problem, we are going to make the additional hypothesis that $\phi$ is increasing since in that case

$$\forall j \in [k] \setminus \{\ell\} : p_\ell(x) \ge p_j(x) \iff w_\ell^\intercal x \mathbf{1}_k - \tilde{x}^\intercal W \in \mathbb{R}_+^k$$

Since the $\ell^{\text{th}}$ coordinate of $\tilde{x}^\intercal W$ is, by definition, equal to $w_\ell^\intercal x$, only the $k-1$ other are interesting. Let us then introduce the Gaussian vector $Z_\ell \in \mathbb{R}^{k-1}$ defined for all $j \in [k] \setminus \ell$ as: $[Z_\ell]_j = (w_\ell - w_j)^\intercal x$. Such a vector $Z_\ell$ has then the mean $^- \equiv^\ell P^\ell$ and the covariance $\bar{K} \equiv P K^\ell P^T$ with:

$$P = \mathbf{1}_{k-1} e_\ell^\intercal - I_k^{-\ell} \in \mathcal{M}_{k-1,k}$$

where $\mathbf{1}_{k-1} \in \mathbb{R}^{k-1}$ is a vector full of 1, $e_\ell$ is the $\ell^{\text{th}}$ vector of the canonical basis of $\mathbb{R}^k$ (full of zeros with a one in the $\ell^{\text{th}}$ coordinate) and $I_k^{-\ell}$ is the identity matrix of $\mathcal{M}_k$ deprived of the $\ell^{\text{th}}$ row. We can then express

$$E_t(x \in \mathcal{C}_\ell) = 1 - \mathbb{P}(Z_\ell \in \mathbb{R}_+^{k-1})$$

#### 5.1.3.2 Experimental validation

This section provides an experimental setup to support our theoretical findings. The input data $X = [x_1, \dots, x_n]$ are independent Resnet15[4] representations of size $p = 512$ [SIVA17] of images generated by the BigGAN generative adversarial network model [BDS18]: as such, being the composition of two neural networks (BigGAN and Resnet15) applied to random standard Gaussian noise (as per the BigGAN model), $X$ is concentrated by construction and satisfies Assumption 12, as requested (see [SLTC20] for a detailed analysis of the Lipschitz properties of these networks). Under this setting, Figure B.9 depicts the learned Softmax weights against their expected large $n, p$ asymptotics as per Theorem 5.3. Despite the finite $p, n$ setting of the simulation, a perfect match is observed between the learned weights and the theoretical predictions. Further experiments are performed on *real images* from the ImageNet dataset [DDS+09], which again show a perfect match between theory and practice, thereby strongly suggesting that the

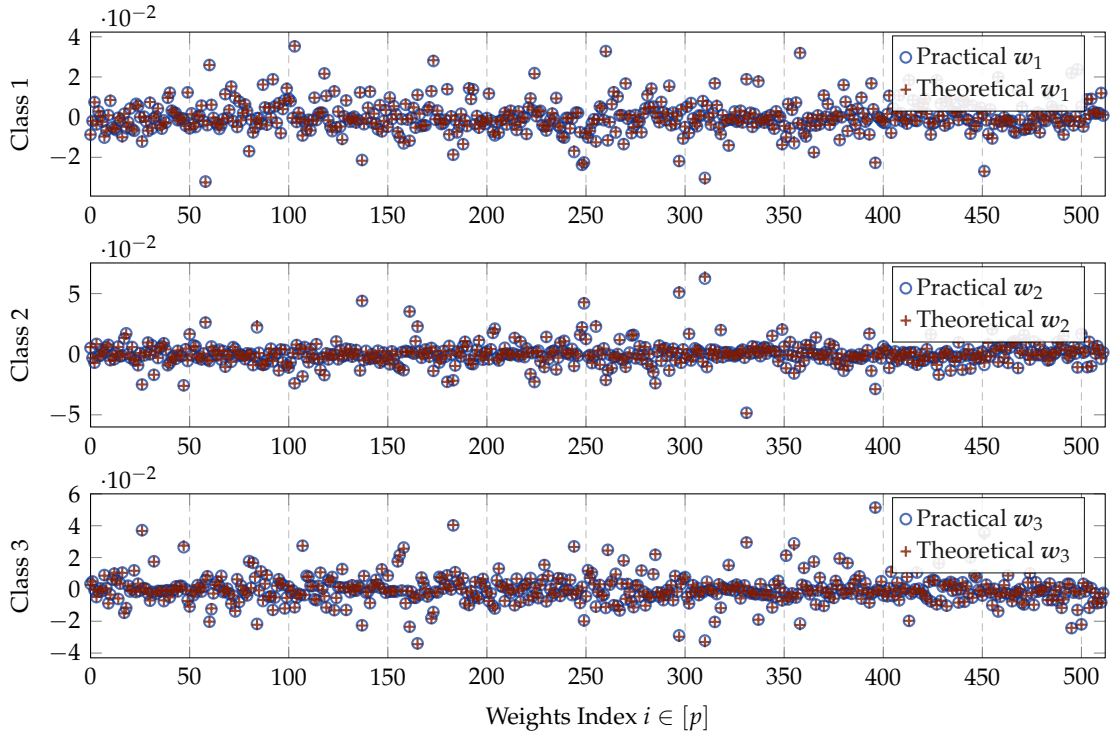



Figure 5.2: Learned weights (in blue circles) versus their theoretical estimations (in red crosses) as per Theorem 5.3. The used data are Resnet18 [SIVA17] representations ($p = 512$) of images generated by the BigGAN model [BDS18] which are concentrated vectors [SLTC20]. We considered $k = 3$ classes which are: *hamburger*, *mushroom*, *pizza*. $n = 3000$ and the regularization constants $\lambda_1, \lambda_2, \lambda_3 = 1.5$. The data are normalized such that their norm is $0.1 \cdot \sqrt{p}$ to ensure $\mathcal{A}_X$.

conclusions of Theorem 5.3 extend to real data.

Figure 5.5 next displays the class-wise scores of a practical Softmax classifier on an independent test set against their estimated statistics according to Corollary 5.1. An almost perfect match is again observed between empirical values and theoretical statistics. Further experiments report similar outputs for real ImageNet data. We importantly stress that, as per Corollary 5.1, the theoretical estimates were obtained using *only the empirical class-wise means and covariances* of the input data. Figure 5.5 thus confirms the theoretically predicted universality of the Softmax classifier. A Python implementation is attached for reproductivity of these experiments.

We provide further experiments using real images from the Imagenet dataset [DDS+09]. Figure B.9 depicts the learned Softmax weights against their expected large $p, n$ asymptotics as predicted by Theorem 3.2. As for GAN generated images, we observe a perfect match between the learned weights and the theoretical predictions. An almost perfect match is also observed for the scores (between the practical scores and their theoretical counterparts) as depicted in Figure 5.5, thereby strongly suggesting that the conclusions of Theorem 5.3 extend to real data.

[4]We used its Pytorch implementation [PGM+19] pre-trained on the Imagenet dataset [DDS+09].

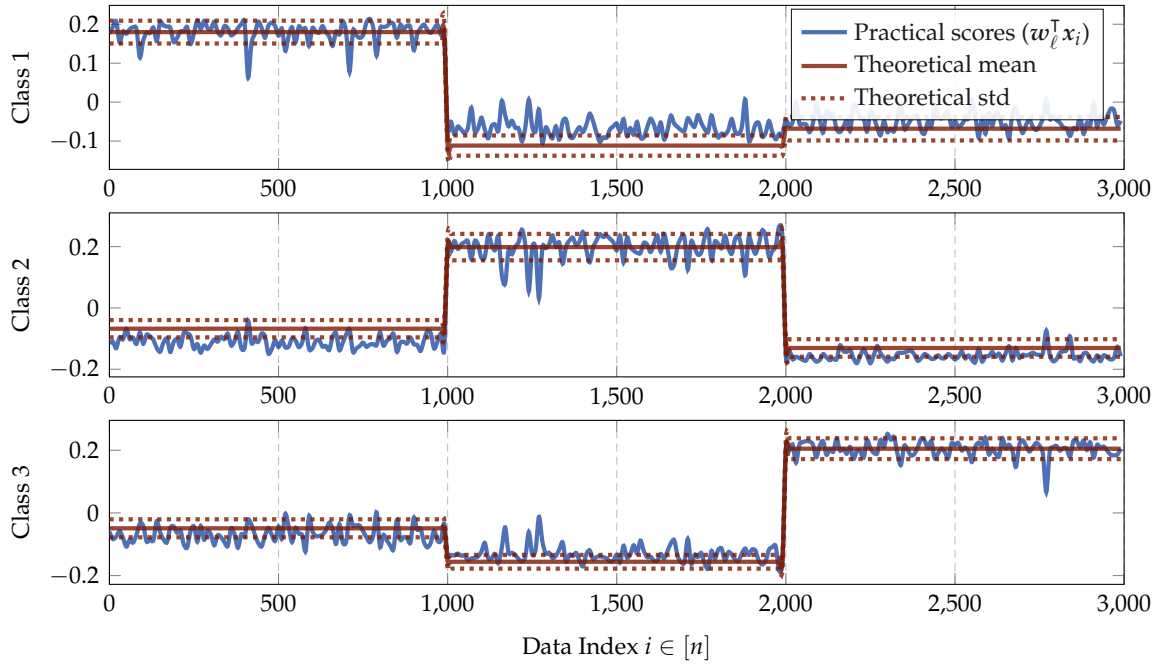



Figure 5.3: Scores (in blue) versus their theoretical estimations (in red) as per Corollary 5.1, with the theoretical means (through $\bar{\kappa}^\ell$) and standard deviations (through $\bar{\boldsymbol{K}}^\ell$), on a test set independent from the training set. The used data are Resnet18 [SIVA17] representations ($p = 512$) of images generated by the BigGAN model [BDS18] which are concentrated vectors [SLTC20]. We considered $k = 3$ classes which are: *hamburger*, *mushroom*, *pizza*. $n = 3000$ and the regularization constants $\lambda_1, \lambda_2, \lambda_3 = 1.5$. The data are normalized such that their norm is $0.1 \cdot \sqrt{p}$ to ensure $\mathcal{A}_X$.

### 5.1.4 Central Contribution

As a consequence of Corollary 5.1, we have demonstrated that, even though the Softmax classifier has a non-linear nature, a property supposedly useful to extract "deep" non-linear features, for $n, p$ rather large, the input data are in fact treated as if they were distributed as a mere Gaussian mixture model. This large dimensional universality phenomenon fundamentally revisits the conventional insights acquired along the years on non-linear classification methods.

As an aftermath, the Softmax classifier being optimal for Gaussian mixture inputs with common covariance, our study is strongly suggestive of the optimality of Softmax as the last layer of a deep neural network classifier.

Yet, the present study assumes a clear-cut separation between a back-end network training isolated from the front-end Softmax layer (as depicted in Figure 5.1). A thorough validation of the equivalence between full network training and this divided approach is a necessary final step to confirm the claimed optimality and anticipate the performances of Softmax classification.

On the other hand, our analysis leverages on the contracting fixed point approach by adding regularization parameters to the the weight vectors. Another approach would be to consider different regularization techniques such as the Dropout operation in neural

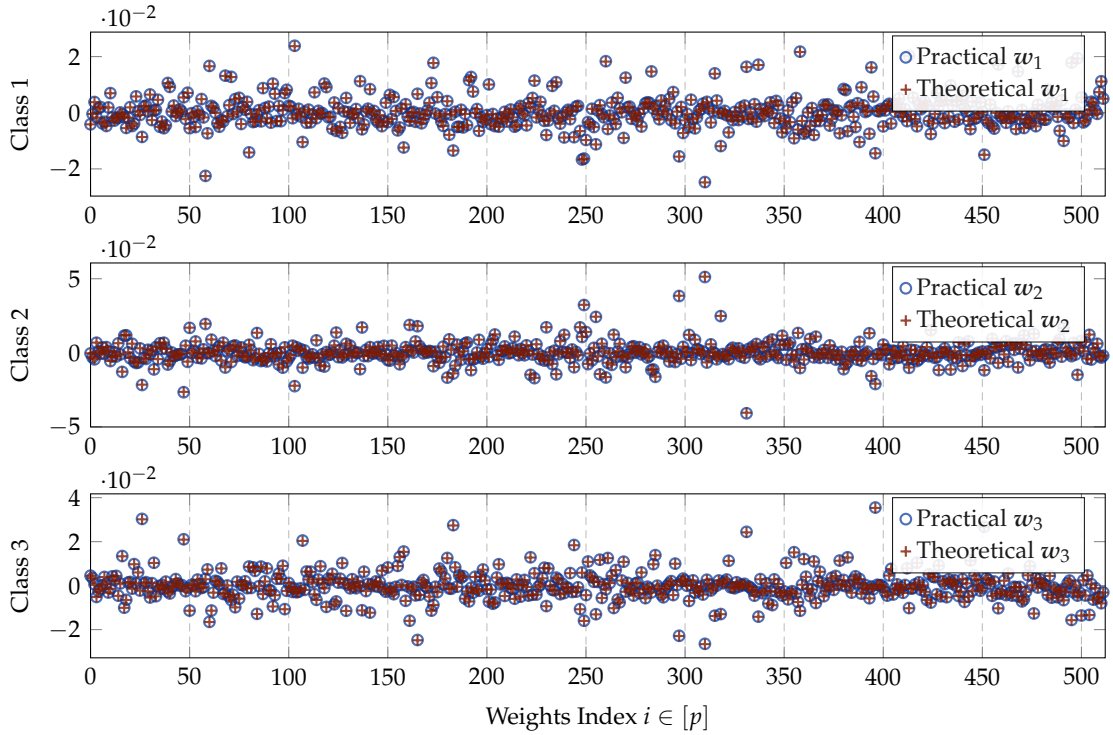



Figure 5.4: Learned weights (in blue circles) versus their theoretical estimations (in red crosses) as per Theorem 3.2 of the Main Paper. The used data are Resnet18 [SIVA17] representations ($p = 512$) of real images from the Imagenet dataset [DDS+09]. We considered $k = 3$ classes which are: *hamburger*, *mushroom*, *pizza*. $n = 3000$ and the regularization constants $\lambda_1, \lambda_2, \lambda_3 = 1.5$. The data are normalized such that their norm is $0.1 \cdot \sqrt{p}$ to ensure $\mathcal{A}_X$.

networks. In the next section, we will provide an analysis of such layer on a one hidden-layer network.

## 5.2 A random matrix analysis of Dropout layers

This section is based on the following work:

**(C5) MEA. Seddik**, R. Couillet, M. Tamaazousti, "*A Random Matrix Analysis of Learning with α-Dropout*", The art of learning with missing values ICML workshop (ICML'20), Online, 2020.

### 5.2.1 Motivation

Many practical datasets contain samples with missing features which impair the behavior of machine learning models. Improperly handling these missing values results in biased predictions. While various imputation techniques exist in the literature, such as imputation of the global mean, the simplest is *zero imputation*, by which the missing features are simply replaced by zeros. Neural networks have been notably shown to be affected when trained on zero-imputed data [HLM15, ŚST+18, YLK+19].

In neural networks, zero imputation can be seen as applying a Dropout [SHK+14] operation to the input data features, or equivalently as applying a binary mask entry-wise

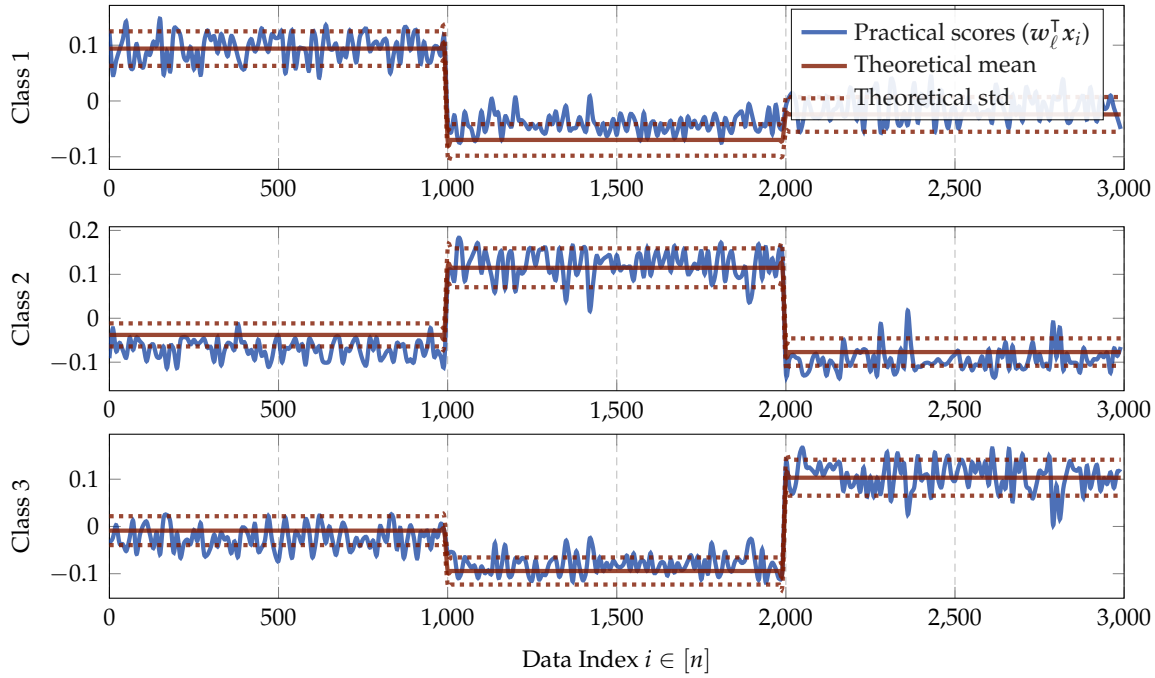



Figure 5.5: Scores (in blue) versus their theoretical estimations (in red) as per Corollary 3.3 of the Main Paper, with the theoretical means (through $\bar{\kappa}^\ell$) and standard deviations (through $\bar{\boldsymbol{K}}^\ell$), on a test set independent from the training set. The used data are Resnet18 [SIVA17] representations ($p = 512$) of real images from the Imagenet dataset [DDS+09]. We considered $k = 3$ classes which are: *hamburger*, *mushroom*, *pizza*. $n = 3000$ and the regularization constants $\lambda_1, \lambda_2, \lambda_3 = 1.5$. The data are normalized such that their norm is $0.1 \cdot \sqrt{p}$ to ensure $\mathcal{A}_X$.

to the data. The Dropout operation is commonly used as a regularization technique applied to certain hidden layers of a neural network during its training phase. However, since zero imputation is known to alter the behavior of neural networks [YLK+19], the Dropout operation must result in the same deleterious effects. Dropping features with other values than zero may thus improve the Dropout in neural networks and mitigate the effects of zero imputation [WWL13, SB16].

To prove and quantify the benefits of a $\alpha$-Dropout approach, in this contribution we study a one hidden layer neural network with $\alpha$-Dropout, i.e., in which the missing or dropped features are replaced by a fixed real value $\alpha$. Training only the output layer, the network (sometimes referred to as an extreme learning machine [HZS06]) reduces to a ridge-regression classifier learnt on $\alpha$-imputed data. Specifically, under the instrumental, yet instructive, setting of a network trained on a set of $n$ data samples of $p$-dimensional features (or equivalently $p$ neurons) distributed in two classes, we retrieve the exact generalization performance when both $p$ and $n$ grow large. A major outcome of our study is the identification of the optimal value of $\alpha$ which maximizes the generalization performances of the classifier.

### 5.2.2 Model and Main Results

Let the training data $\boldsymbol{d}_1, \ldots, \boldsymbol{d}_n \in \mathbb{R}^q$ be independent vectors drawn from two distinct distribution classes $\mathcal{C}_1$ and $\mathcal{C}_2$ of respective cardinality $n_1$ and $n_2$ (and we denote $n =$

$n_1 + n_2$). We suppose the $\boldsymbol{d}_i$'s pass through a first (fixed) random neural network layer with Lipschitz activation $\sigma : \mathbb{R}^q \to \mathbb{R}^p$ in such a way that $\sigma(\boldsymbol{d}_i)$ is a *concentrated random vector* [LC18c]. This random projection is then followed by a random $\alpha$-Dropout, i.e., entries of the feature vector $\sigma(\boldsymbol{d}_i)$ are dropped uniformly at random and replaced by some fixed value $\alpha \in \mathbb{R}$. Letting $\boldsymbol{\mu} \in \mathbb{R}^p$, we further suppose for simplicity of exposition that for $\boldsymbol{d}_i \in \mathcal{C}_a$,

$$\mathbb{E}[\sigma(\boldsymbol{d}_i)] = (-1)^a \boldsymbol{\mu}, \quad \mathbb{E}[\sigma(\boldsymbol{d}_i)\sigma(\boldsymbol{d}_i)^\mathsf{T}] = \boldsymbol{I}_p + \boldsymbol{\mu}\boldsymbol{\mu}^\mathsf{T}.$$

**Remark 5.3.** *As we saw in Chapter 3, this assumption could be largely relaxed but simplifies the interpretation of our results.*

Overall, after the $\alpha$-Dropout layer, the feature vector $\tilde{\boldsymbol{x}}_i \in \mathcal{C}_a$ may thus be written

$$\tilde{\boldsymbol{x}}_i = ((-1)^a \boldsymbol{\mu} + \boldsymbol{z}_i) \odot \boldsymbol{b}_i + \alpha \left(\boldsymbol{1}_p - \boldsymbol{b}_i\right), \tag{5.13}$$

for $a \in \{1, 2\}$, where $\boldsymbol{\mu} \in \mathbb{R}^p$, $\boldsymbol{z}_i$ is a concentrated random vector with zero mean and identity covariance, and $\boldsymbol{b}_i$ is a random binary mask vector with *i.i.d.* entries $b_{ij} \sim \text{Ber}(\varepsilon)$. That is, features are discarded with an average dropout rate $\varepsilon$, as performed in the classical Dropout procedure in neural networks [SHK+14].

The model equation 5.13 thus describes a single hidden layer network with $\alpha$-Dropout (dropped features are replaced by $\alpha$) applied to a two-class mixture of concentrated random vectors of mean $(-1)^a \boldsymbol{\mu}$ for $\boldsymbol{d}_i$ in class $\mathcal{C}_a$ and isotropic covariance. As shown in Chapter 3, from a random matrix perspective, the asymptotic performance of the neural network under study is strictly equivalent to that of features $\tilde{\boldsymbol{x}}_i$ modelled as in equation 5.13 but with $\boldsymbol{z}_i \sim \mathcal{N}(0, \boldsymbol{I}_p)$ thanks to the universality result, an assumption we will make from now on.

In a matrix form, the training features $\tilde{\boldsymbol{X}} = [\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_n] \in \mathcal{M}_{p,n}$ can be compactly written

$$\tilde{\boldsymbol{X}} = \boldsymbol{B}_\varepsilon \odot (\boldsymbol{Z} + \boldsymbol{\mu}\boldsymbol{y}^\mathsf{T}) + \alpha \left(\boldsymbol{1}_p \boldsymbol{1}_n^\mathsf{T} - \boldsymbol{B}_\varepsilon\right), \tag{5.14}$$

where $\boldsymbol{Z}$ has *i.i.d.* $\mathcal{N}(0,1)$ entries, $[\boldsymbol{B}_\varepsilon]_{ij} \sim \text{Ber}(\varepsilon)$ and $\boldsymbol{y} \in \mathbb{R}^n$ stands for the vector of class labels with $y_i = -1$ for $\tilde{\boldsymbol{x}}_i \in \mathcal{C}_1$ and $y_j = 1$ for $\tilde{\boldsymbol{x}}_j \in \mathcal{C}_2$.

For reasons that will be clarified latter, as depicted in Figure 5.6, we shall consider the standardized[5] data matrix $\boldsymbol{X} \equiv \frac{\tilde{\boldsymbol{X}} \boldsymbol{P}_n}{\sqrt{\varepsilon + \alpha^2 \varepsilon(1-\varepsilon)}}$, with $\boldsymbol{P}_n = \boldsymbol{I}_n - \frac{1}{n}\boldsymbol{1}_n \boldsymbol{1}_n^\mathsf{T}$, i.e.,

$$\boldsymbol{X} = \frac{\left(\boldsymbol{B}_\varepsilon \odot (\boldsymbol{Z} + \boldsymbol{\mu}\boldsymbol{y}^\mathsf{T})\right) \boldsymbol{P}_n + \alpha \boldsymbol{B}_\varepsilon \boldsymbol{P}_n}{\sqrt{\varepsilon + \alpha^2 \varepsilon(1-\varepsilon)}}. \tag{5.15}$$

Under the features data model in equation 5.15, we aim in the following to study the generalization performance of a (fully connected) linear layer applied to the features $\boldsymbol{x}_i$'s which is thus equivalent to optimizing (with an $\ell_2$ regularization term)

$$\mathcal{E}(\boldsymbol{w}) = \frac{1}{n} \|\boldsymbol{y} - \boldsymbol{X}^\mathsf{T} \boldsymbol{w}\|^2 + \gamma \|\boldsymbol{w}\|^2, \tag{5.16}$$

[5] Centring by the empirical mean and dividing by the standard deviation $\sqrt{\varepsilon + \alpha^2 \varepsilon(1-\varepsilon)}$ as in batch normalization layers [IS15].

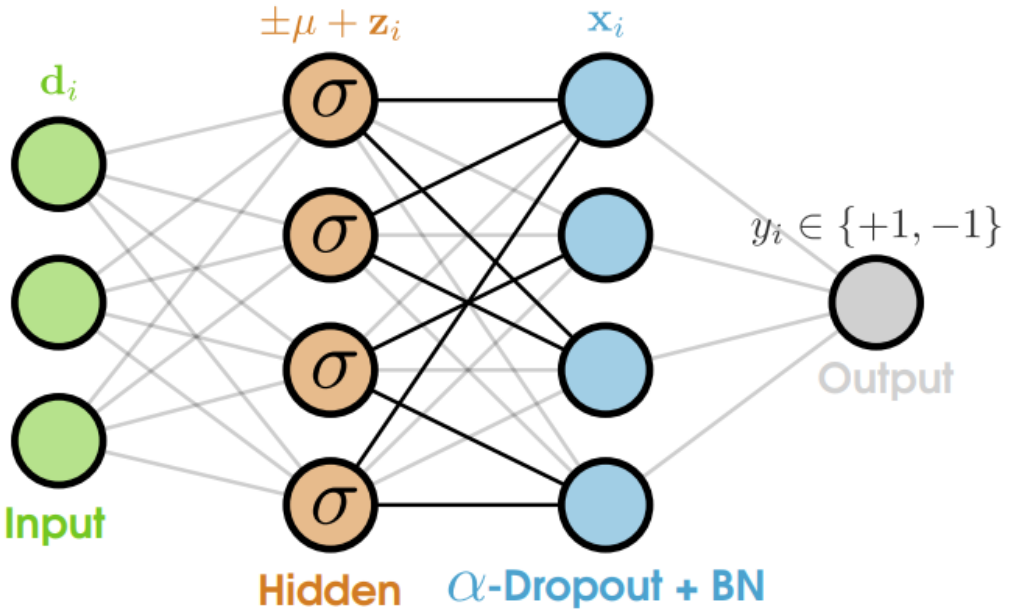



Figure 5.6: Illustration of the model under study in equation 5.15. It consists in a one-hidden-layer network followed by an $\alpha$-Dropout layer and a Batch normalization (BN) layer where the output is given by a linear layer.

the solution of which is explicitly given by, for $z \in \mathbb{C} \setminus \mathbb{R}^-$

$$\boldsymbol{w} = \frac{1}{n}\boldsymbol{Q}(\gamma)\boldsymbol{X}\boldsymbol{y}, \quad \boldsymbol{Q}(z) \equiv \left(\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\intercal + z\boldsymbol{I}_p\right)^{-1}. \tag{5.17}$$

The associated (hard) decision function for a new datum feature vector $\boldsymbol{x} \in \mathcal{C}_a$, for $a \in \{1, 2\}$, then reads

$$g(\boldsymbol{x}) \equiv \boldsymbol{x}^\intercal \boldsymbol{w} = \frac{1}{n}\boldsymbol{x}^\intercal \boldsymbol{Q}(\gamma)\boldsymbol{X}\boldsymbol{y} \underset{\mathcal{C}_2}{\overset{\mathcal{C}_1}{\gtrless}} 0. \tag{5.18}$$

The model in equation 5.15 coupled with the ridge loss function in equation 5.16 is that of an extreme learning machine trained with $\alpha$-Dropout through the random matrix $\boldsymbol{B}_\varepsilon$.

In mathematical terms, studying the generalization performance under a large dimensional network regime consists in studying the statistical behavior of the *resolvent matrix* $\boldsymbol{Q}(z)$ defined in equation 5.17. The main technicality precisely arises from the unconventional presence of the matrix $\boldsymbol{B}_\varepsilon$ in the model. In the following, we derive a deterministic equivalent for $\boldsymbol{Q}(z)$ which is the basic technical ingredient for the further analysis, as a function of $\alpha$ and $\varepsilon$, of the network generalization performance. Let us first present some technical growth rate assumptions before deriving a deterministic equivalent for $\boldsymbol{Q}(z)$.

**Assumption 17** (Growth rate). *As $n \to \infty$,*

1. *$q/n \to r \in (0, \infty)$ and $p/n \to c \in (0, \infty)$;*
2. *For $a \in \{1, 2\}$, $\frac{n_a}{n} \to c_a \in (0, 1)$;*

3. $\|\boldsymbol{\mu}\| = \mathcal{O}(1)$.

**Remark 5.4.** *Assumption 17 along with the standardization in equation 5.15 ensure particularly that the operator norm of $\boldsymbol{Q}(z)$ is bounded asymptotically.*

#### 5.2.2.1 Deterministic equivalent

Now we provide the key technical tool to express the performances of the network under investigation, i.e., a deterministic equivalent for $\boldsymbol{Q}(z)$. Let $\boldsymbol{A} = \frac{\boldsymbol{P}(\boldsymbol{B}_\varepsilon \odot (\boldsymbol{\mu}\boldsymbol{y}^\intercal))}{\sqrt{\varepsilon+\alpha^2\varepsilon(1-\varepsilon)}}$ be the signal part of the the model equation 5.15, form [HLN+07], a deterministic equivalent of $\boldsymbol{Q}(z)$ is given by

$$\bar{\boldsymbol{Q}}(z) \equiv \left( q^{-1}(z) + \frac{1}{1+cq(z)} \frac{1}{n} \mathbb{E}[\boldsymbol{A}\boldsymbol{A}^\intercal] \right)^{-1} \quad \text{with} \quad q(z) \equiv \frac{1+cq(z)}{1+z(1+cq(z))}$$

And by Assumption 17 ($\|\boldsymbol{\mu}\| = \mathcal{O}(1)$), we have

$$\begin{aligned} \frac{1}{n}\mathbb{E}[\boldsymbol{A}\boldsymbol{A}^\intercal] &= \frac{\varepsilon}{1+\alpha^2(1-\varepsilon)} \boldsymbol{\mu}\boldsymbol{\mu}^\intercal + \frac{1-\varepsilon}{1+\alpha^2(1-\varepsilon)} \operatorname{diag}(\boldsymbol{\mu}^{\odot 2}) + \mathcal{O}_{\|\cdot\|}(n^{-\frac{1}{2}}) \\ &= a\boldsymbol{\mu}\boldsymbol{\mu}^\intercal + b \operatorname{diag}(\boldsymbol{\mu}^{\odot 2}) + \mathcal{O}_{\|\cdot\|}(p^{-\frac{1}{2}}) \end{aligned}$$

Therefore, by Lemma 2.6, we have (we denote $r(z) = \frac{1}{1+cq(z)}$)

$$\begin{aligned} \bar{\boldsymbol{Q}}(z) &= \left( q^{-1}\boldsymbol{I}_p + br(z) \operatorname{diag}(\boldsymbol{\mu}^{\odot 2}) + ar(z)\boldsymbol{\mu}\boldsymbol{\mu}^\intercal \right)^{-1} \\ &= \underbrace{\left( q^{-1}\boldsymbol{I}_p + br(z) \operatorname{diag}(\boldsymbol{\mu}^{\odot 2}) \right)^{-1}}_{\mathcal{D}_z} - \frac{ar(z)\mathcal{D}_z\boldsymbol{\mu}\boldsymbol{\mu}^\intercal\mathcal{D}_z}{1+ar(z)\boldsymbol{\mu}^\intercal\mathcal{D}_z\boldsymbol{\mu}} \end{aligned}$$

where $q(z) = \frac{1+cq(z)}{1+z(1+cq(z))}$ is a second order equation in $q(z)$. Therefore, we have

**Proposition 5.3.** *Under Assumption 17,*

$$\boldsymbol{Q}(z) \leftrightarrow \bar{\boldsymbol{Q}}(z) \equiv \mathcal{D}_z - \frac{\frac{\varepsilon}{1+\alpha^2(1-\varepsilon)}\mathcal{D}_z\boldsymbol{\mu}\boldsymbol{\mu}^\intercal\mathcal{D}_z}{1+cq(z)+\frac{\varepsilon}{1+\alpha^2(1-\varepsilon)}\boldsymbol{\mu}^\intercal\mathcal{D}_z\boldsymbol{\mu}},$$

*where* $\mathcal{D}_z \equiv q(z) \operatorname{diag}\left\{ \frac{1+cq(z)}{1+cq(z)+\frac{(1-\varepsilon)q(z)}{1+\alpha^2(1-\varepsilon)}\mu_i^2} \right\}_{i=1}^p$, *and $q(z)$ is given by*

$$q(z) \equiv \frac{c-z-1+\sqrt{(c-z-1)^2+4zc}}{2zc}.$$

Proposition 5.3 shows that the deterministic equivalent $\bar{\boldsymbol{Q}}(z)$ involves two terms: a diagonal matrix $\mathcal{D}_z$ (describing the noise part of the data model) and an informative scaled rank-1 matrix $\mathcal{D}_z\boldsymbol{\mu}\boldsymbol{\mu}^\intercal\mathcal{D}_z$. We see through the expression of $\bar{\boldsymbol{Q}}(z)$ that the informative term is linked to the noise term (through $\mathcal{D}_z$) if $\varepsilon \neq 1$, and for small values of $\varepsilon$ or equivalently large values of $\alpha$ the "energy" of the informative term is transferred to the noise term which will result in a poor classification accuracy on the train set, still we will subsequently see that for a fixed value of $\varepsilon$, there exists a value of $\alpha$ which will provide optimal classification rates on the test set. We will next use the property that $\boldsymbol{a}^\intercal\boldsymbol{Q}(z)\boldsymbol{b} \simeq \boldsymbol{a}^\intercal\bar{\boldsymbol{Q}}(z)\boldsymbol{b}$

for all large $n, p$ and deterministic bounded vectors $\boldsymbol{a}, \boldsymbol{b}$, to exploit $\bar{\boldsymbol{Q}}(z)$ as a proxy for the performance analysis (which is precisely related to a bilinear form on $\boldsymbol{Q}(z)$) of the $\alpha$-Dropout neural network.

