

---

# MODELLING NONLINEAR DEPENDENCIES IN THE LATENT SPACE OF INVERSE SCATTERING

---

**Juliusz Ziomek**

School of Electronics and Computer Science  
University of Southampton

**Katayoun Farrahi**

School of Electronics and Computer Science  
University of Southampton

## ABSTRACT

The problem of inverse scattering proposed by Angles and Mallat in 2018, concerns training a deep neural network to invert the scattering transform applied to an image. After such a network is trained, it can be used as a generative model given that we can sample from the distribution of principal components of scattering coefficients. For this purpose, Angles and Mallat simply use samples from independent Gaussians. However, as shown in this paper, the distribution of interest can actually be very far from normal and non-negligible dependencies might exist between different coefficients. This motivates using models for this distribution that allow for non-linear dependencies between variables. Within this paper, two such models are explored, namely a Variational AutoEncoder and a Generative Adversarial Network. We demonstrate the results obtained can be extremely realistic on some datasets and look better than those produced by Angles and Mallat. The conducted meta-analysis also shows a clear practical advantage of such constructed generative models in terms of the efficiency of their training process compared to existing generative models for images.

## 1 Introduction

Scattering networks have been proposed by Bruna and Mallat [1]. They are a hard-coded transformation designed to be stable with respect to deformations. This implies that a small deformation (which can be non-linear) changing the input should produce a relatively small change in the feature space of its representation. In the paper where scattering networks for image data are proposed, Bruna and Mallat show that a classifier trained on the representation produced by scattering networks on MNIST dataset, outperforms convolutional neural networks in case when limited training data is available. Later, in 2018, Angles and Mallat [2] propose a generative model using the representation produced by scattering networks. In their model, this representation is first obtained for the entire training set, after which Principal Component Analysis is applied to reduce the dimensionality of this representation to 512 features. This representation is then whitened to remove all linear dependencies between features. A deep neural network is subsequently trained to invert this operation and reconstruct an image from the principal components of the scattering network's representation. Such a trained neural network can then be used as a generative model and produce artificial images given artificial principal coefficients. To generate artificial principal coefficients, Angles and Mallat use a distribution of independent Gaussians. They argue that as the scale of the scattering network grows, the distribution of the principal coefficients should approach a Gaussian. This follows directly from the central limit theorem under the assumption that with enough distance pixels will become independent of each other. However, in scattering networks we usually use relatively small scale and in the case of experiments of Angles and Mallat, the averaging occurred over windows of size 16 by 16 pixels while working with images of size 128 by 128. In natural images of such size, locations separated by less than 16 pixels might be highly dependent, meaning that the distribution of interest might be relatively far from Gaussian. The empirical evidence agrees with this conclusion as the results shown by Angles and Mallat do not closely resemble the training distribution. The main aim of this paper is to investigate whether using artificial principal coefficients sampled from models capable of including nonlinear dependencies between features can improve those results.

## 2 Related Work

### 2.1 Variational Autoencoders

Variational autoencoders (VAE) are a class of autoencoders based on the framework developed by Kingma and Welling [3]. They force the model to learn a latent representation with a distribution close to independent Gaussians. The cost function used in these models is a sum of the reconstruction loss which measures how well the image is reconstructed from its representation and a KL-divergence between the distribution of latent space and independent Gaussians. Later, disentangling VAEs [4] were proposed, which additionally weigh the KL-divergence term by a  $\beta$  constant.

### 2.2 Generative Adversarial Networks

Goodfellow et al [5] propose an adversarial process in which two models are trained simultaneously, the generator  $G$  and discriminator  $D$ . The task of the generator is to convert  $H$ -dimensional, random noise vector  $\mathbf{z} \sim p(\mathbf{z})$  to a datapoint resembling those coming from given data distribution  $p(\mathbf{x})$ . The task of the discriminator is to differentiate between the true datapoints  $\mathbf{x}$  and the fake datapoints  $\hat{\mathbf{x}}$  generated by  $G$ . Since their invention, GANs have dominated the landscape of deep generative models and established the state of art on many tasks.

