UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

IRON PRANDO DA SILVA

Normalizing Flows: A Study On Models' Coherence

Work presented in partial fulfillment
of the requirements for the degree of
Bachelor in Computer Science

Advisor: Prof. Dr. João Luiz Dihl Comba
Coadvisor: Prof. Dr. Mariana Recamonde
Mendoza

Porto Alegre
May 2022

ACKNOWLEDGEMENTS

My deep gratitude to my advisor, Professor João Luiz Dihl Comba and co-advisor, Professor Mariana Recamonde Mendoza. Prof. Comba introduced me to the world of research and has always been supportive of new ideas. Both provided insights and fruitful discussions that significantly impacted this work's direction and quality for the better. Thank you very much for your enthusiastic participation in this little project. I would not have been successful without your guidance. I would also like to extend my acknowledgements to the thesis committee members, Professors Claudio Rosito Jung and Thiago Lopes Trugillo da Silveira, who offered very pertinent perspectives and suggestions. I very much appreciate your shared experience.

My gratitude to the Universidade Federal do Rio Grande do Sul and the Instituto de Informática (INF) for enabling such a great education. The INF teaching staff is exceptionally competent and the institute is very well maintained, which certainly is not easy. My gratitude also to the Parque Computacional de Alto Desempenho (PCAD) and its attentive staff. Without the PCAD infrastructure, I do not know what would be of this work. My special thank you also to the Technische Universität Kaiserslautern (TUK) for providing me a second perspective on higher education. This work would much probably not have been conceptualized if it were not for your great Machine Learning courses. My deep appreciation for the exchange program that the TUK maintains with the INF. I am very privileged to have taken part in the program, and I would find it very refreshing to see more people being able to have such experiences. I really hope partnerships like this result in more collaborations between Brazil and Germany in the future.

To all my friends, I am a very lucky person to have you by my side, distant or not, and for all the time we shared, still share and will share together. I am much obliged to you all.

Dedico este trabalho à minha família: à minha mãe, ao meu pai, ao meu irmão, e à minha irmã. Vocês foram os primeiros pilares e efetivamente fundação do que hoje se apresenta. Levo vocês como parte da minha essência aonde for.

"Alternativa pra criança aprender basta quem ensina." - Sabotage

ABSTRACT

Normalizing Flows (NFs) have gathered significant attention from the academic community as a means of embedding a data distribution into a much simpler base distribution. The second belonging to a latent space with the same dimensionality as the data. The Machine Learning models' evolution in the last decades and their now viable industrial use have raised concerns regarding the explainability and maintainability of such models. For example, how private the data used to train such a model remains and how easy it is to modify this model such that it complies with the required data protection guidelines. NFs offer a statistically grounded framework that might help us with both: explainability and maintainability. In this work, the concept of NF coherence is informally presented together with evidence of a known but ignored gap between the learned embedding and the base distribution contained in the latent space. This gap significantly impairs the usage of the base distribution, and further hinders more complex models that could arise from NF-based ones. Guided by the concept of NF coherence, we will asses two adapted models based on the Glow model. Several questions are raised that, to the best of the author's knowledge, have not been considered in the literature. The potential existence of a non-unimodality metric that could improve future assessments of the quality of fit of NFs is also discussed.

Keywords: Normalizing Flows. Deep Generative Models. Statistical Machine Learning. Latent Space Interpretability.

Normalizing Flows: Um Estudo Sobre Coerência de Modelos

RESUMO

Normalizing Flows (NFs) atraíram atenção significativa da comunidade acadêmica como um meio de mergulhar uma distribuição de dados em uma distribuição de base muito mais simples. A segunda pertencendo a um espaço latente de mesma dimensionalidade dos dados. A evolução dos modelos de Aprendizado de Máquina nas últimas décadas e seu agora viável uso industrial trouxe preocupações a respeito da explicabilidade e manutenibilidade de tais modelos. Por exemplo, quão privados são mantidos os dados utilizados no treinamento de tal modelo e quão fácil é modificá-lo de modo a cumprir com as diretrizes de proteção de dados requeridas. NFs oferecem um framework fundado em estatística que talvez possa nos ajudar com ambos: explicabilidade e manutenibilidade. Neste trabalho, o conceito de coerência de NF é informalmente apresentado junto de evidências de uma conhecida, mas ignorada brecha entre o mergulho aprendido e a distribuição de base contida no espaço latente. Essa brecha restringe de modo significativo o uso da distribuição de base e, subsequentemente, prejudica modelos mais complexos que poderiam emergir dos modelos fundamentados em NFs. Guiado pelo conceito de coerência de NF, vamos analisar dois modelos baseados no modelo Glow. no conceito de coerência de NF, diversas questões são levantadas que, no melhor do conhecimento do autor, não foram consideradas na literatura. A potencial existência de uma métrica de não-unimodalidade que pode aprimorar futuras avaliações da qualidade de ajustamento de NFs também é discutida.