Further, let us introduce the following quantities which will be used subsequently. First, we have under Assumption 17, the statistics of the feature vector $\boldsymbol{x}_i$, for $\boldsymbol{x}_i \in \mathcal{C}_a$, are:

$$\begin{aligned} \boldsymbol{m}_a &\equiv \mathbb{E}\left[\boldsymbol{x}_i\right] = (-1)^a \sqrt{\frac{\varepsilon}{1+\alpha^2(1-\varepsilon)}}\boldsymbol{\mu}, \\ \boldsymbol{C}_\varepsilon &\equiv \mathbb{E}\left[\boldsymbol{x}_i \boldsymbol{x}_i^\intercal\right] = \boldsymbol{I}_p + \frac{\varepsilon}{1+\alpha^2(1-\varepsilon)}\boldsymbol{\mu}\boldsymbol{\mu}^\intercal \\ &\quad + \frac{1-\varepsilon}{1+\alpha^2(1-\varepsilon)}\left(\operatorname{diag}(\boldsymbol{\mu}^{\odot 2} + 2\alpha\boldsymbol{\mu}) - \frac{\alpha^2}{p}\mathbf{1}_p\mathbf{1}_p^\intercal\right). \end{aligned} \tag{5.19}$$

The above statistics are derived as follows. Let $\boldsymbol{x} = \boldsymbol{b} \odot (\boldsymbol{z} + \boldsymbol{\mu}) + \alpha \boldsymbol{P}\boldsymbol{b}$. Denote $\mathbb{E}[\boldsymbol{P}\boldsymbol{b}] = \mathbb{E}[\boldsymbol{b} - \frac{1}{p}\sum_{i=1}^n b_i \mathbf{1}_p] = \mathbf{0}$, we have $\mathbb{E}[\boldsymbol{b} \odot (\boldsymbol{z} + \boldsymbol{\mu})] = \boldsymbol{b} \odot \boldsymbol{\mu}$ and $\mathbb{E}[\boldsymbol{b}\boldsymbol{b}^\intercal] = \varepsilon \mathbf{1}_p\mathbf{1}_p^\intercal + (1-\varepsilon)\boldsymbol{I}_p \equiv \boldsymbol{C}_b$. Therefore,

$$\begin{aligned} &\mathbb{E}[\alpha \boldsymbol{P}\boldsymbol{b}(\alpha \boldsymbol{P}\boldsymbol{b})^\intercal] = \alpha^2 \boldsymbol{P}\boldsymbol{C}_b\boldsymbol{P}^\intercal \\ &\mathbb{E}[\boldsymbol{b} \odot (\boldsymbol{z}+\boldsymbol{\mu})(\boldsymbol{b} \odot (\boldsymbol{z}+\boldsymbol{\mu}))^\intercal] = \boldsymbol{C}_b \odot (\boldsymbol{I}_p + \boldsymbol{\mu}\boldsymbol{\mu}^\intercal) \\ &\mathbb{E}[\boldsymbol{b} \odot (\boldsymbol{z}+\boldsymbol{\mu})(\alpha \boldsymbol{P}\boldsymbol{b})^T] = \alpha \operatorname{diag}(\boldsymbol{\mu})\boldsymbol{C}_b\boldsymbol{P} \end{aligned}$$

Hence,

$$\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\intercal] = \boldsymbol{C}_b \odot (\boldsymbol{I}_p + \boldsymbol{\mu}\boldsymbol{\mu}^\intercal) + \alpha \operatorname{diag}(\boldsymbol{\mu})\boldsymbol{C}_b\boldsymbol{P} + \alpha \boldsymbol{P}\boldsymbol{C}_b \operatorname{diag}(\boldsymbol{\mu}) + \alpha^2 \boldsymbol{P}\boldsymbol{C}_b\boldsymbol{P}^\intercal$$

Since $\|\boldsymbol{\mu}\| = \mathcal{O}(1)$, as in Assumption 17, we have

$$\boldsymbol{C}_b \odot (\boldsymbol{I}_p + \boldsymbol{\mu}\boldsymbol{\mu}^\intercal) = \varepsilon \boldsymbol{I}_p + (1-\varepsilon)\boldsymbol{I}_p + \varepsilon \boldsymbol{\mu}\boldsymbol{\mu}^\intercal + (1-\varepsilon)\operatorname{diag}(\boldsymbol{\mu}^{\odot 2}) = \boldsymbol{I}_p + \varepsilon \boldsymbol{\mu}\boldsymbol{\mu}^\intercal + (1-\varepsilon)\operatorname{diag}(\boldsymbol{\mu}^{\odot 2})$$

$$\operatorname{diag}(\boldsymbol{\mu})\boldsymbol{C}_b\boldsymbol{P} = (1-\varepsilon)\operatorname{diag}(\boldsymbol{\mu})\boldsymbol{P} = (1-\varepsilon)\operatorname{diag}(\boldsymbol{\mu}) - \frac{1-\varepsilon}{p}\boldsymbol{\mu}\mathbf{1}_p^\intercal = (1-\varepsilon)\operatorname{diag}(\boldsymbol{\mu}) + \mathcal{O}_{\|\cdot\|}(p^{-\frac{1}{2}})$$

$$\boldsymbol{P}\boldsymbol{C}_b\boldsymbol{P} = (1-\varepsilon)\boldsymbol{P}$$

Thus

$$\begin{aligned} \mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^\intercal\right] &= (1+\alpha^2(1-\varepsilon))\boldsymbol{I}_p + \varepsilon \boldsymbol{\mu}\boldsymbol{\mu}^\intercal - \frac{\alpha^2(1-\varepsilon)}{p}\mathbf{1}_p\mathbf{1}_p^\intercal + (1-\varepsilon)\operatorname{diag}(\boldsymbol{\mu}^{\odot 2}) \\ &\quad + 2\alpha(1-\varepsilon)\operatorname{diag}(\boldsymbol{\mu}) + \mathcal{O}_{\|\cdot\|}(p^{-\frac{1}{2}}) \end{aligned}$$

Therefore,

$$\boldsymbol{m}_a \equiv \mathbb{E}\left[\boldsymbol{x}_i\right] = (-1)^a \sqrt{\frac{\varepsilon}{1+\alpha^2(1-\varepsilon)}}\boldsymbol{\mu} \tag{5.20}$$

$$\boldsymbol{C}_\varepsilon \equiv \mathbb{E}\left[\boldsymbol{x}_i \boldsymbol{x}_i^\intercal\right] = \boldsymbol{I}_p + \frac{\varepsilon}{1+\alpha^2(1-\varepsilon)}\boldsymbol{\mu}\boldsymbol{\mu}^\intercal \frac{1-\varepsilon}{1+\alpha^2(1-\varepsilon)}\left(\operatorname{diag}(\boldsymbol{\mu}^{\odot 2}) + 2\alpha\operatorname{diag}(\boldsymbol{\mu})\right) \tag{5.21}$$

$$- \frac{\alpha^2(1-\varepsilon)}{p(1+\alpha^2(1-\varepsilon))}\mathbf{1}_p\mathbf{1}_p^\intercal + \mathcal{O}_{\|\cdot\|}(p^{-\frac{1}{2}}). \tag{5.22}$$

We will also need in the following the quantity

$$\delta(z) \equiv \frac{1}{n}\operatorname{Tr}\left(\boldsymbol{C}_\varepsilon \bar{\boldsymbol{Q}}(z)\right). \tag{5.23}$$

#### 5.2.2.2 Generalization Performance of $\alpha$-Dropout

The generalization performance of the classifier relates to misclassification errors

$$\mathbb{P}\left(g(\boldsymbol{x}) > 0 \,|\, \boldsymbol{x} \in \mathcal{C}_1\right), \quad \mathbb{P}\left(g(\boldsymbol{x}) < 0 \,|\, \boldsymbol{x} \in \mathcal{C}_2\right)$$

where $g(\cdot)$ is the decision function previously defined in equation 5.18.

Since the Dropout is deactivated at inference time, the statistics of $\boldsymbol{x}$ correspond to the setting where $\varepsilon = 1$, and thus

$$\mathbb{E}[\boldsymbol{x}] = (-1)^a \boldsymbol{\mu}, \quad \boldsymbol{C}_1 = \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\intercal] = \boldsymbol{I}_p + \boldsymbol{\mu}\boldsymbol{\mu}^\intercal.$$

Further, define the following quantities which shall be used subsequently

$$\eta(\boldsymbol{A}) \equiv \frac{(1+\delta(\gamma))\frac{1}{n}\operatorname{Tr}\left(\boldsymbol{C}_\varepsilon \bar{\boldsymbol{Q}}(\gamma)\boldsymbol{A}\bar{\boldsymbol{Q}}(\gamma)\right)}{(1+\delta(\gamma))^2 - \frac{1}{n}\operatorname{Tr}\left(\boldsymbol{C}_\varepsilon \bar{\boldsymbol{Q}}(\gamma)\boldsymbol{C}_\varepsilon \bar{\boldsymbol{Q}}(\gamma)\right)},$$
$$\boldsymbol{\Delta}(\boldsymbol{A}) \equiv \bar{\boldsymbol{Q}}(\gamma)\left(\boldsymbol{A} + \frac{\eta(\boldsymbol{A})}{1+\delta(\gamma)}\boldsymbol{C}_\varepsilon\right)\bar{\boldsymbol{Q}}(\gamma).$$

By Lyapunov's central limit theorem [Bil08], the decision function has the following Gaussian approximation as $n \to \infty$.

**Theorem 5.5** (Gaussian Approximation of $g(\boldsymbol{x})$)**.** *Under Assumption 17, for $\boldsymbol{x} \in \mathcal{C}_a$ with $a \in \{1,2\}$,*

$$\nu^{-\frac{1}{2}}\left(g(\boldsymbol{x}) - m_a\right) \xrightarrow{\mathcal{D}} \mathcal{N}(0,1)$$

*where*

$$m_a \equiv (-1)^a \sqrt{\frac{\varepsilon}{1+\alpha^2(1-\varepsilon)}}\frac{\boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}}(\gamma)\boldsymbol{\mu}}{1+\delta(\gamma)}$$
$$\nu \equiv \frac{1}{(1+\delta(\gamma))^2}\left(\eta(\boldsymbol{C}_1) + \frac{\varepsilon}{1+\alpha^2(1-\varepsilon)} \times \left[\boldsymbol{\mu}^\intercal\left(\boldsymbol{\Delta}(\boldsymbol{C}_1) - \bar{\boldsymbol{Q}}(\gamma)\right)\boldsymbol{\mu} - \frac{2\,\eta(\boldsymbol{C}_1)\boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}}(\gamma)\boldsymbol{\mu}}{1+\delta(\gamma)}\right]\right).$$

*Proof.* See Subsection C.3.2.1. □

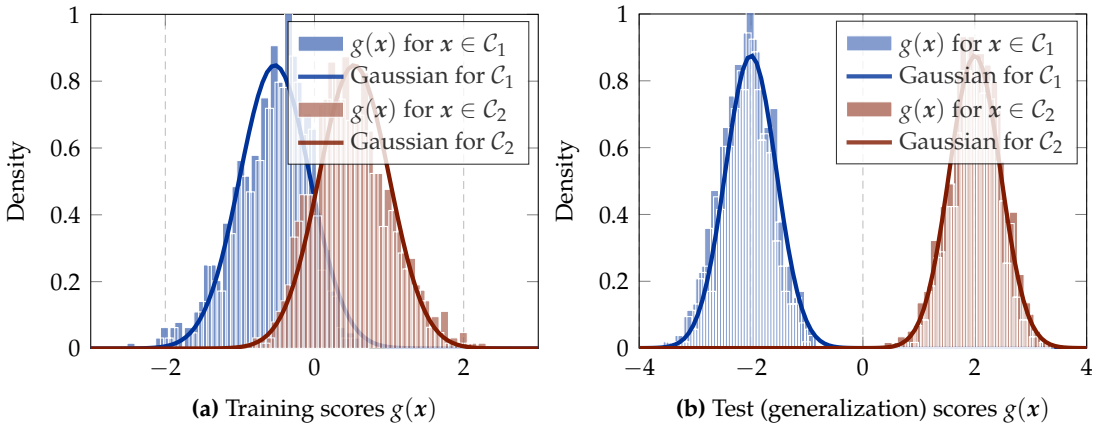
In a nutshell, Theorem 5.5 states that the one hidden layer network classifier with $\alpha$-Dropout is asymptotically equivalent to the thresholding of two monovariate Gaussian random variables, the means and variances of which depend on $\boldsymbol{\mu}$, $\boldsymbol{C}_\varepsilon$ and the parameters $\alpha$ and $\varepsilon$. As such, we have the corresponding (asymptotic) classification errors:

**Corollary 5.2** (Generalization Performance of $\alpha$-Dropout)**.** *Under the setting of Theorem 5.5, for $a \in \{1,2\}$, with probability one*

$$\mathbb{P}\left((-1)^a\, g(\boldsymbol{x}) < 0 \,|\, \boldsymbol{x} \in \mathcal{C}_a\right) - Q\left(\frac{m_a}{\sqrt{\nu}}\right) \to 0$$

*with $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-u^2/2}du$ the Gaussian tail function.*

Corollary 5.2 can therefore be exploited to find the optimal value of $\alpha^*$ which minimizes the test misclassification error, since $Q'(x) < 0$ the optimal value $\alpha^*$ satisfies the equation $\frac{1}{m_a}\frac{\partial m_a}{\partial \alpha} = \frac{1}{\sqrt{\nu}}\frac{\partial \sqrt{\nu}}{\partial \alpha}$ which can be solved numerically.




**(a)** Training scores $g(\boldsymbol{x})$ **(b)** Test (generalization) scores $g(\boldsymbol{x})$

Figure 5.7: Histogram of the decision function $g(\boldsymbol{x})$ when applied to the training data **(a)** and test data **(b)**. The curves represent the Gaussian approximations as per Theorem 5.5 and Theorem 5.6 for test and training data respectively. We used the parameters $\boldsymbol{\mu} = \frac{5 \cdot \boldsymbol{u}}{\|\boldsymbol{u}\|}$ with $\boldsymbol{u} = [10, 10, -10, -10, \boldsymbol{v}]$ where $\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_{p-4})$ , $p = 125$, $n_1 = n_2 = 1000$, $\gamma = 1 \cdot 10^{-2}$, $\varepsilon = 0.25$ and $\alpha = 2$.

#### 5.2.2.3 Training Performance of $\alpha$-Dropout

It is instructive to compare the generalization versus training performances of the network classifier with $\alpha$-Dropout. Following similar arguments as in [LC18c], the central limit argument of the previous section also holds for $g(\boldsymbol{x})$ with $\boldsymbol{x} \in \mathcal{C}_a$ taken from the training set $\boldsymbol{X}$.

**Theorem 5.6** (Training performance of $\alpha$-Dropout)**.** *Under Assumption 17, for $\boldsymbol{x} \in \mathcal{C}_a$ with $a \in \{1, 2\}$ a column of $\boldsymbol{X}$, with probability one,*

$$\mathbb{P}\left((-1)^a g(\boldsymbol{x}) < 0 \,|\, \boldsymbol{x} \in \mathcal{C}_a\right) - Q\left(\frac{\bar{m}_a}{\sqrt{\bar{v} - \bar{m}_a^2}}\right) \to 0$$

*where*

$$\begin{aligned}
\bar{m}_a &\equiv \frac{\delta(\gamma)}{1+\delta(\gamma)} + \frac{(-1)^a \varepsilon}{1+\alpha^2(1-\varepsilon)} \frac{\boldsymbol{\mu}^\mathsf{T} \bar{\boldsymbol{Q}}(\gamma) \boldsymbol{\mu}}{(1+\delta(\gamma))^2} \\
\bar{v} &\equiv \left(\frac{\delta(\gamma)}{1+\delta(\gamma)}\right)^2 + \frac{\eta(\boldsymbol{C}_\varepsilon)}{(1+\delta(\gamma))^4} + \frac{\varepsilon}{1+\alpha^2(1-\varepsilon)} \\
&\quad \times \boldsymbol{\mu}^\mathsf{T} \left( \frac{\delta(\gamma) \bar{\boldsymbol{Q}}(\gamma)}{(1+\delta(\gamma))^3} + \frac{\boldsymbol{\Delta}(\boldsymbol{C}_\varepsilon)}{(1+\delta(\gamma))^4} - \frac{2\eta(\boldsymbol{C}_\varepsilon) \bar{\boldsymbol{Q}}}{(1+\delta(\gamma))^5} \right) \boldsymbol{\mu}.
\end{aligned}$$

*Proof.* See Subsection C.3.2.2. □

### 5.2.3 Experiments

**Gaussian Approximation of the Decision Function.** We complete this study by simulations to validate our theoretical findings. Figure 5.7 depicts histograms showing the distribution of $g(\boldsymbol{x})$ for both **(a)** training and **(b)** test data. As we can see, these distributions are well approximated by monovariate Gaussians as per Theorem 5.5 and Theorem 5.6. Since the $\alpha$-Dropout removes features at random from the training data, the

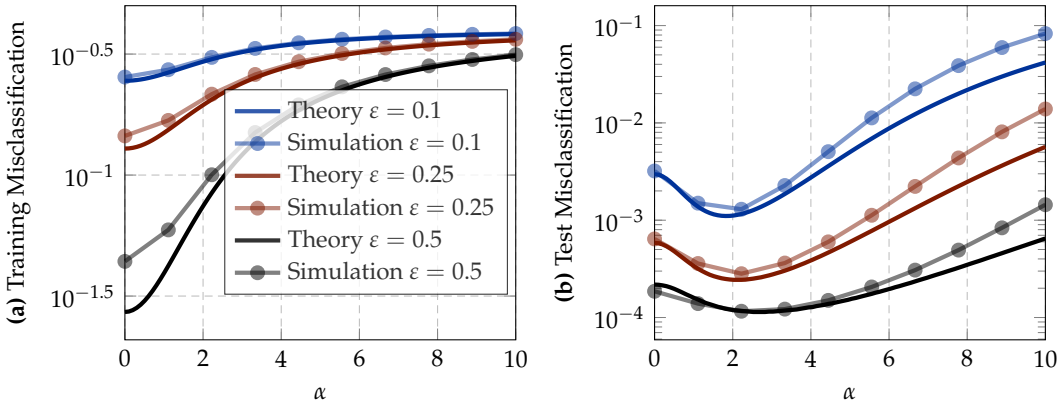



Figure 5.8: **(a)** Training and **(b)** Test misclassification errors as per Theorem 5.6 and Corollary 5.2 respectively. We used the parameters $\boldsymbol{\mu} = 4 \cdot [\frac{1}{\sqrt{4}}, \frac{1}{\sqrt{4}}, -\frac{1}{\sqrt{4}}, -\frac{1}{\sqrt{4}}, \mathbf{0}_{p-4}^\intercal]^\intercal$, $p = 125$, $n_1 = n_2 = 1000$ and $\gamma = 1 \cdot 10^{-2}$. Simulations are obtained through 100 Monte-Carlo runs of independent realizations of the matrix $\boldsymbol{X}$ as in equation 5.15.

misclassification error happens to be larger on the training set compared to the test set. Notably, the difference between the training and test error arises theoretically from the term $\kappa \equiv \sqrt{\frac{\varepsilon}{1+\alpha^2(1-\varepsilon)}}$ as $\bar{m}_a \approx \kappa m_a$, therefore, for small values of $\varepsilon$ the training error is larger than the test error, which shows the regularization effect of the Dropout.

**Training and Test Performances.** Figure 5.8 depicts the theoretical **(a)** training (through Theorem 5.6) and **(b)** test (through Corollary 5.2) misclassification errors, for different values of $\varepsilon$ and in terms of $\alpha$, and their simulated counterparts. We can notice from these plots that the training error increases with $\alpha$ and is minimal for $\alpha = 0$. In contrast, the test misclassifaction error is convex in terms of $\alpha$ and therefore the lowest generalization error corresponds to an optimal value $\alpha \neq 0$. We also remark that the optimal value of $\alpha$ increases in terms of $\varepsilon$ which is counterintuitive since we excpect and $\alpha$ near to 0 for large values of $\varepsilon$, but actually, the test misclassification error in terms of $\alpha$ gets more and more flatter as $\varepsilon$ increases.

### 5.2.4 Central Contribution and Perspectives

Leveraging on random matrix theory, we have analyzed the effect of the $\alpha$-Dropout layer on a one layer neural network, which allowed us to have a deeper understanding of the impact of this layer. We have notably exhibited an optimal Dropout operation (dropping our features with some $\alpha \neq 0$) in terms of the generalization error of the studied classifier. Although, our analysis was presented on a simple binary classification task, it can be straightforwardly generalized to a more realistic data model as the mixture of $k$-class model [LC18b, SLTC20]. Under a $k$-class model it may be beneficial to consider an $\alpha_\ell$ per class $\mathcal{C}_\ell$ as the classes may be constructed with different statistics. Following the same approach one can derive the test misclassification error as per Corollary 5.2 in terms of scalar quantities involving the data statistics, and therefore exploit the formulas to find the optimal values of $\alpha_\ell$'s. Other perspectives of this work concern its application to real data and using multi-layer neural network architectures.

# Chapter 6

# Conclusions & Perspectives

## 6.1 Conclusions

The first notable result from this thesis, presented in Section 3.1, sheds light on the fact that artificially generated data through Generative Adversarial Nets (GANs) are random vectors which fall within the class of *concentrated vectors*. Consequently, real data can be modelled by this class of random vectors – and also since they generalize Gaussian vectors – if we can assimilate them to GAN data. Motivated by this first result, we have further investigated in this thesis three main ML methods, under the Mixture of Concentrated Vectors hypothesis for the input data model, where each data-class is described by its first and second order moments $\{\boldsymbol{\mu}_\ell\}_{\ell=1}^k$ and $\{\boldsymbol{\Sigma}_\ell\}_{\ell=1}^k$. Indeed, relying on random matrix theory (RMT) and supposing the high-dimensional regime when the number of input data and their dimension are both large and comparable, we have first analyzed in Section 3.2 the spectral behavior of large Gram matrices which are at the core of various linear methods. As non-linearities appear in various ML methods, we have also investigated in Section 4.1 kernel methods through the analysis of large kernel matrices. Lastly, we have studied, through the example of the Softmax classifier in Section 5.1, ML methods which are implicitly defined by (convex) optimization problems.

As an aftermath of our analysis of these methods, we have emphasized that their effective performances – when applied to large dimensional data – *solely* depend on the class-wise *means* and *covariances* of the input data, namely the statistics $\{\boldsymbol{\mu}_\ell\}_{\ell=1}^k$ and $\{\boldsymbol{\Sigma}_\ell\}_{\ell=1}^k$, a result which we have empirically verified through extensive experiments using CNN representations of GAN-generated images. This constitutes a major outcome of this thesis in the sense that it highlights the *universality aspect* of large ML classifiers regardless of their input data distribution. As a consequence, the universality aspect notably supports the validity of the Gaussian Mixture Model in the random matrix theory regime, thereby justifying the Gaussian mixture hypothesis on data as assumed in the works [Lia19, Mai19]. From a practical standpoint, RMT allows therefore for systematic analyzes and improvements of a wide range of ML classifiers which rely on the aforementioned methods – through the estimation of their asymptotic performances – on realistic data, such as deep learning representations of the surprising images generated artificially by GANs.

To provide an example of such analyzes and improvements through RMT, we have investigated in Section 4.2 the problem of sparse PCA where we have notably generalized recent ideas to tackle this problem by means of a broader class of random kernel matri-

ces. Specifically, our analysis has provided insights into how the principal components can be consistently recovered through a random kernel matrix $f(\hat{\boldsymbol{\Sigma}})$, where $\hat{\boldsymbol{\Sigma}}$ is a spiked covariance model and $f$ a smooth function. Another example of such RMT analyzes and improvements concerns the analysis of the Dropout in neural networks. Indeed, as we presented in Section 5.2, we saw that the RMT analysis of a one-hidden-layer network with an $\alpha$-Dropout layer has yielded insights about the effect of the Dropout operation, thereby improving it by suggesting dropping out features and replacing them with some $\alpha \neq 0$ in order to minimize the generalization error of the studied network.

## 6.2 Limitations and Perspectives

As we discussed in the previous section, the main outcomes of this thesis are twofold: The first outcome concerns the fact that GAN generated data are concentrated vectors by design, while the second outcome being the universality aspect of the studied ML methods, i.e., ML methods which rely on Gram matrices, kernel matrices or for which their solutions are explicit or implicitly defined through convex optimization problems. Yet, these findings have limitations, so we discuss in the following some potential future research directions in the same vein as our actual preliminary findings.

**Validity of concentrated vectors for data modelling:** So far in our experiments throughout this manuscript, we have considered as data CNN representations of GAN generated images. More fundamentally, we have shown that the framework of concentrated vectors is justified by the fact that GAN data fall within this class of random vectors. Hence, a first limitation concerns the fact that not all types of data can be generated by GANs, therefore modelling them as concentrated vectors is not straightforward. Indeed, texts (and thus their embeddings) are an example of such data types. However, in the natural language processing (NLP) paradigm, *text generation* is performed using Recurrent Neural Networks (RNNs) [SMH11]. Specifically, text data are generated sequentially starting from some hidden state $\boldsymbol{z}_0 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$ and then computing at every time step $t$ the new hidden state as $\boldsymbol{z}_t = \mathrm{RNN}(\boldsymbol{z}_{t-1}, \boldsymbol{w}_{t-1})$ where $\boldsymbol{w}_t$ stands for the embedding of $t$-th word in the processed sentence and RNN is the generator network. Consequently, studying this mechanism along with the RNN dynamics may yield to highlight that the learned word embeddings $\boldsymbol{w}_t$ are concentrated vectors, since the involved operations are Lipschitz transformations and the starting hidden state $\boldsymbol{h}_0$ is a concentrated vector. Another and more revolutionary way of building text embeddings for NLP relies on more involved and huge neural networks architectures known as *transformers* which are based on the so-called attention modules [VSP+17]. The basic and fundamental operation of these architectures is the *self-attention* operation, which is a sequence-to-sequence operation and basically consists in transforming a sequence of input vectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_t$ to another output sequence $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t$ as $\boldsymbol{y}_i = \sum_j \alpha_{ij} \boldsymbol{w}_j$ where the weights $\alpha_{ij}$ are function of the input sequences $\boldsymbol{w}_i$'s (e.g., $\alpha_{ij} = \boldsymbol{w}_i^\intercal \boldsymbol{w}_j$ or as commonly used $\alpha_{ij} = \exp(\boldsymbol{w}_i^\intercal \boldsymbol{w}_j) / \sum_j \exp(\boldsymbol{w}_i^\intercal \boldsymbol{w}_j)$ involving the Softmax activation). Therefore, a first idea would be to study through the tools of RMT the mechanism of this operation which might explain the concentration property of the resulting learned word embeddings $\boldsymbol{w}_i$'s.

**Generalizations of the $k$-class Mixture of Concentrated Vectors Model:** We assumed throughout this manuscript that data are made of distinct classes which are represented by their respective statistical means and covariances. This model does not take into account the between-class correlations or the fact that data have some hierarchical structure (e.g., the class animals is made of several sub-classes: cats, dogs, etc.). A possible generalization of our findings is to include these correlations/hierarchical structure and study their influence so as to understand the behavior of ML methods in some specific applications that are sensitive to between-classe correlations. Concretely, a possible idea is to generalize the notion of hierarchical Gaussian mixture models [OP16, LM07] by relaxing the Gaussianty assumption to the class of concentrated vectors. Such generalization might also find applications in transfer learning in the vein of [YC08] as the target domain is very often different from the source domain, hence this difference shall be considered by modelling the underlying statistical dependencies.

**On the analysis of more complex classifiers relying on non-convex optimization problems:** As we discussed previously, the universality behavior concerns mostly ML methods which rely on Gram matrices, kernel matrices or for which their solutions are explicit or implicitly defined through convex optimization problems. However, the most successive ML methods nowadays rely on non-convex optimization problems which is the case of multi-layer neural networks. The main difficulty when studying these methods is the non-convex nature and the use of the backpropagation algorithm which make the analysis more challenging. However, using tools from *information theory* [SZT17], it has been shown that when building and training a neural network for a given task (e.g., classification), the *mutual information* between its input and successive hidden layers decreases across its internal layers. Roughly speaking, neural networks learn to produce hidden representations that are less *correlated* with the input by basically keeping the most relevant information across layers to solve the considered task, i.e., as long as there is a high correlation between the considered hidden layer and the desired target output. Relying on this intuition, efforts have been made to develop new alternatives to the end-to-end backpropagation algorithm [BEO19, PL20, MLK20, DYP20, LOV19]. Basically, these alternatives suggest to rather train the models layer-wise, i.e., training one layer at a time which might provide more flexibility and accessibility to theoretical analysis. A common approach for such alternatives relies on the *Information Bottlenech* (IB) principle which consists in minimizing the following objective

$$\min_{p(\boldsymbol{h}\,|\,\boldsymbol{x})} \mathcal{I}(\boldsymbol{x};\boldsymbol{h}) - \gamma \mathcal{I}(\boldsymbol{h};\boldsymbol{y})$$

where $\boldsymbol{x}$ stands for the input, $\boldsymbol{y}$ is the target label, $\boldsymbol{h}$ is the hidden feature vector, $\mathcal{I}$ stands for the mutual information and $\gamma > 0$ is an hyperparameter. As such, optimizing IB is equivalent to minimizing the mutual information (i.e., dependence) between the input $\boldsymbol{x}$ and the hidden variable $\boldsymbol{h}$ while maximizing the mutual information between $\boldsymbol{h}$ and the desired target $\boldsymbol{y}$. Specifically, $\boldsymbol{h}$ plays the role of the hidden representations of $\boldsymbol{x}$ in the neural network. Still, from a practical standpoint, a first issue appears from the fact that the mutual information is generally hard to compute. Authors in [MLK20] have very recently proposed an alternative to overcome this issue by considering the Hilbert-Schmidt Independent Criteria (HSIC) instead of the mutual information. Specifically, given some data matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathcal{M}_{p,n}$ and their corresponding targets $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n] \in \mathcal{M}_{k,n}$. Further supposing that the hidden features are given by some parametric transformation $\boldsymbol{H} = f(\boldsymbol{X};\Theta) \in \mathcal{M}_{q,n}$ of the input matrix $\boldsymbol{X}$. The HSIC between $\boldsymbol{X}$ and $\boldsymbol{H}$ is particularly

given by

$$\text{HSIC}(\boldsymbol{X}, \boldsymbol{H}) = \frac{1}{(n-1)^2} \operatorname{tr}(\boldsymbol{K_X K_H})$$

where $\boldsymbol{K_X} \in \mathcal{M}_n$ is the kernel matrix having entries $(\boldsymbol{K_X})_{ij} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for some kernel $\kappa$. Therefore, exploiting this formalism and RMT tools of the analysis of kernel matrices, one could gain new insights about the internal mechanism of this learning approach. In particular, two research directions might be interesting to explore:

1. A first direction concerns the analysis of the dynamics of a two hidden layers (or possibly networks with multiple layers) with the HSIC principale. Concretely, leveraging on RMT, an idea would be to study the involved functionals of the large kernel matrices $\boldsymbol{K_X}$ and $\boldsymbol{K_H}$ to get access to the dynamics of the studied network in the same vein as [LC18a]. Indeed, RMT analyses may provide statistical descriptions of the local or global extrema, thereby leading to a theoretical understanding of the model performances as well as improvements of the studied method through optimal hyperparameter tuning. Such an analysis could be first considered using a one-layer network with non-linear activation in order to analyze and understand the effect of non-linearity on the network dynamics, as investigated in [AS17] with the standard gradient descent approach.

2. Another direction of research, which might be more challenging, is to exploit the layer-wise learning approach to describe the *encoded information* or *encoded sufficient statistics* by neural networks layers weights in the same vein as our analysis of the Softmax layer in Section 5.1. A possible idea is to start by performing preliminary empirical studies to understand the correlations between the learned weights and the corresponding layers representations using the HSIC principle. Since we have identified, through the analysis of the Softmax layer in Section 5.1, that the last layer retrieves information from the network representations through their class-wise means and covariances, one could intuitively obtain the same conclusions layer-wise and characterize the *encoded statistics* in the internal layers weights. Such an analysis might be very useful for the understanding of the mechanism of neural networks and might explain their incredible ability to build hierarchical and discriminative representations.

As a concluding remark, we forcefully believe that the present manuscript scratches the surface of a much more ambitious endeavor. Most effective AI tools nowadays (e.g., deep neural networks) lack explainability even if they have demonstrated incredible super-human performances. However, there are several industrial domains (e.g., medical, transport or military) where explainability is foregrounded and is as crucial as performance, hence such inexplainable AI tools are left aside in these critical domains. To gain more confidence in these tools, RMT creates a new bridge, *from theory and progressively to practice*, which gradually explains and improves AI methods and makes them more accessible, understandable, reliable, hence opening a new way for research and industry along with the current explosion of AI by deep learning. In that sense, pushing forward the development of theoretical tools for the precise understanding of AI methods is of crucial interest to achieve more effective and trustworthy AI.

# Appendix A

# Synthèse de la thèse en Français

L'intelligence artificielle (IA) est connue comme l'ensemble des théories et des techniques utilisées pour créer des machines capables de simuler l'intelligence humaine. L'un des sous-domaines les plus intéressants de l'IA est l'apprentissage machine (ML) qui vise à fournir des algorithmes informatiques qui "apprennent" automatiquement par l'expérience afin de prendre des décisions futures sans être explicitement programmés. Fondamentalement, les algorithmes ML s'appuient sur la construction de modèles mathématiques - très souvent paramétriques - qui seront optimisés sur la base d'échantillons de données d'apprentissage et utilisés ensuite pour effectuer diverses tâches d'IA telles que la classification, la régression, le regroupement, etc.

Tout naturellement, l'IA trouve des applications dans divers domaines et, par conséquent, l'un des défis les plus importants de la ML est de fournir des algorithmes qui peuvent être appliqués à différents types de données (par exemple, des images, des textes, des graphes, etc.). Par construction, ces données peuvent être représentées sous différentes formes et, par conséquent, la performance des algorithmes de ML dépendra largement de la représentation choisie. Cette représentation devrait idéalement contenir des informations pertinentes sur les données afin de permettre l'apprentissage avec des modèles simples et une petite quantité de données. Historiquement, un grand nombre de travaux se sont concentrés sur la conception de représentations (ou de caractéristiques) artisanales, puis sur leur fourniture à des algorithmes de ML simples pour résoudre les tâches souhaitées. Mais pour la plupart des tâches et étant donné les différents types de données, ces approches ne sont pas facilement extensibles pour obtenir une IA efficace.

Depuis l'arrivée des réseaux neuronaux profonds (DNN), l'idée de développer des caractéristiques artisanales a été immédiatement écartée. En effet, les DNN ont surpassé la plupart des approches en démontrant leur incroyable capacité à apprendre automatiquement des représentations pertinentes à partir de données brutes dans un large éventail d'applications, y compris la vision par ordinateur, la reconnaissance de formes et le traitement du langage naturel. Malgré leur succès, de nombreuses questions restent sans réponse concernant les bases théoriques des DNN et qui sont très cruciales notamment pour leur explicabilité. Par exemple, la caractérisation complète de leurs représentations et/ou paramètres appris est encore un problème ouvert.

L'un des principaux aspects qui ont rendu les DNN efficaces dans la pratique est le fait qu'ils soient des modèles sur-paramétrés. En effet, il a été démontré que les architectures profondes de ces modèles surpassent les architectures peu profondes lorsqu'il s'agit de données multi-dimensionnelles (un échantillon de $n$ données de dimension $p$) lorsque ces deux grandeurs $n$ et $p$ sont de grande taille, ce qui est souvent le cas dans les scénarios de

la vie réelle[1]. En outre, les DNN les plus efficaces se trouvent à avoir un certain nombre de paramètres $N$ qui sont au moins de l'ordre de $p$ ou même beaucoup plus grands (par exemple, LeNet-5 [LeC98] contient des paramètres $N = 60000$).

En substance, ces grandes dimensions induisent de nombreux phénomènes contre-intuitifs qui font que les intuitions des petites dimensions s'effondrent complètement. Pour une meilleure compréhension de ces phénomènes, nous fournirons ultérieurement quelques exemples illustratifs qui révèlent ces aspects contre-intuitifs. Dans le cas particulier où les deux $p, n \to \infty$ avec $p/n \to 0 \in (0, \infty)$, la théorie des matrices aléatoires (RMT) fournit des outils puissants pour évaluer la performance de divers algorithmes de ML en tenant compte de l'effet de ces dimensions. En effet, la RMT donne accès au mécanisme interne d'un grand nombre de méthodes de ML, permettant ainsi une compréhension plus approfondie et des améliorations systématiques de ces méthodes. Nous renvoyons le lecteur à la thèse de Z. Liao [Lia19] pour les applications de RMT aux méthodes du noyau, aux réseaux neuronaux aléatoires peu profonds et à la dynamique des réseaux neuronaux ; la thèse de X. Mai [Mai19] qui traite des applications de RMT à l'apprentissage semi-supervisé et aux SVMs.

Les travaux susmentionnés s'appuient largement sur des hypothèses gaussiennes[2] concernant les données traitées. L'un des principaux résultats de cette thèse est d'aller au-delà de l'hypothèse gaussienne pour aborder l'applicabilité de RMT à des données réelles qui ne sont probablement pas proches des vecteurs gaussiens. En particulier, en travaillant sous le modèle statistique plus générique des vecteurs concentrés [LC20], nous fournissons des justifications – en nous appuyant sur les réseaux adversaires générateurs (GAN) – sur la pertinence d'un tel modèle pour la modélisation réaliste des données, et nous analysons en outre, sous l'hypothèse de concentration sur les données, le comportement des grandes matrices de noyaux (qui se trouvent être au coeur de divers algorithmes ML) ainsi que certains composants essentiels des réseaux neuronaux tels que la dernière couche Softmax. Un résultat majeur des travaux développés dans cette thèse est le résultat d'universalité énoncé comme :

> "*seulement les moments d'ordre un et deux importent pour décrire le comportement de ces méthodes*"

justifiant ainsi l'hypothèse de la gaussianité des données selon les résultats de [Lia19, Mai19].

En effet, le premier résultat notable de cette thèse, présenté dans la section 3.1, met en lumière le fait que les données générées artificiellement par le biais des Generative Adversarial Nets (GAN) sont des vecteurs aléatoires qui entrent dans la classe des *vecteurs concentrés*. Par conséquent, les données réelles peuvent être modélisées par cette classe de vecteurs aléatoires – et aussi puisqu'ils généralisent les vecteurs gaussiens – si nous pouvons les assimiler aux données des GAN. Motivés par ce premier résultat, nous avons étudié plus en détail dans cette thèse trois méthodes principales de ML, sous l'hypothèse du mélange de vecteurs concentrés pour le modèle de données d'entrée, où chaque classe de données est décrite par ses moments de premier et de second ordre $\{\boldsymbol{\mu}_\ell\}_{\ell=1}^k$ et $\{\boldsymbol{\Sigma}_\ell\}_{\ell=1}^k$. En effet, en s'appuyant sur la théorie des matrices aléatoires (RMT) et en supposant le régime de grande dimension i.e., lorsque le nombre de données d'entrée et leur dimension sont à la fois grands et comparables, nous avons d'abord analysé dans la section 3.2 le comportement spectral des grandes matrices de Gram qui sont au cœur de diverses méthodes linéaires. Comme des non-linéarités apparaissent dans diverses méthodes de

[1] À titre d'exemple, MNIST [LeC98] contient $n = 70000$ images de dimension $p = 28 \times 28 = 784$.

[2] Modèlisant les données comme un mélange Gaussien de $k$ classes.

ML, nous avons également étudié dans la section 4.1 les méthodes à noyaux par l'analyse de grandes matrices à noyaux. Enfin, nous avons étudié, à travers l'exemple du classificateur Softmax dans la section 5.1, les méthodes de ML qui sont définies par des problèmes d'optimisation (convexes) implicites.

Suite à notre analyse de ces méthodes, nous avons souligné que leurs performances effectives – lorsqu'elles sont appliquées à des données de grande dimension – *seulement* dépendent des classes *moyennes* et *covariances* des données d'entrée, à savoir les statistiques $\{\boldsymbol{\mu}_\ell\}_{\ell=1}^k$ et $\{\boldsymbol{\Sigma}_\ell\}_{\ell=1}^k$, un résultat que nous avons empiriquement vérifié par des expériences approfondies en utilisant les représentations CNN des images générées par un GAN. Ceci constitue un résultat majeur de cette thèse dans le sens où il met en évidence *l'aspect d'universalité* des grands classificateurs ML indépendamment de la distribution des données d'entrée. En conséquence, l'aspect d'universalité soutient notamment la validité du modèle de mélange gaussien dans le régime de la théorie des matrices aléatoires, justifiant ainsi l'hypothèse de mélange gaussien sur les données telle que supposée dans les travaux [Lia19, Mai19]. D'un point de vue pratique, la théorie des matrices aléatoires permet ainsi d'analyser et d'améliorer systématiquement un large éventail de classificateurs ML qui s'appuient sur les méthodes susmentionnées – par l'estimation de leurs performances asymptotiques – sur des données réalistes, telles que des représentations approfondi des images surprenantes générées artificiellement par les GANs.

