Master Degree in Data Science 2020-2021

“Understanding Latent Vector Arithmetic for Attribute Manipulation in Normalizing Flows”

Author: Eduard Gimenez Funes

Directors - Supervisor: Vicenç Gómez (UPF),
Co-Supervisor: Carlos Segura and Ferran Diego (Telefónica Research)

Date

ABSTRACT IN ENGLISH:

Normalizing flows are an elegant approximation to generative modelling. It can be shown that learning a probability distribution of a continuous variable X is equivalent to learning a mapping f from the domain where X is defined to Rn is such that the final distribution is a Gaussian. In “Glow: Generative flow with invertible 1x1 convolutions” Kingma et al introduced the Glow model. Normalizing flows arrange the latent space in such a way that feature additivity is possible, allowing synthetic image generation. For example, it is possible to take the image of a person not smiling, add a smile, and obtain the image of the same person smiling. Using the CelebA dataset we report new experimental properties of the latent space such as specular images and linear discrimination. Finally, we propose a mathematical framework that helps to understand why feature additivity works.

ABSTRACT IN CATALAN/ SPANISH:

Normalizing flows es una elegante aproximación al modelado generativo. Se puede demostrar que aprender una distribución de probabilidad de una variable continua X es equivalente a aprender un mapeo f del dominio donde X se define a Rn de forma que la densidad resultante sea una Gaussiana. En " Glow: Generative flow with invertible 1x1 convolutions", Kingma et al introdujeron el modelo Glow. Los flujos de normalización organizan el espacio latente de tal manera que es posible la adición de características, lo que permite la generación de imágenes sintéticas. Por ejemplo, es posible tomar la imagen de una persona que no sonríe, agregar una sonrisa y obtener la imagen de la misma persona sonriendo. Utilizando el conjunto de datos de CelebA encontramos nuevas propiedades experimentales del espacio latente, como imágenes especulares y discriminación lineal. Finalmente, proponemos un modelo matemático que ayuda a comprender por qué funciona la aditividad de características.

KEYWORDS IN ENGLISH (3)**:** "Generative Models", "Neural Nets", "Deep Learning", "Latent Space"

KEYWORDS IN CATALAN/ SPANISH (3)**:**

“Modelos genrativos”, “ Redes neuronales”, “ Aprendizaje profundo”, “Espacio latente”

Master in Data Science

Barcelona Graduate School of Economics

Understanding Latent Vector Arithmetic for Attribute Manipulation in Normalizing Flows

Eduard Gimenez Funes

Supervisor: Vicenç Gómez

Co-Supervisor: Carlos Segura and Ferran Diego (Telefónica Research)

June 2021

Master in Data Science

Barcelona Graduate School of Economics

Understanding Latent Vector Arithmetic for Attribute Manipulation in Normalizing Flows

Eduard Gimenez Funes

Supervisor: Vicenç Gómez

Co-Supervisor: Carlos Segura and Ferran Diego (Telefónica Research)

June 2021

Contents

1 Introduction **1**

1.1 Generative Adversarial Networks 2

1.2 Variational AutoEncoders 2

1.3 Flow-Based Generative Models 3

2 Normalizing Flows and the Glow Model **5**

2.1 Normalizing Flows 6

2.2 The Glow Model 6

3 The CelebA Dataset **8**

3.1 Image Pre-processing 9

4 Empirical Properties of the Latent Space **11**

4.1 The Latent Space under Gaussianity 11

4.2 Affine Structure of the Latent Space 14

4.2.1 Linear transition norm preserving 14

4.2.2 Feature additivity 16

4.2.3 Specular reflections 19

4.3 Linear Classifiers 20

4.4 Component corruption 24

4.5 Some Failures 25

5 Mathematical Grounding **26**

5.1 Mapping Uniqueness 26

5.2 Affine Structure 28
5.2.1 On the distribution of points on S^{N-1} 28
5.3 A heuristic procedure to understand when does Glow work 29
5.3.1 Locality 29
5.3.2 Feature Uniqueness 29
5.3.3 Feature Sampling 30
5.3.4 How about the opposite? 30
5.3.5 Out of Norm 30
5.4 Domain of the normalizing flow 33

6 Conclusions **34**

List of Figures **35**

Bibliography **37**

Dedication

I would like to dedicate this work to my family, my wife Elena and our two children Marta and Lluís. I miss them and I'm looking forward to spending more time with them.

Acknowledgement

I would like to express my sincere gratitude to Vicenç Gómez for offering me such an exciting project that I have truly enjoyed. His dedication, patience and all the time and great advises he has offered me throughout this project. Additionally, to Ferran Diego Andilla and Carlos Segura from Telefonica Research for their expert advise and... for letting me play with their wonderful GPU grid. To Gabor Lugosi, who has taught me a lot of beautiful Mathematics that have inspired me during this project. An finally, to Andrea Valenzuela, former student of Vicenç who dedicated quite a few hours to allow me build on her previous work. Thanks.

Abstract

Normalizing flows are an elegant approximation to generative modelling. It can be shown that learning a probability distribution of a continuous variable X is equivalent to learning a mapping f from the domain where X is defined to $\mathbb{R}^N$. In [1] the authors introduced the Glow model, pushing forward previous work developed by [2]. Normalizing flows arrange the latent space in such a way that feature additivity is possible, allowing synthetic image generation. For example, it is possible to take the image of a person not smiling, *add a smile*, and obtain the image of the same person smiling. Using the CelebA dataset [3] we report new experimental properties of the latent space such as specular images and linear discrimination. Finally, we propose a mathematical framework that helps to understand why feature additivity works.

Chapter 1

Introduction

Given a class of elements, for example, images of human faces, generative modelling aims at learning its distribution and extract meaningful features with as little, potentially unlabeled, data as possible. This goal is far more ambitious than discriminative modelling, where a model is trained for classification or regression tasks with, usually, labeled data.

The applications of generative modelling are two-fold. On one side, it would allow artificial intelligent systems to better mimic some human cognitive capabilities. For example, humans are able to recognize other humans by just looking at a single picture. Furthermore, humans are able to extract features from a given domain and mentally manipulate the features of a given element of the class. We can see the picture of an unknown person not smiling and we can imagine how this person would smile. On the other side, a wide range of applications are possible such as semi-supervised learning or speech synthesis. In the domain of image synthesis, Deepfakes are paramount. Deepfakes are synthetic media, images or video, in which a person appearing in the image of video is replaced by a different one.

Generative modelling is a broad family of models starting with semi-analytical models such as Naive Bayes or Gaussian Mixture Models. In [4] the authors introduced Latent Dirichlet Allocation, an influential paper that help foster the field of text categorization and its applications, for example, to central banks speech analysis

[5, 6]. However, it has been in recent years when the field of Generative modelling has leaped. Among the most relevant ones, we find Generative Adversarial Networks (GANs), Variational AutoEncoders (VAEs) and Flow-Based Generative Models. The following sections describe each one of them.

1.1 Generative Adversarial Networks

On one side of the spectrum we find smart approaches that are able to turn an unsupervised problem into a supervised one. This is the case of Generative Adversarial Networks (GANs) [7]. The objective in this case is to obtain a system capable of synthetically generate new elements that seem to follow the original distribution. For example, in the case of images, generating realistic ones of people, houses, ..., where the user might be able to guide some features of the synthetic image such as gender or hair style.

To achieve this goal, a Generator and a Discriminator are jointly trained competing among each other. The Generator is entitled to create new data points that resemble as much as possible to the original data distribution. The Discriminator's task is to identify which examples belong to the original data distribution and which have been created by the Generator.