### 2.3 Other generative models utilising scattering networks

Oyallon et al [6] propose a different approach to use scattering networks in generative models than the one used by Angles and Mallat [2]. They use a GAN in the coefficient space to generate artificial scattering coefficients (generator produces artificial scattering coefficients and discriminator tries to distinguish them from real scattering coefficients). However, they work directly with high-dimensional scattering coefficients and not the low-dimensional principal coefficients. They also map the coefficients to the image domain via expensive numerical reconstructions rather than via a neural network.

## 3 Experiment Setup

The experiments in the following sections were conducted on two datasets: MNIST hand-written digits and CelebA [7] containing photos of faces of celebrities. For the MNIST dataset, whole images of shape (28,28) are used. For CelebA dataset, centre-crops of shape (128,128) are used. Before any further work could be done, the representations of images in form of whitened scattering coefficients had to be obtained. The scattering network scale of  $J = 4$  was used for CelebA. The scale of  $J = 2$  was used for MNIST, as this dataset has much smaller images. The experimental setup for CelebA was chosen to faithfully reproduce the work of [2], as they use the same image shapes, same scale of scattering networks and same size of train sets (65,536).

When it comes to training the networks mapping representations to the image domain, we use the same architecture as [2] for CelebA and an architecture with reduced capacity for MNIST, where we set the filter size for all convolutional layers to 3 and the size of the last hidden layer to 32, where previous layers have sizes following geometric progression with a ratio of  $1/2$ .

## 4 Investigating nonlinear dependencies in the latent space

### 4.1 Visualisation of principal components

To better understand what the principal components represent, we visualise the effect of changing the ones with the greatest variance before whitening. For each dataset considered, a "visualisation matrix" was created for the two most significant components. Such a matrix consists of 16 images that are created by mapping different vectors to the image domain. Each column corresponds to increasing the value of the first coefficient and each row to increasing the value of the second coefficient. Values of those components were chosen from a set  $\{-10, -5, 5, 10\}$ . The values of the remaining 510 components were set to 0. The result of this process is shown in Figure 1. The visualisation matrices graphically illustrate the dependencies between the first two principal components. One can see that a given value of the first coefficient produces a realistic image for some specific values of the second coefficient, but not for others. This would suggest that those components must be "jointly realistic" to produce a realistic image, meaning that they are most likely not independent. This conclusion aligns with what is indicated by statistical tests in the next paragraph.

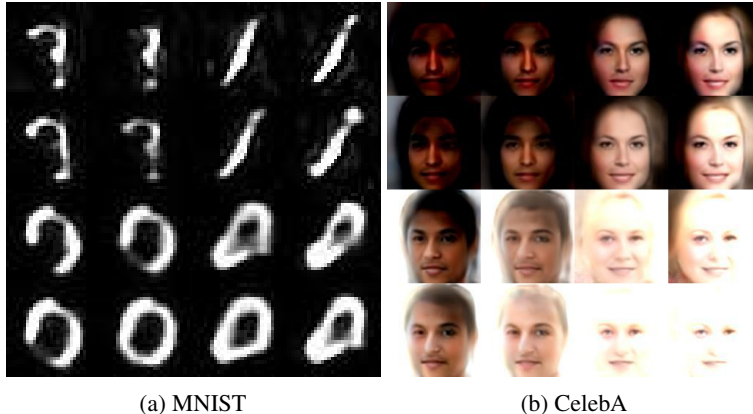


Figure 1: Visualisation matrix of components mapped to the image domain. Each column top to bottom corresponds to increasing the value of the first coefficient. Each row from left to right corresponds to increasing the value of the second coefficient.