Pour fournir un exemple de telles analyses et améliorations par RMT, nous avons étudié dans la section 4.2 le problème de l'ACP parcimonieuse où nous avons notamment généralisé des idées récentes pour s'attaquer à ce problème à travers l'analyse d'une classe plus large de matrices à noyaux aléatoires. Plus précisément, notre analyse a permis de comprendre comment les composantes principales peuvent être récupérés de manière cohérente grâce à une matrice à noyau aléatoire de la forme $f(\hat{\boldsymbol{\Sigma}})$, où $\hat{\boldsymbol{\Sigma}}$ est un modèle de covariance à pics et $f$ une fonction lisse. Un autre exemple d'analyse et d'amélioration à travers la théorie des matrices aléatoires concerne l'analyse du Dropout dans les réseaux de neurones. En effet, comme nous l'avons présenté dans la section 5.2, nous avons vu que l'analyse RMT d'un réseau à une couche cachée avec une couche $\alpha$-Dropout a permis de comprendre l'effet de l'opération Dropout, l'améliorant ainsi en suggérant d'abandonner des caractéristiques des données et de les remplacer par une valeur $\alpha \neq 0$ afin de minimiser l'erreur de généralisation du réseau étudié.

En conclusion générale, les outils d'IA les plus efficaces de nos jours (par exemple, les réseaux de neurones profonds) manquent d'explicabilité même s'ils ont démontré des performances surhumaines incroyables. Cependant, il existe plusieurs domaines industriels (par exemple, le médical, le transport ou le militaire) où l'explicabilité est au premier plan et est aussi cruciale que la performance, c'est pourquoi ces outils d'IA inexplicables sont laissés de côté dans ces domaines critiques. Pour gagner plus de confiance dans ces outils, la théorie des matrices aléatoires crée un nouveau pont, de la théorie et progressivement vers la pratique, qui explique et améliore progressivement les méthodes d'IA et les rend plus accessibles, compréhensibles, fiables, ouvrant ainsi une nouvelle voie pour la recherche et l'industrie en même temps que l'explosion actuelle de l'IA par les méthodes d'apprentissage profond. En ce sens, faire avancer le développement d'outils théoriques pour la compréhension précise des méthodes d'IA est d'un intérêt crucial pour parvenir à une IA plus efficace et plus fiable.

# Appendix B

# Practical Contributions

## Contents



## B.1 Generative Collaborative Networks for Super-resolution

This section is based on the following work:

> **(C6) MEA. Seddik**, M. Tamaazousti, J. Lin, "*Generative Collaborative Networks for Single Image SuperResolution*", Neurocomputing'2019.

### B.1.1 Motivation

The super-resolution problem ($\mathcal{P}_{sr}$) consists in estimating a high resolution (HR) image from its corresponding low resolution (LR) counterpart. $\mathcal{P}_{sr}$ finds a wide range of applications and has attracted much attention within the community of computer vision [NM14, YYDN07, ZY12]. Generally, the considered optimization objective of supervised

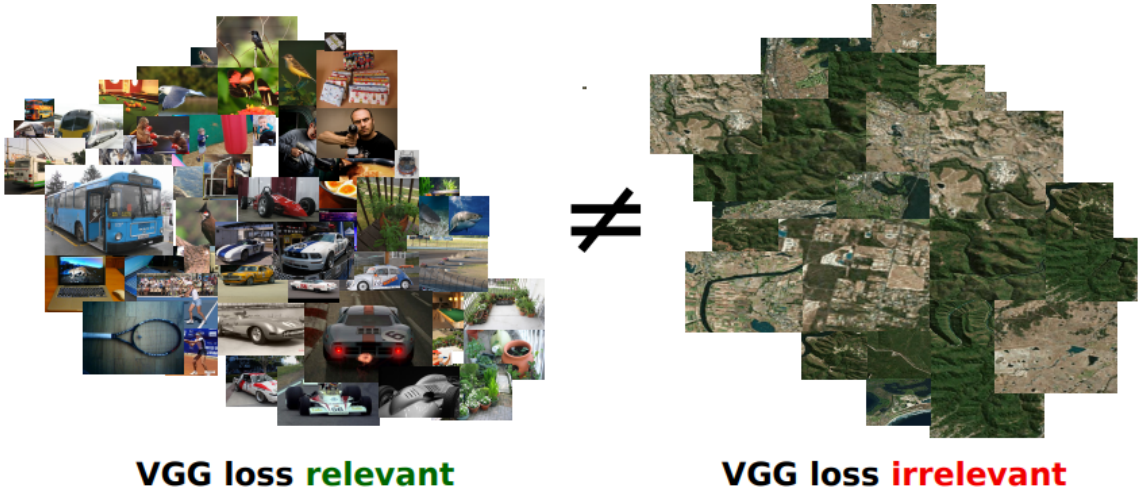



Figure B.1: When super-resolving images from a different domain (*e.g.*, satellite images on the right) than the ImageNet domain (*e.g.*, general objects on the left), the VGG loss introduced by [LTH+16] is no longer relevant. We propose a method that outperforms the SRGAN method [LTH+16] when super-resolving satellite images. Our method falls within a large class of methods which constitutes our proposed *Generative Collaborative Networks* framework.

methods to solve $\mathcal{P}_{sr}$ is the minimization of the mean squared error (MSE) between the recovered HR image and ground-truth. This class of methods are known to be suboptimal to reconstruct texture details at large upscaling factors. In fact, since MSE consists in a pixel-wise images differences, its ability to recover high texture details is limited [LTH+16, GSBB11, WBSS04, WSB03]. Furthermore, the minimization of MSE maximizes the Peak Signal-to-Noise-Ratio (PSNR) metric, which is commonly used for the evaluation of $\mathcal{P}_{sr}$ methods [YMY14].

In order to correctly recover finer texture details when super-resolving at large upscaling factors, a recent (state-of-the-art) work [LTH+16] defined a perceptual loss which is a combination of an adversarial loss and a VGG loss. The former encourages solutions perceptually hard to distinguish from the HR ground-truth images, while the latter consists in using high-level feature maps of the VGG network [SZ14] pre-trained on ImageNet [DDS+09]. When considering the problem of super-resolving images from a target-domain *different* than ImageNet (*e.g.*, satellite images), the features produced by the pre-trained VGG network on the source domain (ImageNet) are suboptimal and no longer relevant for the target domain. In fact, transfer-learning methods are known to be efficient only when the source and target domains are close enough [TLBH17a, TLBH+17b, KQD18]. In this work, we present a general framework which we call *Generative Collaborative Networks* (GCN), where the main idea consists in optimizing the generator (*i.e.*, the mapping of interest) in the feature space of a network which we shall refer to as a *features extractor* network. The two networks are said to be *collaborative* in the sense that the features extractor network "helps" the generator by constructing (here, learning) relevant features. In particular, we applied our framework to the problem of single image super-resolution, and we demonstrated that it results in a method that is more adapted (compared to SRGAN [LTH+16]) when super-resolving images from a domain that is "far" from the ImageNet domain.

The problem of super-resolution has been tackled with a large range of approaches. In the following, we will consider the problem of *single* image super-resolution ($\mathcal{P}_{sisr}$) and thus the approaches that recover HR images from multiple images [BS98b, FREM04] are out of the scope of this paper. First approaches to solve $\mathcal{P}_{sisr}$ were filtering-based methods (*e.g.*, linear, bicubic or Lanczos [Duc79] filtering). Even if these methods are generally very fast, they usually yield overly smooth textures solutions [WBSS04]. Most promising and powerful approaches are learning-based methods which consist in establishing a mapping between LR images and their HR counterparts (supposed to be known). Initial work was proposed by Freeman *et al.* [FJP02]. This method has been improved in [DZSW11, ZEP10] by using compressed sensing approaches. Patch-based methods combined with machine learning algorithms were also proposed: in [TDVG13, TDSVG14] upsampling a LR image by finding similar LR training patches in a low dimensional space (using neighborhood embedding approaches) and a combination of the HR patches counterparts are used to reconstruct HR patches. A more general mapping of example pairs (using kernel ridge regression) was formulated by Kim and Kwon [KK10]. Similar approaches used Gaussian process regression [HS11], trees [SPP15] or Random Forests [SLB15] to solve the regression problem introduced in [KK10]. An ensemble method-based approach was adopted in [DTVG15] by learning multiple patch regressors and selecting the most relevant ones during the test phase.

Convolutional neural networks (CNN)-based approaches outperformed other $\mathcal{P}_{sisr}$ approaches, by showing excellent performance. Authors in [WLY+15] used an encoded sparse representation as a prior in a feed-forward CNN, based on the learned iterative shrinkage and thresholding algorithm of [GL10]. An end-to-end trained three layer deep fully convolutional network, based on bicubic interpolation to upscale the input images, was used in [DLHT14, DLHT16] and achieved good $\mathcal{P}_{sisr}$ performances. Further works suggested that enabling the network to directly learn the upscaling filters, can remarkably increase performance in terms of both time complexity and accuracy [DLT16, SCH+16]. In order to recover visually more convincing HR images, Johnson *et al.* [JAFF16] and Bruna *et al.* bruna2015super used a closer loss function to perceptual similarity. More recently, authors in [LTH+16] defined a perceptual loss which is a combination of an adversarial loss and a VGG loss. The latter consists in minimizing the error between the recovered HR image and ground-truth in the high-level feature space of the pre-trained VGG network [SZ14] on ImageNet [DDS+09]. This method notably outperformed CNN-based methods for the problem $\mathcal{P}_{sisr}$.

## B.1.2 Proposed Methods

### B.1.2.1 Proposed Framework

Consider a problem $\mathcal{P}$ of learning a mapping function $\mathcal{F}$, parameterized by $\theta_{\mathcal{F}}$, that transforms images from a domain $\mathcal{X}$ to a domain $\mathcal{Y}$, given a training set of $N$ pairs $\{(x_i, y_i)\}_{i=1}^{N} \in \mathcal{X} \times \mathcal{Y}$. Denote by $p_{\mathcal{X}}$ and $p_{\mathcal{Y}}$ the probability distributions respectively over $\mathcal{X}$ and $\mathcal{Y}$. In addition, we introduce a given *features extractor* function denoted $\Phi$, parameterized by $\theta_{\Phi}$, that maps an image $y \in \mathcal{Y}$ to a certain euclidean feature space $\mathcal{S}_{\Phi}$ of dimensionality $d$. The mappings $\mathcal{F}$ and $\Phi$ are typically feed-forward Convolutional Neural Networks. The Generative Collaborative Networks (GCN) framework consists in learning the mapping function $\mathcal{F}$ by minimizing a given loss function[1] in the space of

[1] $\ell_2$-loss is considered in the following.

features $\mathcal{S}_\Phi$, between the generated images (through $\mathcal{F}$) and ground-truth. Formally,

$$\hat{\theta}_\mathcal{F} = \arg\min_{\theta_\mathcal{F}} \frac{\lambda_1}{N\,d} \sum_{i=1}^{N} \sum_{j=1}^{d} \left(\Phi_j\left(y_i\right) - \Phi_j\left(\mathcal{F}(x_i)\right)\right)^2 + \lambda_2\,\Omega(\theta_\mathcal{F}), \tag{B.1}$$

where $\Omega(\theta_\mathcal{F})$ is a certain regularization term (detailed below) on the weights $\theta_\mathcal{F}$ and $\lambda_1$ and $\lambda_2$ are summation coefficients. The two networks $\mathcal{F}$ and $\Phi$ are collaborative in the sense that, the latter learns specific features of the domain $\mathcal{Y}$ and "helps" the former, as it is learned in the space $\mathcal{S}_\Phi$. An important question arises about how to learn the mapping $\Phi$. In following, we describe different classes of methods depending on the learning strategy of $\Phi$. In fact, the features extractor function $\Phi$ can take different forms and be learned by different strategies. In particular, we distinguish two learning strategies (illustrated in Figure B.2), which we shall call *disjoint-learning* and *joint-learning*. The four following cases belong to the *disjoint-learning* strategy:

(1.a) When $\Phi$ is the *identity operator* ($\Phi = \mathrm{Id}$). In that case, the objective in equation B.1 becomes a simple pixel-wise MSE loss function. We refer to this class of methods by $\mathcal{P}/mse$.

(1.b) When $\Phi$ corresponds to a *random feature map* neural network, that is to say, the weights $\theta_\Phi$ are set randomly according to a given distribution $\mu$. We refer to this class of methods by $\mathcal{P}/ran$.

(1.c) When $\Phi$ is a part of a model that solves a *reconstruction problem* (jointly with an auxiliary mapping function $\Psi : \mathcal{S}_\Phi \to \mathcal{Y}$), by minimizing the pixel-wise $\ell_2$-loss function between the reconstructed images (through $\Psi$) and ground-truth:

$$(\hat{\theta}_\Phi, \_) = \arg\min_{(\theta_\Phi, \theta_\Psi)} \frac{1}{N\,\dim(\mathcal{Y})} \sum_{i=1}^{N} \sum_{j=1}^{\dim(\mathcal{Y})} \left((y_i)_j - (\Psi \circ \Phi(y_i))_j\right)^2. \tag{B.2}$$

Notably, this strategy allows for the learning of reconstruction features which are different from classification-based features. We refer to this class of methods by $\mathcal{P}/rec$.

(1.d) When $\Phi$ is trained to solve a *multi-label classification problem* [LTH+16], that is to say, when labels are available for the domain $\mathcal{Y}$. More precisely, it exists a dataset $\{(y_i, c_i)\}_{i=1}^n \in \mathcal{Y} \times \{1, \ldots, m\}$ of $n$ images labelled among $m$ classes and $\Phi$ is learned to minimize the following objective:

$$(\hat{\theta}_\Phi, \_) = \arg\max_{(\theta_\Phi, \theta_\Psi)} \mathbb{P}\left\{\Psi \circ \Phi(y_i) = c_i \,|\, y_i\,;\, i \in \{1, \ldots, m\}\right\}, \tag{B.3}$$

where $\Psi : \mathcal{S}_\Phi \to \{1, \ldots, m\}$. We refer to this class of methods by $\mathcal{P}/cla$.

The features extractor function $\Phi$ can also be trained *jointly* with the desired mapping function $\mathcal{F}$. Indeed, as in the GANs paradigm, one can use a discriminator to distinguish the generated images (through $\mathcal{F}$) and ground-truth, and thus learn more relevant and specific features for the problem of interest $\mathcal{P}$. In particular, the *joint-learning* strategy contains two cases:

(2.a) When $\Phi$ is a part of a *discriminator*. $\mathcal{D} = \Psi \circ \Phi : \mathcal{Y} \to \{0, 1\}$ that classifies the generated images (through $\mathcal{F}$) and ground-truth. $\mathcal{D}$ is optimized in an alternating manner along with $\mathcal{F}$ to solve the adversarial min-max problem [SCT+16]:

$$\min_{\theta_\mathcal{F}} \max_{(\theta_\Phi, \theta_\Psi)} \mathbb{E}_{y \sim p_\mathcal{Y}}\left[\log \Psi \circ \Phi(y)\right] + \mathbb{E}_{x \sim p_\mathcal{X}}\left[\log\left\{1 - \Psi \circ \Phi \circ \mathcal{F}(x)\right\}\right]. \tag{B.4}$$

| Standard methods | $\mathcal{P}/mse$ | $\mathcal{P}/cla$ | $\mathcal{P}/rec$ | $\mathcal{P}/dis$ | $\mathcal{P}/dis,rec$ |
|---|---|---|---|---|---|
| Existence | ✓[GSBB11] | ✓[DB16] | ✗ | ✗ | ✗ |
| Adversarial methods | $\mathcal{P}/adv,mse$ | $\mathcal{P}/adv,cla$ | $\mathcal{P}/adv$ | $\mathcal{P}/adv,rec$ | |
| Existence | ✓[YP16] | ✓[LTH+16] | ✗ | ✗ | |

Table B.1: Existent loss functions of the proposed GCN framework.

The adversarial loss (second term of equation B.4) can thus be seen as a regularization of the parameters $\theta_{\mathcal{F}}$ by affecting this quantity to $\Omega(\theta_{\mathcal{F}})$ in equation B.1. This regularization "pushes" the solution of the problem in equation B.1 to the manifold of the images in the domain $\mathcal{Y}$. We refer to this class of methods by $\mathcal{P}/adv$. When $\lambda_2 = 0$, we refer to it by $\mathcal{P}/dis$.

(2.b) When $\Phi$ is a part of a *discriminator* and an *auto-encoder*. Namely, by optimizing its weights $\theta_\Phi$ to solve simultaneously, an *adversarial problem* as in equation B.4; through $\mathcal{D} = \Psi_1 \circ \Phi : \mathcal{Y} \to \{0,1\}$, and a *reconstruction problem* as in equation B.2; through a mapping $\Psi_2 : \mathcal{S}_\Phi \to \mathcal{Y}$. We refer to this class of methods by $\mathcal{P}/adv,rec$ or $\mathcal{P}/dis,rec$ depending on the value of $\lambda_2$ in equation B.1.

#### B.1.2.2 Existing Loss Functions

The natural way to learn a mapping from a manifold to another is to use $\mathcal{P}/mse$ methods. It is well known [GSBB11, LTH+16, WSB03, WBSS04] that this class of methods lead to overly-smooth and poor perceptual quality solutions. In order to handle the mentioned perceptual quality limitation, a variety of methods have been proposed in the literature. First methods used generative adversarial networks (GANs) for generating high perceptual quality images [DCF+15, MCL15], style transfer [LW16] and inpainting [YCL+16], namely the class of methods $\mathcal{P}/adv$ with $\lambda_1 = 0$. Authors in [YP16] proposed to use $\mathcal{P}/mse$ with an adversarial loss ($\lambda_1 > 0$ and $\lambda_2 > 0$) to train a network that super-resolves face images with large upscaling factors. Authors in [BSL15, JAFF16] and in [DB16] used $\mathcal{P}/cla$ by considering respectively $\Phi$ =VGG19 and $\Phi$ =AlexNet networks as fixed features extractors (learned *disjointly* from the mapping of interest), which result in a more perceptually convincing results for both super-resolution and artistic style-transfer [GEB15, GEB16]. More recently, authors in [LTH+16] used $\mathcal{P}/cla,adv$ by considering $\Phi$ =VGG19 as a fixed features extractor combined with an adversarial loss ($\lambda_2 > 0$). To the best of our knowledge, as summarized in table B.1, the use of the other learning strategies of $\Phi$; namely (1.c), (2.a) and (2.b), have not been explored in the literature. We particularly apply these strategies in the context of Single Image Super-Resolution, which results in methods that are more suitable (comparing to the SRGAN method [LTH+16]) to super-resolution domains that differ from the ImageNet domain. The proposed methods as well as the corresponding experiments are presented in the following section.

### B.1.3 Experiments for Single Image Super-Resolution

#### B.1.3.1 Proposed Methods

In this section, we consider the problem of Single Image Super-Resolution ($\mathcal{P}_{sisr}$). In particular, we suppose we are given $N$ pairs $\{(I_i^{LR}, I_i^{HR})\}_{i=1}^N$ of low-resolution images and their high-resolution counterparts. Recalling our GCN framework (presented in the previous section) the proposed methods for the problem $\mathcal{P}_{sisr}$ are: $\mathcal{P}_{sisr}/rec$, $\mathcal{P}_{sisr}/dis$, $\mathcal{P}_{sisr}/dis,rec$, $\mathcal{P}_{sisr}/adv$ and $\mathcal{P}_{sisr}/adv,rec$. We show in the following that the most convincing results are given by $\mathcal{P}_{sisr}/adv,rec$. In particular, we show on a dataset of satellite

images (different from the ImageNet domain) that our method $\mathcal{P}_{sisr}/adv,rec$ outperforms the SRGAN method [LTH+16] by a large margin on the considered domain. Note that, as our goal is to show the irrelevance of the VGG loss for some visual domains (different from ImageNet), we do not consider the well-known SR benchmarks (*e.g.*, Set5, Set14, B100, Urban100) for the evaluation, as these benchmarks are relatively close to the ImageNet domain. The evaluation of the different methods is based on *perceptual metrics* [ZIE+18] which we recall in the following section.

#### B.1.3.2 Evaluation Metrics

The evaluation of super-resolution methods (more generally image regression-based methods) requires comparing visual patterns which remains an open problem in computer vision. In fact, classical metrics such as L2/PSNR, SSIM and FSIM often disagree with human judgments (*e.g.*, blurring causes large perceptual change but small L2 change). Thus, the definition of a *perceptual metric* which agrees with humans perception is an important aspect for the evaluation of $\mathcal{P}_{sisr}$ methods. Zhang *et al.* [ZIE+18] recently evaluated deep features across different architectures (Squeeze [IHM+16], AlexNet [KSH12] and VGG [SZ14]) and tasks (supervised, self-supervised and unsupervised networks) and compared the resulting metrics with traditional ones. They found that deep features outperform all classical metrics (*e.g.*, L2/PSNR, SSIM and FSIM) by large margins on their introduced dataset. As a consequence, deep networks seem to provide an embedding of images which agrees surprisingly well with humans judgments.

Zhang *et al.* [ZIE+18] compute the distance between two images $x, y$ with a network[2] $\Phi$ in the following way:

$$d_{\Phi}(x,y) = \sum_{l} \frac{1}{H_l W_l} \sum_{h,w} \| w_l \odot (\Phi^l(x)_{hw} - \Phi^l(y)_{hw}) \|_2^2, \tag{B.5}$$

where $\Phi^l(\cdot)$ are the extracted features from layer $l$ and unit-normalized in the channel dimension. $w_l$ is a re-scaling vector of the activations channel-wise at layer $l$. $H_l$ and $W_l$ are respectively the height and width of the $l^{th}$ feature map.

Thus, we compute the *perceptual error* (PE) of a $\mathcal{P}_{sisr}$ method (a mapping $\mathcal{F}$) on a given test-set of $N$ low-resolution images and their high-resolution counterparts $\Pi = \{(I_i^{LR}, I_i^{HR})\}_{i=1}^{N}$ as the mean distances between the generated images (through $\mathcal{F}$) and ground-truth as follows:

$$\text{PE}_{\Phi}(\Pi) = \frac{1}{N} \sum_{i=1}^{N} d_{\Phi}(\mathcal{F}(I_i^{LR}), I_i^{HR}). \tag{B.6}$$

Note that we use the implementation of [ZIE+18] to compute the perceptual distances $d_{\Phi}(\cdot,\cdot)$ using six variants which are based on the networks Squeeze [IHM+16], AlexNet [KSH12] and VGG [SZ14] and their "perceptual calibrated" versions. The best method is considered to be the one which minimizes the maximum amount of PEs across different networks $\Phi \in \{\text{Squ, Squ-l, Alex, Alex-l, VGG, VGG-l}\}$.

[2]The considered networks are Squeeze[IHM+16], AlexNet[KSH12] and VGG[SZ14] and their "perceptual calibrated" versions which we refer to respectively as Squeeze-l, AlexNet-l and VGG-l. See [ZIE+18] and the provided Github project within for further details.

#### B.1.3.3 Experiments

The overall goal of this section is to validate our statement about the relevance of the VGG loss when super-resolving images from a different domain than the ImageNet domain. To highlight this aspect, we first present the considered datasets, architectures and training details. Then we select the more appropriate method (across the GCN framework methods) for the $\mathcal{P}_{sisr}$ problem based on perceptual metrics [ZIE+18]. Finally, we compare our proposed method to some baselines and the state-of-the-art SRGAN method [LTH+16], on three different datasets (detailed in the following section). We show in particular that our method outperforms SRGAN on the satellite images domain.

**Datasets.** The idea of replacing the MSE pixel-wise content loss on the image by a loss function that is closer to perceptual similarity is not new. Indeed, [LTH+16] defined a VGG loss on the feature map obtained by a specific layer of the pre-trained VGG19 network and shows that it fixes the inherent problem of overly smooth results which comes with the pixel-wise loss. Nevertheless, VGG19 being trained on ImageNet, their method would not perform particularly well on different images, the distribution of which is far away from that of ImageNet.

Therefore, we propose a similar method where the difference is that our features extractor is not pre-trained, but trained jointly with the generator. This removes the aforementioned limitation since the features extractor is trained on the same dataset as the generator and thus extract relevant features.

To show that, we trained our different networks (*i.e.*, with different features extractors) on three distinct datasets (examples of images of these datasets are shown in Figure B.3):

- A subset of *ImageNet* [DDS+09], for which we sampled 70,000 images. Since VGG19 was trained on ImageNet for many (more than 300K) iterations, we expect to have similar or worse results than the state-of-the-art method SRGAN from [LTH+16] on this database.
- *The Describable Textures Dataset (DTD)* [CMK+14], containing 5,600 images of textural patterns. These data are relatively close to ImageNet and we show that our method gives convincing results relatively close to SRGAN.
- A dataset containing satellite images[3], which we generated by randomly cropping $256 \times 256$ images on a $7205 \times 7205$ satellite image which result in 235,183 images. We particularly show that our method significantly outperforms SRGAN on this dataset. We refer to this dataset by *Sat*.

All experiments are performed with a scale factor of $4\times$ between low- and high-resolutions images and the formers are obtained during the training by down-scaling the original images by a factor $1/4$.

**Architectures.** Our overall goal is to prove that the proposed GCN framework, is adapted to train a generative mapping model and that it surpasses the MSE loss in keeping perceptual similarity in the generated image (whereas the MSE loss tends to smooth things

[3]Can be found in http://www.terracolor.net/sample_imagery.html

out and lose high frequency details). As opposed to [LTH+16]'s work, our framework does not require to have a pre-trained network, like VGG, to extract helpful features for training. In this paper, we focus on the Super Resolution problem. Therefore, we chose our mapping function $\mathcal{F}$, or generator, to be that of Ledig *et al.* [LTH+16]: a feed-forward CNN parametrized by $\theta_{\mathcal{F}}$, composed of 10 residual blocks. These blocks are made of two convolutional layers with $3 \times 3$ kernels and 64 features maps, each followed by batch normalization and PReLU as activation. The image's size is then increased of a factor 4 by two trained upsamplings. The architecture of all the used discriminators follows the guidelines of Radford *et al.* [RMC15] as it is composed of convolutional layers, followed by a batch normalization and a LeakyReLU ($\alpha = 0.2$) activation. This block is repeated eight times and each time the number of $3 \times 3$ kernels increases by a factor 2 (ranging from 64 to 512), a strided convolution is used to reduce the image resolution by 2. Two dense layers and a sigmoid activation then return the discrimination probability. In the case of an auto-encoder (every *Reconstruction* problem), we follow the same architecture for the encoder and a symmetric one for the decoder. Figure B.4 depicts an overview of the architectures for both the generator and the discriminator.

**Training details and parameters.** All networks were trained[4] on a NVIDIA Geoforce GTX 1070 GPU using the considered datasets, which do not contain the $(1000)$ testing images shown as results. We scaled the range of both the LR input images and the HR images to $[-1, 1]$, which explains the tanh activation for the last layer of the generator. All variants of our networks, which differ in their features extractor, were trained from scratch (for the generator and the features extractor) with mini batches of 10 images. We used the Adam optimizer with a learning rate of $2 \cdot 10^{-4}$ and a decay of 0. The generator and the feature extractor are updated alternatively. As we realized training was stable and quite fast, we trained with only $5,000$ update iterations to pinpoint the best method among the different GCNs. Finally, the regularization parameters in our global loss are set by default as $\lambda_1 = 1$ and $\lambda = 10^{-3}$. As a reminder, our goal here is, given a generator architecture (or mapping function $\mathcal{F}$), to find the best strategy to train it, following our GCNs paradigms. The best method is then further compared to baselines.

**Features Extractor Selection.** As we said above, we investigated the ability of different features extractor to construct relevant perceptual feature maps for training and improving the rendering quality of the generator. In order to select the best learning strategy given a certain dataset, we train the generator on each dataset using the different learning strategies: $\mathcal{P}_{sisr}/rec$, $\mathcal{P}_{sisr}/dis$, $\mathcal{P}_{sisr}/dis,rec$, $\mathcal{P}_{sisr}/adv$ and $\mathcal{P}_{sisr}/adv,rec$. Note that, the features extractor for all the considered methods correspond to the first layer of the discriminators (or encoder-decoders). In fact, as the problem $\mathcal{P}_{sisr}$ consists in recovering low-level perceptual cues, we limited our study to the first layer.

Table B.2 summarizes the results of the proposed $\mathcal{P}_{sisr}$ methods in terms of low-level metrics (L2 and SSIM) and perceptual metrics [ZIE+18] which are given by Eq. equation B.6. We notice from this table that the method $\mathcal{P}_{sisr}/adv, rec$ performs relatively well on the datasets ImageNet and Sat in terms of perceptual metrics. While $\mathcal{P}_{sisr}/dis, rec$ gives better results on the DTD dataset. The main difference between these two methods is that the former considers an adversarial loss on the objective function while the latter does not consider the adversarial term. This explains the reason why $\mathcal{P}_{sisr}/adv, rec$ does

[4] A Keras implementation is provided in https://github.com/melaseddik/GCN

not perform well on DTD. In fact, texture images belong to a complex manifold and their distribution is relatively hard to fit by a generative model.

Figure B.5 shows qualitative results of the different proposed methods on the different presented datasets. Generally, the methods which were trained with an additional adversarial loss ($\mathcal{P}_{sisr}/adv$ and $\mathcal{P}_{sisr}/adv,rec$) output images of higher quality (on the datasets ImageNet and Sat) as GANs were introduced to do just so: generate images that follow the distribution of the dataset. Among these two *adversarial* methods, it seems to us (as suggested by the quantitative results of table B.2) that $\mathcal{P}_{sisr}/adv,rec$ (column (c) of Figure B.5) is able to detect and render more details, due to its ability to generate more relevant features as the features extractor $\Phi$ is learned to solve a *multi-task* problem; namely a *discrimination* and a *reconstruction* problem, in particular, this method allows for the learning of both classification and reconstruction-based features. We will thus further investigate the $\mathcal{P}_{sisr}/adv,rec$ method for the comparison to the baseline and the state-of-the-art method SRGAN [LTH+16], on the satellite images domain.

**$\mathcal{P}_{sisr}/adv, rec$ against baseline methods on the satellite images domain.** Our main objective is to show that the VGG loss function (namely, the SRGAN method [LTH+16]) is no longer relevant when super-resolving images from a domain different than the ImageNet domain. In particular, by considering the satellite images domain, we show in this section that the selected method from the previous section ($\mathcal{P}_{sisr}/adv, rec$) outperforms some baselines, which are $\mathcal{P}_{sisr}/mse$ (pixel-wise MSE loss) and $\mathcal{P}_{sisr}/adv,mse$ (pixel wise MSE loss combined with an adversarial loss), and the state-of-the-art super-resolution method, SRGAN [LTH+16]. Note that all the methods use the same architectures (depicted in figure B.4) for the generator and discriminator and are trained on the same domain (here, on satellite images). Our purpose being to show the relevance of the proposed method on a domain "far" from the ImageNet domain, we do not consider standard SR benchmarks, which are raltively "close" to the ImageNet domain.

Table B.3 presents quantitative results, in terms of classical metrics (L2 and SSIM) and perceptual metrics given by equation B.6, of the different methods on the Sat dataset. As we can notice, our method $\mathcal{P}_{sisr}/adv, rec$ outperforms the other methods in terms of perceptual metrics. Knowing that the perceptual metrics agree with human judgments [ZIE+18], these results validate the effectiveness of the $\mathcal{P}_{sisr}/adv, rec$ method. Note also that even if SRGAN [LTH+16] is optimized to minimize a VGG loss, it does not give the lowest perceptual errors in terms of the perceptual metrics VGG and VGG-l, this is due to the fact that the VGG features are not relevant for the satellite images domain. In addition, $\mathcal{P}_{sisr}/adv, rec$ gives the lowest perceptual errors in terms of the perceptual metrics Alex and Alex-l which agrees with a human perception. In fact, AlexNet network may more closely match the architecture of the human visual cortex [YD16].

Figure B.6 shows some qualitative results of different methods on a patch of an image from the Sat dataset. As we can notice, the $\mathcal{P}_{sisr}/adv, rec$ method gives the perceptually closest result to the ground-truth image, which agrees with the quantitative results of table B.3.

**Further results.** In this section, we provide further qualitative and quantitative comparisons to the considered baselines of the previous section. In particular, we consider

all the presented datasets for the comparisons. Qualitative results are provided in figure B.7. SRGAN performs better on ImageNet, which is not that surprising considering our features extractor was trained much less than VGG19 used in [LTH+16] and the VGG features being more relevant for images from the ImageNet domain. Nonetheless, we do have sharper images than the MSE based methods, although we show some artifact (especially on the boat) which we attribute to the competition between the content and adversarial losses. On DTD though, we can see the benefit of our method over a pre-trained VGG loss. Indeed, SRGAN is blurrier on both the house (first row) and the cliff (third row), in spite of having less artifacts than our method. On the "cracks" example (second row), SRGAN even totally obliterates the details in the center. Finally, results on the dataset Sat, which is the most different dataset compared to ImageNet, are the most compelling. Our method generates super resolved images that are really close to the real high resolution images, while we can clearly see imperfections on SRGAN's results because of VGG19 which was not trained to detect perceptual features on satellite images.

Quantitative results are summarized in Table B.4. As shown in [LTH+16, ZIE+18], the standard quantitative measures such as L2 and SSIM fail to highlight image quality according to the human visual system. In fact, while the results of $\mathcal{P}_{sisr}/mse$ are overly smooth perceptually, it has the lowest L2 and SSIM errors on Sat. However, perceptual metrics agree with what we assess qualitatively: SRGAN performs best on ImageNet but not on Sat, the distribution of which is the farthest from ImageNet. Actually, SRGAN ranks third of all four methods on Sat, just before $\mathcal{P}_{sisr}/adv,mse$, while still performing best on DTD which still is pretty close to ImageNet. This shows that the VGG features become less and less relevant as the dataset's distribution part from ImageNet. On the other hand, our training framework allows to construct relevant features on any (never seen) dataset. Thus our method $\mathcal{P}_{sisr}/adv,rec$ performs best on Sat. Our method performing better than $\mathcal{P}_{sisr}/adv,mse$ also shows that our framework helps finding detail preserving features. Figure B.7 provides the results of the different baselines and our method on some examples of the considered datasets. We notice from these images that our method $\mathcal{P}_{sisr}/adv, rec$ recovers finer details on the different datasets while it outperforms the considered baselines on satellite images.

Table B.5 summarizes the results of the different methods on the considered datasets through the paper. From these results, we make the following conclusions:

- When the considered domain is far enough from the ImageNet domain, the VGG loss introduced by [LTH+16] is no longer relevant.
- The VGG network can not be fine-tuned when considering a domain for which there is no available labels for the images (*e.g.*, satellite images). Thus, the SRGAN method cannot be exploited efficiently in this case.
- Our framework results in a method ($\mathcal{P}_{sisr}/adv, rec$) that outperforms some baselines and the SRGAN method on the satellite images domain.
- Even on a domain close to the ImageNet domain (*e.g.*, texture images), one can find within our framework methods which give almost similar results to the SRGAN method, while the later is based on VGG features and thus need to train the VGG network on the whole ImageNet dataset.

### B.1.4 Central Contribution and Discussions

In this work, we have proposed a general framework named Generative Collaborative Networks (GCN) which generalizes the existing methods for the problem of learning a mapping between two domains. The GCN framework highlights that there is a learning strategy of mappings that is not explored in the literature. In particular, the optimization of these mappings in the feature space of a features extractor network, which is mutually learned at the same time as the considered mapping (*joint-learning* strategy). The GCN framework was evaluated in the context of super-resolution on three datasets (ImageNet [DDS+09], *DTD* [CMK+14] and satellite images). We have shown that the proposed *joint-learning* strategy leads to a method that outperforms the state of the art [LTH+16] which uses a pre-trained features extractor network (VGG19 on ImageNet). Specifically, this holds when the domain of interest is "far" from the ImageNet domain (*e.g.*, satellite images or images from the medical domain[5]). However, note that even for domains close to the ImageNet domain, the proposed method gives convincing (almost similar to [LTH+16]) results without using the whole ImageNet dataset to learn the features extractor network (as performed in [LTH+16]).

We systematically designed the proposed methods by using the first layer of the features extractor networks, while it could be interesting to evaluate in more detail the impact of this choice regarding the learning strategy. Moreover, the impact of the selected layer may also depend on the considered dataset. More generally, the GCN framework offers a large vision on the wide variety of existing loss functions used in the literature of learning mappings-based problems (*e.g.*, super-resolution, image completion, artistic style transfer, etc.). In fact, we show that these loss functions can be simply reformulated, in the proposed framework, as a certain combination of a particular type of features extractor networks ($\mathcal{P}$ /*rec*, $\mathcal{P}$ /*dis*, $\mathcal{P}$ /*dis,rec*, $\mathcal{P}$ /*adv* and $\mathcal{P}$ /*adv,rec*) and a particular learning strategies (*joint-learning* or *disjoint-learning*). Therefore it will be interesting to explore this promising framework in other learning mappings-based problems.

## B.2 Neural Networks Compression

This section is based on the following work:

**(C7)** **MEA. Seddik**, H.Essafi, A.Benzine, M.Tamaazousti, "*Lightweight Neural Networks from PCA & LDA Based Distilled Dense Neural Networks*", International Conference on Image Processing (ICIP'20), Online, 2020.

### B.2.1 Motivation

Neural networks are the most effective machine learning methods nowadays, and since they generally require millions of parameters, their implementation in an IoT environment is quite ineffective. Indeed, IoT requires machine learning models with limited amount of parameters since the edge devices have limited resources (computation capacity, storage, bandwidth,...). This requirement among others have triggered intensive research activities and led to the emergence of new computing paradigms, i.e. edge computing [Sat17] which has emerged as an answer to the need for shifting the computing

[5]This domain is particularly relevant for the proposed framework as it seems very far from the ImageNet domain. Unfortunately, we have not found a big amount of publicly available data (to the best of our knowledge) for medical images which prevented us from considering this domain through the paper.

from cloud to decentralized processing units close to the data sources [ASEA19]. The authors in [MMH+19] give a survey of works dealing with machine learning at the network edge. AI at the edge [LLP+19] is a concrete example of leveraging the computing and storage resources near the places where data is produced. The authors in [VTWA18] present a set of approaches proposed for embedding deep learning into the edge computing devices. They also present some applications that can fit with the edge computing paradigm and can take benefit from the edge network. However, the accuracy of deep learning models depends largely on the hyper-parameters of the network in particularly the number and the size of layers.

Nevertheless, big models are resources consuming which can impede the embedding of AI technologies in IoT devices with constrained resources. To tackle this issue some interesting solutions were proposed, model compression was among the first proposed methods. Model compression methods use well proved compression techniques for reducing the storage volume required by neural networks without impacting their performance. For instance the algorithm presented in [HMD15] is composed of three methods applied in pipeline: first pruning (selection of the important weights of the network) method is applied, followed by quantization and Huffman coding (compression without lost). Compression methods are useful for reducing the required storage space of network models but inefficient for reducing the computation power which is one of the main critical aspects of the IoT devices. Recent methods targeting the production of smaller models, with the accuracy near of lager ones, was investigated [MFLG19].