1.2 Variational AutoEncoders

AutoEncoders (AE) can be regarded as a generalization of Principal Component Analysis (PCA) where the goal is finding a pair of functions φ, ψ such that:

$$\begin{aligned} \varphi &: X \longrightarrow L, \\ \psi &: L \longrightarrow X, \\ \varphi, \psi &= \operatorname{argmin}\{|X - (\varphi \circ \psi)X|\}. \end{aligned} \tag{1.1}$$

PCAs have been traditionally used for dimensionality reduction. In the context of financial economics, a popular application is understanding the interest rate curve

variations of a given currency, whose dimensionality can be in the order of 20, by reducing it to a 3-dimensional (slope, parallel, and first order curvature) vector.

In the context of Neural Networks, AEs have been used for many image processing tasks, including denoising, e.g., Figure 1 [8], super-resolution, or image swapping. The face swapping task can be understood as applying $\varphi \circ \psi$ to a new person's face not included in the training phase. The functions φ, ψ are trained such that their composition $\varphi \circ \psi$ projects the initial image to one similar to that included in the training set.

Figure 1: Image of Lena used in many image processing tasks, here for denoising[1].

Originally, in equation 1.1 there is no structure imposed in the latent space L. Therefore, a priori, there is no warranty that all possible configurations of L are meaningful nor that there is some notion of similarity between close points in the feature space. To overcome this difficulty, Kingma and Welling [9] introduced Variational Auto-Encoders (VAE). Within the VAE framework, encoder φ and decoder ψ do not map points into points but rather points into probability distribution functions.

1.3 Flow-Based Generative Models

Finally, we have Flow-based generative models, first described in [10] and extended in [2]. Flow-based generative models have some advantages over prior approaches:

[1] `https://en.wikipedia.org/wiki/Lenna`.

- Exact latent-variable inference and log-likelihood computation are tractable. In VAEs, we need to make approximations in order to calculate the log-likelihood of a given data point. GANs do not have an explicit likelihood function.

- Efficient inference and synthesis. Sampling an image, as well as computing its latent representation can be performed using simple calculations.

The objective of this work is to gain understanding of some properties of the latent space structure of a particular normalizing flow model, the Glow model [11]. For that, we provide experimental results and a theoretical analysis that aims to explain why the Glow latent space structure is convenient for certain tasks.

In Chapter 2, we describe in detail the Glow model and compare it with previous flows. Chapter 3 describes the CelebA data set used to carry on our experimental studies. This data set comprises more that 200K celebrity images annotated with 40 attributes. Chapter 4 reports results on the structure of the latent space. Some of these results are largely based on intuitions, while others provide a more rigorous understanding of the latent space. Beyond the empirical findings, we aimed at providing some theoretical grounding to the experimental results. This is done in Chapter 5. Finally, Chapter 6 summarizes the work.

Chapter 2

Normalizing Flows and the Glow Model

As the dimensionality increases, the task of directly learning the probability distribution from real data, $p(x)$ where $x \in D$ becomes infeasible due to the curse of dimensionality. Firstly, it becomes harder and harder to find sample data in bigger and bigger regions of the domain space. Secondly, the computations become exponentially slower.

To overcome this difficulty, flow-based deep generative models take advantage of an important fact: modelling any continuous probability distribution is equivalent to finding an invertible mapping $f : D \to \mathbb{R}^N$ such that the resulting probability distribution maximizes the following log-likelihood

$$\mathcal{L}(f) = \int_D \log \left(\phi_{0,1} \left(f(x) \right) \left| \frac{\partial f}{\partial x} \right| \right) d\mu, \tag{2.1}$$

where:

- $\phi_{0,1}(\mathbf{z})$ is the density function of an N-dimensional Gaussian distribution.
- $\left|\frac{\partial f}{\partial x}\right|$ is the determinant of the mapping f.
- $d\mu$ is the density in the original space.

Chapter 5 describes with further detail this aspect and proves that *essentially* there is only one such mapping.

2.1 Normalizing Flows

The maximization of the discrete version of equation can be written as

$$\mathcal{L}_d(f) = \sum_{x \in D'} \log(\phi_{0,1}(f(x))) \left| \frac{\partial f}{\partial x} \right|. \tag{2.2}$$

In contrast to equation (2.1), equation (2.2) is a little more *subtle*, since there are uncountable many invertible mappings that bring all $x \in D$ arbitrarily close to the origin with $\left|\frac{\partial f}{\partial x}\right|$ arbitrarily high. That is, the maximization of equation (2.2) is unbounded.

Thus, for optimizing this objective one has to take some regularizing conditions into account. Normalizing flows consider a fixed number of invertible mappings within a class such that $\mathbf{z} = f(x) = f_L \circ \cdots \circ f_1(\mathbf{x})$ and the computation of $\left|\frac{\partial f_i}{\partial f_{i-1}}\right|$ can be done efficiently. In [10] and [2] an initial class of normalizing flows was proposed.

2.2 The Glow Model

The Glow model [11] builds on the previous invertible normalizing flows, NICE [10] and RealNVP [2] introducing the 1×1 invertible convolutions instead of reverse permutations. A 1×1 convolution is a generalization of any permutation of the channel ordering.

Glow creates a new building block consisting of three consecutive sub-layers, an *actnorm* followed by an invertible 1×1 convolution, followed by a coupling layer.

The following table summarizes the elements of the Glow model. The three main components of the proposed flow, their reverses, and their log-determinants. For each one of the proposed building blocks, the table describes the function, its inverse and its determinant.

Description	Function	Reverse Function	Log-determinant
Actnorm. See Section 3.1.	$\forall i,j : \mathbf{y}_{i,j} = \mathbf{s} \odot \mathbf{x}_{i,j} + \mathbf{b}$	$\forall i,j : \mathbf{x}_{i,j} = (\mathbf{y}_{i,j} - \mathbf{b})/\mathbf{s}$	$h \cdot w \cdot \texttt{sum}(\log \lvert\mathbf{s}\rvert)$
Invertible 1×1 convolution. $\mathbf{W} : [c \times c]$.	$\forall i,j : \mathbf{y}_{i,j} = \mathbf{W}\mathbf{x}_{i,j}$	$\forall i,j : \mathbf{x}_{i,j} = \mathbf{W}^{-1}\mathbf{y}_{i,j}$	$h \cdot w \cdot \log \lvert\det(\mathbf{W})\rvert$ or $h \cdot w \cdot \texttt{sum}(\log \lvert\mathbf{s}\rvert)$
Affine coupling layer.	$\mathbf{x}_a, \mathbf{x}_b = \texttt{split}(\mathbf{x})$ $(\log \mathbf{s}, \mathbf{t}) = \texttt{NN}(\mathbf{x}_b)$ $\mathbf{s} = \exp(\log \mathbf{s})$ $\mathbf{y}_a = \mathbf{s} \odot \mathbf{x}_a + \mathbf{t}$ $\mathbf{y}_b = \mathbf{x}_b$ $\mathbf{y} = \texttt{concat}(\mathbf{y}_a, \mathbf{y}_b)$	$\mathbf{y}_a, \mathbf{y}_b = \texttt{split}(\mathbf{y})$ $(\log \mathbf{s}, \mathbf{t}) = \texttt{NN}(\mathbf{y}_b)$ $\mathbf{s} = \exp(\log \mathbf{s})$ $\mathbf{x}_a = (\mathbf{y}_a - \mathbf{t})/\mathbf{s}$ $\mathbf{x}_b = \mathbf{y}_b$ $\mathbf{x} = \texttt{concat}(\mathbf{x}_a, \mathbf{x}_b)$	$\texttt{sum}(\log(\lvert\mathbf{s}\rvert))$

Here, $\mathbf{x}$ denotes the input of the layer, and $\mathbf{y}$ denotes its output. Both $\mathbf{x}$ and $\mathbf{y}$ are tensors of shape $[h \times w \times c]$ with spatial dimensions (h, w) and channel dimension c. With (i, j) we denote spatial indices into tensors $\mathbf{x}$ and $\mathbf{y}$. The function $\texttt{NN}(\cdot)$ is a nonlinear mapping, such as a (shallow) convolutional neural network like in ResNets [2] and RealNVP [12]. Figure 2 represents the compositional architecture.