Palavras-chave: Normalizing Flows. Modelos Gerativos Profundos. Aprendizado de Máquina Estatístico. Interpretabilidade do Espaço Latente.

LIST OF FIGURES

LIST OF ABBREVIATIONS AND ACRONYMS

ML	Machine Learning
DNN	Deep Neural Network
GAN	Generative Adversarial Network
VAE	Variational Autoencoder
NF	Normalizing Flow
SVM	Support Vector Machine
OOD	Out-of-distribution
MLE	Maximum Likelihood Estimation
MAP	Maximum A Posteriori Estimation

LIST OF SYMBOLS

c	Scalar
$\mathbf{x}$	Column vector
$(\mathrm{w}_1, ..., \mathrm{w}_M)$	Row vector
$\mathbf{M}$	Matrix
$\cdot^T$	Matrix transpose
J	Jacobian matrix
$p(\cdot)$	Probability density function or probability mass function
U	Uniform distribution
$\mathcal{N}$	Normal distribution
Gam	Gamma distribution

CONTENTS

1 INTRODUCTION

This chapter presents an overview of how the statistical generative modeling approach and the Deep Neural Network approach complement each other (Section 1.1). Next, a brief motivation for studying Normalizing Flows (NFs) (Section 1.2), some of its known limitations and problems, and works related to this thesis are presented (Section 1.3). The aim, questions, and goals of this work (Section 1.4), as well as this work's limitations (Section 1.5) and contributions (Section 1.6) follow. The overall structure of this document is laid out last (Section 1.7).

1.1 The motivation for Deep Statistical Models

When we want to get to the level of dependable Artificial Intelligence or Machine Learning (ML) based software and cyber-physical systems, we must be able to trace back its decisions and make informed design choices when faced with requirement violations. We are not only interested in a model's capability of performing the tasks it models, but we want it to be auditable and maintainable. We want to be able, for example, to explain the behaviors behind a particular output or to protect confidential data that otherwise could be leaked by the model. In any case of deviation from the intended system function, we must be able to pinpoint the causes and determine the potentially best course of action with enough information to ensure traceability of the rationale behind the model's evolution.

Often, explainable ML models ground themselves in the statistical framework. Statistical models can be approached discriminatively or generatively (BISHOP, 2006). The discriminative approach models the conditional probability distribution of a random variable Y, conditioned on some observation $X = x$. The generative approach seeks to model the joint distribution of these variables, i.e., it also models the probabilities over the random variable X. Thus, in the generative approach, the model learns to approximate the random process behind the manifestations of Y and X together. Even though the statistical framework enables very intricate models (KOLLER; FRIEDMAN, 2009), relying on purely statistical methods can present time and spatial complexities that render models impractical.

With the increase of available computational resources for matrix and tensor operations, we have seen great advances in Deep Neural Network (DNN) -based models. DNNs have proved to learn highly complex functions and are often used for both re-

gression and classification tasks. DNNs' applicability remains in their ability to learn observations' features with little to no human intervention, greatly accelerating the whole process of modeling. But their power comes with a significant problem that often hinders its industrial and commercial applicability: a DNN's parameter space is often too complex to render models readily explainable. Added to that, the correlations implied by the learned subnets may be artefactual.