## 4.2 Statistical tests for normality

Direct testing for the independence of multi-dimensional continuous data is difficult and computationally infeasible. Instead, we measure how much individual distributions of principal components differ from a Gaussian. If those distributions are perfect Gaussians, then they should be completely independent, because they were whitened. If they differ significantly from a Gaussian, whitening does not guarantee independence and it is reasonable to assume there might be (nonlinear) dependencies between them. To test how likely it is that those components come from a Gaussian distribution, we conduct two statistical tests for normality: D’Agostino’s K-squared (K2) and Jarque–Bera (J-B) for each of the component for MNIST and CelebA datasets. The results are shown in Table 1. At the significance level of  $\alpha = 0.05$  depending on dataset and test 35-51 components (7% - 10%) have normality hypothesis rejected and on level of  $\alpha = 0.01$  this number is equal to 12 - 24 (2% - 4%). This is enough to conclude that at least some components have distributions significantly differing from a Gaussian. The main question that remains is what kind of information those components will usually encode and how setting them to unlikely values will affect the reconstructed image. Within the next section we show that using more complex models to sample the principal components can have an overwhelmingly positive effect on the quality of samples generated.

Dataset	Test type	Number of comp. rejected at	
		<b>0.05-level</b>	<b>0.01-level</b>
MNIST	K2	<b>41</b>	<b>12</b>
	JB	<b>35</b>	<b>16</b>
CelebA	K2	<b>46</b>	<b>19</b>
	JB	<b>51</b>	<b>24</b>

Table 1: Number of principal components for which the normality hypothesis can be rejected at a given  $\alpha$ -level (ie. their p-value is lower than  $\alpha$ ). Two types of tests: D’Agostino’s K-squared test (K2) and Jarque–Bera (JB) were conducted.

## 5 Introducing nonlinear models

### 5.1 Models

The first model we use for generating artificial principal coefficients was the variational autoencoder(VAE). Since the principal components are of low-dimensionality, convolutional layers were not needed and the model was constructed using only fully connected layers. We use  $H = 64$  as the size of the latent space. The architecture of VAEs used for experiments is shown in Appendix A. The second model we use was a GAN, and as in the case of a VAE, we use only fully connected layers. The dimensionality of the input noise vector  $H$  was set to  $H = 64$ . The exact architecture of the GAN used for experiments is shown in Appendix B.

## 5.2 Results

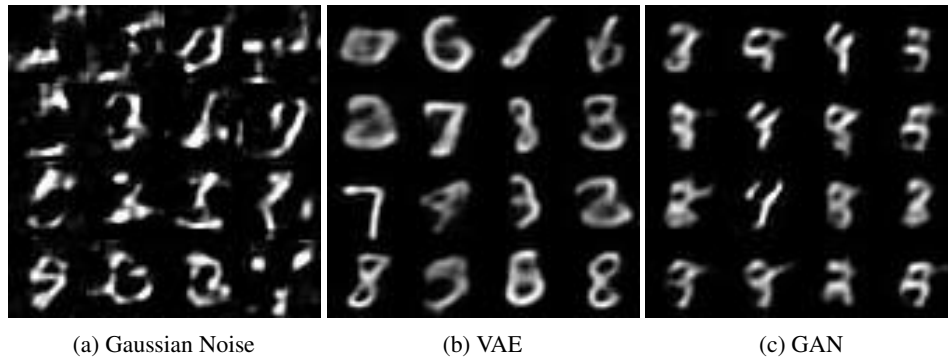


Figure 2: Results of mapping artificial principal components generated by different methods to image domain on the MNIST dataset.

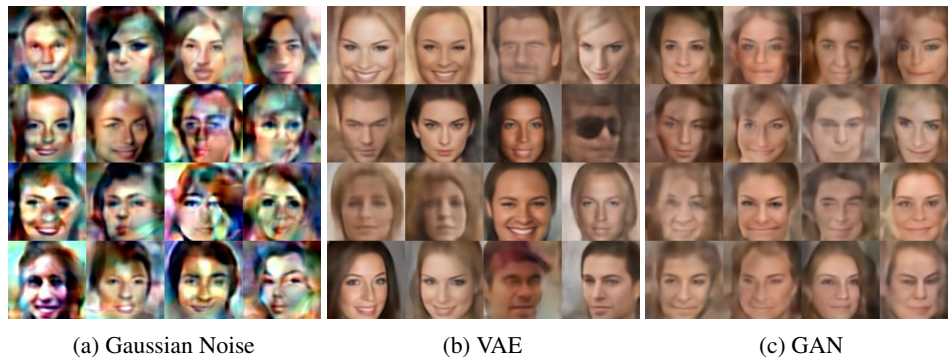


Figure 3: Results of mapping artificial principal components generated by different methods to image domain on the CelebA dataset.