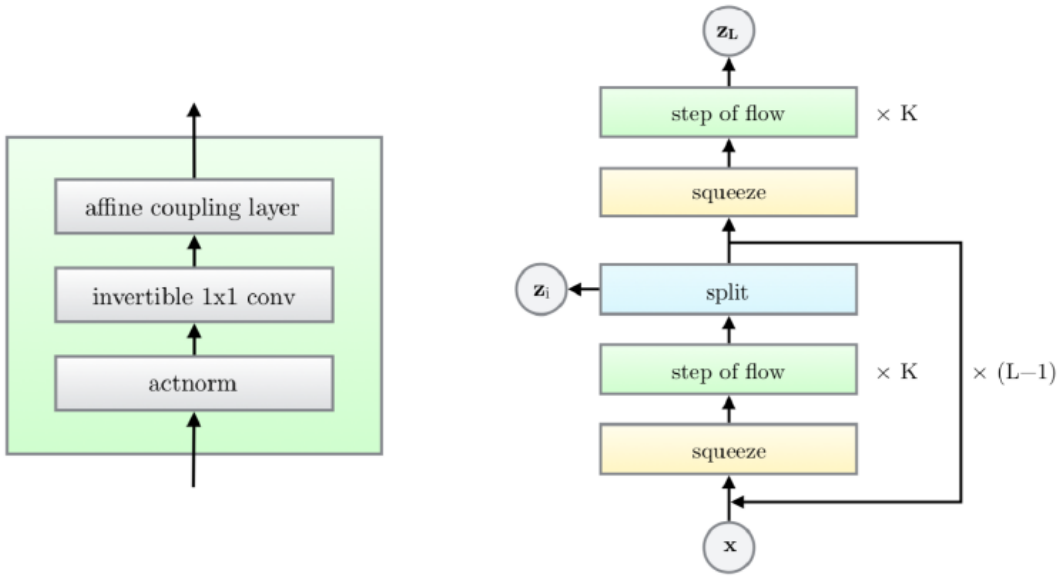

Figure 2: Block diagram corresponding to a layer of the Glow Architecture. (left) One step of the flow. (right) Full architecture. For additional details, see [11].

Chapter 3

The CelebA Dataset

The Celeb Faces Attributes Dataset (CelebA) [3] is a large-scale face attributes dataset with more than 200K celebrity images, each with 40 attribute annotations. CelebA is highly diverse and contains rich annotations, including 10,177 individual identities, 202,599 number of face images, 5 landmark locations, and 40 binary attributes annotations per image. The dataset is used widely employed as a benchmark for computer vision on different tasks such as face attribute recognition, face detection and synthesis.

Figure 3: Sample CelebA impages, illustrating 8 of the 40 annotated attributes.

3.1 Image Pre-processing

The CelebA data set consists of a broad range of images, all of them containing a single person but gestures and poses vary across all of them. Ideally, this could be the original source space to apply generative modelling. In practice, this is a huge space, so it is necessary to take some restricting decisions.

The first one is defining a fixed size for all images. The Glow architecture was calibrated against a set of 256×256 images. The second decision to be made is locate the precise face within the image. For that, the original images are cropped and resized to obtain a 256×256 image where all faces share the same eye location. Figure 3.1 shows a sample of original images and the aimed result after applying cropping and resizing. We can observe, for example, that both processed images share the similar eye location.

Figure 4: Sample images before and after processing.

The former eye alignment has to be done for 200K images. In [3] the authors stated that "Each image in CelebA and LFWA is annotated with forty face attributes and five key points by a professional labeling company". The CelebA includes an additional file with face landmarks that helps the prepocessing task. Using the landmark information, all 200K images were cropped to create a source space ready for generative modelling.

Chapter 4

Empirical Properties of the Latent Space

The latent space learned using the Glow model has several amazing properties such as feature additivity that can be used for synthetic image generation, but we have also found some additional ones.

In this chapter, we start from a learned Glow model and analyze experimentally some properties of the latent representation. This chapter reproduces relevant properties from [2, 9, 7, 13] and describes new findings, namely, specular symmetry and linear discrimination.

4.1 The Latent Space under Gaussianity

As we will later discuss in Chapter 5, a mapping f that minimizes equation (2.1) maps a distribution defined in the source space to a normal distribution ($\mu = 0, \sigma^2 = 1$) in the latent space. Therefore, it is reasonable to assess how close does the normalizing flow gets to fulfill this task by calculating the sample values of μ and σ. Since the latent space has dimension ($256 \times 256 \times 3$) it is hard to comprehend these to values. μ is a $(256 \times 256 \times 3)$ vector and σ is a $(256 \times 256 \times 3) \times (256 \times 256 \times 3)$ matrix. Instead, we will use $\|\mathbb{E}[x]\|$ as proxy for μ. We will compare its sample value versus

a distribution of randomly generated of $\mathbb{E}[\|x\|]$ drawn from a normal distribution. That is, for 1000 trails, we will draw as many samples from a true $(256 \times 256 \times 3)$ dimensional normal distribution as we have drawn from the latent space, calculate the mean and then the norm of that mean. Figure 5 represents the obtained results. The distribution, generated from a normal one, has so little variance that seems like a point mass distribution. This experiment helps us conclude that the normalizing flow is biased.

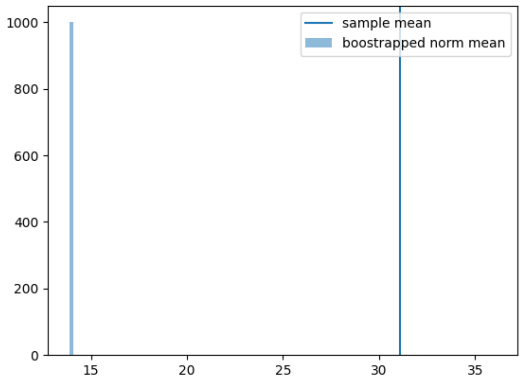

Figure 5: Comparison between the histograms of the norm multi-variate Gaussian distribution with zero mean and unit variance, and the norms of the learned latent representation.

For σ^2 we will make use of the fact that $\mathbb{E}\left[\sum x_i^2\right] = N$ when x_i are independent normal distributions $(\mu = 0, \sigma^2 = 1)$. In high dimensions there is very little convexity so we can expect $\mathbb{E}[\|x\|] \approx \sqrt{256 \times 256 \times 2563}$. Figure 6 represents these calculations. Again, we sample from the final distribution defined in the latent space and for each point we calculate the norm. We compare this distribution to the one than we would obtain by following the same procedure when drawing samples from a normal distribution.

While figure 5 is telling us that the final distribution is biased, figure 6 says that that all points are mapped on a sphere *just a little* bigger that it would theoretically be expected.

As suggested in Figure 5, the *typical* norm for a point that follows a high dimensional normal distribution is of the order of $\sqrt{N}$. We can make use of this fact interpreting

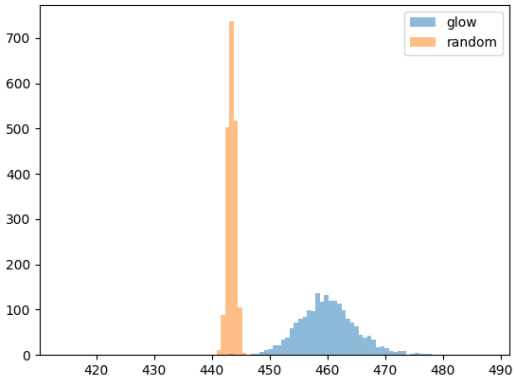

Figure 6: Distribution of norms drawn from a normal distribution vs Distribution of norms drawn from the one generated by the normalizing flow.

that *most* of the points lay on an $(N-1)$-dimensional sphere of radius $\sqrt{N}$. We will denote this sphere as S^{N-1}. Our next goal is to asses whether these points are evenly distributed or not. We will project a set of points drawn from the distribution of the latent space against the line L defined by the origin and the average of the distribution $\bar{\mu}$. We will compare this distribution versus the one generated by drawing points form a normal distribution and projecting them against L. By comparing the results presented in Figure 7, we can conclude that the normalizing flow maps all points to only to one hemisphere of S^{N-1}.

