Random Matrix Theory Proves that Deep Learning Representations of GAN-data Behave as Gaussian Mixtures

Mohamed El Amine Seddik, Cosme Louart, Mohamed Tamaazousti, Romain Couillet

Random Matrix Theory Proves that Deep Learning Representations of GAN-data Behave as Gaussian Mixtures

Mohamed El Amine Seddik [1 2] **Cosme Louart** [1 3] **Mohamed Tamaazousti** [1] **Romain Couillet** [2 3]

Abstract

This paper shows that deep learning (DL) representations of data produced by generative adversarial nets (GANs) are random vectors which fall within the class of so-called *concentrated* random vectors. Further exploiting the fact that Gram matrices, of the type $\boldsymbol{G} = \boldsymbol{X}^\intercal \boldsymbol{X}$ with $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{p \times n}$ and $\boldsymbol{x}_i$ independent concentrated random vectors from a mixture model, behave asymptotically (as $n, p \to \infty$) as if the $\boldsymbol{x}_i$ were drawn from a Gaussian mixture, suggests that DL representations of GAN-data can be fully described by their first two statistical moments for a wide range of standard classifiers. Our theoretical findings are validated by generating images with the BigGAN model and across different popular deep representation networks.

1. Introduction

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. Deep neural networks (Rumelhart et al., 1988) 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 (Bengio et al., 2009).

Specifically, quantifying what makes a given deep learning representation better than another is a fundamental question in the field of *Representation Learning* (Bengio et al., 2013). Relying on (Montavon et al., 2011) 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. (Montavon et al., 2011) 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 ℓ 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 predictor 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 article at theoretically studying 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 (Yeh et al., 2018)– 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. Yet, due to recent advances in generative models since the advent of Generative Adversarial Nets (Goodfellow et al., 2014), 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 (Antoniou et al., 2017). On the other hand, the fundamental concentration of measure phenomenon (Ledoux, 2005) tells us that Lipschitzally 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 vec-

[1]CEA List, Palaiseau, France [2]Centralesupélec, Gif-sur-Yvette, France [3]University Grenoble-Alpes, Grenoble, France. Correspondence to: MEA.Seddik <mohamedelamine.seddik@cea.fr>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 108, 2020.

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

tors and thus an appropriate statistical model of realistic data.

Targeting classification applications by assuming a mixture of concentrated random vectors model, this article studies 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 (Brock et al., 2018) 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 (Deng et al., 2009).

The rest of the paper is organized as follows. In Section 2, we introduce the notion of concentrated vectors and their main properties. Our main theoretical results are then provided in Section 3. In Section 4 we present experimental results. Section 5 concludes the article.

Notation: In the following, we use the notation from (Goodfellow et al., 2016). $[n]$ denotes the set $\{1, \dots, n\}$. Given a vector $\boldsymbol{x} \in \mathbb{R}^n$, the ℓ_2-norm of $\boldsymbol{x}$ is given as $\|\boldsymbol{x}\|^2 = \sum_{i=1}^n \boldsymbol{x}_i^2$. Given a $p \times n$ matrix $\boldsymbol{M}$, its Frobenius norm is defined as $\|\boldsymbol{M}\|_F^2 = \sum_{i=1}^p \sum_{j=1}^n \boldsymbol{M}_{ij}^2$ and its spectral norm as $\|\boldsymbol{M}\| = \sup_{\|\boldsymbol{x}\|=1} \|\boldsymbol{M}\boldsymbol{x}\|$. $\odot$ for the Hadamard product. An application $\mathcal{F} : E \to F$ is said to be $\|\mathcal{F}\|_{lip}$-Lipschitz, if $\forall (\boldsymbol{x}, \boldsymbol{y}) \in E^2$, $\|\mathcal{F}(\boldsymbol{x}) - \mathcal{F}(\boldsymbol{y})\|_F \leq \|\mathcal{F}\|_{lip} \cdot \|\boldsymbol{x} - \boldsymbol{y}\|_E$ and $\|\mathcal{F}\|_{lip}$ is finite.

2. Basic notions of concentrated vectors

Being the central tool of our study, we start by introducing the notion of concentrated vectors. While advanced concentration notions have been recently developed in (Louart & Couillet, 2019) in order to specifically analyze the behavior of large dimensional sample covariance matrices, for simplicity, we restrict ourselves here to the sufficient so-called q-exponentially concentrated random vectors.