In this study, we present two methods for distilling a given large dense neural network into a smaller one. The proposed methods are based on knowledge distillation concept [YJBK17], where a large (teacher) pre-trained network is used to train a smaller (student) network. They have the advantage to produce models that consume less storage and computing resources; the knowledge distillation approach is a kind of transfer learning approach which is commonly used in various machine learning problems [TLBH+19, MTS+19]. In [MFLG19] the authors show that the accuracy of the student network model depends highly on the ratio size between the two network models (teacher and student), higher is this ratio (gap size between teacher and student is large), smaller is the accuracy. To alleviate this problem the authors suggest, instead of distilling the student model directly from the teacher model, to use succession of teacher assistants based knowledge distillation approach where the models are distilled step by step until obtaining the final model. The proposed methods in this paper are notably complementary to the methods in [HMD15, MFLG19].

### B.2.2 Proposed Methods

#### B.2.2.1 Setting & Notations

Consider a dense neural network, which we refer to as the teacher network (TN), composed of $L$ layers and constructed in the following way, for $\ell \in [L]$:

$$\textbf{(TN)}: \begin{cases} \boldsymbol{h}^{(0)} = \boldsymbol{x} \in \mathbb{R}^{p_0}, \\ \boldsymbol{h}^{(\ell)} = f_\ell \left( \boldsymbol{W}^{(\ell)} \boldsymbol{h}^{(\ell-1)} + \boldsymbol{b}^{(\ell)} \right) \in \mathbb{R}^{p_\ell}, \end{cases} \tag{B.7}$$

where, $\boldsymbol{x}$ corresponds to the input data features, $f_\ell$ denotes the $\ell$-th layer activation, $\boldsymbol{h}^{(\ell)}$ stands for the features of $\boldsymbol{x}$ extracted at layer $\ell$, $\boldsymbol{W}^{(\ell)} \in \mathbb{R}^{p_\ell \times p_{\ell-1}}$ and $\boldsymbol{b}^{(\ell)} \in \mathbb{R}^{p_\ell}$ are respectively the weight matrix and bias at each layer $\ell$. TN is typically of large size, meaning

that the hidden features dimensions $p_\ell$ are relatively large. In the following, we will present two methods that construct a small network size, which we refer to as the student network (SN), based on the TN learned features. The two methods target different learning problems, depending if TN solves a supervised problem or an unsupervised one.

#### B.2.2.2 Neural Nets PCA-based Distillation (Net-PCAD)

Given a set of $n$ training samples $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{p_0 \times n}$ on which TN was initially trained to perform some arbitrary learning task. The Net-PCAD method consists in performing a PCA [STC19b] at each hidden layer of TN, and then training a SN to perform the same learning task as TN along with the task of mapping its hidden features at each layer with the reduced features of TN. Formally, we denote by $\boldsymbol{H}_\ell = [\boldsymbol{h}_1^{(\ell)}, \ldots, \boldsymbol{h}_n^{(\ell)}] \in \mathbb{R}^{p_\ell \times n}$ where $\boldsymbol{h}_i^{(\ell)}$ stands for the features of $\boldsymbol{x}_i$ at layer $\ell$. Therefore, a PCA is performed at each layer $\ell$ in order to reduce the dimension of the hidden features $p_\ell$, relying on the top $k_\ell$ largest eigenvectors of the sample covariance matrix:

$$\boldsymbol{C}_\ell = \frac{1}{n} \sum_{i=1}^{n} \bar{\boldsymbol{h}}_i^{(\ell)} \bar{\boldsymbol{h}}_i^{(\ell)\intercal} \tag{B.8}$$

where $\bar{\boldsymbol{h}}_i^{(\ell)} = \boldsymbol{h}_i^{(\ell)} - \frac{1}{n} \sum_{j=1}^{n} \boldsymbol{h}_j^{(\ell)}$ are the centred hidden features. We denote by $\boldsymbol{U}_\ell \in \mathbb{R}^{p_\ell \times k_\ell}$ the matrix containing the $k_\ell$ largest eigenvectors of $\boldsymbol{C}_\ell$. Consequently, the student network (SN) is composed of $L$ layers and has the following structure:

$$\textbf{(SN)}: \begin{cases} \tilde{\boldsymbol{h}}^{(0)} = \boldsymbol{x} \in \mathbb{R}^{p_0}, \\ \tilde{\boldsymbol{h}}^{(\ell)} = f_\ell \left( \tilde{\boldsymbol{W}}^{(\ell)} \tilde{\boldsymbol{h}}^{(\ell-1)} + \tilde{\boldsymbol{b}}^{(\ell)} \right) \in \mathbb{R}^{k_\ell}, \end{cases} \tag{B.9}$$

with the convention $k_0 = p_0$ and where, $\boldsymbol{x}$ corresponds to the input data features, $\tilde{\boldsymbol{h}}^{(\ell)}$ stands for the features of $\boldsymbol{x}$ extracted at layer $\ell$, $\tilde{\boldsymbol{W}}^{(\ell)} \in \mathbb{R}^{k_\ell \times k_{\ell-1}}$ and $\tilde{\boldsymbol{b}}^{(\ell)} \in \mathbb{R}^{k_\ell}$ are respectively the weight matrix and bias at each layer $\ell$.

Given the initial learning problem of TN which corresponds to a loss function $\mathcal{L}_{\text{problem}}$, SN is therefore optimized with the following loss function, where the Homoscedastic loss [KGC18] is considered since the optimization problem for SN can be formulated as a multi-task problem.

$$\mathcal{L} = e^{-\sigma_{\text{problem}}} \mathcal{L}_{\text{problem}} + \sigma_{\text{problem}} + \sum_{\ell=1}^{L-1} e^{-\sigma_\ell} \mathcal{L}_{\text{mse}}(\tilde{\boldsymbol{h}}^{(\ell)}, \boldsymbol{U}_\ell^\intercal \boldsymbol{h}^{(\ell)}) + \sigma_\ell \tag{B.10}$$

where $\mathcal{L}_{\text{mse}}$ denotes the mean squared error loss function, $\sigma_{\text{problem}}$ and $\sigma_\ell$'s are the Homoscedastic loss parameters which are learned during the training of the student network. A full description of the Net-PCAD method is provided as a pseudo-code algorithm in Algorithm 1.

#### B.2.2.3 Neural Nets LDA-based Distillation (Net-LDAD)

If the initial learning problem of the TN is a supervised classification problem, one can take advantage of the fact that the data belong to $K$ different classes $\{\mathcal{C}_j\}_{j=1}^{K}$ and therefore project the hidden features of the TN in structured low-dimensional spaces. Linear Discriminant Analysis (LDA) is a dimension reduction technique that specifically reduces

**Algorithm 1:** Net-PCAD description.

**Input:** A trained teacher network **TN**, a data matrix $\boldsymbol{X}$ and the learning problem loss $\mathcal{L}_{\text{problem}}$.
**Output:** Trained student network **SN**.
**for** $\ell \leftarrow 1$ ***to*** $L-1$ **do**
  1. Extract the representations $\boldsymbol{H}_\ell$ of $\boldsymbol{X}$ from **TN**;
  2. Compute $\boldsymbol{U}_\ell$ through a PCA on $\boldsymbol{H}_\ell$;
**end**
Train the student network **SN** with $\mathcal{L}$ as in equation B.10;

the dimension of data relying on their classes structure [TGIH17]. LDA is closely related to PCA but differs from the latter by the fact that it explicitly attempts to model the difference between the classes of the data, while PCA does not take into account any difference in class labels. Therefore, the idea behind Net-LDAD is to exploit the labels information layer-wise in the training of the student network from the teacher network. Specifically, we compute at each layer $\ell$ of the TN the within class scatter matrix as:

$$\boldsymbol{S}_w^{(\ell)} = \sum_{j=1}^{K} \sum_{\boldsymbol{x} \in \mathcal{C}_j} (\boldsymbol{h}_{\boldsymbol{x}}^{(\ell)} - \boldsymbol{m}_j^{(\ell)})(\boldsymbol{h}_{\boldsymbol{x}}^{(\ell)} - \boldsymbol{m}_j^{(\ell)})^\intercal \tag{B.11}$$

where $\boldsymbol{h}_{\boldsymbol{x}}^{(\ell)}$ is the representation of $\boldsymbol{x}$ at layer $\ell$ of the TN and $\boldsymbol{m}_j^{(\ell)} = \frac{1}{|\mathcal{C}_j|} \sum_{\boldsymbol{x} \in \mathcal{C}_|} \boldsymbol{h}_{\boldsymbol{x}}^{(\ell)}$. And the between class scatter matrix at each layer $\ell$ is given by:

$$\boldsymbol{S}_b^{(\ell)} = \sum_{j=1}^{K} |\mathcal{C}_j| (\boldsymbol{m}_j^{(\ell)} - \boldsymbol{m}^{(\ell)})(\boldsymbol{m}_j^{(\ell)} - \boldsymbol{m}^{(\ell)})^\intercal \tag{B.12}$$

where $\boldsymbol{m}^{(\ell)} = \frac{1}{n} \sum_{\boldsymbol{x} \in \boldsymbol{X}} \boldsymbol{h}_{\boldsymbol{x}}^{(\ell)}$. Therefore, the projection matrix of LDA at each layer $\ell$ is computed as the $k_\ell$ largest eigenvectors of $(\boldsymbol{S}_w^{(\ell)})^{-1}\boldsymbol{S}_b^{(\ell)}$. We denote by $\boldsymbol{V}_\ell \in \mathbb{R}^{p_\ell \times k_\ell}$ such a projection matrix. Similarly to the PCA case, the student network is therefore trained to minimize the following objective:

$$\begin{aligned} \mathcal{L} = {} & e^{-\sigma_{\text{classification}}} \mathcal{L}_{\text{classification}} + \sigma_{\text{classification}} \\ & + \sum_{\ell=1}^{L-1} e^{-\sigma_\ell} \mathcal{L}_{\text{mse}}(\tilde{\boldsymbol{h}}^{(\ell)}, \boldsymbol{V}_\ell^\intercal \boldsymbol{h}^{(\ell)}) + \sigma_\ell \end{aligned} \tag{B.13}$$

where $\mathcal{L}_{\text{classification}}$ is typically a categorical cross entropy loss function since the initial learning problem of the TN is supposed to be a classification problem. A full description of the Net-LDAD method is provided as a pseudo-code algorithm in Algorithm 2.

### B.2.3 Experiments

In this section, we present experiments which highlight the effectiveness of the proposed methods to train student networks that are smaller in size w.r.t.a given large size dense teacher network. In particular, we consider three teacher networks composed of $L = 4$ dense layers which are trained to perform a classification problem respectively on the datasets MNIST [YCC98], Fashion-MNIST [XRV17] and CIFAR10 [KH10]. Therefore, we train the corresponding student networks by successively reducing their hidden dimensions using the presented methods Net-PCAD and Net-LDAD. Note that, for simplicity,

**Algorithm 2:** Net-LDAD description.

**Input:** A trained teacher network **TN**, a data matrix $\boldsymbol{X}$ and the learning problem loss $\mathcal{L}_{\text{problem}}$.
**Output:** Trained student network **SN**.
**for** $\ell \leftarrow 1$ ***to*** $L-1$ **do**
  1. Extract the representations $\boldsymbol{H}_\ell$ of $\boldsymbol{X}$ from **TN**;
  2. Compute $\boldsymbol{V}_\ell$ through a LDA on $\boldsymbol{H}_\ell$;
**end**
Train the student network **SN** with $\mathcal{L}$ as in equation B.13;

we reduce all the hidden dimensions to a constant value $k$ and we vary $k$ in all our experiments. Table B.6 presents the considered architectures for the teacher and student networks.

Figure B.8 depicts the training Homoscedastic loss of the student networks for different values of $k$ and across the different considered datasets using our methods Net-PCAD and Net-LDAD. Note from this figure that both methods yield generally to a stable learning of the student networks, and the classification problem gets much easier as $k$ increases. However, note that a careful choice of $k$ must be made in order to get a smooth learning loss (e.g., see $k = 200$ on the Fashion-MNIST dataset).

Figure B.9 depicts the learned Homoscedastic parameters once the student networks have been trained for different values of $k$. We can observe from this figure (at least for the datasets MNIST and Fashion-MNIST), that the weight $e^{-\sigma_{\text{classification}}}$ is much larger than the weights on the hidden features for the Net-PCAD method, while all weights have the same order of magnitude for the Net-LDAD method. This can be interpreted by the fact that LDA finds layer wise a low dimensional space where data can be classified and therefore "helps" the classification learning problem. This is not the case regarding the curves for the CIFAR10 dataset, since it is a "hard" classification problem given the architecture of the teacher network (TN gets 45% accuracy).

In terms of the test accuracy, we note from Figure B.10 that learning the students networks with our methods improves largely their generalization capacities compared with learning them from scratch, and one can see that they even surpass the teacher network on the CIFAR10 dataset. As a summary, Net-PCAD and Net-LDAD yield to better generalization performances of the student networks as the learning problem gets harder in the sense of the teacher network test accuracy, knowing that classifying MNIST is an easy problem (TN gets 98%), Fashion-MNIST medium (TN gets 88%) while classifying CIFAR10 is a harder learning problem (TN gets 45%).

Table B.7 summarizes the performances of the learned student networks using the Net-PCAD method[6], in terms of the forward execution time and test accuracy. As we can notice, Net-PCAD yields to an accurate speedup of inference time (depending on the choice of $k$) while not degrading the accuracy of the learned student networks w.r.t.the teacher network and even surpassing the teacher network's accuracy in the case of hard classification problems (see CIFAR10).

[6]Net-LDAD gets similar results.

### B.2.4 Central Contribution and Discussions

In this work we presented two methods to distillate a given teacher network (TN) into a student network (SN). Our methods improve the performance of SN compared to learning SN from scratch and even surpasses TN performances when the learning problem gets hard, therefore the resulting learned SN is suited to be implemented in an edge IoT device which requires limited resources. Note that the presented methods need to setup an hyper-parameter $k_\ell$ which will be addressed in an extended version of this work.

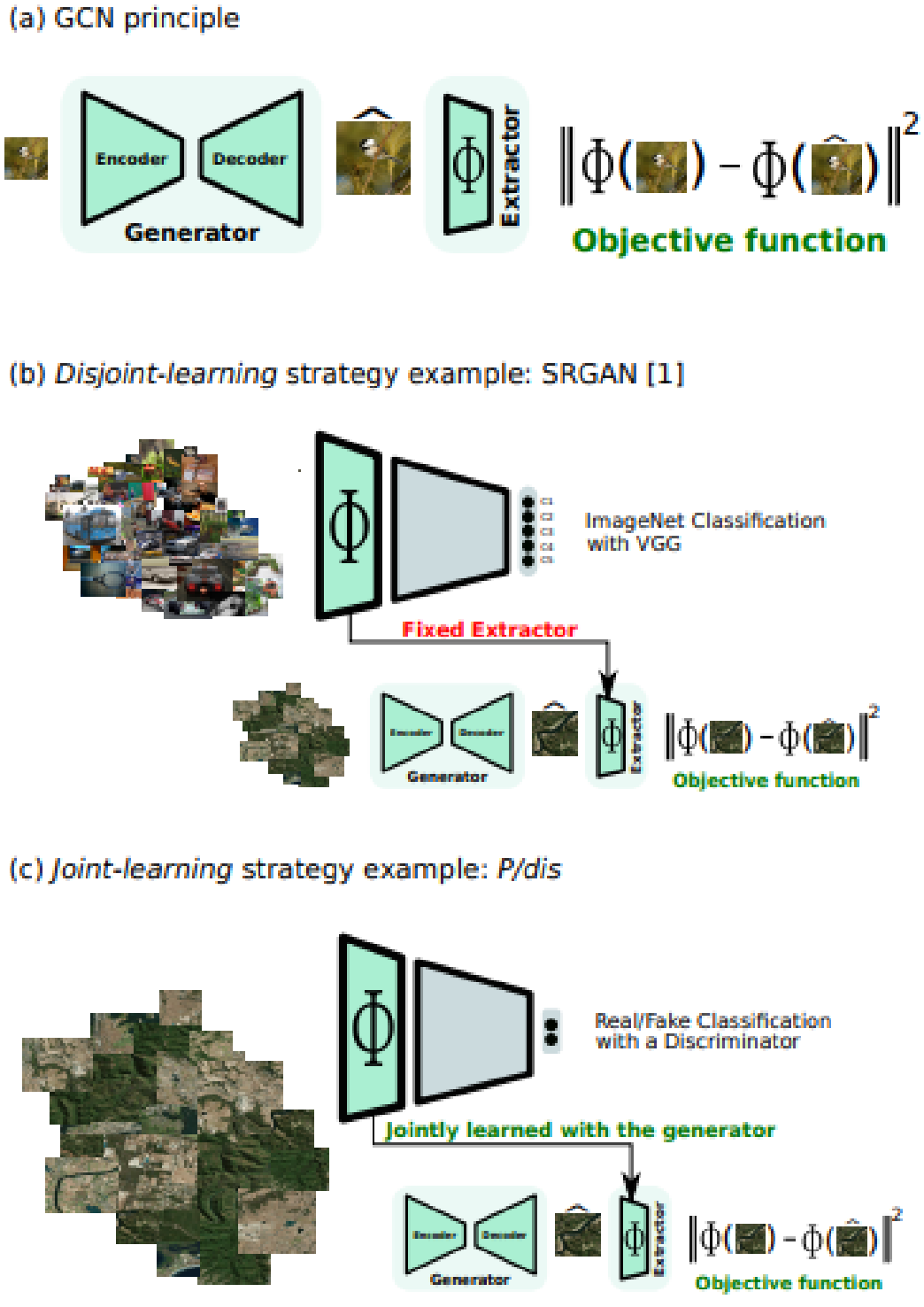



Figure B.2: Overview of the GCN framework with examples of the two learning strategies. The GCN framework consists in optimizing a *generator* in the feature space of an *extractor* as illustrated in **(a)**. The extractor can be trained beforehand and used to optimize the generator, which we refer to as *disjoint-learning* strategy **(b)**. The extractor can also be optimized jointly with the generator, i.e., using a *joint-learning* strategy **(c)**.

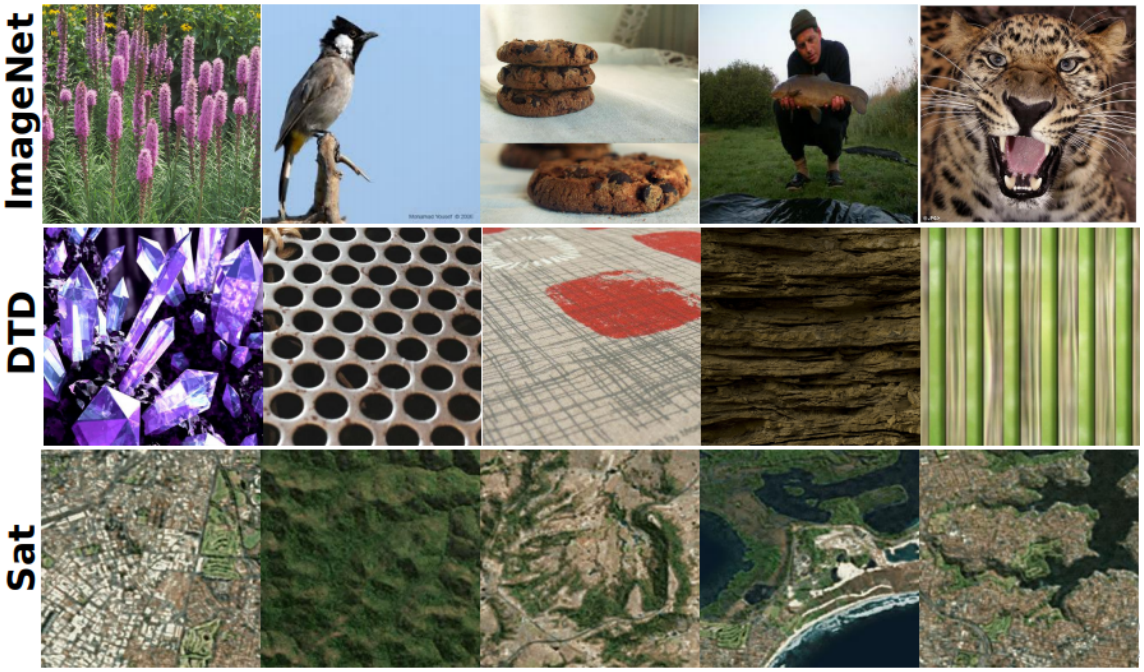



Figure B.3: Examples of images from the considered datasets.

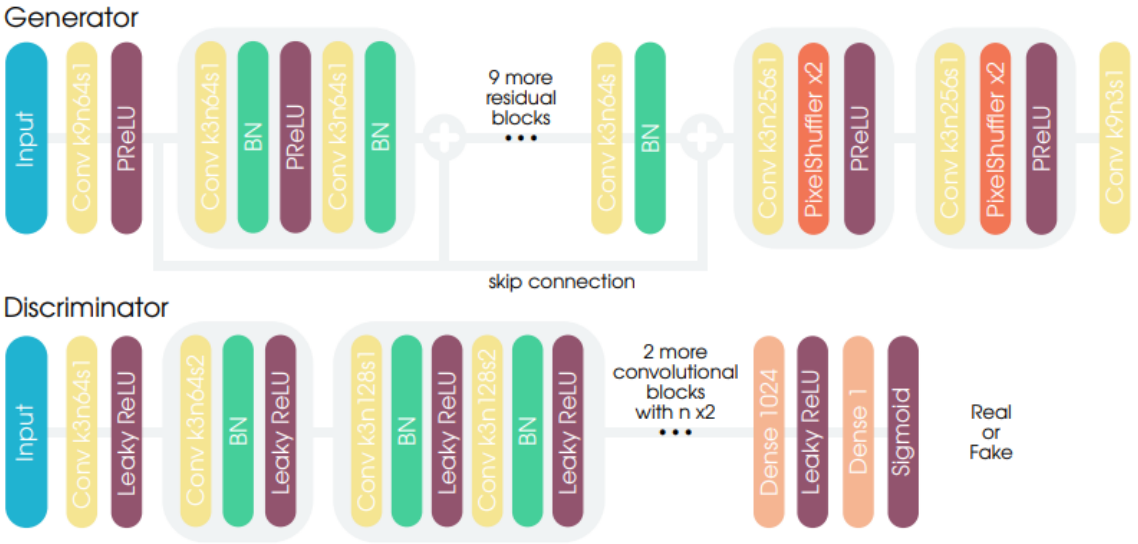



Figure B.4: Overview of the used architectures for the generator and the discriminator. We have considered the same architectures as that of Ledig *et al.* [LTH+16].

| | | Low-level | | Perceptual metrics | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Methods** | L2 | SSIM | Squ | Squ-l | Alex | Alex-l | VGG | VGG-l |
| ImageNet | $\mathcal{P}_{sisr}/dis$ | 0.018 | 0.147 | 1.606 | 0.279 | 1.470 | 0.398 | 2.088 | 0.358 |
| | $\mathcal{P}_{sisr}/rec$ | 0.020 | 0.162 | 1.723 | 0.301 | 1.595 | 0.425 | 2.243 | 0.388 |
| | $\mathcal{P}_{sisr}/dis,rec$ | 0.017 | 0.147 | 1.587 | 0.279 | 1.420 | 0.382 | 2.052 | 0.353 |
| | $\mathcal{P}_{sisr}/adv$ | 0.028 | 0.202 | 1.820 | **0.222** | 1.554 | **0.322** | 2.598 | 0.432 |
| | $\mathcal{P}_{sisr}/adv,rec$ | **0.016** | **0.141** | **1.533** | 0.263 | **1.362** | 0.368 | **1.994** | **0.340** |
| DTD | $\mathcal{P}_{sisr}/dis$ | 0.027 | 0.184 | 1.873 | 0.327 | 1.739 | 0.440 | 2.401 | 0.421 |
| | $\mathcal{P}_{sisr}/rec$ | 0.027 | 0.183 | 1.851 | 0.320 | 1.726 | 0.438 | 2.398 | 0.420 |
| | $\mathcal{P}_{sisr}/dis,rec$ | **0.023** | **0.167** | **1.703** | 0.292 | **1.576** | 0.404 | **2.260** | **0.392** |
| | $\mathcal{P}_{sisr}/adv$ | 0.036 | 0.227 | 2.077 | 0.281 | 1.812 | 0.375 | 2.770 | 0.473 |
| | $\mathcal{P}_{sisr}/adv,rec$ | 0.046 | 0.236 | 2.089 | **0.277** | 1.793 | **0.344** | 2.796 | 0.481 |
| Sat | $\mathcal{P}_{sisr}/dis$ | **0.011** | **0.129** | 1.484 | 0.210 | 1.508 | 0.356 | 2.121 | 0.355 |
| | $\mathcal{P}_{sisr}/rec$ | 0.060 | 0.168 | 1.705 | 0.245 | 1.762 | 0.423 | 2.260 | 0.395 |
| | $\mathcal{P}_{sisr}/dis,rec$ | **0.011** | 0.138 | 1.493 | 0.215 | 1.435 | 0.351 | **2.108** | 0.372 |
| | $\mathcal{P}_{sisr}/adv$ | 0.030 | 0.214 | 1.719 | 0.181 | 1.627 | 0.306 | 2.711 | 0.419 |
| | $\mathcal{P}_{sisr}/adv,rec$ | 0.018 | 0.183 | **1.359** | **0.140** | **1.310** | **0.220** | 2.115 | **0.344** |

Table B.2: Results of the proposed $\mathcal{P}_{sisr}$ methods in terms of traditional metrics (L2 and SSIM) and the *perceptual error* (PE) given by Eq. equation B.6 on different datasets. As we can notice, the method $\mathcal{P}_{sisr}/adv,rec$ outperforms the other methods in the datasets ImageNet and Sat, while $\mathcal{P}_{sisr}/dis,rec$ gives the best results on DTD.

| | | Low-level | | Perceptual metrics | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Methods** | L2 | SSIM | Squ | Squ-l | Alex | Alex-l | VGG | VGG-l |
| Sat | $\mathcal{P}_{sisr}/mse$ | **0.011** | **0.134** | 1.873 | 0.245 | 1.855 | 0.411 | 2.536 | 0.419 |
| | $\mathcal{P}_{sisr}/adv,mse$ | 0.082 | 0.197 | 1.458 | 0.205 | 1.466 | 0.352 | 2.125 | 0.347 |
| | SRGAN [LTH+16] | 0.228 | 0.188 | 1.510 | 0.220 | 1.361 | 0.282 | 2.230 | 0.412 |
| | $\mathcal{P}_{sisr}/adv,rec$ | 0.018 | 0.183 | **1.359** | **0.140** | **1.310** | **0.220** | **2.115** | **0.344** |

Table B.3: Comparison of our method $\mathcal{P}_{sisr}/adv,rec$ with baselines and the SRGAN method [LTH+16] on the satellite images domain, in terms of classical metrics (L2 and SSIM) and perceptual metrics [ZIE+18].

| | | Low-level | | Perceptual metrics | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Methods** | L2 | SSIM | Squ | Squ-l | Alex | Alex-l | VGG | VGG-l |
| ImageNet | $\mathcal{P}_{sisr}/mse$ | 0.017 | 0.146 | 1.568 | 0.280 | 1.435 | 0.391 | 2.064 | 0.349 |
| | $\mathcal{P}_{sisr}/adv,mse$ | 0.020 | 0.156 | 1.634 | 0.241 | 1.397 | 0.329 | 2.223 | 0.384 |
| | SRGAN | 0.028 | 0.170 | **1.303** | **0.177** | **1.084** | **0.225** | 2.045 | 0.342 |
| | $\mathcal{P}_{sisr}/adv,rec$ | **0.016** | **0.141** | 1.533 | 0.263 | 1.362 | 0.368 | **1.994** | **0.340** |
| DTD | $\mathcal{P}_{sisr}/mse$ | 0.029 | 0.185 | 1.972 | 0.342 | 1.856 | 0.470 | 2.479 | 0.434 |
| | $\mathcal{P}_{sisr}/adv,mse$ | 0.025 | 0.188 | 1.880 | 0.268 | 1.586 | 0.349 | 2.512 | 0.430 |
| | SRGAN | 0.031 | 0.191 | **1.557** | **0.209** | **1.298** | **0.241** | 2.308 | 0.393 |
| | $\mathcal{P}_{sisr}/dis,rec$ | **0.023** | **0.167** | 1.703 | 0.292 | 1.576 | 0.404 | **2.260** | **0.392** |

Table B.4: Comparison of our methods $\mathcal{P}_{sisr}/adv,rec$ and $\mathcal{P}_{sisr}/dis,rec$ with baselines and the SRGAN method [LTH+16] on the datasets ImageNet (a subset of 200,000 randomely selected images) and DTD, in terms of classical metrics (L2 and SSIM) and perceptual metrics [ZIE+18].

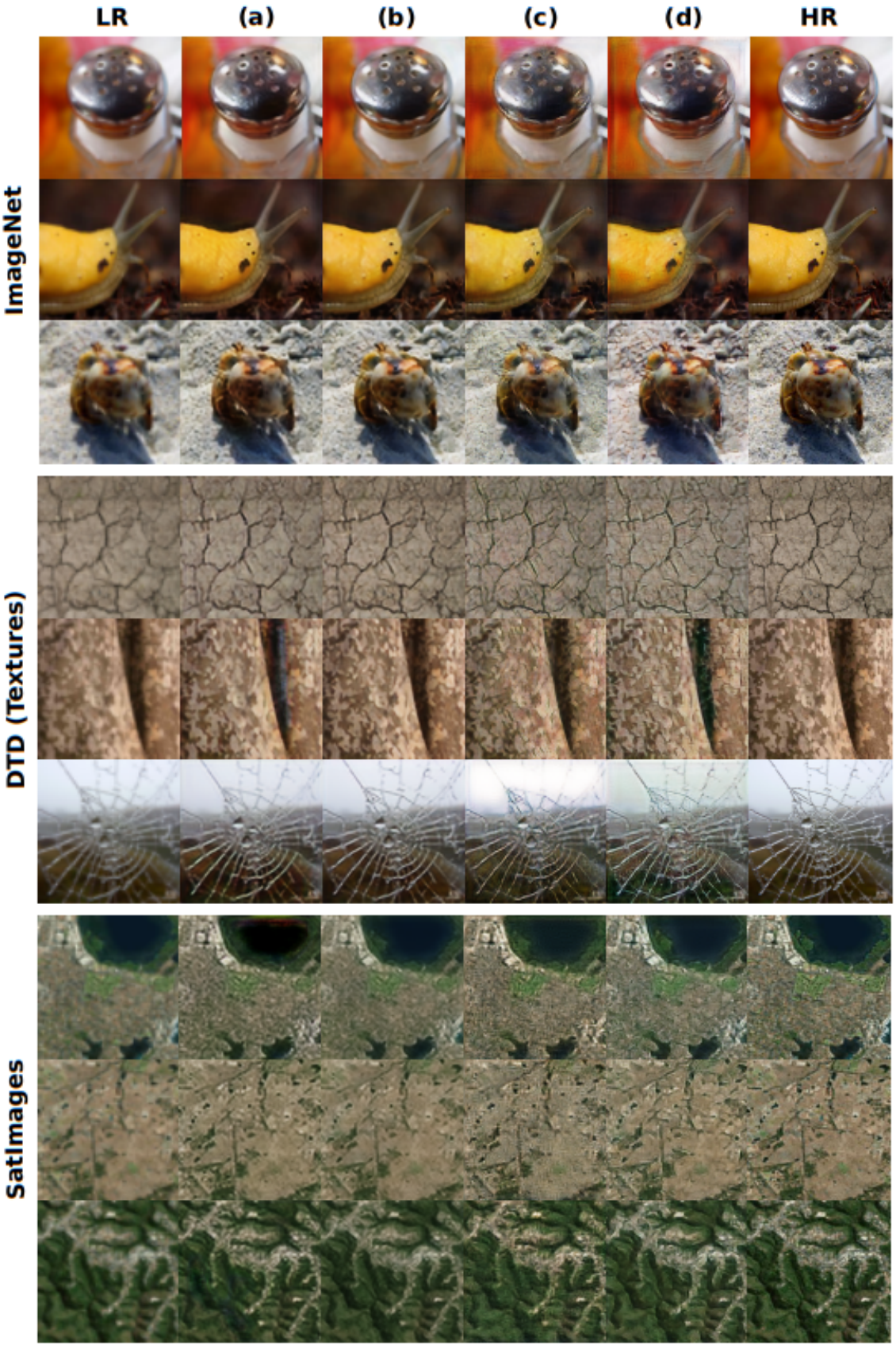



Figure B.5: Rows refer to the different considered Datasets. Columns refer to methods and ground-truth images: **LR** and **HR** refer to the low- and high-resolution pairs. The different used methods are: **(a)** $\mathcal{P}_{sisr}/rec$, **(b)** $\mathcal{P}_{sisr}/dis{,}rec$, **(c)** $\mathcal{P}_{sisr}/adv$ and **(d)** $\mathcal{P}_{sisr}/adv{,}rec$. Best view in PDF.

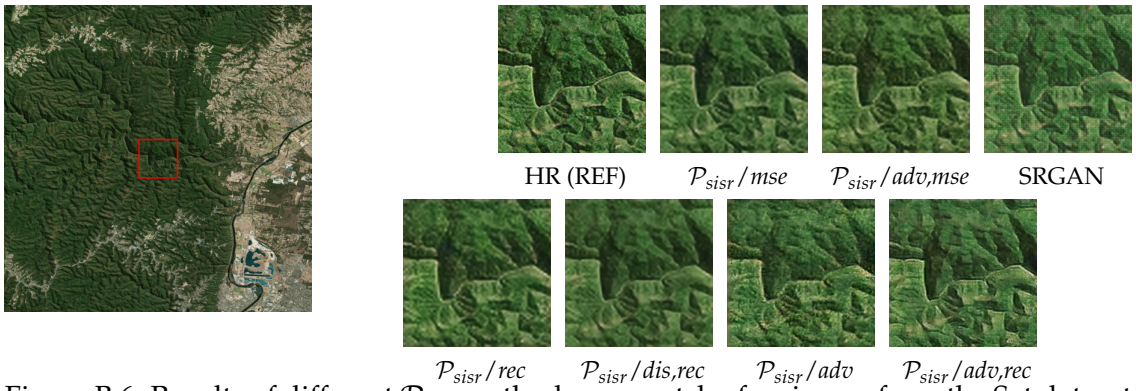



Figure B.6: Results of different $\mathcal{P}_{sisr}$ methods on a patch of an image from the Sat dataset.

| | | Low-level | | Perceptual metrics | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Methods** | L2 | SSIM | Squ | Squ-l | Alex | Alex-l | VGG | VGG-l |
| ImageNet | $\mathcal{P}_{sisr}/mse$ | 0.017 | 0.146 | 1.568 | 0.280 | 1.435 | 0.391 | 2.064 | 0.349 |
| | $\mathcal{P}_{sisr}/adv,mse$ | 0.020 | 0.156 | 1.634 | 0.241 | 1.397 | 0.329 | 2.223 | 0.384 |
| | SRGAN | 0.028 | 0.170 | **1.303** | **0.177** | **1.084** | **0.225** | 2.045 | 0.342 |
| | $\mathcal{P}_{sisr}/dis$ | 0.018 | 0.147 | 1.606 | 0.279 | 1.470 | 0.398 | 2.088 | 0.358 |
| | $\mathcal{P}_{sisr}/rec$ | 0.020 | 0.162 | 1.723 | 0.301 | 1.595 | 0.425 | 2.243 | 0.388 |
| | $\mathcal{P}_{sisr}/dis,rec$ | 0.017 | 0.147 | 1.587 | 0.279 | 1.420 | 0.382 | 2.052 | 0.353 |
| | $\mathcal{P}_{sisr}/adv$ | 0.028 | 0.202 | 1.820 | 0.222 | 1.554 | 0.322 | 2.598 | 0.432 |
| | $\mathcal{P}_{sisr}/adv,rec$ | **0.016** | **0.141** | 1.533 | 0.263 | 1.362 | 0.368 | **1.994** | **0.340** |
| DTD | $\mathcal{P}_{sisr}/mse$ | 0.029 | 0.185 | 1.972 | 0.342 | 1.856 | 0.470 | 2.479 | 0.434 |
| | $\mathcal{P}_{sisr}/adv,mse$ | 0.025 | 0.188 | 1.880 | 0.268 | 1.586 | 0.349 | 2.512 | 0.430 |
| | SRGAN | 0.031 | 0.191 | **1.557** | **0.209** | **1.298** | **0.241** | 2.308 | 0.393 |
| | $\mathcal{P}_{sisr}/dis$ | 0.027 | 0.184 | 1.873 | 0.327 | 1.739 | 0.440 | 2.401 | 0.421 |
| | $\mathcal{P}_{sisr}/rec$ | 0.027 | 0.183 | 1.851 | 0.320 | 1.726 | 0.438 | 2.398 | 0.420 |
| | $\mathcal{P}_{sisr}/dis,rec$ | **0.023** | **0.167** | 1.703 | 0.292 | 1.576 | 0.404 | **2.260** | **0.392** |
| | $\mathcal{P}_{sisr}/adv$ | 0.036 | 0.227 | 2.077 | 0.281 | 1.812 | 0.375 | 2.770 | 0.473 |
| | $\mathcal{P}_{sisr}/adv,rec$ | 0.046 | 0.236 | 2.089 | 0.277 | 1.793 | 0.344 | 2.796 | 0.481 |
| Sat | $\mathcal{P}_{sisr}/mse$ | **0.011** | 0.134 | 1.873 | 0.245 | 1.855 | 0.411 | 2.536 | 0.419 |
| | $\mathcal{P}_{sisr}/adv,mse$ | 0.082 | 0.197 | 1.458 | 0.205 | 1.466 | 0.352 | 2.125 | 0.347 |
| | SRGAN | 0.228 | 0.188 | 1.510 | 0.220 | 1.361 | 0.282 | 2.230 | 0.412 |
| | $\mathcal{P}_{sisr}/dis$ | **0.011** | **0.129** | 1.484 | 0.210 | 1.508 | 0.356 | 2.121 | 0.355 |
| | $\mathcal{P}_{sisr}/rec$ | 0.060 | 0.168 | 1.705 | 0.245 | 1.762 | 0.423 | 2.260 | 0.395 |
| | $\mathcal{P}_{sisr}/dis,rec$ | **0.011** | 0.138 | 1.493 | 0.215 | 1.435 | 0.351 | **2.108** | 0.372 |
| | $\mathcal{P}_{sisr}/adv$ | 0.030 | 0.214 | 1.719 | 0.181 | 1.627 | 0.306 | 2.711 | 0.419 |
| | $\mathcal{P}_{sisr}/adv,rec$ | 0.018 | 0.183 | **1.359** | **0.140** | **1.310** | **0.220** | 2.115 | **0.344** |

Table B.5: Comparison of the proposed $\mathcal{P}_{sisr}$ methods in terms of traditional metrics (L2 and SSIM) and the *perceptual error* (PE) given by equation B.6 on all the considered datasets. In terms of perceptual metrics, the proposed $\mathcal{P}_{sisr}$ methods rank in the second position after SRGAN [LTH+16] on the datasets ImageNet and DTD, while they outperform all the baselines on the satellite images domain which is far from the ImageNet domain.

| Layer | Teacher | Student |
|---|---|---|
| Dense 1 | $p_0 \times 1024$ | $p_0 \times k$ |
| Dense 2 | $1024 \times 512$ | $k \times k$ |
| Dense 3 | $512 \times 256$ | $k \times k$ |
| Dense 4 | $256 \times 10$ | $k \times 10$ |

Table B.6: Architectures of the teacher and student networks. The dimensions of the weight matrix at each dense layer are shown for both networks.