We have seen DNNs and statistical models being mixed in different proportions and structures in the literature. Examples include, among others, Bayesian DNNs (BNNs), Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and Normalizing Flows (NFs). BNNs provide a measure of uncertainty by using a second-order approximation of the learned parameters' distribution (AZEVEDO-FILHO; SHACHTER, 1994). GANs (GOODFELLOW et al., 2014) use a generator DNN and a discriminator DNN to learn how to generate samples from random noise. The generator receives a noise input, generates a data sample, and the discriminator tries to determine if it is synthetic or not. Generator and discriminator are then trained together until their abilities to fool and discern reach a draw. VAEs (KINGMA; WELLING, 2013) make use of an encoder-decoder architecture, comprised of one DNN each, to model a parameterized distribution in an implicit latent space. Its training is based on minimizing the reconstruction loss of the inputs. NFs (DINH; KRUEGER; BENGIO, 2015) apply DNNs to model components of a bijective transform between data space and a base probability distribution space. The VAE's training entails learning a parametric change of variable to a base distribution contained in a latent space, such that the likelihood of the embedded data is maximized. Each of the aforementioned models provides different trade-offs between applicability, generalization power, complexity, interpretability, and maintainability.

1.2 Normalizing Flows

Although we have seen advances regarding the interpretability of the latent space of GANs and VAEs, their latent space is defined by the activation patterns of its composing DNNs. Consequently, the significance of its latent space observations is directly tied to such activation patterns that are themselves products of an arbitrary transformation. The great capacity of these methods to generate high-quality samples comes with a severe impairment: the latent space comprises more than just the observation itself. The transformation often comes with an implicit dimensionality reduction or increase, possibly over

a number of layers.

NFs have seen a steep increase in interest since its popularization through the Non-linear Independent Component Estimation (NICE) model (DINH; KRUEGER; BENGIO, 2015). NFs define a one-to-one embedding transformation between a given continuous data space and a probability distribution space. Effectively, A NF-based model learns a parametric distribution by means of a change of variable between the data space and a base probability distribution in the latent space. By maximizing the likelihood of the data under its parametric distribution, the embedding is limited to regions of high probability in the base distribution. Furthermore, since the transformation is bijective, each embedded observation has a unique representation in both spaces. Added to that, we are able to choose any continuous probability distribution in the latent space. Usually, the choice favors those probability distributions with a closed analytical form and can thus be efficiently computed. NF-based models offer a rich framework for arbitrary continuous distribution modeling due to its sound theoretical roots in statistics. The base distribution is a proper probability distribution, and thus, we may be able to use known results from statistics, such as distribution conditioning, variable marginalization, etc. Or so we would like.

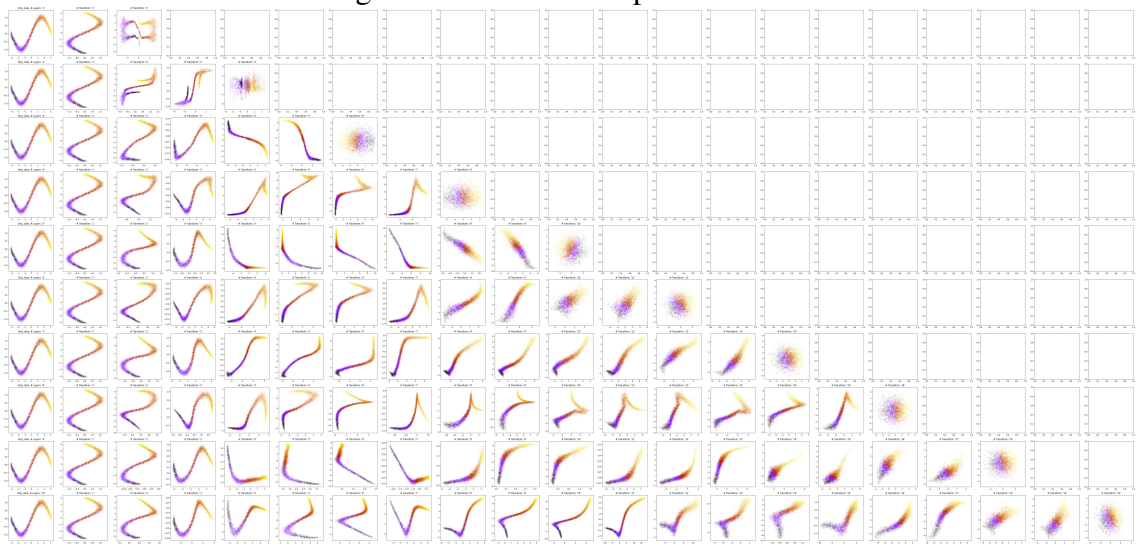
1.3 Known limitations and problems of NFs and related work

Even though NFs are an advancement in transformer-based generative models' interpretability, NF-based models may present drawbacks. For instance, Kirichenko, Izmailov and Wilson (2020) have presented that the Glow model (KINGMA; DHARIWAL, 2018) learns probability distributions over local graphical patterns in opposition to more global semantics. In their work, the authors argued about the tendency of Glow to infer high probabilities to out-of-distribution (OOD) data, rendering them unusable for, e.g., face detection.