Definition 2.1 (q-exponential concentration)**.** *Given a normed space $(E, \|\cdot\|_E)$ and a real q, a random vector $\boldsymbol{x} \in E$ is said to be q-exponentially concentrated if for any 1-Lipschitz real function $f : E \to \mathbb{R}$, there exists $C \geq 0$ independent of $\dim(E)$ and $\sigma > 0$ such that for all $t \geq 0$*

$$\mathbb{P}\{|f(\boldsymbol{x}) - \mathbb{E}f(\boldsymbol{x})| > t\} \leq C\, e^{-(t/\sigma)^q} \tag{1}$$

which we denote $\boldsymbol{x} \in \mathcal{E}_q(\sigma \,|\, E, \|\cdot\|_E)$. We simply write $\boldsymbol{x} \in \mathcal{E}_q(1 \,|\, E, \|\cdot\|_E)$ if the tail parameter σ does not depend on $\dim(E)$, and $x \in \mathcal{E}_q(1)$ for x a scalar real random variable.

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. See (Ledoux, 2005)) for more examples such as uniform and Gamma distribution.

Proposition 2.2 (Gaussian vectors (Ledoux, 2005))**.** *Let $d \in \mathbb{N}$ and $\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{I}_d)$. Then $\boldsymbol{x}$ is a 2-exponentially concentrated vector independently on the dimension d, i.e. $\boldsymbol{x} \in \mathcal{E}_2(1 \,|\, \mathbb{R}^d, \|\cdot\|)$.*

Concentrated vectors have the interesting property of being stable by application of $\mathbb{R}^d \to \mathbb{R}^p$ vector-Lipschitz transformations. Indeed, Lipschitz-ally transformed concentrated vectors remain concentrated according to the following proposition.

Proposition 2.3 (Lipschitz stability)**.** *Let $\boldsymbol{x} \in \mathcal{E}_q(1 \,|\, E, \|\cdot\|_E)$ and $\mathcal{G} : E \to F$ a Lipschitz application with Lipschitz constant $\|\mathcal{G}\|_{lip}$ which may depend on $\dim(F)$. Then the concentration property on $\boldsymbol{x}$ is transferred to $\mathcal{G}(\boldsymbol{x})$, if $\boldsymbol{x} \in \mathcal{E}_q(1 \,|\, E, \|\cdot\|_E)$ then*

$$\mathcal{G}(\boldsymbol{x}) \in \mathcal{E}_q(\|\mathcal{G}\|_{lip} \,|\, F, \|\cdot\|_F). \tag{2}$$

Note importantly for the following that the Lipschitz constant of the transformation $\mathcal{G}$ must be controlled, in order to constrain the tail parameter of the obtained concentration.

In particular, we have the coming corollary to Proposition 2.3 of central importance in the following.

Corollary 2.4. *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,*

$$\begin{aligned} &\boldsymbol{Z} \in \mathcal{E}_q(1 \,|\, \mathbb{R}^{d \times n}, \|\cdot\|_F) \\ &\quad \Rightarrow \;\; \mathcal{G}(\boldsymbol{Z}) \in \mathcal{E}_q\left(\sup_i \|\mathcal{G}_i\|_{lip} \;\middle|\; \mathbb{R}^{p \times n}, \|\cdot\|_F\right). \end{aligned} \tag{3}$$

Proof. This is a consequence of Proposition 2.3 since the map $\mathcal{G}$ is $\sup_i \|\mathcal{G}_i\|_{lip}$-Lipschitz with respect to (w.r.t.) the Frobenius norm. Indeed, for $\boldsymbol{X}, \boldsymbol{H} \in \mathbb{R}^{d \times n}$: $\|\mathcal{G}(\boldsymbol{X} + \boldsymbol{H}) - \mathcal{G}(\boldsymbol{X})\|_F^2 \leq \sum_{i=1}^n \|\mathcal{G}_i\|_{lip}^2 \cdot \|\boldsymbol{H}_{:,i}\|^2 \leq \sup_i \|\mathcal{G}_i\|_{lip}^2 \cdot \|\boldsymbol{H}\|_F^2$. □

3. Main Results

3.1. GAN data: An Example of Concentrated Vectors

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 the ability nowadays to generate random *realistic* data (for instance realistic images) by applying successive Lipschitz operations to standard Gaussian vectors (Goodfellow et al., 2014).