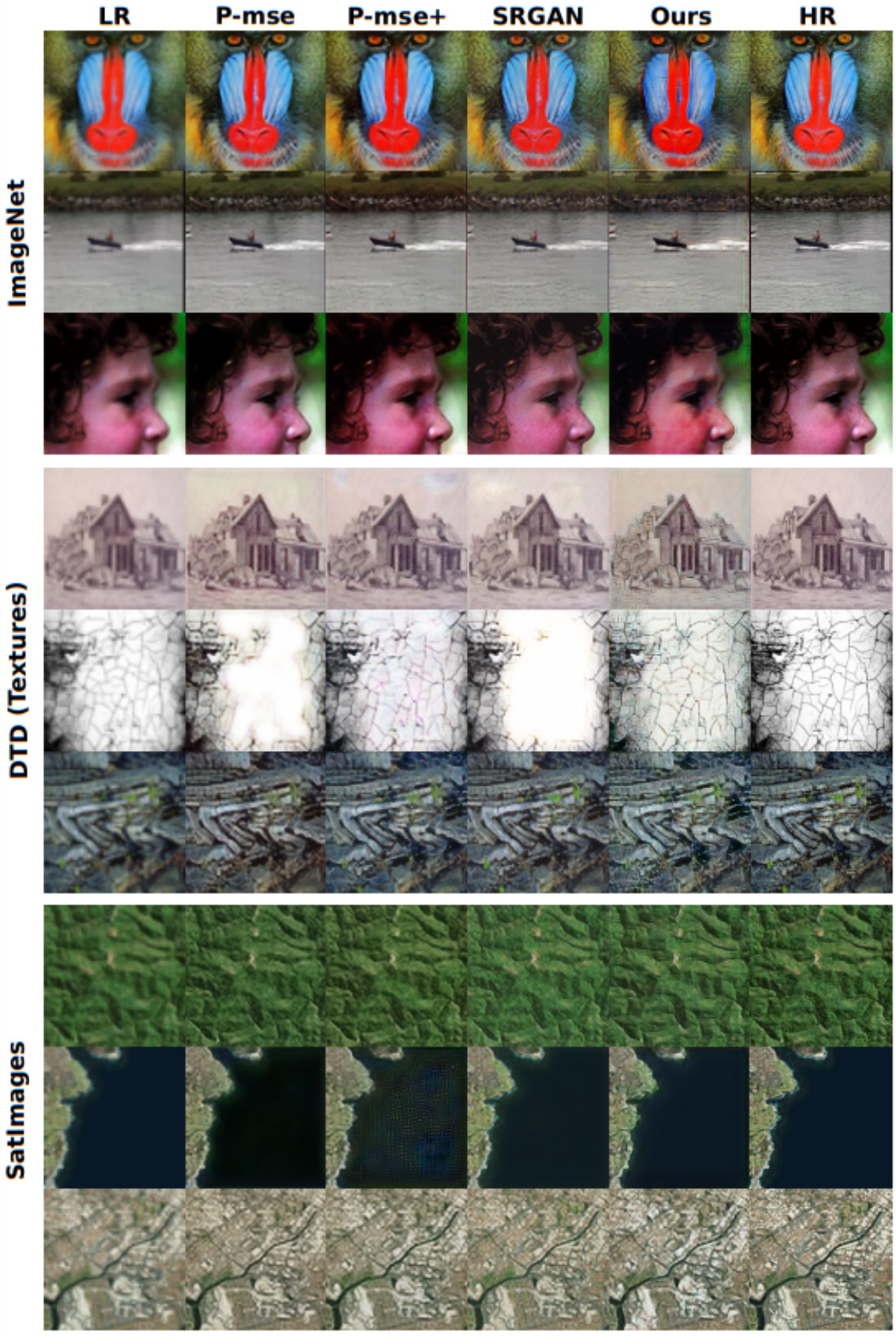



Figure B.7: Rows refer to the different Datasets. Columns refer to methods and ground-truth images: **LR** and **HR** refer to the low- and high-resolution pairs. **P-mse+** refers to the method $\mathcal{P}_{sisr}/mse$ with an adversarial loss ($\lambda_2 > 0$), **SRGAN** for the method in [LTH+16] and our method $\mathcal{P}_{sisr}/adv,rec$. Best view in PDF.

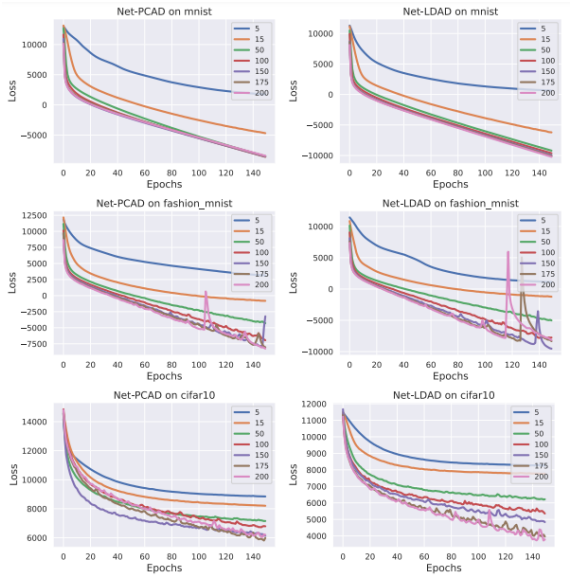



Figure B.8: Training loss of the student network when trained using Net-PCAD **(left)** and Net-LDAD **(right)** in terms of the training epochs, for different values of $k$, and across three different datasets.

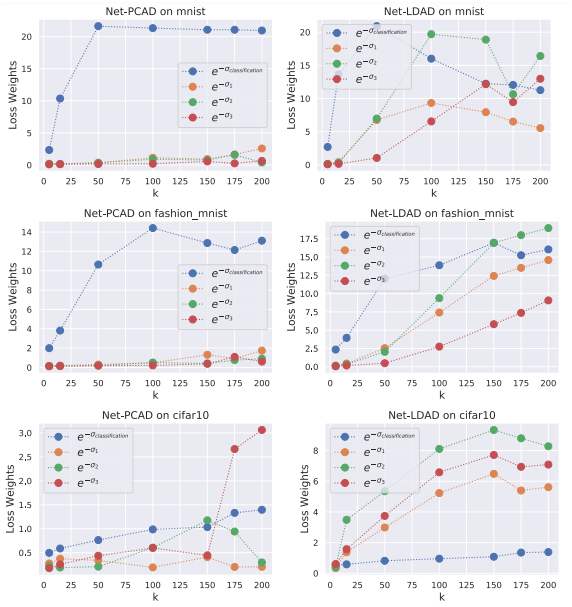



Figure B.9: The learned Homoscedastic loss parameters using Net-PCAD **(left)** and Net-LDAD **(right)** in terms of $k$ and across three different datasets. The weights corresponding to the reduced features mapping loss are of the same order of magnitude for Net-LDAD as the classification loss, which is a consequence of the fact that LDA is classes dependent.

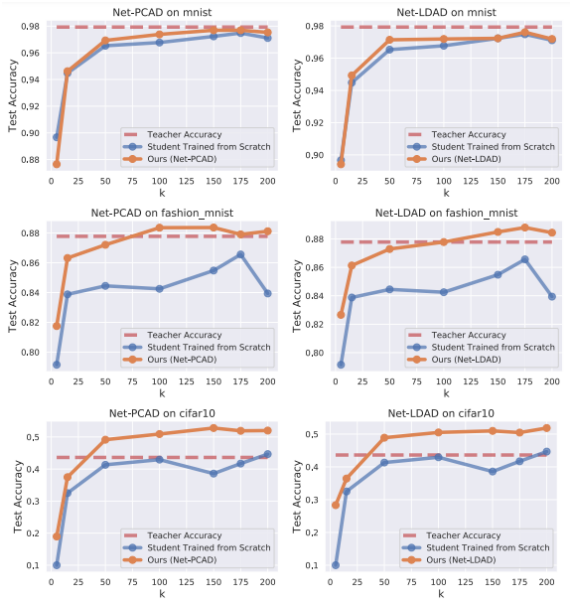



Figure B.10: Test accuracy of the student network in orange trained using Net-PCAD **(left)** and Net-LDAD **(right)**, and test accuracy of the student network in blue trained from scratch, in terms of $k$ and across three different datasets. The test accuracy of the teacher corresponds to the dashed red lines.

| | | Student | | |
|---|---|---|---|---|
| Dataset | Teacher | $k = 50$ | 100 | 200 |
| MNIST | 2.23s<br>98% | 0.38s<br>97% | 0.45s<br>97.5% | 0.65s<br>97.8% |
| FASHION | 2.23s<br>88% | 0.38s<br>87.5% | 0.45s<br>88.5% | 0.65s<br>88.5% |
| CIFAR10 | 4.63s<br>45% | 0.75s<br>50% | 0.92s<br>50.1% | 1.35s<br>50.3% |

Table B.7: Forward execution time in seconds (and corresponding test accuracies in %) of the teacher network and the student network for different values of $k$, the forward pass is applied (on a i7-7700HQ CPU @ 2.80GHz) to the train set of the respective datasets using a batch size of 50000 images.

# Appendix C

# Proofs

**Contents**



## C.1 Proofs of Chapter 3

### C.1.1 Setting of the proof

For simplicity, we will only suppose the case $k = 1$ and we consider the following notations that will be used subsequently.

$$\bar{x} = \mathbb{E}x_i,\ \boldsymbol{C} = \mathbb{E}[x_i x_i^\intercal],\ \boldsymbol{X}_0 = \boldsymbol{X} - \bar{x}\mathbf{1}_n^\intercal,\ \boldsymbol{C}_0 = \mathbb{E}[\boldsymbol{X}_0\boldsymbol{X}_0^\intercal/n].$$

Let

$$\boldsymbol{X}_{-i} = (x_1, \ldots, x_{i-1}, 0, x_i, \ldots, x_n)$$

the matrix $X$ with a vector of zeros at its $i$th column.

Denote the resolvents

$$\boldsymbol{R} = \left(\frac{\boldsymbol{X}^\intercal\boldsymbol{X}}{p} + z\boldsymbol{I}_n\right)^{-1},\ \boldsymbol{Q} = \left(\frac{\boldsymbol{X}\boldsymbol{X}^\intercal}{p} + z\boldsymbol{I}_p\right)^{-1},\ \boldsymbol{Q}_{-i} = \left(\frac{\boldsymbol{X}\boldsymbol{X}^\intercal}{p} - \frac{x_i x_i^\intercal}{p} + z\boldsymbol{I}_p\right)^{-1} \tag{C.1}$$

And let

$$\tilde{\boldsymbol{Q}} = \left( \frac{1}{c} \frac{\boldsymbol{C}}{1+\delta} + z\boldsymbol{I}_p \right)^{-1}, \tag{C.2}$$

where $\delta$ is the solution to the fixed point equation

$$\delta = \frac{1}{p} \operatorname{tr} \left( \boldsymbol{C} \left( \frac{1}{c} \frac{\boldsymbol{C}}{1+\delta} + z\boldsymbol{I}_p \right)^{-1} \right).$$

### C.1.2 Basic tools

**Lemma C.1** ([Led05a]). *Let $\boldsymbol{z} \in \mathcal{E}_q(1 \mid \mathbb{R}^p, \|\cdot\|)$ and $\boldsymbol{M} \in \mathcal{E}_q(1 \mid \mathbb{R}^{p\times n}, \|\cdot\|_F)$. Then, for some numerical constant $C > 0$*

- $\mathbb{E}\,\|\boldsymbol{z}\| \leq \|\mathbb{E}\boldsymbol{z}\| + C\sqrt{p}$, $\mathbb{E}\,\|\boldsymbol{z}\|_\infty \leq \|\mathbb{E}\boldsymbol{z}\|_\infty + C\sqrt{\log p}$.
- $\mathbb{E}\,\|\boldsymbol{M}\| \leq \|\mathbb{E}\boldsymbol{M}\| + C\sqrt{p+n}$, $\mathbb{E}\,\|\boldsymbol{M}\|_F \leq \|\mathbb{E}\boldsymbol{M}\|_F + C\sqrt{pn}$.

**Lemma C.2.** *Denote $\boldsymbol{Q}_{\bar{\boldsymbol{x}}} = (\bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^\intercal + z\boldsymbol{I}_p)^{-1}$, we have:*

$$\boldsymbol{Q}_{\bar{\boldsymbol{x}}}\bar{\boldsymbol{x}} = \frac{\bar{\boldsymbol{x}}}{\|\bar{\boldsymbol{x}}\|^2 + z} \quad \textit{and} \quad \|\tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}}\|,\ \bar{\boldsymbol{x}}\tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}} = \mathcal{O}(1).$$

*Moreover, if $\|\bar{\boldsymbol{x}}\| \geq \sqrt{p}$, $\|\tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}}\| = \mathcal{O}(p^{-1/2})$.*

*Proof.* Since $z\boldsymbol{Q}_{\bar{\boldsymbol{x}}} = \boldsymbol{I}_p - \boldsymbol{Q}_{\bar{\boldsymbol{x}}}\bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^\intercal$ :

$$z\boldsymbol{Q}_{\bar{\boldsymbol{x}}}\bar{\boldsymbol{x}} = \bar{\boldsymbol{x}} - \|\bar{\boldsymbol{x}}\|^2 \boldsymbol{Q}_{\bar{\boldsymbol{x}}}\bar{\boldsymbol{x}},$$

and we recover the first identity of the Lemma.

And since the matrix $\boldsymbol{C}_0$ is nonnegative symmetric, we have :

$$\tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}} = \left( \frac{1}{c} \frac{\boldsymbol{C}_0 + \bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^\intercal}{1+\delta} + z\boldsymbol{I}_p \right)^{-1} \bar{\boldsymbol{x}} \leq \frac{c(1+\delta)\bar{\boldsymbol{x}}}{\|\bar{\boldsymbol{x}}\|^2 + zc(1+\delta)}.$$

Therefore, $\bar{\boldsymbol{x}}\tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}} = \frac{c(1+\delta)\|\bar{\boldsymbol{x}}\|^2}{\|\bar{\boldsymbol{x}}\|^2 + zc(1+\delta)} = \mathcal{O}(1)$ and:

$$\|\tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}}\| = \frac{c(1+\delta)\|\bar{\boldsymbol{x}}\|}{\|\bar{\boldsymbol{x}}\|^2 + zc(1+\delta)} \leq \begin{cases} \dfrac{\|\bar{\boldsymbol{x}}\|}{z} = \mathcal{O}(1) \ \text{ if } \|\bar{\boldsymbol{x}}\| \leq 1, \\ \dfrac{c(1+\delta)}{\|\bar{\boldsymbol{x}}\|} = \mathcal{O}(1) \ \text{ if } \|\bar{\boldsymbol{x}}\| \geq 1. \end{cases}$$

□

**Proposition C.1.** $\bar{\boldsymbol{x}}^\intercal \mathbb{E}[\boldsymbol{Q}]\bar{\boldsymbol{x}} = \bar{\boldsymbol{x}}^\intercal \tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}} + \mathcal{O}\left(\sqrt{\frac{\log p}{p}}\right)$

*Proof.* Let us bound:

$$\left|\bar{\boldsymbol{x}}^\intercal \boldsymbol{Q}\bar{\boldsymbol{x}} - \bar{\boldsymbol{x}}^\intercal \tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}}\right| \leq \frac{c^{-1}}{1+\delta} \left| \mathbb{E}\left[ \bar{\boldsymbol{x}}\boldsymbol{Q}\boldsymbol{x}_i\boldsymbol{x}_i^\intercal \tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}} \left( \frac{1}{p}\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i - \delta \right) \right] + \frac{1}{p}\mathbb{E}\left[ \bar{\boldsymbol{x}}^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i\boldsymbol{x}_i^\intercal \boldsymbol{Q}\boldsymbol{C}\tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}} \right] \right|$$

Now let us consider a supplementary random vector $\boldsymbol{x}_{n+1}$ following the same low as the $\boldsymbol{x}_i$'s and independent of $\boldsymbol{X}$. We divide the set $\mathbb{I} = [n+1]$ into two sets $\mathbb{I}_{\frac{1}{2}}$ and $\mathbb{I}_{\frac{2}{2}}$

of same cardinality ($\lfloor \frac{n+1}{2} \rfloor \leq \#\mathbb{I}_{\frac{1}{2}}, \#\mathbb{I}_{\frac{2}{2}} \leq \lceil \frac{n+1}{2} \rceil$), we note $\boldsymbol{X}_{\frac{1}{2}} = (\boldsymbol{x}_i \,|\, i \in \mathbb{I}_{\frac{1}{2}})$, $\boldsymbol{X}_{\frac{2}{2}} = (\boldsymbol{x}_i \,|\, i \in \mathbb{I}_{\frac{2}{2}})$ and we introduce the diagonal matrices $\boldsymbol{\Delta} = \operatorname{diag}\left(\frac{1}{p}\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i - \delta \,|\, i \in \mathbb{I}_{\frac{1}{2}}\right)$, $\boldsymbol{D} = \operatorname{diag}\left(1 + \frac{1}{p+1}\boldsymbol{x}_i^\intercal \boldsymbol{Q}\boldsymbol{x}_i \,|\, i \in \mathbb{I}_{\frac{2}{2}}\right)$. We have the bound:

$$
\begin{aligned}
&\left|\mathbb{E}\left[\bar{\boldsymbol{x}}\boldsymbol{Q}\boldsymbol{x}_i\boldsymbol{x}_i^\intercal\tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}}\left(\frac{1}{p}\boldsymbol{x}_i^\intercal\boldsymbol{Q}_{-i}\boldsymbol{x}_i - \delta\right)\right]\right| \\
&\quad= \left|\mathbb{E}\left[\left(1 + \frac{1}{p}\boldsymbol{x}_{n+1}^\intercal\boldsymbol{Q}\boldsymbol{x}_{n+1}\right)\boldsymbol{x}_{n+1}\boldsymbol{Q}_{+(n+1)}\boldsymbol{x}_i\boldsymbol{x}_i^\intercal\tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}}\left(\frac{1}{p}\boldsymbol{x}_i^\intercal\boldsymbol{Q}_{-i}\boldsymbol{x}_i - \delta\right)\right]\right| \\
&\quad= \frac{1}{p^2}\left|\mathbb{E}\left[\mathbf{1}^\intercal\boldsymbol{D}\boldsymbol{X}_{\frac{2}{2}}^\intercal\boldsymbol{Q}_{+(n+1)}\boldsymbol{X}_{\frac{1}{2}}\boldsymbol{\Delta}\boldsymbol{X}_{\frac{1}{2}}^\intercal\tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}}\right]\right| \\
&\quad\leq \sqrt{\left|\mathbb{E}\left[\frac{1}{p^3}\mathbf{1}^\intercal\boldsymbol{D}\boldsymbol{X}_{\frac{2}{2}}^\intercal\boldsymbol{Q}_{+(n+1)}\boldsymbol{X}_{\frac{1}{2}}\boldsymbol{\Delta}^2\boldsymbol{X}_{\frac{1}{2}}^\intercal\boldsymbol{Q}_{+(n+1)}\boldsymbol{X}_{\frac{2}{2}}\boldsymbol{D}\mathbf{1}\right]\mathbb{E}\left[\frac{1}{p}\bar{\boldsymbol{x}}^\intercal\tilde{\boldsymbol{Q}}\boldsymbol{X}_{\frac{1}{2}}\boldsymbol{X}_{\frac{1}{2}}^\intercal\tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}}\right]\right|} \\
&\quad\leq \sqrt{\left|\mathbb{E}\left[\left\|\frac{1}{p}\boldsymbol{X}_{\frac{2}{2}}^\intercal\boldsymbol{Q}_{+(n+1)}\boldsymbol{X}_{\frac{1}{2}}\right\|^2\|\boldsymbol{D}\|^2\|\boldsymbol{\Delta}\|^2\right]\mathbb{E}\left[\bar{\boldsymbol{x}}\tilde{\boldsymbol{Q}}\boldsymbol{C}\tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}}\right]\right|} \;\leq\; \mathcal{O}\left(\sqrt{\frac{\log p}{p}}\right),
\end{aligned}
$$

thanks to Lemma C.1 and Lemma C.2 (the spectral norm of $\Delta$ and $D$ is just an infinity norm if we see them as random vectors of $\mathbb{R}^n$). We can bound $\frac{1}{p}\left|\mathbb{E}\left[\bar{\boldsymbol{x}}^\intercal\boldsymbol{Q}_{-i}\boldsymbol{x}_i\boldsymbol{x}_i^\intercal\boldsymbol{Q}\boldsymbol{C}\tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}}\right]\right|$ the same way to obtain the result of the proposition. □

**Proposition C.2.** $\left\|\mathbb{E}[\boldsymbol{x}_i^\intercal\boldsymbol{Q}_{-i}\boldsymbol{X}_{-i}] - \frac{\bar{\boldsymbol{x}}^\intercal\tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}}\mathbf{1}^\intercal}{1+\delta}\right\| = \mathcal{O}(\sqrt{\log p})$

*Proof.* Considering $\boldsymbol{u} \in \mathbb{R}^n$ such that $\|\boldsymbol{u}\| = 1$:

$$
\begin{aligned}
&\left|\mathbb{E}[\boldsymbol{x}_i^\intercal\boldsymbol{Q}_{-i}\boldsymbol{X}_{-i}\boldsymbol{u}] - \frac{\bar{\boldsymbol{x}}^\intercal\tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}}\mathbf{1}^\intercal\boldsymbol{u}}{1+\delta}\right| \\
&\quad= \left|\sum_{\substack{j=1\\ j\neq i}}^n \boldsymbol{u}_j\mathbb{E}\left[\frac{\boldsymbol{x}_i^\intercal\boldsymbol{Q}_{\substack{-i\\-j}}\boldsymbol{x}_j}{1 + \frac{1}{p}\boldsymbol{x}_j^\intercal\boldsymbol{Q}_{\substack{-j\\-i}}\boldsymbol{x}_j} - \frac{\boldsymbol{x}_i^\intercal\tilde{\boldsymbol{Q}}\boldsymbol{x}_j}{1+\delta}\right]\right| \\
&\quad\leq \sqrt{n}\left|\mathbb{E}\left[\frac{\boldsymbol{x}_i^\intercal\boldsymbol{Q}_{\substack{-i\\-j}}\boldsymbol{x}_j}{1 + \frac{1}{p}\boldsymbol{x}_j^\intercal\boldsymbol{Q}_{\substack{-j\\-i}}\boldsymbol{x}_j} - \frac{\boldsymbol{x}_i^\intercal\boldsymbol{Q}_{\substack{-i\\-j}}\boldsymbol{x}_j}{1+\delta}\right]\right| + \left|\frac{1}{1+\delta}\mathbb{E}\left[\boldsymbol{x}_i^\intercal\boldsymbol{Q}_{\substack{-i\\-j}}\boldsymbol{x}_j - \boldsymbol{x}_i^\intercal\tilde{\boldsymbol{Q}}\boldsymbol{x}_j\right]\right| \quad \text{(where } i \neq j\text{)} \\
&\quad\leq \sqrt{n}\left|\mathbb{E}\left[\bar{\boldsymbol{x}}^\intercal\boldsymbol{Q}\boldsymbol{x}_j\left(\frac{1}{p}\boldsymbol{x}_j^\intercal\boldsymbol{Q}_{\substack{-j\\-i}}\boldsymbol{x}_j - \delta\right)\right]\right| + \sqrt{n}\left|\mathbb{E}\left[\bar{\boldsymbol{x}}^\intercal\boldsymbol{Q}_{\substack{-i\\-j}}\bar{\boldsymbol{x}} - \bar{\boldsymbol{x}}^\intercal\tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}}\right]\right|,
\end{aligned}
$$

where the first term is treated the same way as we did in the proof of Proposition C.1 and the second term is bounded thanks to Proposition C.1 □

### C.1.3 Main body of the proof

*Proof of Theorem 3.1.* Recall the definition of the resolvents $\boldsymbol{R}$ and $\boldsymbol{Q}$ in Equation (C.1). The first step of the proof is to show the concentration of $\boldsymbol{R}$. This comes from the fact that the application $\Phi : \boldsymbol{X} \mapsto (\boldsymbol{X}^\intercal\boldsymbol{X} + z\boldsymbol{I}_n)^{-1}$ is $2z^{-3/2}$-Lipschitz *w.r.t.* the Frobenius norm. Indeed, by the matrix identity $\boldsymbol{A} - \boldsymbol{B} = \boldsymbol{A}(\boldsymbol{B}^{-1} - \boldsymbol{A}^{-1})\boldsymbol{B}$, we have

$$
\Phi(\boldsymbol{X}) - \Phi(\boldsymbol{X} + \boldsymbol{H}) = \Phi(\boldsymbol{X})(\boldsymbol{H}^\intercal\boldsymbol{X} + (\boldsymbol{X} + \boldsymbol{H})^\intercal\boldsymbol{H})\Phi(\boldsymbol{X} + \boldsymbol{H})
$$

And by the bounds $\|AB\|_F \leq \|A\| \cdot \|B\|_F$, $\|\Phi(X)X^\intercal\| \leq z^{-1/2}$ and $\|\Phi(X)\| \leq z^{-1}$, we have

$$\|\Phi(X+H) - \Phi(X)\|_F \leq \frac{2}{z^{3/2}}\|H\|_F.$$

Therefore, given $X \in \mathcal{E}_q(1 \,|\, \mathbb{R}^{p\times n}, \|\cdot\|_F)$ and since the application $X \mapsto R = \Phi(X/\sqrt{p})$ is $2z^{-3/2}p^{-1/2}$-Lipschitz, we have by Proposition 2.4 that $R \in \mathcal{E}_q(p^{-1/2} \,|\, \mathbb{R}^{n\times n}, \|\cdot\|_F)$.

The second step consists in estimating $\mathbb{E}R(z)$ through a deterministic matrix $\tilde{R}$. Indeed, by the identity $(M^\intercal M + zI)^{-1}M^\intercal = M^\intercal(MM^\intercal + zI)^{-1}$, the resolvent $R$ can be expressed in function of $Q$ as follows

$$R = \frac{1}{z}\left(I_n - \frac{X^\intercal Q X}{p}\right), \tag{C.3}$$

thus a deterministic equivalent for $R$ can therefore be obtained through a deterministic equivalent of the matrix $X^\intercal Q X$. However, as demonstrated in [LC19], the matrix $Q$ has as a deterministic equivalent the matrix $\tilde{Q}$ defined in equation C.2. In the following, we aim at deriving a deterministic equivalent for $\frac{1}{p}X^\intercal Q X$ in function of $\tilde{Q}$. Let $u$ and $v$ be two unitary vectors in $\mathbb{R}^n$, and let us estimate

$$\Delta \equiv \mathbb{E}\left[u^\intercal\left(\frac{X^\intercal Q X}{p} - \frac{X^\intercal \tilde{Q} X}{p}\right)v\right] = \frac{1}{p}\mathbb{E}\left[\frac{u^\intercal X^\intercal Q C \tilde{Q} X v}{1+\delta} - \frac{1}{p}u^\intercal X^\intercal Q X X^\intercal \tilde{Q} X v\right]$$

With the following matrix identities (to explore the independence of the columns of $X$):

$$Q = Q_{-i} - \frac{1}{p}Q_{-i}x_i x_i^\intercal Q\,, \qquad Qx_i = \frac{Q_{-i}x_i}{1+\frac{1}{p}x_i^\intercal Q_{-i}x_i}, \qquad A - B = A(B^{-1} - A^{-1})B$$

and the decomposition $QXX^\intercal = \sum_{i=1}^n Qx_i x_i^\intercal$, we obtain:

$$\begin{aligned}
\Delta &= \frac{1}{p^2}\mathbb{E}\left[\sum_{i=1}^n \frac{u^\intercal X^\intercal Q_{-i} C \tilde{Q} X v}{1+\delta} - \frac{u^\intercal X^\intercal Q_{-i}x_i x_i^\intercal \tilde{Q} X v}{1+\frac{1}{p}x_i^\intercal Q_{-i}x_i} - \frac{1}{p}\frac{u^\intercal X^\intercal Q_{-i}x_i x_i^\intercal Q C \tilde{Q} X v}{1+\delta}\right] \\
&= \frac{1}{p^2}\sum_{i=1}^n \mathbb{E}\left[\frac{u^\intercal X_{-i}^\intercal Q_{-i} C \tilde{Q} X_{-i} v}{1+\delta} - \frac{u^\intercal X_{-i}^\intercal Q_{-i}x_i x_i^\intercal \tilde{Q} X_{-i} v}{1+\frac{1}{p}x_i^\intercal Q_{-i}x_i}\right. \\
&\qquad + \frac{u_i x_i^\intercal Q_{-i} C \tilde{Q} X_{-i} v}{1+\delta} + \frac{v_i u^\intercal X_{-i}^\intercal Q_{-i} C \tilde{Q} x_i}{1+\delta} + u_i v_i \frac{x_i^\intercal Q_{-i} C \tilde{Q} x_i}{1+\delta} \\
&\qquad - \frac{u_i x_i^\intercal Q_{-i}x_i x_i^\intercal \tilde{Q} X_{-i} v}{1+\frac{1}{p}x_i^\intercal Q_{-i}x_i} - \frac{v_i u^\intercal X_{-i}^\intercal Q_{-i}x_i x_i^\intercal \tilde{Q} x_i}{1+\frac{1}{p}x_i^\intercal Q_{-i}x_i} - u_i v_i \frac{x_i^\intercal Q_{-i}x_i x_i^\intercal \tilde{Q} x_i}{1+\frac{1}{p}x_i^\intercal Q_{-i}x_i} \\
&\qquad \left. - \frac{1}{p}\frac{u^\intercal X^\intercal Q_{-i}x_i x_i^\intercal Q C \tilde{Q} X v}{1+\delta}\right]
\end{aligned}$$

We can show with Holder's inequality and the concentration bounds (mainly the fact that $\frac{1}{p}x_i^\intercal Q_{-i}x_i$ concentrates around $\delta$) developed in [LC19], that most of the above quantities

vanish asymptotically. As a toy example, we consider the following term:

$$\begin{aligned}
&\left|\frac{1}{p^2}\sum_{i=1}^{n}\mathbb{E}\left[\frac{\boldsymbol{u}^\intercal \boldsymbol{X}_{-i}^\intercal \boldsymbol{Q}_{-i}\boldsymbol{C}\tilde{\boldsymbol{Q}}\boldsymbol{X}_{-i}\boldsymbol{v}}{1+\delta}-\frac{\boldsymbol{u}^\intercal \boldsymbol{X}_{-i}^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i\boldsymbol{x}_i^\intercal \tilde{\boldsymbol{Q}}\boldsymbol{X}_{-i}\boldsymbol{v}}{1+\frac{1}{p}\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i}\right]\right| \\
&\quad=\left|\frac{1}{p^2}\sum_{i=1}^{n}\mathbb{E}\left[\boldsymbol{u}^\intercal \boldsymbol{X}_{-i}^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i\boldsymbol{x}_i^\intercal \tilde{\boldsymbol{Q}}\boldsymbol{X}_{-i}\boldsymbol{v}\frac{\delta-\frac{1}{p}\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i}{(1+\delta)(1+\frac{1}{p}\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i)}\right]\right| \\
&\quad\le\left|\frac{1}{p^2}\sum_{i=1}^{n}\mathbb{E}\left[(\boldsymbol{u}^\intercal \boldsymbol{X}_{-i}^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i)(\boldsymbol{x}_i^\intercal \tilde{\boldsymbol{Q}}\boldsymbol{X}_{-i}\boldsymbol{v})\left(\delta-\frac{1}{p}\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i\right)\right]\right| \\
&\quad\le\left|\frac{1}{p}\sum_{i=1}^{n}\left(\mathbb{E}\left[\left(\frac{1}{\sqrt{p}}\boldsymbol{u}^\intercal \boldsymbol{X}_{-i}^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i\right)^3\right]\mathbb{E}\left[\left(\frac{1}{\sqrt{p}}\boldsymbol{x}_i^\intercal \tilde{\boldsymbol{Q}}\boldsymbol{X}_{-i}\boldsymbol{v}\right)^3\right]\mathbb{E}\left[\left(\delta-\frac{1}{p}\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i\right)^3\right]\right)^{\frac{1}{3}}\right| \\
&\quad=\mathcal{O}\left(\frac{1}{\sqrt{p}}\right)
\end{aligned}$$

Similarly, we can show that:

$$\begin{aligned}
\left|\frac{1}{p^2}\sum_{i=1}^{n}\mathbb{E}\left[\frac{\boldsymbol{u}_i\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}\boldsymbol{C}\tilde{\boldsymbol{Q}}\boldsymbol{X}_{-i}\boldsymbol{v}}{1+\delta}+\frac{\boldsymbol{v}_i\boldsymbol{u}^\intercal \boldsymbol{X}_{-i}^\intercal \boldsymbol{Q}_{-i}\boldsymbol{C}\tilde{\boldsymbol{Q}}\boldsymbol{x}_i}{1+\delta}\right.\right. \\
\left.\left.+\boldsymbol{u}_i\boldsymbol{v}_i\frac{\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}\boldsymbol{C}\tilde{\boldsymbol{Q}}\boldsymbol{x}_i}{1+\delta}-\frac{1}{p}\frac{\boldsymbol{u}^\intercal \boldsymbol{X}^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i\boldsymbol{x}_i^\intercal \boldsymbol{Q}\boldsymbol{C}\tilde{\boldsymbol{Q}}\boldsymbol{X}\boldsymbol{v}}{1+\delta}\right]\right|=\mathcal{O}\left(\frac{1}{\sqrt{p}}\right)
\end{aligned}$$

Finally, the remaining terms in $\Delta$ can be estimated as follows:

$$\begin{aligned}
\Delta&=\frac{1}{p^2}\sum_{i=1}^{n}\mathbb{E}\left[-\frac{\boldsymbol{u}_i\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i\boldsymbol{x}_i^\intercal \tilde{\boldsymbol{Q}}\boldsymbol{X}_{-i}\boldsymbol{v}}{1+\frac{1}{p}\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i}\right. \\
&\qquad\left.-\frac{\boldsymbol{v}_i\boldsymbol{u}^\intercal \boldsymbol{X}_{-i}^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i\boldsymbol{x}_i^\intercal \tilde{\boldsymbol{Q}}\boldsymbol{x}_i}{1+\frac{1}{p}\boldsymbol{x}_i^\intercal \boldsymbol{Q}\boldsymbol{x}_i}-\boldsymbol{u}_i\boldsymbol{v}_i\frac{\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i\boldsymbol{x}_i^\intercal \tilde{\boldsymbol{Q}}\boldsymbol{x}_i}{1+\frac{1}{p}\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i}\right]+\mathcal{O}\left(\frac{1}{\sqrt{p}}\right) \\
&=-\frac{2}{p}\frac{\delta\boldsymbol{u}^\intercal \mathbf{1}\bar{\boldsymbol{x}}^\intercal \tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}}\mathbf{1}^\intercal \boldsymbol{v}}{1+\delta}-\frac{\delta^2\boldsymbol{u}^\intercal \boldsymbol{v}}{1+\delta}+\mathcal{O}\left(\sqrt{\frac{\log p}{p}}\right)
\end{aligned}$$

Where the last equality is obtained through the following estimation:

$$\begin{aligned}
\frac{1}{p^2}\sum_{i=1}^{n}\mathbb{E}\left[\frac{\boldsymbol{v}_i\boldsymbol{u}^\intercal \boldsymbol{X}_{-i}^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i\boldsymbol{x}_i^\intercal \tilde{\boldsymbol{Q}}\boldsymbol{x}_i}{1+\frac{1}{p}\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i}\right]&=\frac{1}{p}\sum_{i=1}^{n}\mathbb{E}\left[\frac{\boldsymbol{v}_i\boldsymbol{u}^\intercal \boldsymbol{X}_{-i}^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i\left(\frac{1}{p}\boldsymbol{x}_i^\intercal \tilde{\boldsymbol{Q}}\boldsymbol{x}_i(1+\delta)-\delta\left(1+\frac{1}{p}\boldsymbol{x}_i^\intercal \tilde{\boldsymbol{Q}}\boldsymbol{x}_i\right)\right)}{\left(1+\frac{1}{p}\boldsymbol{x}_i^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i\right)(1+\delta)}\right] \\
&\quad+\frac{1}{p}\sum_{i=1}^{n}\frac{\boldsymbol{v}_i\delta\mathbb{E}[\boldsymbol{u}^\intercal \boldsymbol{X}_{-i}^\intercal \boldsymbol{Q}_{-i}\boldsymbol{x}_i]}{(1+\delta)}
\end{aligned}$$