Such NF-based models may also suffer from problems arising from an ill-defined parameterized transformation or an architecture not capable of effectively transforming data observations to the base distribution space. In this case, we may end up with an embedding where the codomain is constrained by the base distribution but with distinct properties such as skewness and number of modes. Figure 1.1 presents how ten different NFs with a different number of transform compositions and the same standard Gaussian base distribution behave when presented with the same data set. Each row represents a

single model with a distinct number of layers. The first column is the original data. Each subsequent column depicts the result data after the application of each transform. The transforms interleave an axis permutation and an affine transform. For example, if we look at the image's first row, we have the original data in the first column, the result after applying the axis permutation in the second column, and the final output after applying the affine transform. Notice, looking at the last plot of each line, that for NFs that are not powerful enough, we end with embedded data that is not well modeled by the standard Gaussian base distribution. Furthermore, the gaps in the embedded data distribution within high probability regions of the base distribution might result in unexpected generated samples. In the context of image generation, this is the same of recovering incoherent images from high probability regions of the base distribution.

Figure 1.1: Inside a simple NF model



The usage of distinct models' latent spaces has seen some recent advances. Traditionally, vector arithmetic has been used to demonstrate the generative capabilities of models and are present in many works, for instance, Real NVP (DINH; SOHL-DICKSTEIN; BENGIO, 2016) and Glow (KINGMA; DHARIWAL, 2018). In the context of GANs, the work of Shen et al. (2019) presents a Support Vector Machine classifier approach to learn the directions of each attribute's manifestation in the latent space. Valenzuela et al. (2021) present a facial expression transfer procedure using the Glow model. Their goal was to manipulate a target image such that it presents the same expression from a source image while maintaining the face's identity. The average latent vector of multiple images of a single individual's identity is calculated. These are then used to neutralize the source identity, leaving a latent representation of the expression. This latent expression vector is

then linearly combined with the target image.

Although the advances regarding the usage of the embedded data, studies focusing on how well the base distribution fits the embedded data distribution are unsatisfyingly dim. Although my research on NFs is very small, only one work was found that discussed how well the embedded data distribution and the base distribution relate. In the work of Funes (2021), the embedding of the Glow model is briefly assessed. His work suggests that the expected distribution of the model's embedding deviated from the base distribution. This may be the effect of a limited embedding transformation (See Figure 1.1), bias in the data used for the assessment, or both. It should be emphasized that this was the only work found that attempts to determine how the embedded data distribution deviates from the base distribution.

During the writing of this thesis, although not covered here, no other work was found that used two NF models with distinct data domains or learning goals in an attempt to use the more tractable functional form of the base distribution to compose joint distributions. This might be a novelty idea in the context of NFs, and it may have been hindered by the lack of proper attention to the quality of fit between the embedded data and the base distributions.

1.4 Research goals and significance

The focus of this work is to explore the coherence of an adaption of the NF-based Glow model (KINGMA; DHARIWAL, 2018) in the context of facial synthesis. Coherence, in this context, is a combination of two properties. Property one is: given a high probability region of the embedded data distribution, regardless of the base distribution, we recover coherently structured data. This is the main property that is explored in the NFs' literature for the generative capabilities it entails. Property two is: given a high probability sub-region of the base distribution in the latent space, the NF transform must recover coherently structured sample data. This would mean that the base distribution adequately approximates the embedded data distribution and thus offers a proxy for the data distribution in the latent space. Given the importance of these properties and the potential implications behind the coherence concept introduced by this work, it has earned its own section: Section 2.5.

One example of the potential benefits of having a closed-form proxy base distribution, more specifically a Gaussian one, lies in the efficient inference and in the mainte-

nance of the functional form under more complex compositions. Probabilistic Graphical Models (KOLLER; FRIEDMAN, 2009) offer a very rich and compelling framework for the modeling of joint distributions. A probabilistic graphical model encodes conditional independencies of a potentially broader probability distribution's random variables by means of network graphs, where each node corresponds to a single random variable and edges indicate dependencies. By having analytically tractable proxy distributions in the latent space for distinct random variables, we may define more complex models that learn and execute inferences in the latent space much more efficiently than we would have with non-analytically tractable ones.