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) or fake (for the generated data). The discriminator is updated directly through a binary classification problem, whereas the generator is updated through the discriminator. 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 (Goodfellow et al., 2014).

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:

$$\text{Real Data} \approx \text{GAN Data} = \mathcal{F}_N \circ \cdots \circ \mathcal{F}_1(\boldsymbol{z}), \qquad (4)$$

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,

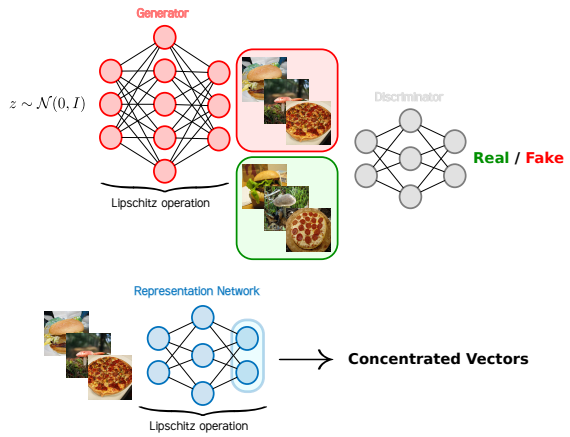

Figure 1. Deep learning representations of GAN-data are constructed by applying successive Lipschitz operations to Gaussian vectors, therefore they are *concentrated* vectors by design, since Gaussian vectors are concentrated and thanks to the Lipschitz stability in Proposition 2.3.

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

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

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 (Ioffe & Szegedy, 2015) the vectors of a small batch $\mathcal{B} = \{\boldsymbol{x}_i\}_{i=1}^b \subset \mathbb{R}^d$ as follows: for each $\boldsymbol{x}_k \in \mathcal{B}$

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

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 $\mathbf{diag}(\boldsymbol{v})$ 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 1, since standard Gaussian vectors are concentrated vectors as mentioned in Proposition 2.2 and since the notion of concentrated vectors is stable by Lipschitz transformations thanks to Proposition 2.3, GAN-data (and their DL representations) are concentrated vectors by design given the construction in Equation (4).

Moreover, in order to generate data belonging to a specific class, Conditional GANs have been introduced (Mirza & Osindero, 2014); once again data generated by these models are concentrated vectors as a consequence of Corollary 2.4. 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 (Brock et al., 2018)).

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 (Roth et al., 2017; Brock et al., 2018). The control of the Lipschitz constant of representation networks is also needed in practice in order to make them robust against adversarial examples (Szegedy et al., 2013; Gulrajani et al., 2017). This control is particularly ensured through spectral normalization of the affine layers (Brock et al., 2018), such as Fully Connected Layers, Convolutional Layers and Batch Normalization. Indeed, spectral normalization (Miyato et al., 2018) 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-propagation algorithm, where $\sigma_1(\boldsymbol{W})$ stands for the largest singular value of the weight matrix $\boldsymbol{W}$. (Brock et al., 2018), 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 $\boldsymbol{W} \leftarrow \boldsymbol{W}/\sigma_1(\boldsymbol{W})$ (Miyato et al., 2018)– to the affine layers:

$$\boldsymbol{W} \leftarrow \boldsymbol{W} - (\sigma_1(\boldsymbol{W}) - \sigma_*)\, \boldsymbol{u}_1(\boldsymbol{W})\boldsymbol{v}_1(\boldsymbol{W})^\intercal \qquad (5)$$

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 σ_* 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 (5). 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 (Antognini & Sohl-Dickstein, 2018).