With the following bound:

$$\begin{aligned}
&\left|\frac{1}{p}\boldsymbol{x}_i^\intercal \tilde{\boldsymbol{Q}}\boldsymbol{x}_i(1+\delta)-\delta\left(1+\frac{1}{p}\boldsymbol{x}_i^\intercal \tilde{\boldsymbol{Q}}\boldsymbol{x}_i\right)\right| \\
&\quad=\left|\frac{1}{p}\boldsymbol{x}_i^\intercal \tilde{\boldsymbol{Q}}\boldsymbol{x}_i(1+\delta)-\delta(1+\delta)+\delta(1+\delta)-\delta\left(1+\frac{1}{p}\boldsymbol{x}_i^\intercal \tilde{\boldsymbol{Q}}\boldsymbol{x}_i\right)\right| \\
&\quad\le\left|\frac{1}{p}\boldsymbol{x}_i^\intercal \tilde{\boldsymbol{Q}}\boldsymbol{x}_i-\delta\right|(1+2\delta),
\end{aligned}$$

we have again with Holder's inequality and Proposition C.2:

$$\frac{1}{p^2}\sum_{i=1}^{n}\mathbb{E}\left[\frac{v_i\boldsymbol{u}^\mathsf{T}\boldsymbol{X}_{-i}^\mathsf{T}\boldsymbol{Q}_{-i}\boldsymbol{x}_i\boldsymbol{x}_i^\mathsf{T}\tilde{\boldsymbol{Q}}\boldsymbol{x}_i}{1+\frac{1}{p}\boldsymbol{x}_i^\mathsf{T}\boldsymbol{Q}\boldsymbol{x}_i}\right]=\frac{1}{p}\sum_{i=1}^{n}\frac{v_i\delta\boldsymbol{u}^\mathsf{T}\mathbf{1}\bar{\boldsymbol{x}}^\mathsf{T}\tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}}}{1+\delta}+\mathcal{O}\left(\sqrt{\frac{\log p}{p}}\right)$$

Now that we estimated $\Delta$, it remains to estimate $\mathbb{E}[\frac{1}{p}\boldsymbol{X}^\mathsf{T}\tilde{\boldsymbol{Q}}\boldsymbol{X}]$. Indeed, given two unit norm vectors $u, v \in \mathbb{R}^n$ we have:

$$\begin{aligned}\mathbb{E}\left[\frac{1}{p}\boldsymbol{u}^\mathsf{T}\boldsymbol{X}^\mathsf{T}\tilde{\boldsymbol{Q}}\boldsymbol{X}\boldsymbol{v}\right]&=\frac{1}{p}\sum_{i,j=1}^{n}\boldsymbol{u}_i\boldsymbol{v}_j\mathbb{E}[\boldsymbol{x}_i^\mathsf{T}\tilde{\boldsymbol{Q}}\boldsymbol{x}_j]=\frac{1}{p}\sum_{i=1}^{n}\sum_{\substack{j=1\\j\neq i}}^{n}\boldsymbol{u}_i\boldsymbol{v}_j\bar{\boldsymbol{x}}^\mathsf{T}\tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}}+\sum_{i=1}^{n}\boldsymbol{u}_i\boldsymbol{v}_i\delta\\&=\frac{1}{p}\bar{\boldsymbol{x}}^\mathsf{T}\tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}}\boldsymbol{u}^\mathsf{T}\mathbf{1}\mathbf{1}^\mathsf{T}\boldsymbol{v}+(\delta-\frac{1}{p}\bar{\boldsymbol{x}}^\mathsf{T}\tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}})\boldsymbol{u}^\mathsf{T}\boldsymbol{v}=\frac{1}{p}\bar{\boldsymbol{x}}^\mathsf{T}\tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}}\boldsymbol{u}^\mathsf{T}\boldsymbol{M}_1\boldsymbol{v}^\mathsf{T}+\delta\boldsymbol{u}^\mathsf{T}\boldsymbol{v}+\mathcal{O}\left(\frac{1}{p}\right)\end{aligned}$$

since we have $\bar{\boldsymbol{x}}^\mathsf{T}\tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}}=\mathcal{O}(1)$ by Lemma C.2; we introduced the matrix $\boldsymbol{M}_1=\mathbf{1}\mathbf{1}^\mathsf{T}$. Therefore we have the following estimation:

$$\frac{1}{p}\mathbb{E}\left[\boldsymbol{X}^\mathsf{T}\boldsymbol{Q}\boldsymbol{X}\right]=\frac{\delta}{1+\delta}\boldsymbol{I}_n+\frac{1}{p}\left(\frac{1-\delta}{1+\delta}\right)\bar{\boldsymbol{x}}^\mathsf{T}\tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}}\boldsymbol{M}_1+\mathcal{O}_{\|\cdot\|}\left(\sqrt{\frac{\log p}{p}}\right)$$

where $\boldsymbol{A}=\boldsymbol{B}+\mathcal{O}_{\|\cdot\|}(\alpha(p))$ means that $\|\boldsymbol{A}-\boldsymbol{B}\|=\mathcal{O}(\alpha(p))$. Finally, since $\boldsymbol{R}$ concentrates around its mean, we can then conclude:

$$\boldsymbol{R}=\frac{1}{z}\left(\boldsymbol{I}_n-\frac{1}{p}\boldsymbol{X}^\mathsf{T}\boldsymbol{Q}\boldsymbol{X}\right)=\frac{1}{z}\frac{1}{1+\delta}\boldsymbol{I}_n+\frac{\delta-1}{pz(\delta+1)}\bar{\boldsymbol{x}}^\mathsf{T}\tilde{\boldsymbol{Q}}\bar{\boldsymbol{x}}\boldsymbol{M}_1+\mathcal{O}_{\|\cdot\|}\left(\sqrt{\frac{\log p}{p}}\right).$$

□

## C.2 Proofs of Chapter 4

### C.2.1 Proofs of Section 4.1

Let us Taylor-expand $K_{ij}$ in the vicinity of $\tau$, *i.e.*,

$$\begin{aligned}K_{ij}=f(\tau)+f'(\tau)\left(\frac{1}{p}\|x_i-x_j\|^2-\tau\right)+\frac{1}{2}f''(\tau)\left(\frac{1}{p}\|x_i-x_j\|^2-\tau\right)^2\\+\frac{1}{6}f^{(3)}(\zeta_{ij}^n)\left(\frac{1}{p}\|x_i-x_j\|^2-\tau\right)^3,\end{aligned}$$

where $\zeta_{ij}^n\in\lfloor\frac{1}{p}\|x_i-x_j\|^2,\tau\rfloor$.

• *Control of the third order random matrix*: First, we show that the third order matrix term vanishes in operator norm. Indeed, exploiting the concentration

$$\frac{1}{p}\|x_i-x_j\|^2-\tau\in\mathcal{O}\left(e^{-(n\cdot)^{q/2}}+e^{-(\sqrt{n}\cdot)^q}+e^{-(n\cdot)^q}\right),$$

we have for all $s\in 2\mathbb{N}^*$

$$\mathbb{E}|\frac{1}{p}\|x_i-x_j\|^2-\tau|^s=\mathcal{O}(n^{-s/2}).$$

Let $K^{(3)}$ be the matrix with entries $\left(\frac{1}{p}\|x_i - x_j\|^2 - \tau\right)^3$, we have in particular

$$\mathbb{E}\|K^{(3)}\|_F^2 = \sum_{i,j=1}^{n} \mathbb{E}|\frac{1}{p}\|x_i - x_j\|^2 - \tau|^6 = \mathcal{O}(n^2\, n^{-3}) = \mathcal{O}(n^{-1}).$$

On the other hand, if $f^{(3)}$ is bounded in a neighborhood of $\tau$, we have with probability at least $1-\delta$ that $\max_{i\neq j} f^{(3)}(\zeta_{ij}^n)$ is bounded. Indeed, since

$$\max_{1\leq i\neq j\leq n}\left\{\frac{1}{p}\|x_i - x_j\|^2 - \tau\right\} = \mathcal{O}\left(n^{-1/2}\log\left(\frac{n}{\sqrt{\delta}}\right)^{1/q}\right),$$

we also have

$$\max_{1\leq i\neq j\leq n}|\zeta_{ij}^n - \tau| = \mathcal{O}\left(n^{-1/2}\log\left(\frac{n}{\sqrt{\delta}}\right)^{1/q}\right),$$

which shows that $\zeta_{ij}^n$ is bounded, so is $\max_{i\neq j} f^{(3)}(\zeta_{ij}^n)$ with probability at least $1-\delta$. Thus

$$\mathbb{E}\|f^{(3)}(\zeta^n)\odot K^{(3)}\|_F^2 \leq \max_{i\neq j} f^{(3)}(\zeta_{ij}^n)^2\,\mathbb{E}\|K^{(3)}\|_F^2 = \mathcal{O}(n^{-1}),$$

in particular, involving Markov's inequality and since the operator norm is bounded by the Frobenius norm, we conclude that $\|f^{(3)}(\zeta^n)\odot K^{(3)}\|_{op}$ is $\mathcal{O}(n^{-1/2})$ with probability at least $1-\delta$.

• *Control of the second order random matrix*: First, let us show that the matrix with entries $E_{ij} = \mathbf{1}_{i\neq j}(\psi_i + \psi_j)z_i^\intercal z_j$ is vanishing in operator norm. Denote $\Sigma = ZZ - \operatorname{diag}(ZZ)$, we can then express $E$ as

$$E = \Sigma\operatorname{diag}(\psi) + \operatorname{diag}(\psi)\Sigma$$

Fix $\gamma\in(0,1/2)$, we first show that $\|\Sigma\|_{op} = \mathcal{O}\left(\log(n)^\gamma\right)$. Indeed, since $\|\operatorname{diag}(Z'Z)\|_{op}$ is bounded it remains to control $Z'Z$ in operator norm. Involving an $\varepsilon$-net argument, we have

$$\|Z'Z\|_{op} \leq \frac{1}{1-2\varepsilon}\max_{u\in\mathcal{E}} u'Z'Zu = \frac{1}{1-2\varepsilon}\max_{u\in\mathcal{E}}\|Zu\|_2^2$$

Recalling the $q$-exponential concentration, we have $Z\in\mathcal{O}(e^{-(\sqrt{n}\,\cdot)^q})$, and since $Z\mapsto\|Zu\|_2$ is 1-Lipschitz, $\|Zu\|_2\in\mathcal{O}(e^{-(\sqrt{n}\,\cdot)^q})$ and thanks to Proposition 2.5, we have

$$\|Zu\|_2^2 \in \mathcal{O}(e^{-(n\,\cdot)^{q/2}} + e^{-(\sqrt{n}\,\cdot)^q})$$

Thus, there exits two absolute constants $M$ and $\sigma$ such that for all $t\geq 2\mathbb{E}\|Zu\|_2^2$, we have

$$\mathbb{P}\left\{\|Zu\|_2^2\geq t\right\} \leq M(e^{-(n\,t/\sigma)^{q/2}} + e^{-(\sqrt{n}\,t/\sigma)^q})$$

To control the maximum over the $\varepsilon$-net $\mathcal{E}$, we need to bound its cardinality. In fact, we have the following Lemma

**Lemma C.3** (Cardinality of an $\varepsilon$-net)**.** *There exists an $\varepsilon$-net $\mathcal{E}$ of the unit sphere in $n$ dimentsions, satisfying:*

$$|\mathcal{E}| \leq \left(1+\frac{2}{\varepsilon}\right)^n$$

Thus, we have by the union bound

$$\mathbb{P}\left\{\max_{u\in\mathcal{E}}\|Zu\|_2^2 \geq t\right\} \leq Me^{n\log(1+2/\varepsilon)}(e^{-(nt/\sigma)^{q/2}} + e^{-(\sqrt{n}t/\sigma)^q}) \equiv p(t)$$

In particular, for $t = \mathcal{O}(\log(n)^\gamma)$ and for all $q \geq 2$

$$\frac{n\log(1+2/\varepsilon)}{(n\frac{t}{\sigma})^{q/2}} = \mathcal{O}\left(\frac{n^{1-q/2}}{\log(n)^{q\gamma/2}}\right) \to 0 \text{ and } \frac{n\log(1+2/\varepsilon)}{(\sqrt{n}\frac{t}{\sigma})^q} = \mathcal{O}\left(\frac{n^{1-q}}{\log(n)^{q\gamma}}\right) \to 0$$

which shows that $\|Z^\intercal Z\|_{op} = \mathcal{O}(\log(n)^\gamma)$ with probability $1 - \mathcal{O}(e^{-(n\log(n)^\gamma)^{q/2}})$.

It remains to control $\mathrm{diag}(\psi)$ in operator norm, we have for all $k \in 2\mathbb{N}$

$$\mathbb{E}\,\mathrm{tr}\left(\mathrm{diag}(\psi)^k\right) = \sum_{i=1}^n \mathbb{E}\psi_i^k = \mathcal{O}(n^{1-k/2}),$$

thus by Markov's inequality, we can show that $\|\,\mathrm{diag}(\psi)\|_{op} = \mathcal{O}(n^{-1/2+1/k})$ for all $k \in 2\mathbb{N}$. Finally, since the operator norm is subadditive, we conclude that

$$\|\left\{\mathbf{1}_{i\neq j}(\psi_i+\psi_j)z_i^\intercal z_j\right\}_{i,j=1}^n\|_{op} = \mathcal{O}\left(n^{-1/2+\gamma}\log(n)^\gamma\right) \to 0$$

The kernel random matrix $K$ can thus be expanded as

$$\begin{aligned}
K = f(\tau)' + f'(\tau)\Bigg[&\psi^\intercal + \psi^\intercal + \left\{t_a\frac{1_{n_a}}{\sqrt{p}}\right\}_{a=1}^k{}^\intercal + \left\{t_b\frac{1_{n_b}^\intercal}{\sqrt{p}}\right\}_{b=1}^k \\
&+ \left\{\|\bar{m}_a-\bar{m}_b\|^2\frac{1_{n_a}1_{n_b}^\intercal}{p}\right\}_{a,b=1}^k + 2\left\{\frac{1}{\sqrt{p}}Z_a^\intercal(\bar{m}_a-\bar{m}_b)1_{n_b}^\intercal\right\}_{a,b=1}^k \\
&-2\left\{\frac{1}{\sqrt{p}}1_{n_a}(\bar{m}_a-\bar{m}_b)^\intercal Z_b\right\}_{a,b=1}^k - 2Z^\intercal Z\Bigg] \\
&+ \frac{f''(\tau)}{2}\Bigg[\psi^{2\intercal} + (\psi^2)^\intercal + \left\{t_a^2\frac{1_{n_a}}{\sqrt{p}}\right\}_{a=1}^k{}^\intercal + \left\{t_b^2\frac{1_{n_b}^\intercal}{\sqrt{p}}\right\}_{b=1}^k \\
&+ 2\left\{t_at_b\frac{1_{n_a}1_{n_b}^\intercal}{p}\right\}_{a,b=1}^k + 2\psi\left\{t_b\frac{1_{n_b}^\intercal}{\sqrt{p}}\right\}_{b=1}^k + 2\left\{t_a\frac{1_{n_a}}{\sqrt{p}}\right\}_{a=1}^k\psi^\intercal \\
&+ \frac{2}{\sqrt{p}}\,\mathrm{diag}\{t_a1_{n_a}\}_{a=1}^k\psi^\intercal + \frac{2}{\sqrt{p}}\psi^\intercal\,\mathrm{diag}\{t_b1_{n_b}\}_{b=1}^k \\
&+4\left\{\mathrm{tr}(C_aC_b)\frac{1_{n_a}1_{n_b}^\intercal}{p^2}\right\}_{a,b=1}^k + 2\psi\psi^\intercal\Bigg] \\
&+ (f(0)-f(\tau)+\tau f'(\tau))I_n + \mathcal{O}_{\|\cdot\|}(n^{-1/2+\gamma}\log(n)^\gamma)
\end{aligned}$$

where $\psi \equiv [\psi_1,\ldots,\psi_n]^\intercal \in \mathbb{R}^n$, $t_\ell \equiv p^{-\frac{1}{2}}\,\mathrm{tr}\,\bar{C}_\ell$, $Z \equiv [z_1,\ldots,z_n] \in \mathbb{R}^{p\times n}$ and $Z_\ell = [z_{n_1+\ldots+n_{\ell-1}+1},\ldots,z_{n_1+\ldots+n_\ell}]$ the restriction of $Z$ to the mixture $\mu_\ell$.

In particular, the centered kernel matrix expresses as follows:

$$\begin{aligned}\bar{K} = PKP = f'(\tau)P\Bigg[&\left\{\|\bar{m}_a - \bar{m}_b\|^2 \frac{1_{n_a}1_{n_b}^\mathsf{T}}{p}\right\}_{a,b=1}^k + 2\left\{\frac{1}{\sqrt{p}} Z_a^\mathsf{T}(\bar{m}_a - \bar{m}_b)1_{n_b}^\mathsf{T}\right\}_{a,b=1}^k \\ &-2\left\{\frac{1}{\sqrt{p}} 1_{n_a}(\bar{m}_a - \bar{m}_b)^\mathsf{T} Z_b\right\}_{a,b=1}^k - 2Z^\mathsf{T}Z\Bigg] P \\ &+ \frac{f''(\tau)}{2} P\Bigg[2\left\{t_a t_b \frac{1_{n_a}1_{n_b}^\mathsf{T}}{p}\right\}_{a,b=1}^k + 2\psi\left\{t_b \frac{1_{n_b}^\mathsf{T}}{\sqrt{p}}\right\}_{b=1}^k + 2\left\{t_a \frac{1_{n_a}}{\sqrt{p}}\right\}_{a=1}^k \psi^\mathsf{T} \\ &+4\left\{\mathrm{tr}(C_a C_b)\frac{1_{n_a}1_{n_b}^\mathsf{T}}{p^2}\right\}_{a,b=1}^k + 2\psi\psi^\mathsf{T}\Bigg] P \\ &+ (f(0) - f(\tau) + \tau f'(\tau))P + \mathcal{O}_{\|\cdot\|}(n^{-1/2+\gamma}\log(n)^\gamma)\end{aligned}$$

where $P = I_n - \frac{1}{n}^\mathsf{T}$.

Introduce the following notations:

$$M = [\bar{m}_1, \ldots, \bar{m}_k] \in \mathbb{R}^{p\times k}, \;\; t = \left\{\frac{1}{\sqrt{p}}\,\mathrm{tr}\,\bar{C}_\ell\right\}_{\ell=1}^k \in \mathbb{R}^k$$

$$T = \left\{\frac{1}{p}\,\mathrm{tr}\,\bar{C}_a\bar{C}_b\right\}_{a,b=1}^k \in \mathbb{R}^{k\times k}, \;\; J = [j_1, \ldots, j_k] \in \mathbb{R}^{n\times k}$$

$$\Phi = Z^\mathsf{T}M - \{Z_\ell^\mathsf{T}\bar{m}_\ell\}_{\ell=1}^k \in \mathbb{R}^{n\times k}, \;\; U = P\left[\frac{1}{\sqrt{p}}J, \Phi, \psi\right] \in \mathbb{R}^{n\times(2k+1)}$$

$$A_{11} = M^\mathsf{T}M - \frac{f''(\tau)}{2f'(\tau)}tt^\mathsf{T} - \frac{f''(\tau)}{f'(\tau)}T - \left\{\frac{\|\bar{m}_a\|^2 + \|\bar{m}_b\|^2}{2}\right\}_{a,b=1}^k$$

$$A = \begin{bmatrix} A_{11} & I_k & -\frac{f''(\tau)}{2f'(\tau)}t \\ I_k & \mathbf{0}_{k\times k} & \mathbf{0}_{k\times 1} \\ -\frac{f''(\tau)}{2f'(\tau)}t^\mathsf{T} & \mathbf{0}_{1\times k} & -\frac{f''(\tau)}{2f'(\tau)} \end{bmatrix}$$

The centred kernel matrix $\bar{K}$ can be approximated by

$$\tilde{K} = -2f'(\tau)\left[PZ^\mathsf{T}ZP + UAU^\mathsf{T}\right] + (f(0) - f(\tau) + \tau f'(\tau))P \tag{C.4}$$

such that

$$\|\bar{K} - \tilde{K}\|_2 = \mathcal{O}(n^{-1/2+\gamma}\log(n)^\gamma) \to 0 \tag{C.5}$$

### C.2.2 Proofs of Section 4.2

#### C.2.2.1 Proof of Theorem 4.3

*Proof.* We first need the following key Lemma.

**Lemma C.4** (A Moment Result). *For $g_{ij}(X) \equiv [\Sigma_p^{1/2}]_{i,\cdot}\left(\frac{1}{n}XX^\mathsf{T} - I_p\right)[\Sigma_p^{1/2}]_{\cdot,j}$, we have, for all $k \in \mathbb{N}$ and for some absolute constant $C_k > 0$,*

$$\mathbb{E}|g_{ij}(X)|^{2k} \le C_k \frac{\beta_p^{4k}}{n^k}. \tag{C.6}$$

*Proof.* Given a random variable $Z$, we have

$$\forall m > 0,\ \mathbb{E}|Z|^m = \int_0^\infty m\, t^{m-1} \mathbb{P}\{|Z| \geq t\} dt,$$

whenever the right hand side is finite. Applying this identity to the random variable $g_{ij}(X)$ with $m = 2k$ and exploiting the concentration property in Lemma 4.2 yields the result. □

The proof starts by a Taylor expansion of $F_{ij}$ in the vicinity of $[\Sigma_p]_{ij}$, *i.e.*,

$$F_{ij} = \sum_{k=0}^{2} \frac{f^{(k)}(\sigma_{ij})}{k!} F_{ij}^{(k)} + \frac{f^{(3)}(\xi_{ij}^n)}{6} F_{ij}^{(3)}$$

where $\sigma_{ij} = [\Sigma_p]_{ij}$, $\xi_{ij}^n \in \left\lfloor [YY^\intercal/n]_{ij}, \sigma_{ij} \right\rfloor$,[1] and $F^{(k)}$ is the matrix with entries

$$F_{ij}^{(k)} \equiv [\Sigma_p^{1/2}(n^{-1}XX^\intercal - I_p)\Sigma_p^{1/2}]_{ij}^k = g_{ij}(X)^k.$$

We have by Lemma 4.2 that $[YY^\intercal/n]_{ij}$ concentrates around $\sigma_{ij}$, so that $\xi_{ij}^n$ is bounded by $\sigma_{ij} + \varepsilon$, for all $\varepsilon > 0$, with high probability[2] (note that the condition $\max_{ij} |\sigma_{ij}| < B$ ensures that $\sigma_{ij}$ is bounded and the condition on $\beta_p$ ensures the quasi-exponential concentration of $[YY^\intercal/n]_{ij}$ around $\sigma_{ij}$; see considered Assumptions above), formally

$$\begin{aligned}\mathbb{P}\left\{|\xi_{ij}^n| \geq \sigma_{ij} + \varepsilon\right\} &\leq \mathbb{P}\left\{|g_{ij}(X)| \geq \varepsilon\right\} \leq K e^{-\frac{n}{\beta_p^2}\min(c_1\varepsilon, \frac{c_2\varepsilon^2}{\beta_p^2})} \\ &\leq K e^{-K' n^{\frac{1}{2}+2\epsilon}\min(c_1\varepsilon, K' c_2 \varepsilon^2 n^{-\frac{1}{2}+2\epsilon})} \equiv p_n \to 0,\end{aligned}$$

where $K' > 0$. And since $f^{(3)}$ is continuous, we deduce that $f^{(3)}(\xi_{ij}^n)$ is in particular bounded by

$$A \equiv \max_{x \in [\sigma_{ij}-\varepsilon, \sigma_{ij}+\varepsilon]} |f^{(3)}(x)|,$$

with probability $1 - p_n$. Knowing that the operator norm is bounded by the Frobenius norm, we look for a control of the Frobenius norm of the tailing term. We have

$$\|f^{(3)}(\xi^n) \odot F^{(3)}\|_F^2 \leq A^2 \|F^{(3)}\|_F^2. \tag{C.7}$$

By Lemma C.4, for all $k \in \mathbb{N}$

$$\mathbb{E}\|F^{(k)}\|_F^2 = \sum_{i,j=1}^{p} \mathbb{E}\left[|g_{ij}(X)|^{2k}\right] \leq C_k \frac{p^2 \beta_p^{4k}}{n^k},$$

for some absolute constant $C_k > 0$. Thus, by *Markov's inequality*, we have for all $\eta > 0$

$$\mathbb{P}\left\{\|F^{(k)}\|_F \geq \frac{p\,\beta_p^{2k}}{n^{\frac{k}{2}}} \sqrt{\frac{C_k}{\eta}}\right\} \leq \eta.$$

Recalling Eq. equation C.7, we have with probability at least $1 - \eta$

$$\|f^{(3)}(\xi^n) \odot F^{(3)}\|_F \leq C\, \frac{p\,\beta_p^6}{n^{\frac{3}{2}}\sqrt{\eta}}.$$

□

[1] The notation $\lfloor a, b \rfloor$ stands for the interval $[a, b]$ if $a < b$ or $[b, a]$ otherwise.

[2] For a given asymptotic variable $n$, we say that an event $E_n$ occurs with high probability when it exist a function $\psi(n)$ quasi-exponentially decreasing in $n$ such that $\mathbb{P}\{E_n\} \geq 1 - \psi(n)$.

#### C.2.2.2 Proof of Theorem 4.4

*Proof.* The proof needs the introduction of the following two lemmas, that can be found in [EK08] and which are a consequence of the $\varepsilon$-sparsity notion[3]

**Lemma C.5.** *Given an $\varepsilon$-sparse $p\times p$ real symmetric matrix $M$ and calling $m = \max_{ij}|M_{ij}|$, we have, for all $k \in 2\mathbb{N}$*

$$\|M\|_{op} \leq (M^k)^{1/k} = \mathcal{O}(m\, p^{\varepsilon(1-1/k)+1/k}). \tag{C.8}$$

**Lemma C.6.** *Given two real symmetric matrices $M$ and $N$ with $|M_{ij}| \leq N_{ij}$. Then, we have $\|M\|_{op} \leq \|N\|_{op}$.*

First, we show that when $\Sigma_p$ is $\varepsilon$-sparse, the Hadamard product $f^{(k)}(\Sigma_p) \odot F^{(k)}$ is of vanishing operator norm for $k \geq 1$, precisely

**Lemma C.7.** *Let $\mu > 0$, suppose $\Sigma_p$ is a $\frac{1}{2+\mu}$-sparse matrix. For $f$ a real and differentiable function, $k \in \{1,2\}$ such that $f^{(k)}(0) = 0$ and for $\eta > 0$, we have for all $\epsilon \in (0, \frac{k(2+\mu)-2}{2(3+2\mu)})$*

$$\|f^{(k)}(\Sigma_p) \odot F^{(k)}\|_{op} = \mathcal{O}_\eta^{\lfloor 1/\epsilon \rfloor}\left(n^{\frac{2-k(2+\mu)}{2(2+\mu)}+\epsilon\left(2-\frac{1}{2+\mu}\right)}\right).$$

*Proof.* We start by proving that the matrix $F^{(k)}$ has entries of order $\mathcal{O}(n^{-k/2})$. In fact, we have by Lemma C.4, for all $m \in \mathbb{N}^*$

$$\mathbb{E}|F_{ij}^{(k)}|^{2m} = \mathbb{E}|g_{ij}(X)|^{2km} = \mathcal{O}(n^{-km}),$$

thus applying *Markov's inequality* to the random variable $|F_{ij}^{(k)}|^{2m}$ yields to the following tail control.

$$\mathbb{P}\{|F_{ij}^{(k)}| \geq t\} \leq \frac{\mathbb{E}|F_{ij}^{(k)}|^{2m}}{t^{2m}} \leq C\, n^{-km}\, t^{-2m},$$

where $C$ is an absolute constant. Recalling Assumption **A1** and by the union bound, we have

$$\mathbb{P}\{\max_{ij}|F_{ij}^{(k)}| \geq t\} \leq \sum_{i,j=1}^{p}\mathbb{P}\{|F_{ij}^{(k)}| \geq t\} \leq p^2\,\mathbb{P}\{|F_{ij}^{(k)}| \geq t\} \leq C\, n^{2-km}\, t^{-2m},$$

which implies for $\eta > 0$ and for all $m > 0$

$$\max_{ij}|F_{ij}^{(k)}| = \mathcal{O}_\eta^m\left(n^{-\frac{k}{2}+\frac{1}{m}}\right) \tag{C.9}$$

Besides, let $M$ be the matrix defined as $M \equiv \max_{ij}|F_{ij}^{(k)}| \cdot f^{(k)}(\Sigma_p)$, we have

$$|[f^{(k)}(\Sigma_p) \odot F^{(k)}]_{ij}| \leq M_{ij},$$

[3]Through the identity $(M^k) \leq \max_{ij}|M_{ij}|^k \cdot |\mathcal{C}_p(k)|$.

thus, one has by Lemma C.6

$$\|f^{(k)}(\Sigma_p) \odot F^{(k)}\|_{op} \leq \|M\|_{op} = \max_{ij} |F_{ij}^{(k)}| \cdot \|f^{(k)}(\Sigma_p)\|_{op}.$$

In particular, since $f^{(k)}(\Sigma_p)$ is $\frac{1}{2+\mu}$-sparse (by Remark 4.1), we have by Lemma C.5 and by equation C.9, for some $\eta > 0$

$$\|f^{(k)}(\Sigma_p) \odot F^{(k)}\|_{op} = \mathcal{O}_\eta^{2m}\left(n^{\frac{1}{2+\mu}(1-\frac{1}{2m})+\frac{1}{2m}-\frac{k}{2}+\frac{1}{2m}}\right),$$

choosing $\epsilon = \frac{1}{2m} < \frac{k(2+\mu)-2}{2(3+2\mu)}$ yields the final result. □

When considering $f$ such that $f'(0) = f''(0) = 0$, the result holds by Corollary 4.1 and Lemma C.7. In fact, the dominant order corresponds to $k = 1$ in Lemma C.7. Which completes the proof.

□

## C.3 Proofs of Chapter 5

### C.3.1 Proofs of Section 5.1

This section provides the exact computation of the fixed point equation in Theorem 5.3. We start by introducing the main tools to perform the calculations.

Theorem C.1 giving the central limit theorem for concentrated vectors was originally proven for uniform distributions on convex subspaces of $\mathbb{R}^p$, but it was quickly understood that the result is true for a larger class of random vectors satisfying a so-called "thin shell property" (see [Fre19] for a simple and complete proof of this inference). The thin shell property expresses the fact that a random vector $\boldsymbol{X}$ lies principally on a thin shell around a sphere with the following inequality satisfied for some $\varepsilon > 0$

$$\mathbb{P}\left(\left|\frac{\|X\|}{\sqrt{p}} - 1\right| \geq \varepsilon\right) \leq \varepsilon.$$

For a concentrated random vector $X \propto \mathcal{E}_2$, such that $\mathbb{E}[XX^T] = I_p$ and $\sqrt{p} = O(\mathbb{E}[\|X\|])$[4] the norm being a 1-Lipschitz observation, we know (as in equation 5.3) that there exist two constants $C, c > 0$ such that:

$$\mathbb{P}\left(|\|X\| - \mathbb{E}[\|X\|]| \geq \varepsilon\right) \leq Ce^{-c\varepsilon^2}.$$

Integrating this concentration inequality for $\varepsilon \in [0, \infty)$ with Fubini Theorem, we have $\forall r \geq 2$:

$$\mathbb{E}\left[\left|\frac{\|X\|}{\mathbb{E}[\|X\|]} - 1\right|^r\right] \leq C(r/2c)^{r/2}\mathbb{E}[\|X\|]^r.$$

Therefore, since $\mathbb{E}[\|X\|] \leq \sqrt{\mathbb{E}[\|X\|^2]} = \sqrt{p} = O(\mathbb{E}[\|X\|])$, we can deduce from Hölder's inequality that:

$$\mathbb{E}\left[\left|\frac{\|X\|^2}{\mathbb{E}[\|X\|]^2} - 1\right|\right] \leq \sqrt{\mathbb{E}\left[\left|\frac{\|X\|}{\mathbb{E}[\|X\|]} - 1\right|^2\right]\mathbb{E}\left[\left|\frac{\|X\|}{\mathbb{E}[\|X\|]} + 1\right|^2\right]} = O(1/\sqrt{p}),$$

[4] This is a very loose hypothesis needed to set that $\frac{\|X\|}{\mathbb{E}[\|X\|]}$ is sufficiently concentrated. Generally if $\sqrt{p} \ll \mathbb{E}[\|X\|]$, that means that one can obtain a better concentration than $X \propto \mathcal{E}_2$

from which we conclude that: $\mathbb{E}\left[\|X\|\right]^2 = \mathbb{E}\left[\|X\|^2\right] + O(\sqrt{p})$ and therefore $\mathbb{E}\left[\|X\|\right] = \sqrt{p} + O(p^{1/4})$. Choosing $\varepsilon = p^{-1/4}$ yields from the concentration of $\|X\|$ to the existence of some constant $K > 0$ such that:

$$\mathbb{P}\left(\left|\frac{\|X\|}{\sqrt{p}} - 1\right| \geq Kp^{-1/4}\right) \leq Ce^{-cp^{1/2}} \leq Kp^{-1/4},$$

for $p$ large enough. We can then infer (see [Fre19]), that the projections on small dimensional vector spaces of a concentrated vector are Gaussian vectors with high probability.

**Theorem C.1** (CLT for concentrated vectors [Kla07, FGP07])**.** *Given a random vector $\boldsymbol{X} \in \mathbb{R}^p$, and noting $G$, the cumulative distribution function of a Gaussian variable of zero mean and unit variance. If $\boldsymbol{X} \propto \mathcal{E}_2$, $\mathbb{E}[\boldsymbol{X}] = 0$ and $\mathbb{E}[\boldsymbol{X}\boldsymbol{X}^T] = I_p$, then for any integer $k_i n \mathbb{N}$, small compared to $p$, for any $\eta \in (0,1)$, there exists two constants $C, c > 0$ and a set $\Theta \subset \mathcal{S}^{p-1}$ such that $\sigma(\Theta) \geq 1 - \sqrt{p}Ce^{-c\sqrt{p}}$ and $\forall \boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k) \in \Theta^k$, there exists a Gaussian vector $\boldsymbol{Z} \sim \mathcal{N}(0, \boldsymbol{\theta}^\intercal \boldsymbol{\theta})$ such that:*

$$\forall a \in \mathbb{R}^k : \sup_{t \in \mathbb{R}} |\mathbb{P}(\boldsymbol{a}^\intercal \boldsymbol{\theta}^\intercal \boldsymbol{X} \geq t) - \mathbb{P}(\boldsymbol{a}^\intercal \boldsymbol{Z} \geq t)| \leq Cp^{-1/4}.$$

We need a simple preliminary Lemma to state our Stein-like formula for concentrated vectors from Theorem C.1.

**Lemma C.8.** *Given two random variables $X, Y \in \mathbb{R}^k$, if:*

$$\sup_{t \in \mathbb{R}} |\mathbb{P}(a^\intercal X \geq t) - \mathbb{P}(a^\intercal Y \geq t)| \leq \varepsilon,$$

*then for any differentiable mapping $f : \mathbb{R}^k \to \mathbb{R}$ integrable and bounded around $\infty$ by $f_\infty$:*

$$|\mathbb{E}\left[f(X)\right] - \mathbb{E}\left[f(Y)\right]| \leq \frac{f_\infty + \int |f|}{\varepsilon}$$

*Proof.* Let us prove it in the case $k = 1$:

$$\begin{aligned}
|\mathbb{E}\left[f(X)\right] - \mathbb{E}\left[f(Y)\right]| &\leq \left|\int_{t=-\infty}^{\infty} f(t)(d\mathbb{P}_X(t) - d\mathbb{P}_Y(t))\right| \\
&\leq |[f(t)(\mathbb{P}(X \geq t) - \mathbb{P}(Y \geq t))]_{-\infty}^{\infty}| \\
&\quad + \left|\int_{t=-\infty}^{\infty} f'(t)((\mathbb{P}(X \geq t) - \mathbb{P}(Y \geq t))dt\right| \\
&\leq \frac{f_\infty + \int |f|}{\varepsilon}
\end{aligned}$$

□

We have the following Stein-like [LN08] theorem for concentrated vectors which is the central tool to express the fixed point mappings in equation (8) of the Main Paper.