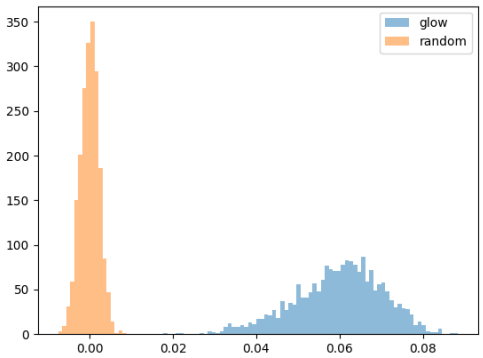

Figure 7: Distribution of points drawn from a normal distribution projected against the line L and distribution of points drawn from the distribution defined by the normalizing flow. L is the line defined by *zero* and the average of the distribution defined by the normalizing flow.

4.2 Affine Structure of the Latent Space

We say that the latent space has an affine structure in the sense that one can take a picture of a person who is not smiling, *add a smile* vector, and will obtain a picture of the same person smiling. Furthermore, given two images, any convex linear combination of them will create a realistic picture. This properties are well known and reported in [10] and [2]. In this section we report experiments reproducing this properties and some other that given the *affine structure* also hold.

4.2.1 Linear transition norm preserving

Figure 8 represents the result obtained by taking two given images, mapping them to the latent space as $image_A$ and $image_B$, creating in the latent space a linear path from $image_A$ to $image_B$, and mapping a set of points in this linear path back to the original space. We can see that the resulting images, in general, are fairly realistic.

$$p_t = (1 - t) * image_A + t * image_B \tag{4.1}$$

Figure 8: Linear transitions among two given images

As these images are farther apart, this straight line is farther and farther away from S^{N-1}. Therefore, one reasonable experiment would be to try to find a path that goes from $image_A$ to $image_B$ but as close as possible to S^{N-1}. We generate such

path p_t using the following equation:

$$\begin{aligned} n_a &= \|image_A\| \\ n_b &= \|image_B\| \\ x_t &= (1-t) * image_A + t * image_B \\ n_t &= \|x_t\| \\ p_t &= \frac{(1-t) * n_a + t * n_b}{n_t} * x_t. \end{aligned} \tag{4.2}$$

Figure 9 presents the obtained results. Unfortunately, the results are slightly worse. This result is some how counter-intuitive since we are generating points whose norm is within the distribution of norms.

Figure 9: Norm preserving transitions among two given images. The obtained images are nor as realistic as using the *straight* line path

Figure 10 compares the evolution of the norms of the point within the linear path versus the norm of the points in the norm-preserving path. The linear path contains points of norm close to 300, completely outside of the range of sampled norms $[450, 475]$.

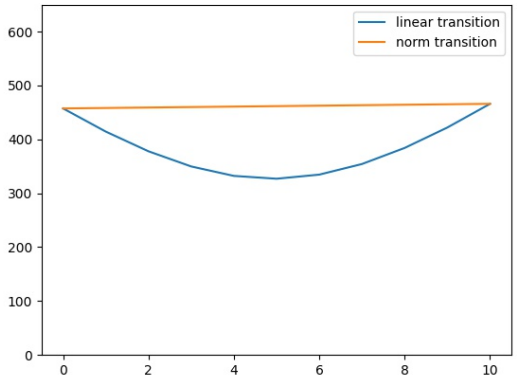

Figure 10: Norms for Linear and Norm preserving paths

4.2.2 Feature additivity

The CelebA data set contains 200K images, all labelled on 40 features. Some of these features are, for example, smiling, young, or mustache. The first thing we can do is calculate the mean of a class in the latent space and map this average to the original space. Figure 11 shows the mean values of the *bald, smiling, mustache* and *glasses* classes while figure 12 shows the mean of the elements of each opposite class. We can see that while in figure 11 the average values differ from each other, in figure 12 the average values of the complementary classes look very much alike. This is so because for these classes, $True$ and $False$ are not equally represented. That is, the class *mustache* has very few representatives compared to *not mustache*. The same happens for the classes Bald and Glasses.

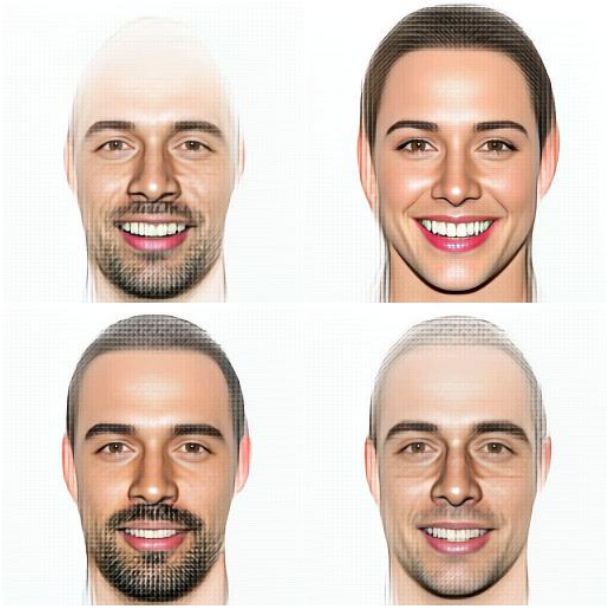

Figure 11: Average values for: Bald, Smiling, Mustache, Glasses

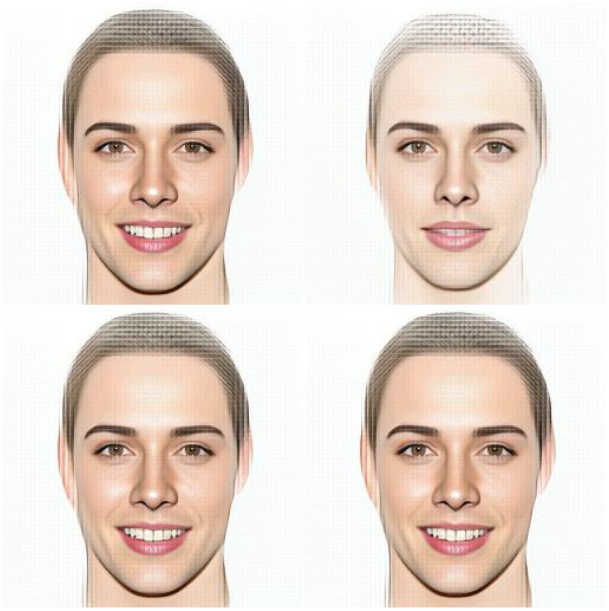

Figure 12: Average values for: Not-Bald, Not-Smiling, Not-Mustache, Not-Glasses

This does not happen for the male and female classes since these two sets are equally represented.