Provided that we already have evidence of the first property of NF coherence's holding, at least to some extent, the goal of this work is to briefly assess my Glow adaption from the perspective of this second property of NF coherence (Section 2.5). This is done by means of the inspected structural coherence of sampled data and their assigned probabilities in the data and in the base distribution spaces.

1.5 Disclaimer and research limitations

Resources are always a very significant potential limitation of any work. This one is no different. The overall topic of this bachelor thesis was conceptualized, developed, and written in approximately three months of part-time work. Added to such time constraints is the lack of prior knowledge or experience of the author with respect to NFs. Needless to say, the direction to which take this thesis has changed a number of times until it converged into its final form. In this time window, the author has collected information on the subject and digested it, found the potential problem in which this work focuses: that the distribution of the embedded data deviates from the base distribution; justified why this may hinder progress with such models and its potential applications; and realized experiments enough to, finally, write this bachelor thesis.

Another gap that limits significantly the usability of this work is that the Glow model was developed in-house by use of a third-party library, and some adaptations of the model were required. This breaks the flow of composable research, and the findings may be limited by the quality of the model at hand. This choice was taken based more on my own interest to learn about the *nflows* library (DURKAN et al., 2020) and the potential future work it might enable. To mitigate this problem, the code used in this work can

be accessed in GitHub[1]. Furthermore, the overall scope of the literature covered by this work is relatively narrow when compared with the volume of research available, and the new models that were proposed after Glow (2018) are not accounted for. Thus, results may be too specific to this work and may not be valid for most recent NF-based models or for other data domains. More experiments are required to validate this work's overall idea with different models and data.

This thesis is very small for the magnitude of the problem. This work does not answer a number of very significant questions about the topic: is the goal of having an embedding properly modeled by the base distribution achievable? If so, to what extent? How does it relate to the dimensionality of the data? Is there a minimum number of samples for us to achieve such a model? Also, does it affect the quality of the generative process? And does it improve the interpretability and further usage of the embedding in any way? These and other questions are laid out in Section 6.2.

1.6 Contributions

To the best of the author's very limited knowledge:

- This is the first work to discuss the importance of a good fit between the base distribution and the embedded data and why it may be hindering further applications that build from NFs.
- This is the first work to explicitly define the properties that compose the possibly new NF coherence concept.
- This is the first work that suggests the use of a non-unimodality metric as a means for the quality of fit assessment between the embedded data distribution and the base distribution of NFs.
- This is also the first work that has reportedly used Support Vector Machine (SVM) linear classification in an attempt to determine a direction of attribute manifestation in the latent space of NFs.
- This is the first work that attempts to approximate the embedded data's individual class distributions from a Bayesian perspective with the goal of assessing the overall embedded data distribution. This is also the first work to show we can bias our

[1]The code used in this work is available in GitHub: <https://github.com/IronPS/NF-coherence_BScThesis>.

samples by using such distributions.

A number of open questions pertaining to NF coherence and related future work not considered in the NFs' literature assessed are presented in Section 6.2.

1.7 Document structure

This work is structured as follows. Chapter 2 introduces NFs, the NF-based model in which this work grounds itself, and the properties behind the concept of NF coherence. The data set used, the model, and training specifications, as well as the training results, are contained in Chapter 3. The description of the experiments executed in this thesis and their results are respectively in Chapters 4 and 5. A discussion about possible improvements, questions related to NF coherence, and other tangent questions are presented in Chapter 6. Finally, Chapter 7 contains the concluding remarks.

2 NORMALIZING FLOWS

This chapter is designed to present the necessary concepts about Normalizing Flows (NFs) and the related work that built up to the Glow model (KINGMA; DHARIWAL, 2018), on which this work is based.

NFs are a family of methods that provide an easy-to-model representation of a provided data set's distribution. NF-based models aim to model the generative process of the data by learning the variables' joint distribution through a parameterized invertible and differentiable transformation guided by a much more easily tractable base distribution. Its learning process is unsupervised, and the learned distribution could be used for both sampling and inferring information about data.

This chapter proceeds as indicated. Section 2.1 presents a brief overview of the previous work that preceded the Glow model. Section 2.2 lays down the definition of a NF. Its functional form, as well as other relevant information with respect to the transforms that compose the NF are discussed in Section 2.3. The Glow model and the properties of NF coherence are introduced in Sections 2.4 and 2.5, respectively.