**Proposition C.3.** *Let $\boldsymbol{x} \in \mathbb{R}^p$ be a random vector satisfying the concentration $\boldsymbol{x} \propto \mathcal{E}_2$ (denote $\boldsymbol{m} \equiv \mathbb{E}[\boldsymbol{x}]$ and $\boldsymbol{C} \equiv \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\intercal]$) and let $f : \mathbb{R}^p \to \mathbb{R}$ be some three times differentiable function such that $f$, $f'$ and $f''$ satisfy the hypotheses of Lemma C.8 with $f_\infty, \int |f|, f'_\infty, \int |f'|, f''_\infty, \int |f''| = O(1)$. Then, there exists a subset $\Theta \subset \mathbb{S}^{p-1}$ such that: $\sigma(\Theta) \geq 1 - Ce^{-cp/\log p}$ and $\forall \boldsymbol{w}, \boldsymbol{v}, \boldsymbol{u} \in \Theta$:*

$$\mathbb{E}[f(\boldsymbol{w}^\intercal \boldsymbol{x})\boldsymbol{v}^\intercal \boldsymbol{x}] = \mathbb{E}[f(\boldsymbol{w}^\intercal \boldsymbol{x})]\boldsymbol{v}^\intercal \boldsymbol{m} + \mathbb{E}[f'(\boldsymbol{w}^\intercal \boldsymbol{x})]\boldsymbol{v}^\intercal \boldsymbol{C}\boldsymbol{w} + O\left(\frac{1}{n^{1/4}}\right)$$

$$\begin{aligned}
\mathbb{E}[f(\boldsymbol{w}^\intercal \boldsymbol{x})\boldsymbol{v}^\intercal \boldsymbol{x}\boldsymbol{x}^\intercal \boldsymbol{u}] &= \mathbb{E}[f(\boldsymbol{w}^\intercal \boldsymbol{x})]\boldsymbol{v}^\intercal (\boldsymbol{m}\boldsymbol{m}^\intercal + \boldsymbol{C})\boldsymbol{u} + \mathbb{E}[f'(\boldsymbol{w}^\intercal \boldsymbol{x})]\boldsymbol{v}^\intercal \left(\boldsymbol{C}\boldsymbol{w}\boldsymbol{m}^\intercal + \boldsymbol{m}\boldsymbol{w}^\intercal \boldsymbol{C}\right)\boldsymbol{u} \\
&\quad + \mathbb{E}[f''(\boldsymbol{w}^\intercal \boldsymbol{x})]\boldsymbol{v}^\intercal \boldsymbol{C}\boldsymbol{w}\boldsymbol{w}^\intercal \boldsymbol{C}\boldsymbol{u} + O\left(\frac{1}{n^{1/4}}\right)
\end{aligned}$$

*Proof.* Let us first consider a random vector $z$ with zero mean and identity covariance satisfying $z \propto \mathcal{E}_2$. Considering the subset $\Theta \subset \mathbb{S}^{p-1}$ mentioned in Theorem C.1 (for $k = 3$), we know that $\sigma(\Theta) \geq 1 - Ce^{-cp^{1-\varepsilon}}$ for two constants $C, c > 0$ and furthermore for any $\boldsymbol{\theta} = (w, v, u) \in \Theta^3$, and some Gaussian random vector $\boldsymbol{Z} \sim \mathcal{N}(0, \boldsymbol{\theta}^\intercal \boldsymbol{\theta})$:

$$\forall a \in \mathbb{R}^2 : \sup_{t \in \mathbb{R}} \left| \mathbb{P}(a^\intercal \boldsymbol{\theta}_i^\intercal z \geq t) - \mathbb{P}(a^\intercal \boldsymbol{Z} \geq t) \right| = O\left(\frac{1}{n^{1/4}}\right).$$

Given a mapping $g : \mathbb{R} \to \mathbb{R}$, we know thanks to Lemma C.8 and Stein's identity:

$$w^\intercal \mathbb{E}[g(w^\intercal z)z] = \mathbb{E}[g(e_1^\intercal \boldsymbol{Z}) e_1^\intercal \boldsymbol{Z}] + O\left(\frac{1}{n^{1/4}}\right) = \mathbb{E}[g'(w^\intercal z)]\|w\|^2 + O\left(\frac{1}{n^{1/4}}\right),$$

where $e_1 = (1, 0)$. Second, if we note $\tilde{\boldsymbol{\theta}} = (w, \tilde{v}, \tilde{u}) = \boldsymbol{QR}$, the QR- decomposition of $\boldsymbol{\theta}$, and $\boldsymbol{Q} = (q_1, q_2, q_3)$ (of course, $q_1 = e_1$), we know that $e_1^\intercal \boldsymbol{Z}$ and $q_2^\intercal \boldsymbol{Z}$ are independent ($\boldsymbol{Z}$ is Gaussian and $\mathbb{E}[e_1^\intercal \boldsymbol{Z} q_2^\intercal \boldsymbol{Z}] = e_1^\intercal \boldsymbol{\theta}^\intercal \boldsymbol{\theta} q_2 = w^\intercal \tilde{v} = 0$). We can therefore estimate:

$$\tilde{v}^\intercal \mathbb{E}[g(w^\intercal z)z] = \mathbb{E}[g(e_1^\intercal \boldsymbol{Z})]\mathbb{E}[q_2^\intercal \boldsymbol{Z}] + O\left(\frac{1}{n^{1/4}}\right) = O\left(\frac{1}{n^{1/4}}\right).$$

Combing those 2 estimations, we see that for any differentiable function $g$ $\mathbb{E}[g(w^\intercal z)v^\intercal z] = \mathbb{E}[g'(w^\intercal z)]v^\intercal w$. Therefore if we take for $g$ the mapping $t \mapsto f(w^\intercal m + t)$ (satisfying $f(w^\intercal x) = g(w^\intercal C^{1/2} z)$), we get the the identity:

$$\begin{aligned}\mathbb{E}[f(w^\intercal x)v^\intercal x] &= \mathbb{E}[g(w^\intercal C^{1/2} z)v^\intercal m] + \mathbb{E}[g(w^\intercal C^{1/2} z)]v^\intercal C^{1/2} w + O\left(\frac{1}{n^{1/4}}\right) \\ &= \mathbb{E}[f(w^\intercal x)]v^\intercal m + \mathbb{E}[g'(w^\intercal C^{1/2} z)]v^\intercal C w + O\left(\frac{1}{n^{1/4}}\right)\end{aligned}$$

With the same method, let us first compute:

$$\tilde{v}^\intercal \mathbb{E}[g(w^\intercal z)zz^\intercal]w = \mathbb{E}[g(w^\intercal z)wz]\mathbb{E}[z^\intercal \tilde{v}] = O\left(\frac{1}{n^{1/4}}\right) = w^\intercal \mathbb{E}[g(w^\intercal z)zz^\intercal]\tilde{u}.$$

Second:

$$\tilde{v}^\intercal \mathbb{E}[g(w^\intercal z)zz^\intercal]\tilde{u} = \mathbb{E}[g(w^\intercal z)]\mathbb{E}[\tilde{v}^\intercal zz^\intercal \tilde{u}] + O\left(\frac{1}{n^{1/4}}\right) = \mathbb{E}[g(w^\intercal z)]\tilde{v}^\intercal \tilde{u} + O\left(\frac{1}{n^{1/4}}\right).$$

Third:

$$\begin{aligned}w^\intercal \mathbb{E}[g(w^\intercal z)zz^\intercal]w &= \mathbb{E}[g(w^\intercal z)w^\intercal z w^\intercal z] = \mathbb{E}[g(w^\intercal z) + g'(w^\intercal z)w^\intercal z]\|w\|^2 + \\ &= \mathbb{E}[g(w^\intercal z)]\|w\|^2 + \mathbb{E}[g''(w^\intercal z)]\|w\|^4 + O\left(\frac{1}{n^{1/4}}\right).\end{aligned}$$

Therefore, $\mathbb{E}[g(w^\intercal z)\tilde{v}^\intercal zz^\intercal \tilde{v}] = \mathbb{E}[(g(w^\intercal z)]\tilde{v}^\intercal \tilde{u} + \mathbb{E}[g''(w^\intercal z)]\tilde{v}^\intercal w w^\intercal \tilde{u}$, and we can conclude as before that:

$$\begin{aligned}\tilde{v}^\intercal \mathbb{E}[f(w^\intercal x)xx^\intercal]u &= \tilde{v}^\intercal m \mathbb{E}[f(w^\intercal x)]m^\intercal u + \tilde{v}^\intercal m \mathbb{E}[g(w^\intercal C^{1/2} z)z^\intercal]C^{1/2} u \\ &\quad + \tilde{v}^\intercal C^{1/2}\mathbb{E}[g(w^\intercal C^{1/2} z)z]mu + \tilde{v}^\intercal C^{1/2}\mathbb{E}[g(w^\intercal C^{1/2} z)zz^\intercal]C^{1/2} u + O\left(\frac{1}{n^{1/4}}\right) \\ &= \mathbb{E}[f(w^\intercal x)]\tilde{v}^\intercal (mm^\intercal + C)u \\ &\quad + \mathbb{E}[f'(w^\intercal x)]\tilde{v}^\intercal (Cwm^\intercal + mCw^\intercal)u \\ &\quad + \mathbb{E}[f''(w^\intercal x)]\tilde{v}^\intercal Cww^\intercal Cu + O\left(\frac{1}{n^{1/4}}\right)\end{aligned}$$

□

Let us now employ Proposition C.3 to express the mappings (defined here for a random vector $\boldsymbol{x}_i$ in the class $\mathcal{C}_\ell$)

$$m_\ell(\boldsymbol{m}_W, \boldsymbol{C}_W) = \boldsymbol{\Lambda}^{-1}\mathbb{E}[\tilde{\boldsymbol{x}}_i h_i(\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}_{-i})]$$
$$c_\ell(\boldsymbol{m}_W, \boldsymbol{C}_W) = \frac{1}{n}\boldsymbol{\Lambda}^{-1}\mathbb{E}[\tilde{\boldsymbol{x}}_i h_i(\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}_{-i}) h_i(\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}_{-i})^\intercal \tilde{\boldsymbol{x}}_i^\intercal]\boldsymbol{\Lambda}^{-1},$$

from the class-wise means and covariances of $x_1, \ldots, x_n$ and $\boldsymbol{m}_W$ and $\boldsymbol{C}_W$ (that are respectively the mean and covariance of $\boldsymbol{W}$ but also of $\boldsymbol{W}_{-i}$). We are going to fix successively $\boldsymbol{x}_i$ and $\boldsymbol{W}_{-i}$ to be able to compute the expectations appearing in the formulations of $m_\ell$ and $c_\ell$ (it is made possible since $\boldsymbol{x}_i$ and $\boldsymbol{W}_{-i}$ are independent). Although it is not fully rigorous, we employ Proposition C.3 as if the estimations of $\mathbb{E}[f(\boldsymbol{w}^\intercal\boldsymbol{x})\boldsymbol{v}^\intercal\boldsymbol{x}]$ and $\mathbb{E}[f(\boldsymbol{w}^\intercal\boldsymbol{x})\boldsymbol{v}^\intercal\boldsymbol{x}\boldsymbol{x}^\intercal\boldsymbol{u}]$ for $\boldsymbol{w}, \boldsymbol{v}$ and $\boldsymbol{u}$ belonging to a big subset of $\mathbb{S}^{p-1}$ of measure bigger than $1 - Ce^{-cp-1/4}$ implied that the result would be true for all vectors $\boldsymbol{w}, \boldsymbol{v}, \boldsymbol{u} \in \mathbb{S}^{p-1}$. It is of course not rigorously correct, however, in practice, for the vectors $\boldsymbol{w}, \boldsymbol{v}$ and $\boldsymbol{u}$ we are considering, it appears to be valid.

To simplify the expression of the derivative in $\tilde{\boldsymbol{x}}_i$ of $h_i(\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W})$, let us replace our couple of variables $(\tilde{\boldsymbol{x}}_i, \boldsymbol{W}) \in \mathbb{R}^{p\times k} \times \mathbb{R}^{pk}$ by the variables $(\boldsymbol{x}_i, \tilde{\boldsymbol{W}}) \in \mathbb{R}^p \times \mathbb{R}^{p\times k}$ where $\tilde{\boldsymbol{W}} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k)$. We have then $h_i(\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}) = h_i(\tilde{\boldsymbol{W}}^\intercal \boldsymbol{x}_i)$. Given a twice differentiable mapping $\phi : \mathbb{R}^k \to \mathbb{R}$ and $\psi : \boldsymbol{v} \mapsto \phi(\tilde{\boldsymbol{W}}^\intercal \boldsymbol{v})$ we have the identities:

$$\nabla\psi(\boldsymbol{v}) = \tilde{\boldsymbol{W}}\nabla\phi(\tilde{\boldsymbol{W}}^\intercal\boldsymbol{v}) \qquad \text{and} \qquad d^2\boldsymbol{v} = \tilde{\boldsymbol{W}}d^2\phi\tilde{\boldsymbol{W}}^\intercal\boldsymbol{v}\tilde{\boldsymbol{W}}^\intercal$$

Now, let us follow our new notations and try to compute $m_\ell(\boldsymbol{m}_W, \boldsymbol{C}_W) \equiv \mathbb{E}_{-i}\left[\sqrt{n}\mathbb{E}_i[\boldsymbol{x}_i h^\ell(\tilde{\boldsymbol{W}}_{-i}^\intercal \boldsymbol{x}_i)^\intercal]\right]$, where we noted $h^\ell : \mathbb{R}^k \to \mathbb{R}^k$, the mapping $h_i$ for $k(i) = \ell$ (recall that $k(i)$ provides the class of $\boldsymbol{x}_i$. Let us decompose the matrix $\boldsymbol{S}_W = \mathbb{E}[\boldsymbol{W}\boldsymbol{W}^T]$ followingly:

$$\boldsymbol{S}_W = \left(\begin{array}{c|c|c} \boldsymbol{S}_W^{1,1} & \ldots & \boldsymbol{S}_W^{1,k} \\ \hline \vdots & & \vdots \\ \hline \boldsymbol{S}_W^{k,1} & \ldots & \boldsymbol{S}_W^{k,k} \end{array}\right) \in \mathcal{M}_{kp},$$

where for all $a, b \in [k]$, $\boldsymbol{S}_W^{a,b} \in \mathcal{M}_p$, so that we can introduce the low-dimensional random vector $z \sim \mathcal{N}(^\ell, \boldsymbol{K}^\ell)$ with:

$$^\ell \equiv \boldsymbol{\mu}_\ell^\intercal \boldsymbol{m}_W \qquad \text{and} \qquad \boldsymbol{K}^\ell \equiv (\mathrm{Tr}((\boldsymbol{\Sigma}_\ell + \boldsymbol{\mu}_\ell\boldsymbol{\mu}_\ell^\intercal)\boldsymbol{S}_W^{a,b}))_{1\le a,b\le k} - \boldsymbol{\mu}_\ell^\intercal \boldsymbol{m}_W \boldsymbol{m}_W^\intercal \boldsymbol{\mu}_\ell. \tag{C.10}$$

With such a choice, $z$ has the same distribution as $\tilde{\boldsymbol{x}}_i^\intercal \boldsymbol{W}_{-i}$ for $k(i) = \ell$. Then we can compute thanks to Proposition C.3:

$$\begin{aligned} m_\ell(\boldsymbol{m}_W, \boldsymbol{C}_W) &= \boldsymbol{\mu}_\ell\mathbb{E}\left[h^\ell(\tilde{\boldsymbol{W}}^\intercal\boldsymbol{x}_i)^\intercal\right] + \boldsymbol{\Sigma}_\ell\mathbb{E}_{-i}\left[\tilde{\boldsymbol{W}}_{-i}\mathbb{E}_i[dh_i^\intercal\tilde{\boldsymbol{W}}^\intercal\boldsymbol{x}_i]\right] \\ &= \boldsymbol{\mu}_\ell\mathbb{E}[h^\ell(z)^\intercal] + \boldsymbol{\Sigma}_\ell\boldsymbol{m}_W\mathbb{E}[dh_i^\intercal z] + O_{\|\cdot\|}\left(\frac{1}{n^{1/4}}\right), \end{aligned}$$

since $\|\boldsymbol{C}_W\| \le O(\sqrt{\log n/n})$ (because $\|\boldsymbol{W}\| \propto \mathcal{E}_2(\sqrt{\log n/n})$ on $\mathcal{A}_X$).

To estimate the mapping $\sigma_\ell$, let us note for simplicity $H : \boldsymbol{v} \mapsto h^\ell(\tilde{\boldsymbol{W}}_{-i}^\intercal\boldsymbol{v})h^\ell(\tilde{\boldsymbol{W}}_{-i}^\intercal\boldsymbol{v})^\intercal \in \mathcal{M}_{k,k}$, we know that: $\nabla H_{a,b}(\boldsymbol{v}) = \tilde{\boldsymbol{W}}_{-i}J(\tilde{\boldsymbol{W}}_{-i}^\intercal\boldsymbol{v})_{a,b}$ and $d^2H_{a,b}\boldsymbol{v} = \tilde{\boldsymbol{W}}_{-i}K(\tilde{\boldsymbol{W}}_{-i}^\intercal\boldsymbol{v})_{a,b}\tilde{\boldsymbol{W}}_{-i}^\intercal$, where we introduce for any $\boldsymbol{u} \in \mathbb{R}^k$ the objects:

$$J(\boldsymbol{u})_{a,b} = h_a^\ell(\boldsymbol{u})\nabla h_b^\ell(\boldsymbol{u}) + h_b^\ell(\boldsymbol{u})\nabla h_a^\ell(\boldsymbol{u}) \in \mathbb{R}^k$$
$$K(\boldsymbol{u})_{a,b} = \nabla h_a^\ell(\boldsymbol{u})\nabla h_b^\ell(\boldsymbol{u})^\intercal + h_b^\ell(\boldsymbol{u})d^2h_a^\ell\boldsymbol{u} + h_a^\ell(\boldsymbol{u})d^2h_b^\ell\boldsymbol{u} \in \mathcal{M}_k$$

Following the same strategy as previously, we can show thanks to Proposition C.3 that with the decomposition:

$$c_\ell(\boldsymbol{m}_W, \boldsymbol{C}_W) = \left(\begin{array}{c|c|c} c_\ell(\boldsymbol{m}_W, \boldsymbol{C}_W)_{1,1} & \dots & c_\ell(\boldsymbol{m}_W, \boldsymbol{C}_W)_{1,k} \\ \hline \vdots & & \vdots \\ \hline c_\ell(\boldsymbol{m}_W, \boldsymbol{C}_W)_{k,1} & \dots & c_\ell(\boldsymbol{m}_W, \boldsymbol{C}_W)_{k,k} \end{array}\right) \in \mathcal{M}_{kp},$$

for any $a, b \in [k]$ and $\ell \in [k]$ such that $k(i) = \ell$:

$$\begin{aligned} \boldsymbol{\Lambda} c_\ell(\boldsymbol{m}_W, \boldsymbol{C}_W)_{a,b} \boldsymbol{\Lambda} &= \frac{1}{n}\mathbb{E}[H_{a,b}(\boldsymbol{x}_i)\boldsymbol{x}_i\boldsymbol{x}_i^\intercal] \\ &= \frac{1}{n}\boldsymbol{\Sigma}_\ell \tilde{\boldsymbol{m}}_W \mathbb{E}[J(z)_{a,b}]\boldsymbol{\mu}_\ell^\intercal + \frac{1}{n}\boldsymbol{\mu}_\ell \mathbb{E}[J(z)_{a,b}]^\intercal \tilde{\boldsymbol{m}}_W^\intercal \boldsymbol{\Sigma}_\ell \\ &\quad + \frac{1}{n}\boldsymbol{\Sigma}_\ell \left( \tilde{\boldsymbol{m}}_W \mathbb{E}[K(z)_{a,b}]\tilde{\boldsymbol{m}}_W^\intercal + \sum_{1 \le c,d \le k} S_W^{c,d} (\mathbb{E}[K(z)_{a,b}])_{c,d} \right) \boldsymbol{\Sigma}_\ell \\ &\quad + \frac{1}{n}\mathbb{E}[h_a^\ell(z) h_b^\ell(z)](\boldsymbol{\mu}_\ell\boldsymbol{\mu}_\ell^\intercal + \boldsymbol{\Sigma}_\ell) + O_{\|\cdot\|}\left(\frac{1}{n^{1/4}}\right) \end{aligned}$$

We are then left to estimating the derivatives of $h$ to be able to compute $m_\ell$ and $c_\ell$. From the implicit expression of $g_i$ given by Proposition 4.3 one can deduce the formulas, for $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^k$:

$$dg_i\boldsymbol{v} = -(\Delta_i df_i g(\boldsymbol{v}) - \boldsymbol{I}_k)^{-1} \qquad \text{and} \qquad d^2 g_i \boldsymbol{v}\cdot\boldsymbol{u} = dg_i\boldsymbol{v}\Delta_i(d^2 f_i g(\boldsymbol{v})\cdot\boldsymbol{u})dg_i\boldsymbol{v}$$

Then, recalling the identity $h_i = f_i \circ g_i$, one can derive from the upper formulas the expression of the differentiates of $h_i$ thanks to the identities for $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^k$:

$$dh_i\boldsymbol{v} = \Delta_i^{-1}(dg_i\boldsymbol{v} - \boldsymbol{I}_k) \qquad \text{and} \qquad d^2 h_i \boldsymbol{v}\cdot\boldsymbol{u} = \Delta_i^{-1} d^2 g_i \boldsymbol{v}\cdot\boldsymbol{u}$$

### C.3.2 Proofs of Section 5.2

#### C.3.2.1 Proof of Theorem 5.5

*Proof of Theorem 5.5.* For clarity, we simply write $\boldsymbol{Q}(z) = \boldsymbol{Q}$ and $\delta(z) = \delta$ removing the dependence on $z$, and let the resolvent $\boldsymbol{Q}_{-i}$ which is $\boldsymbol{Q}$ without the $i$-th datum $\boldsymbol{x}_i$ defined as

$$\boldsymbol{Q}_{-i} = \left(\frac{1}{n}\boldsymbol{X}\boldsymbol{X}^\intercal - \frac{1}{n}\boldsymbol{x}_i\boldsymbol{x}_i^\intercal + z\boldsymbol{I}_p\right)^{-1} \tag{C.11}$$

And let

$$\boldsymbol{m} \equiv \sqrt{\frac{\varepsilon}{1+\alpha^2(1-\varepsilon)}}\boldsymbol{\mu} \tag{C.12}$$

**Estimation of $\mathbb{E}[g(\boldsymbol{x})]$** Using the identity $\boldsymbol{Q}\boldsymbol{x}_i = \frac{\boldsymbol{Q}_{-i}\boldsymbol{x}_i}{1+\frac{1}{n}\boldsymbol{x}_i\boldsymbol{Q}_{-i}\boldsymbol{x}_i}$, we have for $\boldsymbol{x} \in \mathcal{C}_a$

$$\begin{aligned} \mathbb{E}\left[\boldsymbol{x}^\intercal\boldsymbol{w}\right] &= \frac{1}{n}\mathbb{E}\left[\boldsymbol{x}^\intercal\boldsymbol{Q}\boldsymbol{X}\boldsymbol{y}\right] = \frac{1}{n}\sum_{i=1}^n y_i \mathbb{E}\left[\boldsymbol{x}^\intercal\boldsymbol{Q}\boldsymbol{x}_i\right] = \frac{1}{n}\sum_{i=1}^n y_i \mathbb{E}\left[\frac{\boldsymbol{x}^\intercal\boldsymbol{Q}_{-i}\boldsymbol{x}_i}{1+\frac{1}{n}\boldsymbol{x}_i^\intercal\boldsymbol{Q}_{-i}\boldsymbol{x}_i}\right] \\ &= \frac{1}{n}\sum_{i=1}^n y_i \mathbb{E}\left[\frac{\boldsymbol{x}^\intercal\boldsymbol{Q}_{-i}\boldsymbol{x}_i}{1+\delta}\right] + \mathcal{O}\left(\frac{1}{\sqrt{n}}\right) = (-1)^a \frac{\boldsymbol{\mu}^\intercal\bar{\boldsymbol{Q}}\boldsymbol{m}}{1+\delta} + \mathcal{O}\left(\frac{1}{\sqrt{n}}\right) \end{aligned}$$

**Estimation of $\mathbb{E}[g(x)^2]$**

$$\begin{aligned}
\mathbb{E}\left[(x^\intercal w)^2\right] &= \frac{1}{n^2}\mathbb{E}\left[y^\intercal X^\intercal Q x x^\intercal Q X y\right] = \frac{1}{n^2}\mathbb{E}\left[y^\intercal X^\intercal Q C_1 Q X y\right] \\
&= \frac{1}{n^2}\sum_{i,j=1}^{n} y_i y_j \mathbb{E}\left[x_i^\intercal Q C_1 Q x_j\right] = \frac{1}{n^2}\sum_{i=1}^{n} y_i^2 \mathbb{E}\left[x_i^\intercal Q C_1 Q x_i\right] + \frac{1}{n^2}\sum_{i\neq j} y_i y_j \mathbb{E}\left[x_i^\intercal Q C_1 Q x_j\right] \\
&= \frac{1}{n^2}\sum_{i=1}^{n} y_i^2 \mathbb{E}\left[\frac{x_i^\intercal Q_{-i} C_1 Q_{-i} x_i}{(1+\delta)^2}\right] + \frac{1}{n^2}\sum_{i\neq j} y_i y_j \mathbb{E}\left[\frac{x_i^\intercal Q_{-i} C_1 Q_{-j} x_j}{(1+\delta)^2}\right] + \mathcal{O}\left(\frac{1}{\sqrt{n}}\right) \\
&= \frac{1}{n}\frac{\operatorname{Tr}\left(C\mathbb{E}\left[Q_{-i} C_1 Q_{-i}\right]\right)}{(1+\delta)^2} + \frac{1}{n^2}\sum_{i\neq j} y_i y_j \mathbb{E}\left[\frac{x_i^\intercal Q_{-i} C_1 Q_{-j} x_j}{(1+\delta)^2}\right] + \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)
\end{aligned}$$

And using the identity $Q = Q_{-i} - \frac{Q_{-i}\frac{1}{n}x_i x_i^\intercal Q_{-i}}{1+\frac{1}{n}x_i^\intercal Q_{-i} x_i}$, the second term develops as

$$\begin{aligned}
&\frac{1}{n^2}\sum_{i\neq j} y_i y_j \mathbb{E}\left[\frac{x_i^\intercal Q_{-i} C_1 Q_{-j} x_j}{(1+\delta)^2}\right] \\
&= \frac{1}{n^2}\sum_{i\neq j} y_i y_j \mathbb{E}\left[\frac{x_i^\intercal Q_{-ij} C_1 Q_{-ji} x_j}{(1+\delta)^2}\right] - \frac{1}{n^3}\sum_{i\neq j} y_i y_j \mathbb{E}\left[\frac{x_i^\intercal Q_{-ij} C_1 Q_{-ji} x_i x_i^\intercal Q_{-ji} x_j}{(1+\delta)^3}\right] \\
&- \frac{1}{n^3}\sum_{i\neq j} y_i y_j \mathbb{E}\left[\frac{x_i^\intercal Q_{-ji} x_j x_j^\intercal Q_{-ij} C_1 Q_{-ji} x_j}{(1+\delta)^3}\right] + \frac{1}{n^4}\sum_{i\neq j} y_i y_j \mathbb{E}\left[\frac{x_i^\intercal Q_{-ij} x_j x_j^\intercal Q_{-ij} C_1 Q_{-ji} x_i x_i^\intercal Q_{-ij} x_j}{(1+\delta)^4}\right] + \mathcal{O}\left(\frac{1}{\sqrt{n}}\right) \\
&= \frac{m^\intercal \mathbb{E}[Q_{-ij} C_1 Q_{-ji}] m}{(1+\delta)^2} - \frac{2\operatorname{Tr}(\mathbb{E}[CQC_1 Q])}{n(1+\delta)^3} m^\intercal \bar{Q} m + \frac{1}{n^2(1+\delta)^4}(m^\intercal \bar{Q} m)^2 m^\intercal \mathbb{E}[QC_1 Q] m + \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)
\end{aligned}$$

where the term $\mathbb{E}[QAQ]$ is handled by

$$\begin{aligned}
\eta(A) &\equiv \frac{1}{n}\operatorname{Tr}\left(C_\varepsilon \mathbb{E}[QAQ]\right) = \frac{(1+\delta)\frac{1}{n}\operatorname{Tr}\left(C_\varepsilon \bar{Q} A \bar{Q}\right)}{(1+\delta)^2 - \frac{1}{n}\operatorname{Tr}\left(C_\varepsilon \bar{Q} C_\varepsilon \bar{Q}\right)} \\
\Delta(A) &\equiv \mathbb{E}[QAQ] = \bar{Q} A \bar{Q} + \frac{\eta(A)}{1+\delta}\bar{Q} C_\varepsilon \bar{Q}
\end{aligned}$$

Putting all together yields to

$$\mathbb{E}\left[(x^\intercal w)^2\right] = \frac{\eta(C_1)}{(1+\delta)^2} + \frac{m^\intercal \Delta(C_1) m}{(1+\delta)^2} - \frac{2\eta(C_1) m^\intercal \bar{Q} m}{(1+\delta)^3} + \mathcal{O}\left(\frac{1}{n^2}\right)$$

□

### C.3.2.2 Proof of Theorem 5.6

*Proof of Theorem 5.6.* Using the previous notations and matrix identities, for $x_i \in \mathcal{C}_a$ a sample from the training set $X$, we have:

**Estimation of $\mathbb{E}[g(x_i)]$**

$$\begin{aligned}\mathbb{E}[x_i^\intercal w] &= \frac{1}{n}\mathbb{E}[x_i^\intercal QXy] = \frac{1}{n}\sum_{j=1}^n y_j \mathbb{E}[x_i^\intercal Q x_j] \\ &= \frac{1}{n}\mathbb{E}\left[\frac{x_i^\intercal Q_{-i} x_i}{1+\frac{1}{n}x_i^\intercal Q_{-i}x_i}\right] + \frac{1}{n}\sum_{j\neq i} y_j \mathbb{E}\left[\frac{x_i^\intercal Q_{-ji}x_j}{(1+\delta)^2}\right] + \mathcal{O}\left(\frac{1}{\sqrt{n}}\right) \\ &= \frac{\delta}{1+\delta} + (-1)^a \frac{m^\intercal \bar{Q} m}{(1+\delta)^2} + \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)\end{aligned}$$

**Estimation of $\mathbb{E}[g(x_i)^2]$**

$$\begin{aligned}\mathbb{E}\left[(x_i^\intercal w)^2\right] &= \frac{1}{n^2}\mathbb{E}\left[y^\intercal X^\intercal Q x_i x_i^\intercal Q X y\right] = \frac{1}{n^2}\mathbb{E}\left[\frac{y^\intercal X^\intercal Q_{-i} x_i x_i^\intercal Q_{-i} X y}{(1+\delta)^2}\right] + \mathcal{O}\left(\frac{1}{\sqrt{n}}\right) \\ &= \frac{1}{n^2}\sum_{j,k=1}^n y_j y_k \mathbb{E}\left[\frac{x_j^\intercal Q_{-i} x_i x_i^\intercal Q_{-i} x_k}{(1+\delta)^2}\right] + \mathcal{O}\left(\frac{1}{\sqrt{n}}\right) \\ &= \frac{1}{n^2}\sum_{j=1}^n y_j^2 \mathbb{E}\left[\frac{x_j^\intercal Q_{-i} x_i x_i^\intercal Q_{-i} x_j}{(1+\delta)^2}\right] + \frac{1}{n^2}\sum_{j\neq k} y_j y_k \mathbb{E}\left[\frac{x_j^\intercal Q_{-i} x_i x_i^\intercal Q_{-i} x_k}{(1+\delta)^2}\right] + \mathcal{O}\left(\frac{1}{\sqrt{n}}\right) \\ &= \frac{1}{n^2}\mathbb{E}\left[\frac{(x_i^\intercal Q_{-i} x_i)^2}{(1+\delta)^2}\right] + \frac{1}{n^2}\sum_{j\neq i} y_j^2 \mathbb{E}\left[\frac{x_j^\intercal Q_{-i} x_i x_i^\intercal Q_{-i} x_j}{(1+\delta)^2}\right] + \frac{1}{n^2}\sum_{j\neq i} y_j y_i \mathbb{E}\left[\frac{x_j^\intercal Q_{-i} x_i x_i^\intercal Q_{-i} x_i}{(1+\delta)^2}\right] \\ &+ \frac{1}{n^2}\sum_{i\neq j\neq k} y_j y_k \mathbb{E}\left[\frac{x_j^\intercal Q_{-i} x_i x_i^\intercal Q_{-i} x_k}{(1+\delta)^2}\right] + \mathcal{O}\left(\frac{1}{\sqrt{n}}\right) \\ &= \left(\frac{\delta}{1+\delta}\right)^2 + \frac{1}{n^2}\sum_{j\neq i}\mathbb{E}\left[\frac{x_j^\intercal Q_{-ij} x_i x_i^\intercal Q_{-ij} x_j}{(1+\delta)^4}\right] + \frac{1}{n^2}\sum_{j\neq i} y_j y_i \mathbb{E}\left[\frac{x_j^\intercal Q_{-ij} x_i x_i^\intercal Q_{-i} x_i}{(1+\delta)^3}\right] \\ &+ \frac{1}{n^2}\sum_{i\neq j\neq k} y_j y_k \mathbb{E}\left[\frac{x_j^\intercal Q_{-ij} x_i x_i^\intercal Q_{-ik} x_k}{(1+\delta)^4}\right] + \mathcal{O}\left(\frac{1}{\sqrt{n}}\right) \\ &= \left(\frac{\delta}{1+\delta}\right)^2 + \frac{\frac{1}{n}\operatorname{Tr}\left(C_\varepsilon \mathbb{E}[Q_{-ij} C_\varepsilon Q_{-ij}]\right)}{(1+\delta)^4} + \frac{\delta m^\intercal \bar{Q} m}{(1+\delta)^3} \\ &+ \frac{1}{n^2}\sum_{i\neq j\neq k} y_j y_k \frac{\mathbb{E}\left[x_j^\intercal Q_{-ij} C_\varepsilon Q_{-ik} x_k\right]}{(1+\delta)^4} + \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)\end{aligned}$$

where we have previously estimated the term $\frac{1}{n^2}\sum_{i\neq j\neq k} y_j y_k \mathbb{E}[x_j^\intercal Q_{-ij} C_\varepsilon Q_{-ik} x_k]$ as

$$\frac{1}{n^2}\sum_{i\neq j\neq k} y_j y_k \mathbb{E}[x_j^\intercal Q_{-ij} C_\varepsilon Q_{-ik} x_k] = m^\intercal \mathbb{E}[Q_{-ijk} C_\varepsilon Q_{-ijk}] m - \frac{2\operatorname{Tr}(\mathbb{E}[C_\varepsilon Q C_\varepsilon Q])}{n(1+\delta)} m^\intercal \bar{Q} m + \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$$

Hence, putting all together we get

$$\begin{aligned}\mathbb{E}[g(x_i)^2] &= \left(\frac{\delta}{1+\delta}\right)^2 + \frac{\eta(C_\varepsilon)}{(1+\delta)^4} + \frac{\delta m^\intercal \bar{Q} m}{(1+\delta)^3} + \frac{m^\intercal \Delta(C_\varepsilon) m}{(1+\delta)^4} - \frac{2\eta m^\intercal \bar{Q} m}{(1+\delta)^5} + \mathcal{O}\left(\frac{1}{\sqrt{n}}\right) \\ &= \left(\frac{\delta}{1+\delta}\right)^2 + \frac{\eta(C_\varepsilon)}{(1+\delta)^4} + m^\intercal\left(\frac{\delta \bar{Q}}{(1+\delta)^3} + \frac{\Delta(C_\varepsilon)}{(1+\delta)^4} - \frac{2\eta(C_\varepsilon)\bar{Q}}{(1+\delta)^5}\right) m + \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)\end{aligned}$$

and the CLT is obtained with similar arguments than louart2018random. □

#### C.3.2.3 Optimality

Denote

$$a = \frac{\varepsilon}{1+\alpha^2(1-\varepsilon)}, \quad b = 1+cq, \quad d = \frac{1-\varepsilon}{\varepsilon}$$

And suppose, for some $\beta > 0$ and some integer $s$

$$\boldsymbol{\mu} = \beta \sum_{i=1}^{s} \frac{(-1)^i}{\sqrt{s}} \boldsymbol{e}_i$$

where $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_s$ are the $s$-first canonical vectors of $\mathbb{R}^p$, with $[e_i]_i = 1$. The expression of the test missclassification error involves the following terms which depend on $\alpha$:

$$\boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}} \boldsymbol{\mu}, \quad \boldsymbol{\mu}^\intercal \boldsymbol{\Delta} \boldsymbol{\mu}$$

Let us first expression $\boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}} \boldsymbol{\mu}$, which develops as

$$q_1 \equiv \boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}} \boldsymbol{\mu} = \boldsymbol{\mu}^\intercal \mathcal{D}_z \boldsymbol{\mu} - \frac{a(\boldsymbol{\mu}^\intercal \mathcal{D}_z \boldsymbol{\mu})^2}{b + a\boldsymbol{\mu}^\intercal \mathcal{D}_z \boldsymbol{\mu}}$$

where

$$d_1 \equiv \boldsymbol{\mu}^\intercal \mathcal{D}_z \boldsymbol{\mu} = qb \sum_{i=1}^{p} \frac{\mu_i^2}{b + aqd\mu_i^2} = \frac{qb\beta^2}{b + aqd\frac{\beta^2}{s}}$$

The term $\boldsymbol{\mu}^\intercal \boldsymbol{\Delta} \boldsymbol{\mu}$ involves the quantity $\eta$ which reduces to

$$\eta = \frac{(1+\delta)cq^2}{(1+\delta)^2 - cq^2}$$

And thus $\boldsymbol{\mu}^\intercal \boldsymbol{\Delta} \boldsymbol{\mu}$ develops as

$$\boldsymbol{\mu}^\intercal \boldsymbol{\Delta} \boldsymbol{\mu} = \boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}} \boldsymbol{C}_1 \bar{\boldsymbol{Q}} \boldsymbol{\mu} + \frac{\eta}{1+\delta} \boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}} \boldsymbol{C}_\varepsilon \bar{\boldsymbol{Q}} \boldsymbol{\mu}$$

where $\boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}} \boldsymbol{C}_1 \bar{\boldsymbol{Q}} \boldsymbol{\mu}$ is given by

$$\boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}} \boldsymbol{C}_1 \bar{\boldsymbol{Q}} \boldsymbol{\mu} = \boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}}^2 \boldsymbol{\mu} + (\boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}} \boldsymbol{\mu})^2$$

And

$$\begin{aligned} q_2 \equiv \boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}}^2 \boldsymbol{\mu} &= \boldsymbol{\mu}^\intercal \left( \mathcal{D}_z - \frac{a\mathcal{D}_z \boldsymbol{\mu}\boldsymbol{\mu}^\intercal \mathcal{D}_z}{b + a\boldsymbol{\mu}^\intercal \mathcal{D}_z \boldsymbol{\mu}} \right)^2 \boldsymbol{\mu} \\ &= \boldsymbol{\mu}^\intercal \mathcal{D}_z^2 \boldsymbol{\mu} - \frac{2a\boldsymbol{\mu}^\intercal \mathcal{D}_z^2 \boldsymbol{\mu}\boldsymbol{\mu}^\intercal \mathcal{D}_z \boldsymbol{\mu}}{b + a\boldsymbol{\mu}^\intercal \mathcal{D}_z \boldsymbol{\mu}} + \frac{a^2 \boldsymbol{\mu}^\intercal \mathcal{D}_z^2 \boldsymbol{\mu} (\boldsymbol{\mu}^\intercal \mathcal{D}_z \boldsymbol{\mu})^2}{(b + a\boldsymbol{\mu}^\intercal \mathcal{D}_z \boldsymbol{\mu})^2} \end{aligned}$$

where

$$d_2 \equiv \boldsymbol{\mu}^\intercal \mathcal{D}_z^2 \boldsymbol{\mu} = q^2 b^2 \sum_{i=1}^{p} \frac{\mu_i^2}{(b + aqd\mu_i^2)^2} = \frac{q^2 b^2 \beta^2}{(b + aqd\frac{\beta^2}{s})^2}$$