Dataset	Used $\beta$	Epochs	Time Elapsed
MNIST	0.001	31	9 mins
CelebA	0.0025	96	17 mins

Table 2: Summary of the VAE training process for different datasets.

Dataset	Epochs	Time Elapsed
MNIST	30	5 mins
CelebA	38	10 mins

Table 3: Summary of the GAN training process for different datasets.

The results of mapping artificial principal components generated by different methods to the image domain is shown in Figure 2 for the MNIST dataset and in Figure 3 for the CelebA dataset. Additionally, we report training times and  $\beta$  values used for the VAE models in Table 2 and training times of GAN models in Table 3.

## 6 Discussion

Both VAEs and GANs are able to outperform Gaussian Noise on MNIST dataset (compare Figure 2 a) with b) and c)). VAEs are also able to obtain excellent results on the CelebA dataset, where some samples are arguably indistinguishable from real people. Together with the results of statistical tests this is enough to claim that principal

components containing information about important image features can have nonlinear dependencies between each other. GANs also show an improvement over generation from Gaussians, producing more consistent images, however, their improvement is not as significant as that of VAEs. GANs also generate very similar samples, which is due to a phenomenon called the "mode-collapse". However, as argued by Ian Goodfellow [8], this is a general tendency of GAN models and while it can be reduced, it cannot be completely avoided.

Possibly the most important advantage of the generative models constructed within the scope of this work is not their performance but their efficiency. Although the performance of some methods is excellent on particular datasets, it is likely that a generative model trained end-to-end will produce samples of better quality. Such an end-to-end model, however, will have one significant issue - training time. Training a deep, convolutional network often takes hours or even days. Moreover, training a generative model is a very unstable process, which is highly sensitive to hyperparameters. Therefore, such expensive training has to be repeated multiple times, before a sensible model is obtained. A tremendous advantage of the presented approach is the separation of expensive deep, convolutional network used to map to image domain from cheap and shallow networks used to generate principal components. The mapping network is trained to map a given set of inputs to corresponding outputs, and this training has no generative aspect. Such a process is straightforward to perform and multiple efficient methods have been developed to highly optimise it. Such non-generative training is also stable and not that sensitive to hyperparameters, meaning that in general, it does not have to be repeated. All of the instability lies in training a completely independent generating network, which can be a shallow, fully connected network as the principal coefficients have low dimensionality. As mentioned before, this training usually has to be repeated multiple times, however, since the network is lightweight, such repetitions are not computationally expensive. A meta-analysis was conducted to estimate typical times of generative training for different datasets. The results are shown together with methods proposed within this work in Table 4. It already shows that the total generative training time of a generative scattering model is at least an order of magnitude lower than that of a typical VAE and two orders of magnitude lower than that of a typical GAN. In a case where many different hyperparameters of the generative model have to be tested, the total time of all experiments would be significantly shorter for a generative scattering model.

Source	Dataset (dimensionality)	Method	Generative training	Non-generative training
This project	CelebA (128x128x3)	VAE in scatter. space	<b>17 mins</b>	<b>27 h 45 mins</b>
[9]	MR Brain (4x80x96x64)	VAE	<b>10 h</b>	-
[10]	CMB (400x400)	VAE	<b>6 h</b>	-
[11]	Fashion img. (180x240x3)	VAE	<b>4 h</b>	-
This project	CelebA (128x128x3)	GAN in scatter. space	<b>10 mins</b>	<b>27 h 45 mins</b>
[12]	CelebA (64x64x3)	GAN	<b>10 h</b>	-
[13]	Landscape img. (256x256x3)	GAN	<b>73 h</b>	-

Table 4: Results of the meta-analysis regarding training time of generative models.