2.1 Related work

The normalizing flow's framework was first proposed in the work of Tabak and Turner (2013) and popularized by Dinh, Krueger and Bengio (2015), where its potential was demonstrated over image data sets to problems of image generation and image inpainting. Since then, multiple works have extended the proposed model. In the work of Dinh, Sohl-Dickstein and Bengio (2016), the model was generalized to work with non-volume preserving transformations, enabling a much larger class of parameterized distributions. To reduce the requirements of human intervention and model architecture manipulation, Kingma and Dhariwal (2018) designed a 1x1 invertible convolution layer that, instead of being fixed, is learned during the training process and presented the Glow model (KINGMA; DHARIWAL, 2018). Since then, the NFs framework has been applied to a myriad of problems: image and video generation, noise modeling, computer graphics, physics, and more (KOBYZEV; PRINCE; BRUBAKER, 2021).

2.2 Definition

Let $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_D\}$ be a dataset comprised of D vectors $\mathbf{x}_d \in \mathbb{R}^M$. A Normalizing Flow maps a vector $\mathbf{x}$ to another vector $\mathbf{u} \in \mathbb{R}^M$, with $\mathbf{u} \sim p_u(\mathbf{u})$, by means of an invertible and differentiable transformation $\mathbf{T}$, such that

$$\mathbf{x} = \mathbf{T}(\mathbf{u}) \tag{2.1}$$

The distribution $p_u(\mathbf{u})$ is deemed base distribution of the flow-based model (PAPAMAKARIOS et al., 2019). It is usually selected by its desirable properties, e.g., a standard Gaussian distribution for its analytical closed-form.

Given that $\mathbf{T}$ is invertible and both $\mathbf{T}$ and $\mathbf{T}^{-1}$ are differentiable, the densities $p_x(\mathbf{x}_d)$ are well defined and can be determined by the change of variables formula:

$$\begin{aligned} p_x(\mathbf{x}) &= p_u(\mathbf{T}^{-1}(\mathbf{u}))\,|\det J_{\mathbf{T}^{-1}}(\mathbf{x})| \\ &= p_u(\mathbf{u})\,|\det J_{\mathbf{T}}(\mathbf{u})|^{-1} \end{aligned} \tag{2.2}$$

where $J_{\mathbf{T}}$ denotes the Jacobian of the transformation $\mathbf{T}$ and has the form

$$J_{\mathbf{T}}(\mathbf{u}) = \begin{bmatrix} \frac{\partial T_1}{\partial \mathrm{u}_1} & \cdots & \frac{\partial T_1}{\partial \mathrm{u}_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial T_M}{\partial \mathrm{u}_1} & \cdots & \frac{\partial T_M}{\partial \mathrm{u}_M} \end{bmatrix} \tag{2.3}$$

Furthermore, a sequence of transformations $\mathbf{T}_1, ..., \mathbf{T}_N$ that are all invertible and differentiable is also composable, i.e., we can build a transformation $T = \mathbf{T}_1 \circ \mathbf{T}_2 \circ ... \circ \mathbf{T}_N$ and it will also be invertible and differentiable. Its inverse follows the form

$$\mathbf{T}^{-1} = \mathbf{T}_N^{-1} \circ \mathbf{T}_{N-1}^{-1} ... \circ \mathbf{T}_1^{-1} \tag{2.4}$$

and its Jacobian determinant is given by the chain rule

$$\det J_{\mathbf{T}_N \circ ... \circ \mathbf{T}_1}(\mathbf{u}) = \det J_{\mathbf{T}_N}\left(\mathbf{T}_{N-1} \circ ... \circ \mathbf{T}_1(\mathbf{u})\right) \cdot ... \cdot \det J_{\mathbf{T}_1}(\mathbf{u}) \tag{2.5}$$