And the remaining term $\boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}} \boldsymbol{C}_\varepsilon \bar{\boldsymbol{Q}} \boldsymbol{\mu}$ develops as

$$\boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}} \boldsymbol{C}_\varepsilon \bar{\boldsymbol{Q}} \boldsymbol{\mu} = \boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}}^2 \boldsymbol{\mu} + a(\boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}} \boldsymbol{\mu})^2 + ad\boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}} \operatorname{diag}\left(\boldsymbol{\mu}^{\odot 2} + 2\alpha\boldsymbol{\mu}\right) \bar{\boldsymbol{Q}} \boldsymbol{\mu}$$

where it remains to express $\boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}} \operatorname{diag}\left(\boldsymbol{\mu}^{\odot 2} + 2\alpha\boldsymbol{\mu}\right) \bar{\boldsymbol{Q}}\boldsymbol{\mu}$ which develops as

$$\boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}} \operatorname{diag}\left(\boldsymbol{\mu}^{\odot 2} + 2\alpha\boldsymbol{\mu}\right) \bar{\boldsymbol{Q}}\boldsymbol{\mu} = \boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}} \operatorname{diag}\left(\boldsymbol{\mu}^{\odot 2}\right) \bar{\boldsymbol{Q}}\boldsymbol{\mu} + 2\alpha\boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}} \operatorname{diag}\left(\boldsymbol{\mu}\right) \bar{\boldsymbol{Q}}\boldsymbol{\mu}$$

with

$$\begin{aligned}
q_3 \equiv \boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}} \operatorname{diag}\left(\boldsymbol{\mu}^{\odot 2}\right) \bar{\boldsymbol{Q}}\boldsymbol{\mu} &= \boldsymbol{\mu}^\intercal \left(\mathcal{D}_z - \frac{a\mathcal{D}_z\boldsymbol{\mu}\boldsymbol{\mu}^\intercal\mathcal{D}_z}{b + a\boldsymbol{\mu}^\intercal\mathcal{D}_z\boldsymbol{\mu}}\right) \operatorname{diag}\left(\boldsymbol{\mu}^{\odot 2}\right) \left(\mathcal{D}_z - \frac{a\mathcal{D}_z\boldsymbol{\mu}\boldsymbol{\mu}^\intercal\mathcal{D}_z}{b + a\boldsymbol{\mu}^\intercal\mathcal{D}_z\boldsymbol{\mu}}\right) \boldsymbol{\mu} \\
&= \boldsymbol{\mu}^\intercal \left(\mathcal{D}_z \operatorname{diag}\left(\boldsymbol{\mu}^{\odot 2}\right) - \frac{a\mathcal{D}_z\boldsymbol{\mu}\boldsymbol{\mu}^\intercal\mathcal{D}_z}{b + a\boldsymbol{\mu}^\intercal\mathcal{D}_z\boldsymbol{\mu}} \operatorname{diag}\left(\boldsymbol{\mu}^{\odot 2}\right)\right) \left(\mathcal{D}_z - \frac{a\mathcal{D}_z\boldsymbol{\mu}\boldsymbol{\mu}^\intercal\mathcal{D}_z}{b + a\boldsymbol{\mu}^\intercal\mathcal{D}_z\boldsymbol{\mu}}\right) \boldsymbol{\mu} \\
&= \boldsymbol{\mu}^\intercal \left(\mathcal{D}_z \operatorname{diag}\left(\boldsymbol{\mu}^{\odot 2}\right) \mathcal{D}_z - \frac{a\mathcal{D}_z\boldsymbol{\mu}\boldsymbol{\mu}^\intercal\mathcal{D}_z}{b + a\boldsymbol{\mu}^\intercal\mathcal{D}_z\boldsymbol{\mu}} \operatorname{diag}\left(\boldsymbol{\mu}^{\odot 2}\right) \mathcal{D}_z - \mathcal{D}_z \operatorname{diag}\left(\boldsymbol{\mu}^{\odot 2}\right) \frac{a\mathcal{D}_z\boldsymbol{\mu}\boldsymbol{\mu}^\intercal\mathcal{D}_z}{b + a\boldsymbol{\mu}^\intercal\mathcal{D}_z\boldsymbol{\mu}}\right. \\
&\qquad \left. + \frac{a\mathcal{D}_z\boldsymbol{\mu}\boldsymbol{\mu}^\intercal\mathcal{D}_z}{b + a\boldsymbol{\mu}^\intercal\mathcal{D}_z\boldsymbol{\mu}} \operatorname{diag}\left(\boldsymbol{\mu}^{\odot 2}\right) \frac{a\mathcal{D}_z\boldsymbol{\mu}\boldsymbol{\mu}^\intercal\mathcal{D}_z}{b + a\boldsymbol{\mu}^\intercal\mathcal{D}_z\boldsymbol{\mu}}\right) \boldsymbol{\mu} \\
&= \boldsymbol{\mu}^\intercal\mathcal{D}_z \operatorname{diag}\left(\boldsymbol{\mu}^{\odot 2}\right) \mathcal{D}_z\boldsymbol{\mu} - \boldsymbol{\mu}^\intercal \frac{a\mathcal{D}_z\boldsymbol{\mu}\boldsymbol{\mu}^\intercal\mathcal{D}_z}{b + a\boldsymbol{\mu}^\intercal\mathcal{D}_z\boldsymbol{\mu}} \operatorname{diag}\left(\boldsymbol{\mu}^{\odot 2}\right) \mathcal{D}_z\boldsymbol{\mu} - \boldsymbol{\mu}^\intercal\mathcal{D}_z \operatorname{diag}\left(\boldsymbol{\mu}^{\odot 2}\right) \frac{a\mathcal{D}_z\boldsymbol{\mu}\boldsymbol{\mu}^\intercal\mathcal{D}_z}{b + a\boldsymbol{\mu}^\intercal\mathcal{D}_z\boldsymbol{\mu}}\boldsymbol{\mu} \\
&\qquad + \boldsymbol{\mu}^\intercal \frac{a\mathcal{D}_z\boldsymbol{\mu}\boldsymbol{\mu}^\intercal\mathcal{D}_z}{b + a\boldsymbol{\mu}^\intercal\mathcal{D}_z\boldsymbol{\mu}} \operatorname{diag}\left(\boldsymbol{\mu}^{\odot 2}\right) \frac{a\mathcal{D}_z\boldsymbol{\mu}\boldsymbol{\mu}^\intercal\mathcal{D}_z}{b + a\boldsymbol{\mu}^\intercal\mathcal{D}_z\boldsymbol{\mu}}\boldsymbol{\mu}
\end{aligned}$$

and

$$d_3 \equiv \boldsymbol{\mu}^\intercal\mathcal{D}_z \operatorname{diag}\left(\boldsymbol{\mu}^{\odot 2}\right) \mathcal{D}_z\boldsymbol{\mu} = q^2b^2 \sum_{i=1}^{p} \frac{\mu_i^4}{(b + aqd\mu_i^2)^2} = \frac{q^2b^2\frac{\beta^2}{s}}{(b + aqd\frac{\beta^2}{s})^2}$$

Similarly $\boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}} \operatorname{diag}\left(\boldsymbol{\mu}\right) \bar{\boldsymbol{Q}}\boldsymbol{\mu}$ develops as

$$\begin{aligned}
q_4 \equiv \boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}} \operatorname{diag}\left(\boldsymbol{\mu}\right) \bar{\boldsymbol{Q}}\boldsymbol{\mu} &= \boldsymbol{\mu}^\intercal\mathcal{D}_z \operatorname{diag}\left(\boldsymbol{\mu}\right) \mathcal{D}_z\boldsymbol{\mu} - \boldsymbol{\mu}^\intercal \frac{a\mathcal{D}_z\boldsymbol{\mu}\boldsymbol{\mu}^\intercal\mathcal{D}_z}{b + a\boldsymbol{\mu}^\intercal\mathcal{D}_z\boldsymbol{\mu}} \operatorname{diag}\left(\boldsymbol{\mu}\right) \mathcal{D}_z\boldsymbol{\mu} \\
&\qquad - \boldsymbol{\mu}^\intercal\mathcal{D}_z \operatorname{diag}\left(\boldsymbol{\mu}\right) \frac{a\mathcal{D}_z\boldsymbol{\mu}\boldsymbol{\mu}^\intercal\mathcal{D}_z}{b + a\boldsymbol{\mu}^\intercal\mathcal{D}_z\boldsymbol{\mu}}\boldsymbol{\mu} + \boldsymbol{\mu}^\intercal \frac{a\mathcal{D}_z\boldsymbol{\mu}\boldsymbol{\mu}^\intercal\mathcal{D}_z}{b + a\boldsymbol{\mu}^\intercal\mathcal{D}_z\boldsymbol{\mu}} \operatorname{diag}\left(\boldsymbol{\mu}\right) \frac{a\mathcal{D}_z\boldsymbol{\mu}\boldsymbol{\mu}^\intercal\mathcal{D}_z}{b + a\boldsymbol{\mu}^\intercal\mathcal{D}_z\boldsymbol{\mu}}\boldsymbol{\mu}
\end{aligned}$$

with

$$d_4 \equiv \boldsymbol{\mu}^\intercal\mathcal{D}_z \operatorname{diag}\left(\boldsymbol{\mu}\right) \mathcal{D}_z\boldsymbol{\mu} = q^2b^2 \sum_{i=1}^{p} \frac{\mu_i^3}{(b + aqd\mu_i^2)^2} = \frac{q^2b^2\frac{\beta^2}{\sqrt{s}}}{(b + aqd\frac{\beta^2}{s})^2}$$

Thus, putting all together, we have

$$\begin{aligned}
k_\varepsilon \equiv \boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}}\boldsymbol{C}_\varepsilon \bar{\boldsymbol{Q}}\boldsymbol{\mu} &= q_2 + aq_1^2 + ad\left(q_3 + 2q_4\sqrt{\frac{\frac{\varepsilon}{a} - 1}{1 - \varepsilon}}\right) \\
k_1 \equiv \boldsymbol{\mu}^\intercal \bar{\boldsymbol{Q}}\boldsymbol{C}_1 \bar{\boldsymbol{Q}}\boldsymbol{\mu} &= q_2 + q_1^2 \\
h \equiv \boldsymbol{\mu}^\intercal \boldsymbol{\Delta}\boldsymbol{\mu} &= k_1 + \frac{\eta}{1 + \delta}k_\varepsilon
\end{aligned}$$

where

$$\begin{aligned}
q_1 &= d_1 - \frac{ad_1^2}{b + ad_1}, \quad q_2 = d_2 - \frac{2ad_2d_1}{b + ad_1} + \frac{a^2d_2d_1^2}{(b + ad_1)^2} \\
q_3 &= d_3 - \frac{2ad_1d_3}{b + ad_1} + \frac{a^2d_1^2d_3}{(b + ad_1)^2}, \quad q_4 = d_4 - \frac{2ad_1d_4}{b + ad_1} + \frac{a^2d_1^2d_4}{(b + ad_1)^2}
\end{aligned}$$

And finally

$$d_1 = \frac{qb\beta^2}{b + aqd\frac{\beta^2}{s}}, \quad d_2 = \frac{q^2b^2\beta^2}{(b + aqd\frac{\beta^2}{s})^2}, \quad d_3 = \frac{q^2b^2\frac{\beta^2}{s}}{(b + aqd\frac{\beta^2}{s})^2}, \quad d_4 = \frac{q^2b^2\frac{\beta^2}{\sqrt{s}}}{(b + aqd\frac{\beta^2}{s})^2}$$

Therefore, the mean and variance of the decision are given by

$$\begin{aligned} m_\ell &= (-1)^\ell \sqrt{a}\frac{q_1}{1+\delta} \\ \nu &= \frac{1}{(1+\delta)^2}\left(\eta + a\left[h - q_1 - \frac{2\eta q_1}{1+\delta}\right]\right) \end{aligned}$$

with the change of variable $a \leftarrow \frac{\varepsilon}{1+\alpha^2(1-\varepsilon)}$, the optimal $\alpha$ is provided by

$$\frac{\partial}{\partial a} Q\left(\frac{m_\ell}{\sqrt{\nu}}\right) = 0$$

Since $Q'(x) = -\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$, the above equation reduces to

$$\sqrt{\nu}\frac{\partial m_\ell}{\partial a} - m_\ell \frac{\partial \sqrt{\nu}}{\partial a} = 0$$

# Bibliography


[A+02] Suykens Johan AK et al. *Least squares support vector machines*. World Scientific, 2002.

[AKC18] Hafiz Tiomoko Ali, Abla Kammoun, and Romain Couillet. Random matrix-improved kernels for large dimensional spectral clustering. In *IEEE Statistical Signal Processing Workshop 2018*, 2018.

[AMGS12] Konstantin Avrachenkov, Alexey Mishenin, Paulo Gonçalves, and Marina Sokol. Generalized optimization framework for graph-based semi-supervised learning. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 966–974. SIAM, 2012.

[And63] Theodore Wilbur Anderson. Asymptotic theory for principal component analysis. *The Annals of Mathematical Statistics*, 34(1):122–148, 1963.

[APD14] Megasthenis Asteris, Dimitris Papailiopoulos, and Alexandros Dimakis. Nonnegative sparse pca with provable guarantees. In *International Conference on Machine Learning*, pages 1728–1736, 2014.

[AS17] Madhu S Advani and Andrew M Saxe. High-dimensional dynamics of generalization error in neural networks. *arXiv preprint arXiv:1710.03667*, 2017.

[ASD18] Joseph Antognini and Jascha Sohl-Dickstein. Pca of high dimensional random walks with comparison to neural network training. In *Advances in Neural Information Processing Systems*, pages 10307–10316, 2018.

[ASE17] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

[ASEA19] Ali Alnoman, Shree Krishna Sharma, Waleed Ejaz, and Alagan Anpalagan. Emerging edge computing technologies for distributed iot systems. *IEEE Network*, 33(6):140–147, 2019.

[B+09] Yoshua Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[BAP+05] Jinho Baik, Gérard Ben Arous, Sandrine Péché, et al. Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices. *The Annals of Probability*, 33(5):1643–1697, 2005.

[BCV13] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, Aug 2013.

[BDS18] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[BEO19] Eugene Belilovsky, Michael Eickenberg, and Edouard Oyallon. Greedy layerwise learning can scale to imagenet. In *International Conference on Machine Learning*, pages 583–593, 2019.

[BGC16] Florent Benaych-Georges and Romain Couillet. Spectral analysis of the gram matrix of mixture models. *ESAIM: Probability and Statistics*, 20:217–237, 2016.

[BGN11] Florent Benaych-Georges and Raj Rao Nadakuditi. The eigenvalues and eigenvectors of finite, low rank perturbations of large random matrices. *Advances in Mathematics*, 227(1):494–521, 2011.

[BGN12] Florent Benaych-Georges and Raj Rao Nadakuditi. The singular values and vectors of low rank perturbations of large rectangular random matrices. *Journal of Multivariate Analysis*, 111:120–135, 2012.

[Bil08] Patrick Billingsley. *Probability and measure*. John Wiley & Sons, 2008.

[BL08] Peter J Bickel and Elizaveta Levina. Covariance regularization by thresholding. *The Annals of Statistics*, pages 2577–2604, 2008.

[BS+98a] Zhi-Dong Bai, Jack W Silverstein, et al. No eigenvalues outside the support of the limiting spectral distribution of large-dimensional sample covariance matrices. *The Annals of Probability*, 26(1):316–345, 1998.

[BS98b] Sean Borman and Robert L Stevenson. Super-resolution from image sequences-a review. In *Circuits and Systems, 1998. Proceedings. 1998 Midwest Symposium on*, pages 374–378. IEEE, 1998.

[BS08] Zhidong D Bai and Jack W Silverstein. Clt for linear spectral statistics of large-dimensional sample covariance matrices. In *Advances In Statistics*, pages 281–333. World Scientific, 2008.

[BS10] Zhidong Bai and Jack W Silverstein. *Spectral analysis of large dimensional random matrices*, volume 20. Springer, 2010.

[BSL15] Joan Bruna, Pablo Sprechmann, and Yann LeCun. Super-resolution with deep convolutional sufficient statistics. *arXiv preprint arXiv:1511.05666*, 2015.

[CBG+16] Romain Couillet, Florent Benaych-Georges, et al. Kernel spectral clustering of large dimensional data. *Electronic Journal of Statistics*, 10(1):1393–1454, 2016.

[CDMF09] Mireille Capitaine, Catherine Donati-Martin, and Delphine Féral. The largest eigenvalues of finite rank deformation of large wigner matrices: convergence and nonuniversality of the fluctuations. *The Annals of Probability*, pages 1–47, 2009.

[CHM+15] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Artificial intelligence and statistics*, pages 192–204, 2015.

[CJ95] Jorge Cadima and Ian T Jolliffe. Loading and correlations in the interpretation of principle compenents. *Journal of Applied Statistics*, 22(2):203–214, 1995.

[CMK+14] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[CS13] Xiuyuan Cheng and Amit Singer. The spectrum of random inner-product kernel matrices. *Random Matrices: Theory and Applications*, 2(04):1350010, 2013.

[CSZ09] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[DB16] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems*, pages 658–666, 2016.

[DCF+15] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494, 2015.

[dCPS+18] Remi Tachet des Combes, Mohammad Pezeshki, Samira Shabanian, Aaron Courville, and Yoshua Bengio. On the learning dynamics of deep neural networks. *arXiv preprint arXiv:1809.06848*, 2018.

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[dGJL05] Alexandre d'Aspremont, Laurent E Ghaoui, Michael I Jordan, and Gert R Lanckriet. A direct formulation for sparse pca using semidefinite programming. In *Advances in neural information processing systems*, pages 41–48, 2005.

[DK70] Chandler Davis and William Morton Kahan. The rotation of eigenvectors by a perturbation. iii. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

[DLHT14] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision*, pages 184–199. Springer, 2014.

[DLHT16] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2016.

[DLT16] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *European Conference on Computer Vision*, pages 391–407. Springer, 2016.

[DM14] Yash Deshpande and Andrea Montanari. Sparse pca via covariance thresholding. In *Advances in Neural Information Processing Systems*, pages 334–342, 2014.

[DM16] Yash Deshpande and Andrea Montanari. Sparse pca via covariance thresholding. *J. Mach. Learn. Res.*, 17(1):4913–4953, January 2016.

[DTVG15] Dengxin Dai, Radu Timofte, and Luc Van Gool. Jointly optimized regressors for image super-resolution. In *Computer Graphics Forum*, volume 34, pages 95–104. Wiley Online Library, 2015.

[Duc79] Claude E Duchon. Lanczos filtering in one and two dimensions. *Journal of applied meteorology*, 18(8):1016–1022, 1979.

[DYP20] Shiyu Duan, Shujian Yu, and Jose Principe. Modularizing deep learning via pairwise learning with kernels. *arXiv preprint arXiv:2005.05541*, 2020.

[DZSW11] Weisheng Dong, Lei Zhang, Guangming Shi, and Xiaolin Wu. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE Transactions on Image Processing*, 20(7):1838–1857, 2011.

[EI98] Stanley C Eisenstat and Ilse CF Ipsen. Three absolute perturbation bounds for matrix eigenvalues imply relative bounds. *SIAM Journal on Matrix Analysis and Applications*, 20(1):149–158, 1998.

[EK08] Noureddine El Karoui. Operator norm consistent estimation of large-dimensional sparse covariance matrices. *The Annals of Statistics*, pages 2717–2756, 2008.

[EK10a] Noureddine El Karoui. On information plus noise kernel random matrices. *The Annals of Statistics*, 38(5):3191–3216, 2010.

[EK+10b] Noureddine El Karoui et al. The spectrum of kernel random matrices. *The Annals of Statistics*, 38(1):1–50, 2010.

[FGP07] B. Fleury, O. Guédon, and G. Paouris. A stability result for mean width of l p -centroid bodies. *Advances in Mathematics*, 214:865–877, 2007.

[Fin92] Helmut Finner. A generalization of holder's inequality and some probability inequalities. *The Annals of probability*, pages 1893–1901, 1992.

[FJP02] William T Freeman, Thouis R Jones, and Egon C Pasztor. Example-based super-resolution. *IEEE Computer graphics and Applications*, 22(2):56–65, 2002.

[FP07] Delphine Féral and Sandrine Péché. The largest eigenvalue of rank one deformation of large wigner matrices. *Communications in mathematical physics*, 272(1):185–228, 2007.

[Fre19] Daniel J. Fresen. A simplified proof of clt for convex bodies. *arXiv preprint arXiv:1907.06785*, 2019.

[FREM04] Sina Farsiu, M Dirk Robinson, Michael Elad, and Peyman Milanfar. Fast and robust multiframe super resolution. *IEEE transactions on image processing*, 13(10):1327–1344, 2004.

[GAA+17] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems 30*, pages 5767–5777. 2017.

[GCM18] Samantha Guerriero, Barbara Caputo, and Thomas Mensink. Deep nearest class mean classifiers. In *International Conference on Learning Representations, Worskhop Track*, 2018.

[GEB15] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 262–270, 2015.

[GEB16] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR).*, pages 2414–2423. IEEE, 2016.

[GL10] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 399–406. Omnipress, 2010.

[GP17] Bolin Gao and Lacra Pavel. On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805*, 2017.

[GPAM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

[GSBB11] Prateek Gupta, Priyanka Srivastava, Satyam Bhardwaj, and Vikrant Bhateja. A modified psnr metric based on hvs for quality assessment of color images. In *Communication and Industrial Application (ICCIA), 2011 International Conference on*, pages 1–4. IEEE, 2011.

[HLM15] Elad Hazan, Roi Livni, and Yishay Mansour. Classification with low rank and missing data. In *ICML*, pages 257–266, 2015.

[HLN+07] Walid Hachem, Philippe Loubaton, Jamal Najim, et al. Deterministic equivalents for certain functionals of large random matrices. *The Annals of Applied Probability*, 17(3):875–930, 2007.

[HMD15] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[HRU+17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017.

[HS11] He He and Wan-Chi Siu. Single image super-resolution using gaussian process regression. In *Computer Vision and Pattern Recognition (CVPR).*, pages 449–456. IEEE, 2011.

[HZS06] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006.

[IHM+16] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[IS15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[JAFF16] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.

[JL09] Iain M Johnstone and Arthur Yu Lu. On consistency and sparsity for principal components analysis in high dimensions. *Journal of the American Statistical Association*, 104(486):682–693, 2009.

[KC17] Abla J Kammoun and Romain Couillet. Subspace kernel clustering of large dimensional data. *(submitted to) Annals of Applied Probability*, 2017.

[KGC18] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

[KH10] Alex Krizhevsky and Geoff Hinton. Convolutional deep belief networks on cifar-10. *Unpublished manuscript*, 40(7), 2010.

[KK10] Kwang In Kim and Younghee Kwon. Single-image super-resolution using sparse regression and natural image prior. *IEEE transactions on pattern analysis and machine intelligence*, 32(6):1127–1133, 2010.

[Kla07] B. Klartag. A central limit theorem for convex sets. *Inventiones mathematicae volume*, 168:pages91–131, 2007.

[KNV15] Robert Krauthgamer, Boaz Nadler, and Dan Vilenchik. Do semidefinite relaxations solve sparse pca up to the information limit? *Ann. Statist.*, 43(3):1300–1322, 06 2015.

[KQD18] Alireza Karbalayghareh, Xiaoning Qian, and Edward R Dougherty. Optimal bayesian transfer learning. *IEEE Transactions on Signal Processing*, 2018.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[KXR+19] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition, 2019.

[KY13] Antti Knowles and Jun Yin. The isotropic semicircle law and deformation of wigner matrices. *Communications on Pure and Applied Mathematics*, 66(11):1663–1749, 2013.

[LC17] Zhenyu Liao and Romain Couillet. Random matrices meet machine learning: A large dimensional analysis of ls-svm. In *ICASSP*, pages 2397–2401. IEEE, 2017.

[LC18a] Zhenyu Liao and Romain Couillet. The dynamics of learning: A random matrix approach. *arXiv preprint arXiv:1805.11917*, 2018.

[LC18b] Cosme Louart and Romain Couillet. Concentration of measure and large random matrices with an application to sample covariance matrices. *arXiv preprint arXiv:1805.08295*, 2018.

[LC18c] Cosme Louart and Romain Couillet. A random matrix and concentration inequalities framework for neural networks analysis. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4214–4218. IEEE, 2018.

[LC19] Cosme Louart and Romain Couillet. Concentration of measure and large random matrices with an application to sample covariance matrices. *submitted*, 2019.

[LC20] Cosme Louart and Romain Couillet. Concentration of measure and large random matrices with an application to sample covariance matrices. *submitted to Random Matrices: Theory and Applications*, 2020.

[LeC98] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[Led05a] Michel Ledoux. *The concentration of measure phenomenon*. Number 89. American Mathematical Soc., 2005.

[Led05b] Michel Ledoux. *The concentration of measure phenomenon*. Number 89. American Mathematical Soc., 2005.

[Lia19] Zhenyu Liao. *A random matrix framework for large dimensional machine learning and neural networks*. Theses, Université Paris-Saclay, September 2019.

[LLP+19] Lauri Lovén, Teemu Leppänen, Ella Peltonen, Juha Partala, Erkki Harjula, Pawani Porambage, Mika Ylianttila, and Jukka Riekki. Edge ai: A vision for distributed, edge-native artificial intelligence in future 6g networks. *The 1st 6G Wireless Summit*, pages 1–2, 2019.

[LM07] Neil D Lawrence and Andrew J Moore. Hierarchical gaussian process latent variable models. In *Proceedings of the 24th international conference on Machine learning*, pages 481–488, 2007.

[LN08] Zinoviy Landsman and Johanna Nešlehová. Stein's lemma for elliptical random vectors. *Journal of Multivariate Analysis*, 99(5):912–927, 2008.

[LOV19] Sindy Löwe, Peter O'Connor, and Bastiaan Veeling. Putting an end to end-to-end: Gradient-isolated learning of representations. In *Advances in Neural Information Processing Systems*, pages 3039–3051, 2019.

[LTH+16] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint*, 2016.

[LW16] Chuan Li and Michael Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2479–2486, 2016.

[LWL+17] Xuezhi Liang, Xiaobo Wang, Zhen Lei, Shengcai Liao, and Stan Z Li. Soft-margin softmax for deep classification. In *International Conference on Neural Information Processing*, pages 413–421. Springer, 2017.

[M+13] Zongming Ma et al. Sparse principal component analysis and iterative thresholding. *The Annals of Statistics*, 41(2):772–801, 2013.

[Mai19] Xiaoyi Mai. *Methods of random matrices for large dimensional statistical learning*. PhD thesis, Université Paris-Saclay, 2019.

[MBM11] Gregoire Montavon, Mikio L Braun, and Klaus-Robert Miller. Kernel analysis of deep networks. *Journal of Machine Learning Research*, 12(Sep):2563–2581, 2011.

[MC17] Xiaoyi Mai and Romain Couillet. A random matrix analysis and improvement of semi-supervised learning for large dimensional data. *arXiv preprint arXiv:1711.03404*, 2017.

[MCL15] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[MFLG19] Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher. *arXiv preprint arXiv:1902.03393*, 2019.

[MKKY18] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[MLC19] Xiaoyi Mai, Zhenyu Liao, and Romain Couillet. A large scale analysis of logistic regression: Asymptotic performance and new insights. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3357–3361. IEEE, 2019.

[MLK20] Kurt Wan-Duo Ma, JP Lewis, and W Bastiaan Kleijn. The hsic bottleneck: Deep learning without back-propagation. In *AAAI*, pages 5085–5092, 2020.

[MMH+19] MG Murshed, Christopher Murphy, Daqing Hou, Nazar Khan, Ganesh Ananthanarayanan, and Faraz Hussain. Machine learning at the network edge: A survey. *arXiv preprint arXiv:1908.00080*, 2019.

[MO14] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[MP67] Vladimir A Marčenko and Leonid Andreevich Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457, 1967.

[MTS+19] Sara Meftah, Youssef Tamaazousti, Nasredine Semmar, Hassane Essafi, and Fatiha Sadat. Joint learning of pre-trained and random units for domain adaptation in part-of-speech tagging. *arXiv preprint arXiv:1904.03595*, 2019.

[MVPC13] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2624–2637, 2013.

[MWA06] Baback Moghaddam, Yair Weiss, and Shai Avidan. Spectral bounds for sparse pca: Exact and greedy algorithms. In *Advances in neural information processing systems*, pages 915–922, 2006.

[NBA+18] Roman Novak, Yasaman Bahri, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study. *arXiv preprint arXiv:1802.08760*, 2018.

[NJW02] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

[NM14] Kamal Nasrollahi and Thomas B Moeslund. Super-resolution: a comprehensive survey. *Machine vision and applications*, 25(6):1423–1468, 2014.

[OP16] Lukasz P Olech and Mariusz Paradowski. Hierarchical gaussian mixture model with objects attached to terminal and non-terminal dendrogram nodes. In *Proceedings of the 9th International Conference on Computer Recognition Systems CORES 2015*, pages 191–201. Springer, 2016.

[Pau07] Debashis Paul. Asymptotics of sample eigenstructure for a large dimensional spiked covariance model. *Statistica Sinica*, 17:1617–1642, 2007.

[PB17] Jeffrey Pennington and Yasaman Bahri. Geometry of neural network loss surfaces via random matrix theory. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2798–2806. JMLR. org, 2017.

[PDK13] Dimitris Papailiopoulos, Alexandros Dimakis, and Stavros Korokythakis. Sparse pca through low-rank approximations. In *International Conference on Machine Learning*, pages 747–755, 2013.

[PGM+19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

[PL20] Roman Pogodin and Peter E Latham. Kernelized information bottleneck leads to biologically plausible 3-factor hebbian learning in deep networks. *arXiv preprint arXiv:2006.07123*, 2020.

[PRU+18] Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Günter Klambauer. Fréchet chemblnet distance: A metric for generative models for molecules. *arXiv preprint arXiv:1803.09518*, 2018.

[PW17] Jeffrey Pennington and Pratik Worah. Nonlinear random matrix theory for deep learning. In *Advances in Neural Information Processing Systems*, pages 2637–2646, 2017.

[RHW+88] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[RLNH17] Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. In *Advances in Neural Information Processing Systems 30*, pages 2018–2028. 2017.

[RMC15] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[Sat17] Mahadev Satyanarayanan. The emergence of edge computing. *Computer*, 50(1):30–39, 2017.

[SB95] Jack W Silverstein and ZD Bai. On the empirical distribution of eigenvalues of a class of large dimensional random matrices. *Journal of Multivariate analysis*, 54(2):175–192, 1995.

[SB16] Suraj Srinivas and R Venkatesh Babu. Generalized dropout. *arXiv preprint arXiv:1611.06791*, 2016.

[SC95] Jack W Silverstein and Sang-Il Choi. Analysis of the limiting spectral distribution of large dimensional random matrices. *Journal of Multivariate Analysis*, 54(2):295–309, 1995.

[SCH+16] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1874–1883, 2016.

[SCT+16] Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár. Amortised map inference for image super-resolution. *arXiv preprint arXiv:1610.04490*, 2016.

[SEBT20] Mohamed El Amine Seddik, Hassan Essafi, Abdallah Benzine, and Mohamed Tamaazousti. Lightweight neural networks from pca lda based distilled dense neural networks. In *International Conference on Image Processing*, 2020.

[SH08] Haipeng Shen and Jianhua Z Huang. Sparse principal component analysis via regularized low rank matrix approximation. *Journal of multivariate analysis*, 99(6):1015–1034, 2008.

[SHK+14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[SIVA17] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[SLB15] Samuel Schulter, Christian Leistner, and Horst Bischof. Fast and accurate image upscaling with super-resolution forests. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3791–3799, 2015.

[SLCT20] Mohamed El Amine Seddik, Cosme Louart, Romain Couillet, and Mohamed Tamaazousti. The unexpected deterministic and universal behavior of large softmax classifiers. 2020.

[SLTC20] Mohamed El Amine Seddik, Cosme Louart, Mohamed Tamaazousti, and Romain Couillet. Random matrix theory proves that deep learning representations of gan-data behave as gaussian mixtures. In *International Conference on Machine Learning*, 2020.

[SMG13] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[SMH11] Ilya Sutskever, James Martens, and Geoffrey E Hinton. Generating text with recurrent neural networks. In *ICML*, 2011.

[SPP15] Jordi Salvador and Eduardo Perez-Pellitero. Naive bayes super-resolution forest. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 325–333, 2015.

[ŚST+18] Marek Śmieja, Lukasz Struski, Jacek Tabor, Bartosz Zieliński, and Przemysław Spurek. Processing of missing data by neural networks. In *Advances in Neural Information Processing Systems*, pages 2719–2729, 2018.

[STC19a] Mohamed El Amine Seddik, Mohamed Tamaazousti, and Romain Couillet. Kernel random matrices of large concentrated data: The example of gan-generated images. In *International Conference on Acoustics, Speech, and Signal Processing*, 2019.

[STC19b] Mohamed El Amine Seddik, Mohamed Tamaazousti, and Romain Couillet. A kernel random matrix-based approach for sparse PCA. In *International Conference on Learning Representations*, 2019.

[STC19c] Mohamed El Amine Seddik, Mohamed Tamaazousti, and Romain Couillet. Pourquoi les matrices aléatoires expliquent l'apprentissage ? un argument d'universalité offert par les gans. In *Colloque francophone de traitement du signal et des images*, 2019.

[STL19] Mohamed El Amine Seddik, Mohamed Tamaazousti, and John Lin. Generative collaborative networks for single image super-resolution. *arXiv:1902.10467*, 2019.

[SV99] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[SZ14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[SZS+13] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[SZT17] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

[TAKC18] Hafis Tiomoko Ali, Abla Kammoun, and Romain Couillet. Random matrix asymptotics of inner product spectral clustering. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[Tao12] Terence Tao. *Topics in random matrix theory*, volume 132. American Mathematical Soc., 2012.

[TDSVG14] Radu Timofte, Vincent De Smet, and Luc Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *Asian Conference on Computer Vision*, pages 111–126. Springer, 2014.

[TDVG13] Radu Timofte, Vincent De, and Luc Van Gool. Anchored neighborhood regression for fast example-based super-resolution. In *Computer Vision (ICCV).*, pages 1920–1927. IEEE, 2013.

[TGIH17] Alaa Tharwat, Tarek Gaber, Abdelhameed Ibrahim, and Aboul Ella Hassanien. Linear discriminant analysis: A detailed tutorial. *AI Commun.*, 30:169–190, 2017.

[TLBH17a] Youssef Tamaazousti, Hervé Le Borgne, and Céline Hudelot. Mucale-net: Multi categorical-level networks to generate more discriminating features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6711–6720, 2017.

[TLBH+17b] Youssef Tamaazousti, Hervé Le Borgne, Céline Hudelot, Mohamed El Amine Seddik, and Mohamed Tamaazousti. Learning more universal representations for transfer-learning. *arXiv:1712.09708*, 2017.

[TLBH+19] Youssef Tamaazousti, Hervé Le Borgne, Céline Hudelot, Mohamed El Amine Seddik, and Mohamed Tamaazousti. Learning more universal representations for transfer-learning. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[VLBB08] Ulrike Von Luxburg, Mikhail Belkin, and Olivier Bousquet. Consistency of spectral clustering. *The Annals of Statistics*, pages 555–586, 2008.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[VTWA18] Sahar Voghoei, Navid Hashemi Tonekaboni, Jason G Wallace, and Hamid R Arabnia. Deep learning at the edge. In *2018 International Conference on*

*Computational Science and Computational Intelligence (CSCI)*, pages 895–901. IEEE, 2018.

[WBSS04] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[WEG87] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[WGR+09] John Wright, Arvind Ganesh, Shankar Rao, Yigang Peng, and Yi Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in neural information processing systems*, pages 2080–2088, 2009.

[WLY+15] Zhaowen Wang, Ding Liu, Jianchao Yang, Wei Han, and Thomas Huang. Deep networks for image super-resolution with sparse prior. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 370–378, 2015.

[WSB03] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on*, volume 2, pages 1398–1402. Ieee, 2003.

[WWL13] Stefan Wager, Sida Wang, and Percy S Liang. Dropout training as adaptive regularization. In *Advances in neural information processing systems*, pages 351–359, 2013.

[XRV17] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[YC08] Kai Yu and Wei Chu. Gaussian process models for link analysis and transfer learning. In *Advances in Neural Information Processing Systems*, pages 1657–1664, 2008.

[YCC98] LeCun Yann, Cortes Corinna, and J Christopher. The mnist database of handwritten digits. *URL http://yhann. lecun. com/exdb/mnist*, 1998.

[YCL+16] Raymond Yeh, Chen Chen, Teck Yian Lim, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with perceptual and contextual losses. *arXiv preprint arXiv:1607.07539*, 2016.

[YD16] Daniel LK Yamins and James J DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, 19(3):356, 2016.

[YJBK17] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4133–4141, 2017.

[YKYR18] Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K Ravikumar. Representer point selection for explaining deep neural networks. In *Advances in Neural Information Processing Systems*, pages 9291–9301, 2018.

[YLK+19] Joonyoung Yi, Juhyuk Lee, Kwang Joon Kim, Sung Ju Hwang, and Eunho Yang. Why not to use zero imputation? correcting sparsity bias in training neural networks. In *International Conference on Learning Representations*, 2019.

[YMY14] Chih-Yuan Yang, Chao Ma, and Ming-Hsuan Yang. Single-image super-resolution: A benchmark. In *European Conference on Computer Vision*, pages 372–386. Springer, 2014.

[YP16] Xin Yu and Fatih Porikli. Ultra-resolving face images by discriminative generative networks. In *European Conference on Computer Vision*, pages 318–333. Springer, 2016.

[YW19] Yaqiong Yao and HaiYing Wang. Optimal subsampling for softmax regression. *Statistical Papers*, 60(2):235–249, 2019.

[YYDN07] Qingxiong Yang, Ruigang Yang, James Davis, and David Nistér. Spatial-depth super resolution for range images. In *Computer Vision and Pattern Recognition, CVPR.*, pages 1–8. IEEE, 2007.

[YZ13] Xiao-Tong Yuan and Tong Zhang. Truncated power method for sparse eigenvalue problems. *Journal of Machine Learning Research*, 14(Apr):899–925, 2013.

[ZEP10] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pages 711–730. Springer, 2010.

[ZHT06] Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006.

[ZIE+18] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *arXiv preprint*, 2018.

[ZS07] Ron Zass and Amnon Shashua. Nonnegative sparse pca. In *Advances in Neural Information Processing Systems*, pages 1561–1568, 2007.

[ZY12] Wilman WW Zou and Pong C Yuen. Very low resolution face recognition problem. *IEEE Transactions on Image Processing*, 21(1):327–340, 2012.