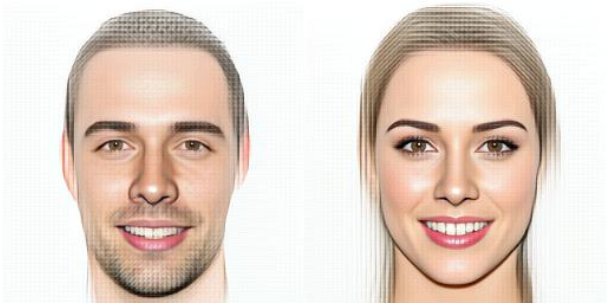

Figure 13: Male and Female prototypes

Feature additivity is a phenomenon that has been observed in different areas of Machine Learning, for example, in Natural Language Processing, the authors of Glove [14], reported their well-known formula:

$$Queen = King - man + woman. \tag{4.3}$$

In our domain, we aim at operating in a similar fashion but we would interpret subtracting Man and adding $Woman$ as subtracting the mean of the elements of the class Man and adding the mean of the elements of the class $Woman$. For any given class representing a feature, i.e. images of people smiling, we will calculate the average of the the elements within that class $\mu_{feature}$ and the average of the elements in the complementary class $\mu_{not_feature}$. Ideally, we would like to perform the following operation for any image without a given feature.

$$image_{feature} = image_{not\_feature} - \mu_{not\_feature} + \mu_{feature}. \tag{4.4}$$

Notice that this formula is *reversible*, given an image with a given feature, we could also try to remove it.

$$image_{not\_feature} = image_{feature} - \mu_{feature} + \mu_{not\_feature}. \tag{4.5}$$

We have found that adding a scaling factor improves the effectiveness of the resulting operation.

$$image_{feature} = image_{not\_feature} + \alpha * (\mu_{feature} - \mu_{not\_feature}). \tag{4.6}$$

Figures 14, 15 and 16 show the result of applying feature additivity for *smiling*, *mustache* and *old* with $\alpha = [0, 0.5, 1, 2, 3]$. We can conclude that a scaling factor improves the final image, but finding the optimal α becomes a challenge.

Figure 14: Smile transition, $\alpha = [0, 0.5, 1, 2, 3]$

Figure 15: Mustache transition, $\alpha = [0, 0.5, 1, 2, 3]$

Figure 16: Old transition, $\alpha = [0, 0.5, 1, 2, 3]$

4.2.3 Specular reflections

New now present a new experiment consistent with the affine structure of the latent space, specular reflections. For any given *image*, we wonder what $-image$ would be. Figure 17 shows this exercise for values of $\alpha = 0.5$. Using a value of $\alpha = 1$ proved

Figure 17: Results for specular imaging

to be off limits. That is, we recovered corrupted images. The specular image tends to swap hair, gender, and pose. Images staring to the right have specular images staring to the left and so on.

4.3 Linear Classifiers

When we approached ourselves to the challenge of understanding what was going on under the hood of normalizing flows, one of the topics that came to our minds was classification. If a normalizacing flow did indeed turn a given image distribution into a multi-normal one, any correlation structure should be lost. Loosing correlation would mean, for example, that CNN architectures would not work for tasks such as classification. Linear classifiers should still work, but even this class seemed too broad. The underlying reason is that the CelebA has 200K images, which is approximately the dimension of the latent space. Thus, any partition of the image set into two subsets is linearly separable. We needed an even smaller class. Given a binary partition of the latent space, i.e., mustache and non-mustache, smiling and non-smiling, we can calculate the average vector of the class $\mu_{feature}$ and $\mu_{non\_feature}$ and project any point of the distribution into the straight line defined by $\mu_{feature}$ and $\mu_{non\_feature}$. We will refer to this projection as *opposite-projection*. We can also define the straight line that goes through $\mu_{feature}$ and *zero*. We will refer to the

projection as *zero-projection*. We wondered if such a simple method would be useful for classification tasks. The answer turned out to be positive. This linear classifiers have predictive power and it's higher in the latent space than in the original space. Figures 18 and 19 report the obtained results for the *smiling* classification task. For each image, a set of four histograms are shown, the top two histograms represent the *opposite-projection* and *zero-projection* of both classes. The bottom two histograms represent the *opposite-projection* and *zero-projection* of the hold image distribution compared to a random normal distribution in the case of latent space and a uniform distribution in the case of original space. These projection exercises also allow us to understand a little bit more where the images are set on S^{N-1}.

Generally speaking, discrimination of classes is best performed using the mean projection in the latent space. Furthermore, as we saw in figure 6, the image distribution in the latent space fails to follow a normal distribution.