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 η:*

$$\begin{aligned} &\boldsymbol{W} \leftarrow \boldsymbol{W} - \eta \boldsymbol{E}, \text{ with } \boldsymbol{E}_{i,j} \sim \mathcal{N}(0,1) \\ &\boldsymbol{W} \leftarrow \boldsymbol{W} - \max(0, \sigma_1(\boldsymbol{W}) - \sigma_*)\, \boldsymbol{u}_1(\boldsymbol{W})\boldsymbol{v}_1(\boldsymbol{W})^\intercal. \end{aligned} \qquad (6)$$

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

$$\|\mathcal{G}\|_{lip} \leq \prod_{i=1}^{N} \left(\varepsilon + \sqrt{\sigma_*^2 + \eta^2 d_i d_{i-1}}\right). \qquad (7)$$

Proof. The proof is provided in the supplementary material. □

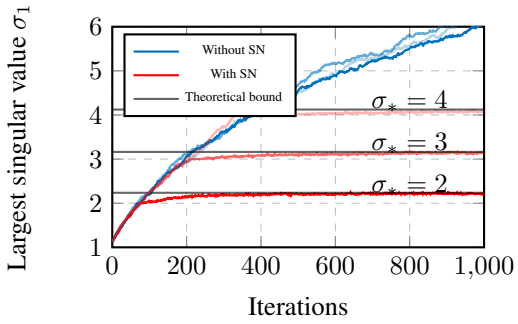

Figure 2. Behavior of the largest singular value of a weight matrix in terms of the iterations of a random walk (see proposition 3.1), without spectral normalization in (blue) and with spectral normalization in (red). The **(black)** lines correspond to the theoretical bound $\sqrt{\sigma_*^2 + \eta^2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

Proposition 3.1 shows that the Lipschitz constant of a neural network is controlled when trained with the spectral normalization in Equation (5). 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 σ_* is constant. Therefore, with this spectral normalization, it can be assumed that $\|\mathcal{G}\|_{lip} = \mathcal{O}(1)$ when dimensions grow. Figure 2 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. 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 $\mu_1, \ldots, \mu_k$ with distinct means $\{\boldsymbol{m}_\ell\}_{\ell=1}^k$ and "covariances" $\{\boldsymbol{C}_\ell\}_{\ell=1}^k$ defined respectively as

$$\boldsymbol{m}_\ell = \mathbb{E}_{\boldsymbol{x}_i \sim \mu_\ell}[\boldsymbol{x}_i], \ \boldsymbol{C}_\ell = \mathbb{E}_{\boldsymbol{x}_i \sim \mu_\ell}[\boldsymbol{x}_i \boldsymbol{x}_i^\intercal]. \tag{8}$$

For some $q > 0$, we consider a q-exponential concentration property on the laws μ_ℓ, in the sense that for any family of independent vectors $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_s$ sampled from μ_ℓ, $[\boldsymbol{y}_1, \ldots, \boldsymbol{y}_s] \in \mathcal{E}_q(1 \,|\, \mathbb{R}^{p \times s}, \|\cdot\|_F)$. 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 $\ell \in [k]$, $\boldsymbol{x}_{1+\sum_{j=1}^{\ell-1} n_j}, \ldots, \boldsymbol{x}_{\sum_{j=1}^{\ell} n_j} \sim \mu_\ell$, where n_ℓ stands for the number of $\boldsymbol{x}_i$'s sampled from μ_ℓ. In particular, we have the concentration of $\boldsymbol{X}$ as

$$\boldsymbol{X} \in \mathcal{E}_q(1 \,|\, \mathbb{R}^{p \times n}, \|\cdot\|_F). \tag{9}$$

Such a data matrix $\boldsymbol{X}$ can be constructed through Lipschitzally transformed Gaussian vectors ($q = 2$), with controlled Lipschitz constant, thanks to Corollary 2.4. 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.

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

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

3.3. Gram Matrices 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 (AK et al., 2002), Semi-supervised Learning (Chapelle et al., 2009) and Spectral Clustering (Ng et al., 2002). Indeed, the performance evaluation of these methods has already been studied under GMM models in (Liao & Couillet, 2017; Mai & Couillet, 2017; Couillet & Benaych-Georges, 2016) through RMT. On the other hand, analyzing the spectral behavior of $\boldsymbol{G}$ for DL representations quantifies their quality –through its principal subspace (Montavon et al., 2011)– as we have discussed in the introduction. In particular, the Gram matrix decomposes as

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

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 (Benaych-Georges & Couillet, 2016)– the existence of a threshold ξ 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).