## 7 Conclusions

In this paper we have provided strong arguments for the thesis that principal components of outputs of scattering networks for natural images, might have nonlinear dependencies and be not modelled accurately by independent Gaussians. We also show that using a model capable of capturing those nonlinear dependencies greatly improves the quality of images obtaining after mapping the components to image domain. We also compare our proposed method to existing ones and identify an improvement in terms of efficiency, as in our method the unstable generative training only concerns a very small and shallow neural network.

## References

- [1] Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *CoRR*, abs/1203.1513, 2012.
- [2] Tomás Angles and Stéphane Mallat. Generative networks as inverse problems with scattering transforms. *CoRR*, abs/1805.06621, 2018.
- [3] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [4] Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.
- [5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- [6] Edouard Oyallon, Sergey Zagoruyko, Gabriel Huang, Nikos Komodakis, Simon Lacoste-Julien, Matthew Blaschko, and Eugene Belilovsky. Scattering networks for hybrid representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP:1–1, 07 2018.
- [7] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild, 2015.
- [8] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks, 2016.
- [9] Scott Lee, Nikesh Mishra, and Rodrigo Nieto. Brain tumor segmentation on clinical datasets. *Stanford University: Technical Report*, 2020.
- [10] Kai Yi, Yi Guo, Yanan Fan, Jan Hamann, and Yu Guang Wang. Cosmovae: Variational autoencoder for cmb image inpainting. *arXiv preprint arXiv:2001.11651*, 2020.
- [11] James-Andrew Sarmiento. Exploiting latent codes: Interactive fashion product generation, similar image retrieval, and cross-category recommendation using variational autoencoders. *arXiv preprint arXiv:2009.01053*, 2020.
- [12] Weidong Yin, Yanwei Fu, Leonid Sigal, and Xiangyang Xue. Semi-latent gan: Learning to generate and modify facial images from attributes. *arXiv preprint arXiv:1704.02166*, 2017.
- [13] Akshat Gautam, Muhammed Sit, and Ibrahim Demir. Realistic river image synthesis using deep generative adversarial networks. *arXiv preprint arXiv:2003.00826*, 2020.