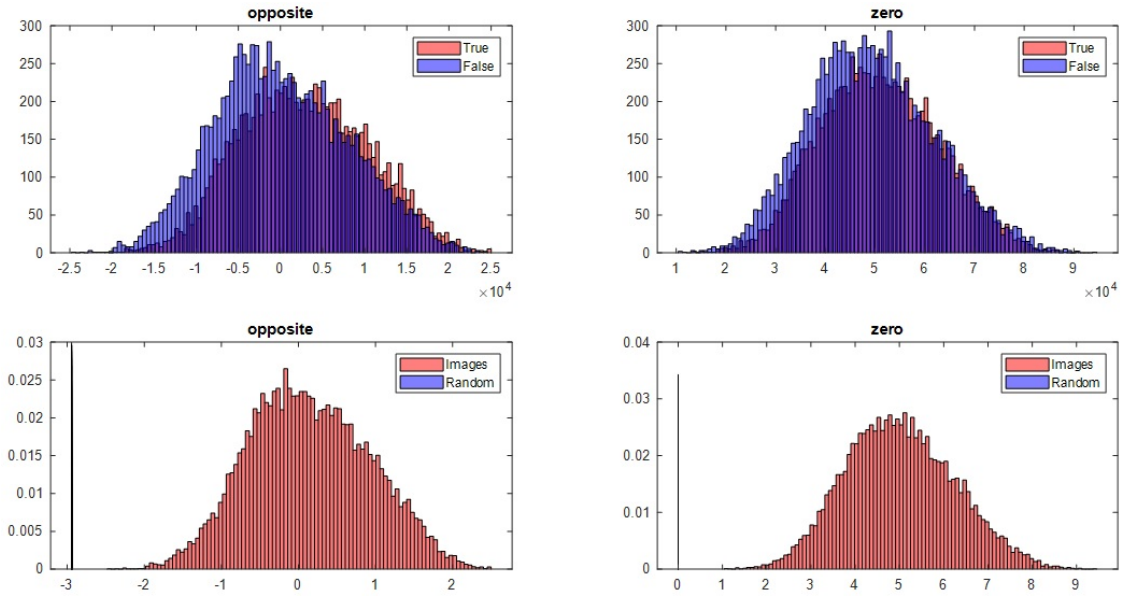

Figure 18: Smiling Image Space

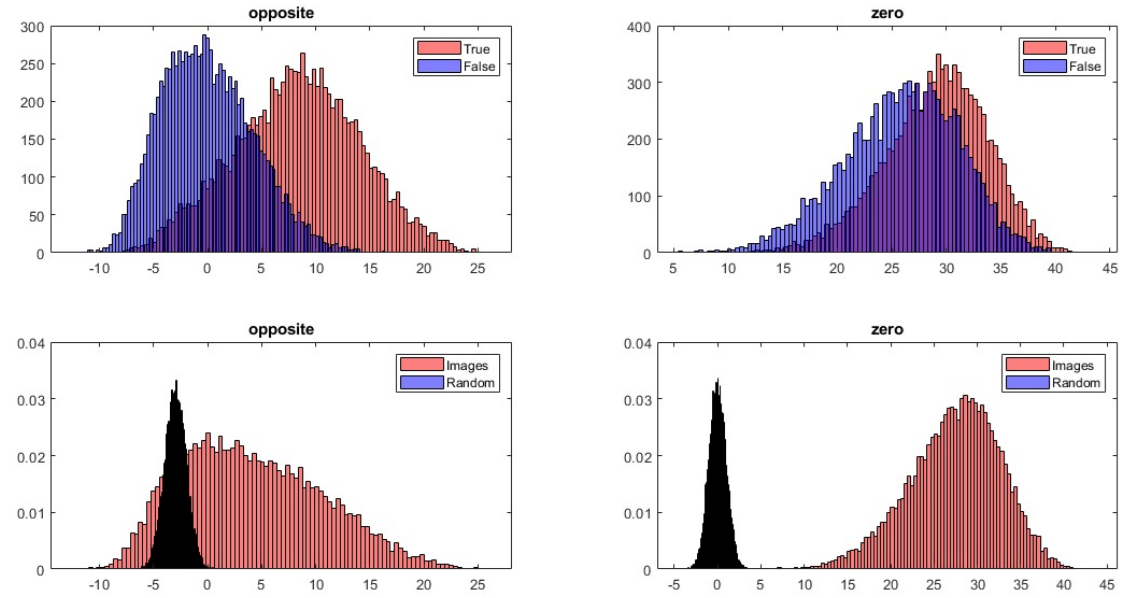

Figure 19: Smiling Glow Space

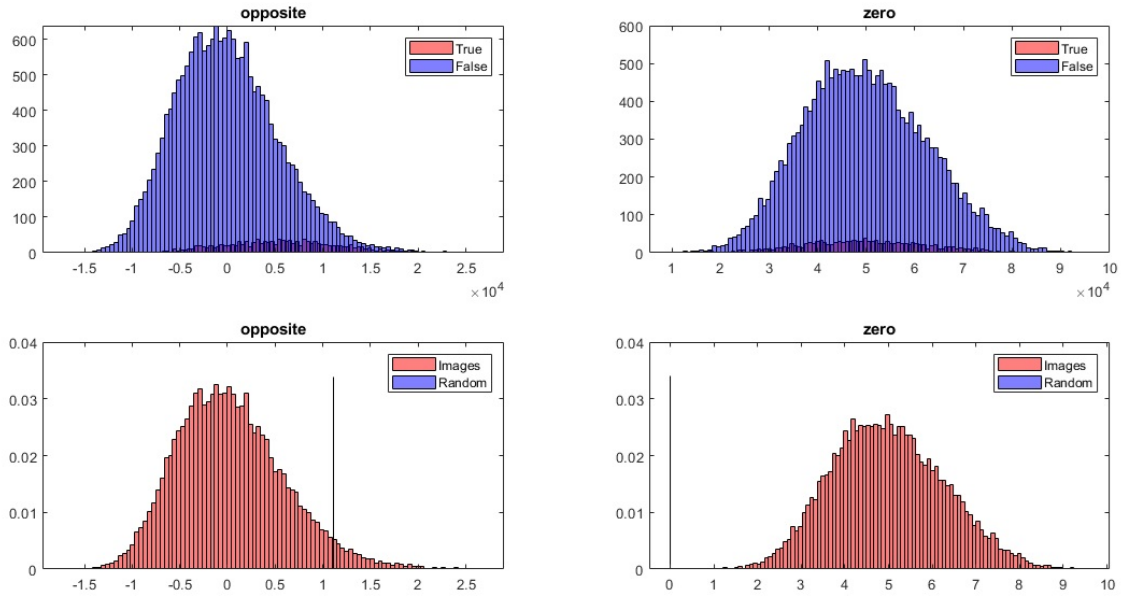

Figure 20: Eyeglasses Image Space

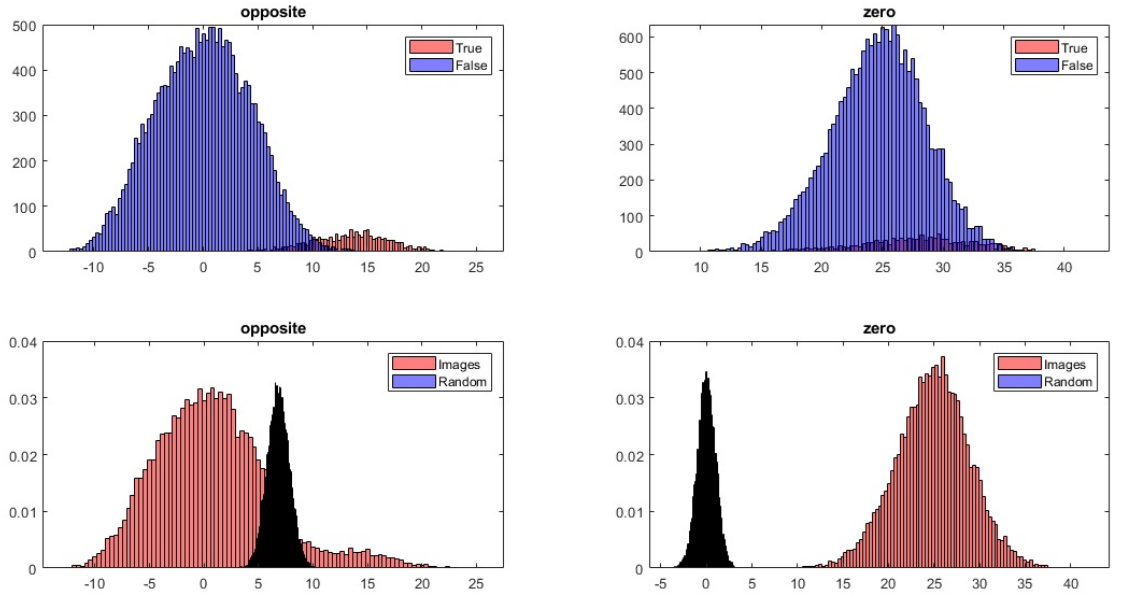

Figure 21: Eyeglasses Glow Space

4.4 Component corruption

In this section, we introduce the concept of component corruption. For a given image, we will take a random component of the latent representation and assign a value of 100. This is a value absolutely out of range since each component should follow a distribution as close as possible to a Gaussian one. Most of the times the result will be a tiny spot as shown in the left part of figure 22 a few will have a global effect.

Figure 22: Result of corrupting one single component

Another interesting property is the *additivity* of component corruptions. In figure 23 we take a set of components and corrupt them all. This property will help us understand what is going on in the following subsections.

Figure 23: Result of corrupting many components

4.5 Some Failures

Sometimes, it is equality important to understand why something is working than to understand why it is not. Figure 24 depicts the local structure of normalizing flows. When the normalizing flow is asked to add the *smiling* feature it tends to place it within the image coordinate system not relative to the given face.

Figure 24: Failed smile transition

Chapter 5

Mathematical Grounding

In chapter 4 we have reported some empirical properties of the Glow normalizing flow. In this chapter we would like to address the following challenges:

- **Gaussianity of the latent distribution:** We would like to understand why by minimizing expression 2.1, we should effectively obtain a normal distribution and how many different solutions might exist for this minimizing problem.
- **Feature additivity:** In a nutshell, why $King - man + woman = Queen$

To answer these questions, we first prove that given a continuous distribution defined in a connected domain of R^N, *essentially*, there exists only one f that maps such distribution to a normal distribution $\mathcal{N}(0, 1)$. Then we will discuss on the affine structure of the latent space. In particular why feature additivity works by sketching a simple calculation that helps us understand what is going on. Finally, we compile a set of use cases that are useful to predict if a feature transition task might work or not.