In order to characterize the spectral behavior (*i.e.*, eigenvalues and leading eigenvectors) of $\boldsymbol{G}$ under the concentration assumption in Equation (9) on $\boldsymbol{X}$, we will be interested in determining the spectral distribution $L = \frac{1}{n}\sum_{i=1}^n \delta_{\lambda_i}$ of $\boldsymbol{G}$, with $\lambda_1, \ldots, \lambda_n$ the eigenvalues of $\boldsymbol{G}$, where δ_x stands for the Dirac measure at point x. Essentially, to determine the limiting eigenvalue distribution as $p, n \to \infty$ and $p/n \to c \in (0, \infty)$, a conventional approach in RMT consists in determining an estimate of the *Stieltjes transform* (Silverstein & Choi, 1995) m_L of L, which is defined for some $z \in \mathbb{C} \setminus \mathrm{Supp}(L)$

$$m_L(z) = \int_\lambda \frac{dL(\lambda)}{\lambda - z} = \frac{1}{n}\operatorname{tr}\left((\boldsymbol{G} - z\boldsymbol{I}_n)^{-1}\right). \tag{10}$$

Hence, quantifying the behavior of the *resolvent* of $\boldsymbol{G}$ defined as $\boldsymbol{R}(z) = (\boldsymbol{G} + z\boldsymbol{I}_n)^{-1}$ determines the limiting measure of L through $m_L(z)$. Furthermore, since $\boldsymbol{R}(z)$ and $\boldsymbol{G}$ share the same eigenvectors with associated eigenvalues $\frac{1}{\lambda_i + z}$, 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 γ is an oriented complex contour surrounding the top m eigenvalues of $\boldsymbol{G}$.

To study the behavior of $\boldsymbol{R}(z)$, we look for a so-called *deterministic equivalent* (Hachem et al., 2007) $\tilde{\boldsymbol{R}}(z)$ for $\boldsymbol{R}(z)$, which is a deterministic matrix that satisfies for all $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ and all $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$ of respectively bounded spectral and Eucildean norms, $\frac{1}{n}\operatorname{tr}(\boldsymbol{A}\boldsymbol{R}(z)) - \frac{1}{n}\operatorname{tr}(\boldsymbol{A}\tilde{\boldsymbol{R}}(z)) \to 0$ and $\boldsymbol{u}^\intercal(\boldsymbol{R}(z) - \tilde{\boldsymbol{R}}(z))\boldsymbol{v} \to 0$ almost surely as $n \to \infty$. In the following, we present our main result which gives such a deterministic equivalent under the concentration assumption on $\boldsymbol{X}$ in Equation (9) and under the following assumptions.

Assumption 3.2. *As $p \to \infty$,*

1. *$p/n \to c \in (0, \infty)$,*
2. *The number of classes k is bounded,*
3. *$\|\boldsymbol{m}_\ell\| = \mathcal{O}(\sqrt{p})$.*

Theorem 3.3 (Deterministic Equivalent for $\boldsymbol{R}(z)$)**.** *Under the model described in Section 3.2 and Assumptions 3.2, we*

have $\boldsymbol{R}(z) \in \mathcal{E}_q(p^{-1/2} \,|\, \mathbb{R}^{n\times n}, \|\cdot\|_F)$. *Furthermore,*

$$\left\| \mathbb{E}\boldsymbol{R}(z) - \tilde{\boldsymbol{R}}(z) \right\| = \mathcal{O}\left(\sqrt{\frac{\log(p)}{p}} \right)$$
$$\tilde{\boldsymbol{R}}(z) = \frac{1}{z} \operatorname{diag}\left\{ \frac{\boldsymbol{I}_{n_\ell}}{1+\delta_\ell^*(z)} \right\}_{\ell=1}^{k} + \frac{1}{p\, z} \boldsymbol{J} \boldsymbol{\Omega}_z \boldsymbol{J}^\intercal \tag{11}$$

with $\boldsymbol{\Omega}_z = \boldsymbol{M}^\intercal \tilde{\boldsymbol{Q}}(z) \boldsymbol{M} \odot \operatorname{diag}\left\{ \frac{\delta_\ell^*(z)-1}{\delta_\ell^*(z)+1} \right\}_{\ell=1}^{k}$ *and* $\tilde{\boldsymbol{Q}}(z) = \left(\frac{1}{c\,k} \sum_{\ell=1}^{k} \frac{\boldsymbol{C}_\ell}{1+\delta_\ell^*(z)} + z\boldsymbol{I}_p \right)^{-1}$,

where $\delta^*(z) = [\delta_1^*(z), \ldots, \delta_k^*(z)]^\intercal$ *is the unique fixed point of the system of equations for each* $\ell \in [k]$