## A VAE architecture

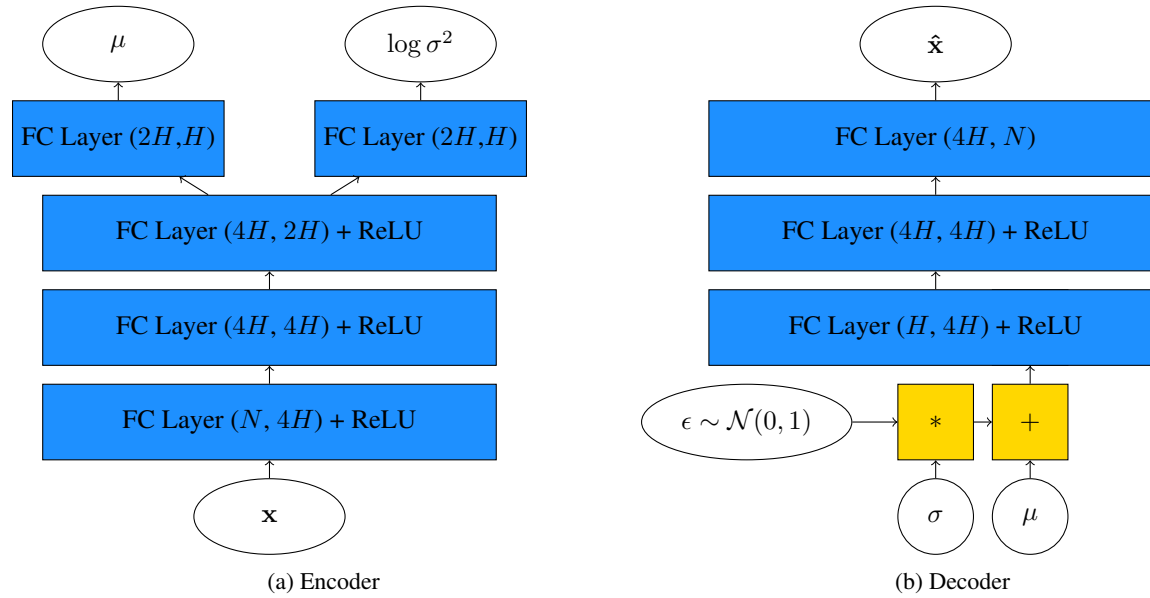


Figure 4: Schematic of the used VAE architecture.  $N$  represents the number of input features and  $H$  is the size of the latent representation.  $\mathbf{x}$  denotes the input.

## B GAN architecture

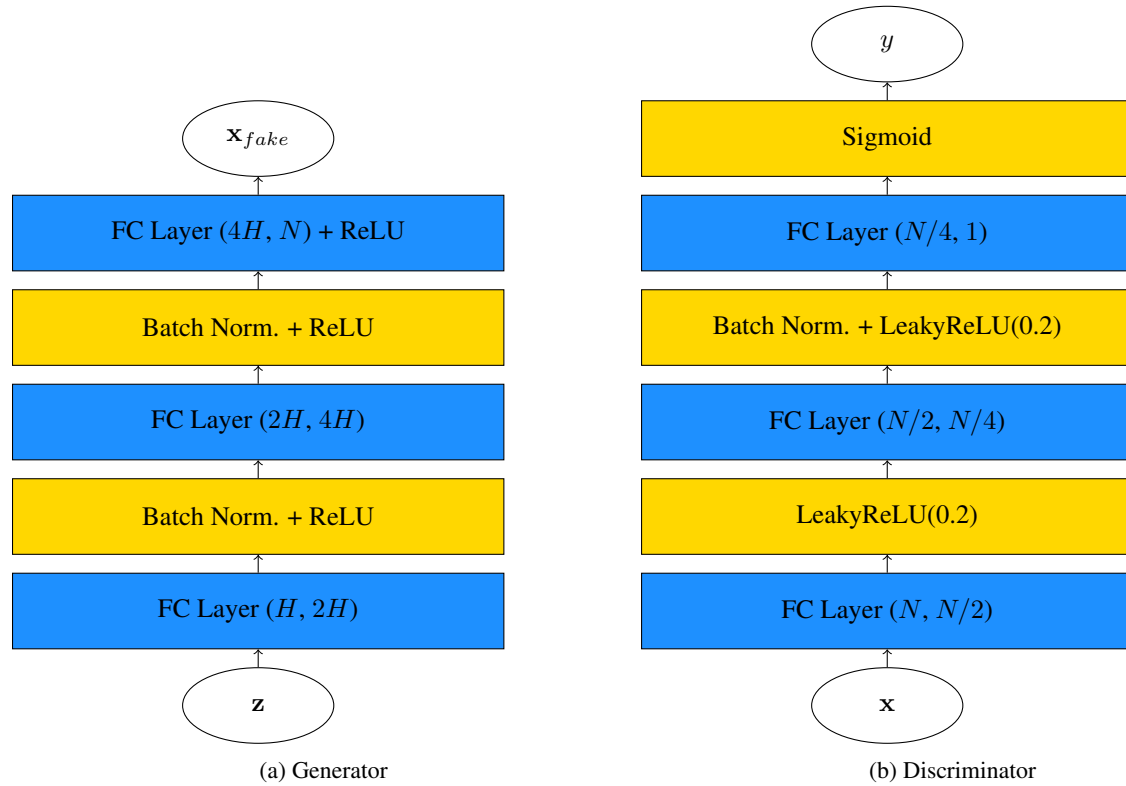


Figure 5: Schematic of the used GAN architecture.  $N$  represents the input size and  $H$  is the dimensionality of the input, random noise vector.