5.1 Mapping Uniqueness

Let $\phi_{0,1}(x)$ be a normal distribution defined in $\mathbb{R}^N$. A mapping $f : D \to \mathbb{R}^N$ induces a probability distribution in D as defined by equation 5.1. We will call such

distribution, the pull-back of $\phi_{0,1}$ on f.

$$df(x) = \phi_{0,1}(f(x)) \left|\frac{\partial f}{\partial x}\right| \tag{5.1}$$

When calibrating the Glow model we are effectively finding a mapping such that:

$$\hat{f} = \arg\max_f \int_D \log(\phi_{0,1}(f(x)) \left|\frac{\partial f}{\partial x}\right|) d\mu \tag{5.2}$$

$$= \arg\min_f - \int_D \log(\phi_{0,1}(f(x)) \left|\frac{\partial f}{\partial x}\right|) d\mu \tag{5.3}$$

Adding a constant won't change the minimizer $\hat{f}$.

$$\hat{f} = \arg\min_f - \int_D \log(\phi_{0,1}(f(x))) \left|\frac{\partial f}{\partial x}\right| d\mu + \int_D \log(\mu(x)) d\mu \tag{5.4}$$

$$= \arg\min_f - \int_D \log(\phi_{0,1}(f(x))) \left|\frac{\partial f}{\partial x}\right| d\mu + \int_D \log(\mu(x)) d\mu \tag{5.5}$$

$$= \arg\min_f - \int_D \log(\frac{\mu(x)}{\phi_{0,1}(f(x))) \left|\frac{\partial f}{\partial x}\right|}) d\mu \tag{5.6}$$

$$= \arg\min_f D_{KL}(\mu(x) \| \phi_{0,1}(f(x))) \left|\frac{\partial f}{\partial x}\right|) \tag{5.7}$$

Where $D_K L(P\|Q) = \int p \log \frac{p}{q} dx$ is the Kullback-Leibler divergence. The Kullback-Leibler divergence has two important properties. First is positive. Second, 0 is only reach when $P = Q$. Now let us suppose that we have two probability distributions $d\mu_1$ and $d\mu_2$ defined over the same domain D. Then if both are absolutely continuous, there exists a 1-to-1 homeomorphism $h : D \to D$ such that $d\mu_1$ is the pullback of $d\mu_2$ on h and vice versa. Thus if $\hat{f}$ minimizes expression 5.7, the pull-back distribution of $\phi_{0,1}$ on $\hat{f}$ is effectively $d\mu$. This means, among other things, that all the correlation structure that $d\mu$ might have is captured in $\hat{f}$.

Now, let us suppose there are two mapping f and g such that both pull-backs induce

$d\mu$ then $g \circ f^{-1}\mathbb{R}^N :\to \mathbb{R}^N$ induces a pull-back from $\phi_{0,1}$ to $\phi_{0,1}$. However, the set of maps with such properties is well know. Is the orthogonal group $O(N)$. Therefore, up to an element $o \in O(N)$, there exists only one minimizing mapping.

5.2 Affine Structure

In this section, we develop sketchy calculations that will help us to understand what is going on. Mainly, why most points lay on the surface S^{N-1} and why feature additivity works.

5.2.1 On the distribution of points on S^{N-1}

As we saw in figure 6 of chapter 4 the distribution of the norms of the points in the latent space is fairly concentrated on $\sqrt{265 \times 256 \times 3}$. This is consistent with the fact that maximizing 2.1 produces a normal distribution. If $X \sim \mathcal{N}(\mu = 0, \sigma^2 = 1)$ the sum of squares of its components follows a χ^2 distribution and the χ^2 concentrates its probability around N as $N \to \infty$

Now, let us suppose that we have a continuous distribution over $\mathbb{R}^{\mathbb{N}}$, an let $f : \mathbb{R}^{\mathbb{N}} \to \mathbb{R}^{\mathbb{N}}$ normalizing-flow. Then if A and B are two sets defined over $\mathbb{R}^N$ with equal probability then $f(A)$ and $f(B)$ will have the same volume.

Furthermore, suppose we have a partition of $\mathbb{R}^N$ in two half-spaces. $A \cup \overline{A} = \mathbb{R}^N$. Then:

$$|\mathbb{E}_A[x]| = \frac{e}{2\sqrt{2\pi}} \approx 0.6 \tag{5.8}$$

Equation 5.8 comes from the fact that:

$$\int_0^\infty x \frac{e^{-x^2}}{\sqrt{2\pi}} = -\frac{e^{-x^2}}{2\sqrt{2\pi}}\Big|_0^\infty \tag{5.9}$$

$$= \frac{e}{2\sqrt{2\pi}} \tag{5.10}$$

This value is independent of the dimension of N. So as $N \to \infty$ the ratio between expression 5.8 and the radius of S^{N-1} becomes smaller and smaller. Recall that the radius of S^{N-1} tends to $\sqrt{N}$ and $N \to \infty$.

Let $B_{N+\epsilon}(0) = \{x \in \mathbb{R}^N \,|\, \|x\| \leq N + \epsilon\}$. As the dimension of N grows, for any given any partition of $B_{N+\epsilon}(0) = A \cup \overline{A}$ defined by a hyper-plane that contains the origin, with high probability we will have that if $X \in A$ then $X + \mathbb{E}_{\overline{A}}[X] - \mathbb{E}_A[X] \in \overline{A}$.

To summarize the former derivations. If we define a Gaussian distribution over R^N, for high N, although the expected norm is N if we define a hyper-plane that contains the origin. The expected distance of any point to that plane is $\frac{e}{2\sqrt{2\pi}}$. This means that, on average, if we take any point and add a vector orthogonal to the hyper-plane of *just* $\frac{e}{\sqrt{2\pi}}$ pointing to the opposite site, we will effectively cross to the other side.

5.3 A heuristic procedure to understand when does Glow work

Glow is about deep learning, so any mathematics we might do is a poor man's intent to understand something that it is way too complex. However, we have compiled this cheat sheet to know when feature translation should work.

5.3.1 Locality

Is the feature you would like to manipulate local? Such as smile, eyeglasses, or mustache? Then feature additivity might work. Unlike hairstyle, eyeglasses is fairly a local concept that occurs in a defined place on the face. Figure 27 shows that while the eyeglasses placed are far from optimal, they are placed in the right place.

5.3.2 Feature Uniqueness

When you think about that feature, does it come a unique or very few images of it? Like smiling or mustache? Then feature additivity might work. It is not the

case of eyeglasses, necklaces, earrings, or hats, where there is a broad range of them. Figure 28 shows that since mustache is quite a unique and *local* concept, mustaches are correctly placed although there are no prior images with such a characteristic.

5.3.3 Feature Sampling

The feature is not local, might not be unique but it has been well sampled. Then is when the mathematical machinery that we have presented kicks in. It should work. Figure30 shows one of the most existing results where the color of the model is changed without changing its shape.

5.3.4 How about the opposite?

Maybe the feature is not local, not unique, or is not well sampled, but the opposite might be well sampled or unique. Then, *removing* that feature might be possible. Figure 29 proves that it is possible to remove the eyeglasses of the model with reasonable results.

5.3.5 Out of Norm

If the feature that we are trying to transfer has very little representation, things will go wrong. Suppose an extreme case, we consider as feature a single person and we want to transfer the *person's characteristics* to another one. We would still perform the following manipulation.

$$result = image_1 - mean\_images\_not\_image_2 + image_2 \tag{5.11}$$

Figure 25 depicts for a given image the different results obtained when *adding all the characteristic of another person* versus adding a smile.