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

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 2.3 that $\boldsymbol{R}(z) \in \mathcal{E}_q(p^{-1/2} \,|\, \mathbb{R}^{n\times n}, \|\cdot\|_F)$. 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 Lemmas 1.1 and 1.2 (see supplementary material) and finally exploiting the result of (Louart & Couillet, 2019) which shows that $\mathbb{E}\boldsymbol{Q}(z)$ can be estimated through $\tilde{\boldsymbol{Q}}(z)$, we obtain the estimator $\tilde{\boldsymbol{R}}(z)$ for $\mathbb{E}\boldsymbol{R}(z)$. A more detailed proof is provided in the supplementary material. □

This result allows specifically to (i) describe the limiting eigenvalues distribution of $\boldsymbol{G}$, (ii) determine the spectral detectability threshold mentioned above, (iii) evaluate the asymptotic "content" of the leading eigenvectors of $\boldsymbol{G}$ and, much more fundamentally, (iv) infer the asymptotic performances of machine learning algorithms that are based on simple functionals of $\boldsymbol{G}$ (*e.g.*, LS-SVM, spectral clustering etc.). Looking carefully at Theorem 3.3 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 μ_ℓ (their means $\boldsymbol{m}_\ell$ and "covariances" $\boldsymbol{C}_\ell$). 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* 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.

Figure 3. **(Top)** GAN generated images using the BigGAN model (Brock et al., 2018). **(Bottom)** Real images selected from the Imagenet dataset (Deng et al., 2009). We considered $n = 1500$ images from $k = 3$ classes which are {mushroom, pizza, hamburger}.

4. Application to CNN Representations of GAN-generated Images

This section presents experiments that confirm the result of Theorem 3.3. 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 (Montavon et al., 2011). In the following, all representation networks are standard convolutional neural networks pre-trained on the Imagenet dataset (Deng et al., 2009), in particular, we used pre-trained models of the Pytorch deep learning framework.

4.1. Spectrum and Dominant Eigenspace of the Gram Matrix

In this section, we consider $n = 1500$ data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^p$ as CNN representations –across popular CNN architectures of different sizes p– of GAN-generated images using the generator of the Big-GAN model (Brock et al., 2018). We further use real images from the Imagenet dataset (Deng et al., 2009) for comparison. In particular, we empirically compare the spectrum of the Gram matrix of this data with the 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

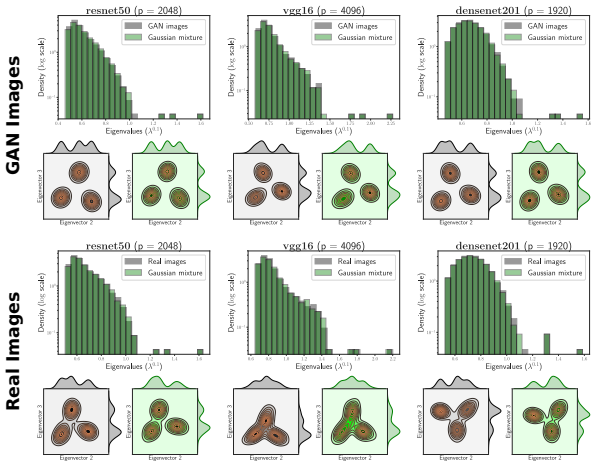

Figure 4. **(Top)** Spectrum and leading eigenspace of the Gram matrix for CNN representations of GAN generated images using the BigGAN model (Brock et al., 2018). **(Bottom)** Spectrum and leading eigenspace of the Gram matrix for CNN representations of real images selected from the Imagenet dataset (Deng et al., 2009). 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.