Following these properties, we can define a transformation $\mathbf{T}$ composed of multiple transformations $\mathbf{T}_i$, each parameterized by a set of parameters Ψ_i. We can specify this function to be as simple or complex as we need. Furthermore, we may let the base

distribution be defined by a set of parameters θ. Let $\Omega = \{\Psi, \theta\}$ be the parameters that guide the model. We can then use maximum likelihood estimation to find the parameters that maximize the data set log-likelihood as follows:

$$\begin{aligned}
\Omega_{MLE} &= \underset{\Omega}{\operatorname{argmax}}\, \mathcal{L}(\mathcal{D}|\Omega) \\
&= \underset{\Omega}{\operatorname{argmax}}\, \frac{1}{D}\sum_{d=1}^{D} \ln p_x(\mathbf{x}_d|\Omega) \\
&= \underset{\Omega}{\operatorname{argmax}}\, \frac{1}{D}\sum_{d=1}^{D} \left[\ln p_u(\mathbf{T}^{-1}(\mathbf{x}_d|\Psi)|\theta) + \ln \left|\det J_{\mathbf{T}^{-1}}(\mathbf{T}^{-1}(\mathbf{x}_d|\Psi))\right|\right] \\
&= \underset{\Omega}{\operatorname{argmin}} -\frac{1}{D}\sum_{d=1}^{D} \left[\ln p_u(\mathbf{T}^{-1}(\mathbf{x}_d|\Psi)|\theta) + \ln \left|\det J_{\mathbf{T}^{-1}}(\mathbf{T}^{-1}(\mathbf{x}_d|\Psi))\right|\right]
\end{aligned} \tag{2.6}$$

Using the above, we can train our model by means of gradient descent.

Overall, since evaluating the above log-likelihood involves computing the transform and its Jacobian determinant, we are interested in a sequence of transformations that are easy to invert and with a Jacobian determinant that is easy to evaluate. Different types of transformations that observe these properties have been proposed in the literature. Some of the proposed transforms are described in Section 2.3. After training, we may use the base distribution and the identity given by equation (2.1) to sample from the model. In case we want to evaluate the probability of a new observation in the data space, we may use equation (2.2).

It is important to note that other optimization methods have been proposed for NF training. For instance, Rezende and Mohamed (2015) present a variational inference method of learning by defining a variational distribution $q(\mathbf{u}) \approx p(\mathbf{u}|\mathbf{x})$ and by maximizing its Evidence Lower Bound through the coordinate ascent. Papamakarios et al. (2019) present two KL-divergence approaches to train the flow in both directions since, depending on the application, we may be interested in exchanging one direction's transform tractability for sampling or density estimation capability. The optimization criterion described in this thesis is the same as the forward KL-divergence and maximum likelihood estimation described in Section 2.3.1 of Papamakarios et al. (2019).

2.3 Functional form

From the previous section, we have seen that the transformations used to compose the model architecture will determine how efficient its learning, sampling, and inference will be. We are generally interested in easy to invert transformations with an easily tractable Jacobian determinant. Furthermore, the functional form of such transforms will affect how general the model can become. Consequently, affecting the quality of the approximation learned by the flow-based model given the complexity of the data distribution.

2.3.1 Coupling layer

Coupling layers are bijective transforms with a general functional form that enforces the Jacobian matrix to be lower triangular. Lower triangular matrices have straightforward determinant computations, simply given by the product of its diagonal elements. A general coupling layer was proposed in (DINH; KRUEGER; BENGIO, 2015). Intuitively, a given observation is partitioned such that one partition is directly fed into the next layer, whereas the other partition is transformed by a function, deemed by Dinh, Krueger and Bengio (2015) as the *coupling law*. Such a functional form effectively makes the Jacobian lower triangular.

More formally, the general coupling layer is an invertible function $f : \mathbb{R}^M \rightarrow \mathbb{R}^M$, differentiable almost everywhere, and defined over a fixed vector partition $\mathbf{x} = (\mathbf{x}_{1:m}, \mathbf{x}_{m+1:M})^T$, such that its codomain is a partitioned vector $\mathbf{y}$ guided by the following equations:

$$\begin{aligned} \mathbf{y}_{1:m} &= \mathbf{x}_{1:m} \\ \mathbf{y}_{m+1:M} &= g\left(\mathbf{x}_{m+1:M}, h(\mathbf{x}_{1:m})\right) \end{aligned} \tag{2.7}$$

where $g : \mathbb{R}^{M-m} \times \mathbb{R}^m \rightarrow \mathbb{R}^{M-m}$ is the coupling law and h is a function with the appropriate domain and codomain dimensionality called *coupling function*. Notice that the function g is required to be a differentiable bijection, whereas this requirement can be waived for h. The coupling layer inversion is then defined as

$$\begin{aligned} \mathbf{x}_{1:m} &= \mathbf{y}_{1:m} \\ \mathbf{x}_{m+1:M} &= g^{-1}\left(\mathbf{y}_{m+1:M}, h(\mathbf{x}_{1:m})\right) \end{aligned} \tag{2.8}$$