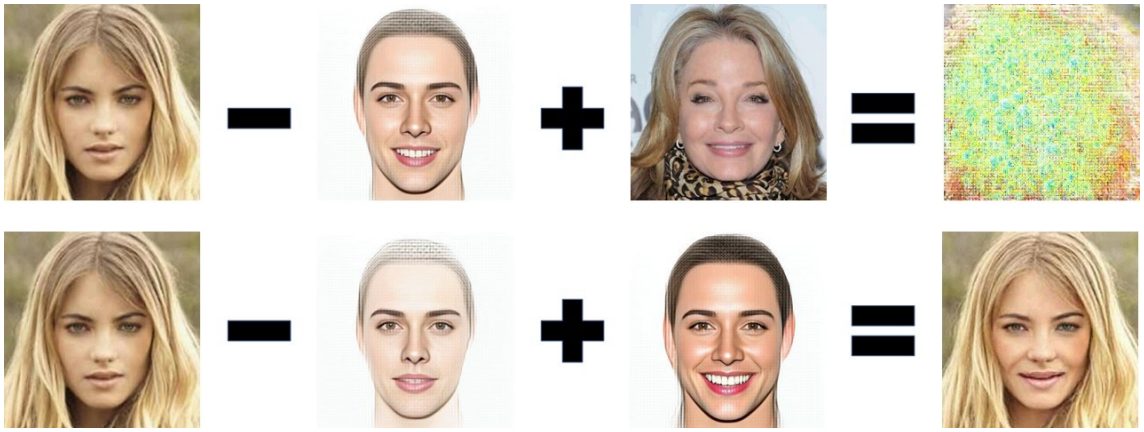

Figure 25: Adding a single person as a feature vs adding a smile

If we take a look at the norms we immediately see what is going on. In the first operation, the obtained *image* has a norm outside of the learned range, while in the second operation, the result's norm is within range.

	Source	Mean Class	Mean Feature	Result
Single Person	457	28	466	834
Smile	457	27	29	457

Figure 26: Resulting norms

Figure 27: Eyeglasses are a local concept but not unique

Figure 28: Mustache is a local and fairly unique concept

Figure 29: Not glasses is a unique well sampled concept

Figure 30: Blond, Black and Brown hair are well sampled classes

5.4 Domain of the normalizing flow

In chapter 4 we have made an empirical study of the properties of the Glow normalizing-flow. In chapter 5, we have discussed some of the theoretical properties that the limit normalizing flow should have. This let us depict, figure 31, where Glow is currently working. The vertexes represent each image properly mapped. On chapter 4, figures 8 and 10 showed us that while the original images are mapped on S^{N-1} transition from one image to another one worked best on linear paths than paths that went from the original image to the source image on S^{N-1}. Finally, specular images worked but we had to correct for a $\alpha \leq 1$.

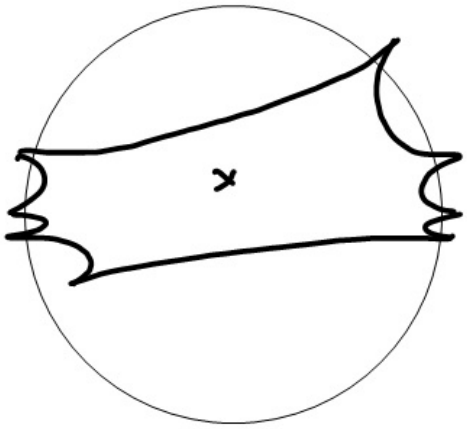

Figure 31: Representation of where Glow works

Chapter 6

Conclusions

Our original goal was to understand why feature additivity works in the Glow normalizing flow. We have been able to spot a great deal of strengths and some possible improvements. Chapter 4 presents some amazing properties such as feature additivity or image translation. We have reported two new property consistent with the affine structure of the latent space, specular reflections. We have also reported that while normalizing flows lose correlation structure, they exhibit great linear separability properties. We have also seen that although the optimizing flow should have zero sample mean, the current normalizing flow is biased and all images are mapped to one single hemisphere. This leaves some room for improvement. Nevertheless, most of the images from the source space are properly mapped on S^{N-1}. Performing different paths from $image_a$ to $image_b$ showed us that Glow in not perfectly defined in all S^{N-1} this is also some room for improvement. Finanly, in chapter 5 we developed some mathematical grounding of why should the latent space have and affine structure.

List of Figures

1 Image of Lena used in many image processing tasks, here for denoising[1]. 3

2 Block diagram corresponding to a layer of the Glow Architecture. (left) One step of the flow. (right) Full architecture. For additional details, see [11]. 7

3 Sample CelebA impages, illustrating 8 of the 40 annotated attributes. 9

4 Sample images before and after processing. 10

5 Comparison between the histograms of the norm multi-variate Gaussian distribution with zero mean and unit variance, and the norms of the learned latent representation. 12

6 Distribution of norms drawn from a normal distribution vs Distribution of norms drawn from the one generated by the normalizing flow. 13

7 Distribution of points drawn from a normal distribution projected against the line L and distribution of points drawn from the distribution defined by the normalizing flow. L is the line defined by *zero* and the average of the distribution defined by the normalizing flow. . 13

8 Linear transitions among two given images 14

9 Norm preserving transitions among two given images. The obtained images are nor as realistic as using the *straight* line path 15

10 Norms for Linear and Norm preserving paths 16

11 Average values for: Bald, Smiling, Mustache, Glasses 17

12 Average values for: Not-Bald, Not-Smiling, Not-Mustache, Not-Glasses 17

13 Male and Female prototypes 18

14 Smile transition, $\alpha = [0, 0.5, 1, 2, 3]$ 19
15 Mustache transition, $\alpha = [0, 0.5, 1, 2, 3]$ 19
16 Old transition, $\alpha = [0, 0.5, 1, 2, 3]$ 19
17 Results for specular imaging 20
18 Smiling Image Space 22
19 Smiling Glow Space 22
20 Eyeglasses Image Space 23
21 Eyeglasses Glow Space 23
22 Result of corrupting one single component 24
23 Result of corrupting many components 25
24 Failed smile transition 25

25 Adding a single person as a feature vs adding a smile 31
26 Resulting norms 31
27 Eyeglasses are a local concept but not unique 31
28 Mustache is a local and fairly unique concept 32
29 Not glasses is a unique well sampled concept 32
30 Blond, Black and Brown hair are well sampled classes 33
31 Representation of where Glow works 33

Bibliography

[1] Kingma, D. P. & Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions (2018). 1807.03039.

[2] Dinh, L., Sohl-Dickstein, J. & Bengio, S. Density estimation using real NVP **abs/1605.08803** (2016). URL http://arxiv.org/abs/1605.08803. 1605.08803.

[3] Liu, Z., Luo, P., Wang, X. & Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)* (2015).

[4] Blei, D. M., Ng, A. Y. & Jordan, M. I. Latent dirichlet allocation. *J. Mach. Learn. Res.* **3**, 993–1022 (2003). URL http://portal.acm.org/citation.cfm?id=944937.

[5] Klejdysz, J. *Shifts in ECB communication: a text mining approach.* Master's thesis (2018). URL http://hdl.handle.net/2105/44119.

[6] Bholat, D., Hansen, S., Santos, P. & Schonhardt-Bailey, C. Text mining for central banks. *SSRN* (2015). URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2624811.

[7] Goodfellow, I. J. *et al.* Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, 2672–2680 (MIT Press, Cambridge, MA, USA, 2014).

[8] Vincent, P., Larochelle, H., Bengio, Y. & Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, 1096–1103 (2008).

[9] Kingma, D. P. & Welling, M. Auto-encoding variational bayes (2014). 1312.6114.

[10] Dinh, L., Krueger, D. & Bengio, Y. Nice: Non-linear independent components estimation (2015). 1410.8516.

[11] Kingma, D. P. & Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, vol. 31, 10215–10224 (2018).

[12] Oord, A. V., Kalchbrenner, N. & Kavukcuoglu, K. Pixel recurrent neural networks. In Balcan, M. F. & Weinberger, K. Q. (eds.) *Proceedings of The 33rd International Conference on Machine Learning*, vol. 48 of *Proceedings of Machine Learning Research*, 1747–1756 (PMLR, New York, New York, USA, 2016). URL http://proceedings.mlr.press/v48/oord16.html. 1601.06759.

[13] Valenzuela, A., Segura, C., Diego, F. & Gómez, V. Expression transfer using flow-based generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (2021).

[14] Pennington, J., Socher, R. & Manning, C. D. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543 (2014). URL http://www.aclweb.org/anthology/D14-1162.