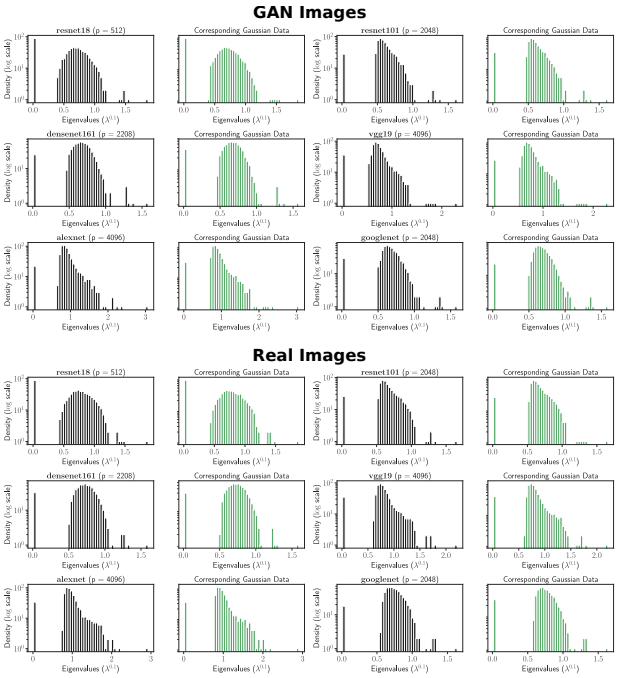

Figure 5. Spectrum of the Gram matrix for CNN representations in **(black)** and the corresponding Gaussian data (in green) for GAN generated images using the BigGAN model (Brock et al., 2018) **(Top)** and for real images randomly selected from the Imagenet dataset (Deng et al., 2009) **(Bottom)**. 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).

information as detailed in the previous section. Figure 3 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 3.3.

Figure 4 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 3.3 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 5 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 3.3. 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 (Montavon et al., 2011), 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 3.3, 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 γ 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.

4.2. Linear SVM Performance on the Dominant k-dimensional Eigenspace of $\boldsymbol{G}$

In this section, 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

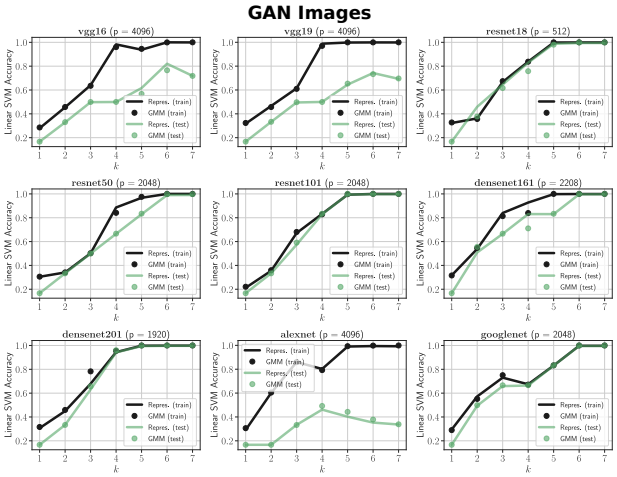

Figure 6. **Train** and test accuracy of a linear SVM model trained on the top k eigenvectors of the Gram matrix, computed on the representations of GAN generated images. We generated with the BigGAN model (Brock et al., 2018) $n = 6000$ images belonging to the 6 classes {hamburger, mushroom, pizza, 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.

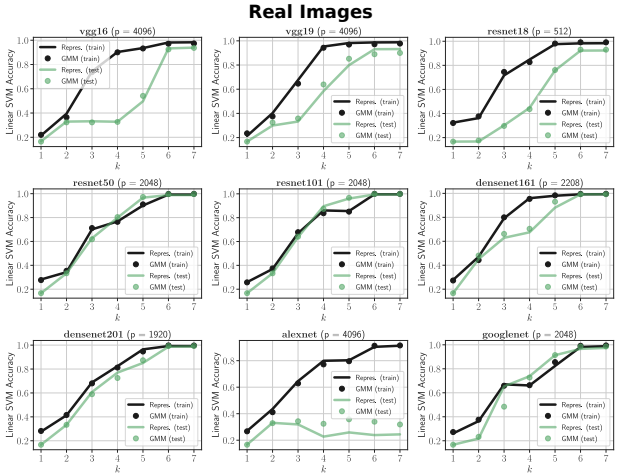

Figure 7. **Train** and test accuracy of a linear SVM model trained on the top k eigenvectors of the Gram matrix, computed on the representations of Real images. We randomly sampled $n = 6000$ images belonging to the 6 classes {hamburger, mushroom, pizza, strawberry, coffee, daisy} (1000 images per class) of the Imagenet dataset (Deng et al., 2009). 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.

order moments. Experiments were made in the following settings:

- Data types: We do the experiments for both GAN generated images using the BigGAN model (Brock et al., 2018) and for real images randomly selected for the Imagenet dataset (Deng et al., 2009). 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 (Deng et al., 2009) 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 6 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 3.3. This matching seems to extend beyond GAN images (which are concentrated vectors) to real images as depicted in Figure 7. As a consequence, our results suggest that the *quality* of a given representation network can be quantified through their first two statistical moments.

5. Conclusion

Leveraging on random matrix theory (RMT) and the concentration of measure phenomenon, we have shown through this paper that DL representations of GAN-data behave as Gaussian mixtures for linear classifiers, a fundamental *universal* property which is only valid in high-dimension of data. To the best of our knowledge, this result constitutes a new approach towards the theoretical understanding of complex objects such as DL representations, as well as the understanding of the behavior of more elaborate machine learning algorithms for complex data structures. In addition, the article explicitly demonstrated our ability, through RMT, to anticipate the behavior of a certain range of widely used standard classifiers for data as complex as DL representations of the realistic and surprising images generated by GANs. This opens the way to a more systematic analysis and improvement of machine learning algorithms on real datasets by means of large dimensional statistics.

References

AK, S. J. et al. *Least squares support vector machines*. World Scientific, 2002.

Antognini, J. and Sohl-Dickstein, J. Pca of high dimensional random walks with comparison to neural network training. In *Advances in Neural Information Processing Systems*, pp. 10307–10316, 2018.

Antoniou, A., Storkey, A., and Edwards, H. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

Benaych-Georges, F. and Couillet, R. Spectral analysis of the gram matrix of mixture models. *ESAIM: Probability and Statistics*, 20:217–237, 2016.

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, Aug 2013. ISSN 0162-8828. doi: 10.1109/TPAMI.2013.50.

Bengio, Y. et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

Brock, A., Donahue, J., and Simonyan, K. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Chapelle, O., Scholkopf, B., and Zien, A. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

Couillet, R. and Benaych-Georges, F. Kernel spectral clustering of large dimensional data. *Electronic Journal of Statistics*, 10(1):1393–1454, 2016.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *NIPS*, pp. 2672–2680, 2014.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. *Deep learning*, volume 1. MIT Press, 2016.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems 30*, pp. 5767–5777. 2017.

Hachem, W., Loubaton, P., Najim, J., et al. Deterministic equivalents for certain functionals of large random matrices. *The Annals of Applied Probability*, 17(3):875–930, 2007.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Ledoux, M. *The concentration of measure phenomenon*. Number 89. American Mathematical Soc., 2005.

Liao, Z. and Couillet, R. Random matrices meet machine learning: A large dimensional analysis of ls-svm. In *ICASSP*, pp. 2397–2401. IEEE, 2017.

Louart, C. and Couillet, R. Concentration of measure and large random matrices with an application to sample covariance matrices. *submitted*, 2019.

Mai, X. and Couillet, R. A random matrix analysis and improvement of semi-supervised learning for large dimensional data. *arXiv preprint arXiv:1711.03404*, 2017.

Mirza, M. and Osindero, S. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Montavon, G., Braun, M. L., and Miller, K.-R. Kernel analysis of deep networks. *Journal of Machine Learning Research*, 12(Sep):2563–2581, 2011.

Ng, A. Y., Jordan, M. I., and Weiss, Y. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pp. 849–856, 2002.

Roth, K., Lucchi, A., Nowozin, S., and Hofmann, T. Stabilizing training of generative adversarial networks through regularization. In *Advances in Neural Information Processing Systems 30*, pp. 2018–2028. 2017.

Rumelhart, D. E., Hinton, G. E., Williams, R. J., et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

Silverstein, J. W. and Choi, S.-I. Analysis of the limiting spectral distribution of large dimensional random matrices. *Journal of Multivariate Analysis*, 54(2):295–309, 1995.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Yeh, C.-K., Kim, J., Yen, I. E.-H., and Ravikumar, P. K. Representer point selection for explaining deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 9291–9301, 2018.