From the Equations 2.7 and 2.8, we see that the Jacobian is the matrix

$$J_f = \frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} I_m & 0 \\ \frac{\partial \mathbf{y}_{m+1:M}}{\partial \mathbf{x}_{1:m}} & \frac{\partial \mathbf{y}_{m+1:M}}{\partial \mathbf{x}_{m+1:M}} \end{bmatrix} \tag{2.9}$$

where I_m is the $m \times m$ identity matrix. Given its lower triangular form, its determinant can be calculated in linear time as

$$\det J_f = \prod_{i=1}^{M} J_{f\,ii} \tag{2.10}$$

Given the functional form of the coupling layer, it is not necessary to evaluate the derivatives of the coupling function h with respect to the input vector $\mathbf{x}$ when calculating the Jacobian determinant. As a consequence, the coupling function can be as complex as one may need. Added to the fact that its inverse is not required, we can choose such a function independently of how difficult it may be to invert.

2.3.2 Input vector partitioning

From the nature of the coupling layer, presented in Equation 2.7, of its Jacobian determinant (2.9) and the chain rule presented in Equation 2.5, we can observe that at least three layers are required such that all dimensions influence all others when alternating the partitions after each layer. During the optimization process, the first layer inflicts influences of the partitioned vector $\mathbf{x}_{m+1:M}$ onto itself. The second layer inflicts influences of the partition $\mathbf{x}_{1:m}$ onto itself. And finally, the third layer encompasses the influences of one partition over the other.

In the work of Dinh, Krueger and Bengio (2015), the partitions are simply alternated after each layer. Within the context of images, a posterior work proposes a manually defined spatial checkerboard pattern mask to exploit local correlation structures (DINH; SOHL-DICKSTEIN; BENGIO, 2016). In their work, Dinh, Sohl-Dickstein and Bengio (2016) propose a recursive multi-scale treatment of the channels, where each alternation of the vector partitions outputs two times the number of channels, each with halved spatial dimensions. The effect of such treatment is that, as we go from the image domain to the base distribution domain, the variables of the base distribution vector corresponding to the later transformations become more and more global. I.e., they capture characteristics

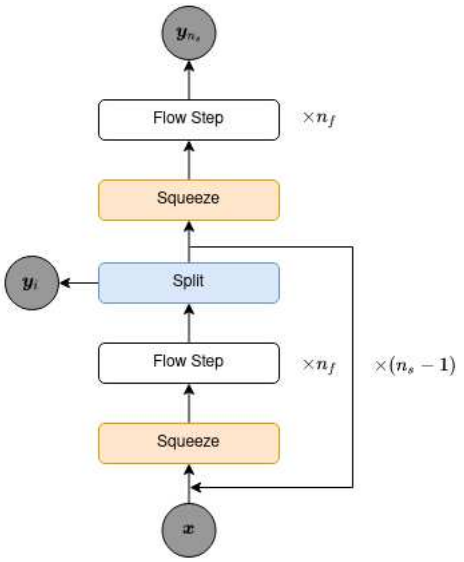
of a greater area of the image, in opposition to smaller local patches.

Envisioning the reduction of the requirements of human manipulation of features and further generalization of the previous models, the work of Kingma and Dhariwal (2018) presents a learnable invertible and differentiable 1×1 convolution that acts as a permutation of channels. Such 1×1 convolutions are typically used in neural network architectures before another, more expensive, $n \times n$ convolution as a means of reducing the number of channels that the filter must operate on. The convolution will operate over all channels of a single pixel and output another single pixel with the same or a different number of channels. Kingma and Dhariwal (2018) make use of a 1×1 convolution with equal input and output dimensions to generalize the permutation operation.

2.4 The Glow model

This thesis is based on an adaptation of the Glow model proposed by Kingma and Dhariwal (2018). Glow is a multi-scale model composed of n_s scales. Each scale performs a channel squeezing operation on its input and feeds it to a transformation composed of n_{f} flow steps, described in Section 2.4.2. The result is then split in half, where one part is output as a vector in the base distribution space, and the other part is forwarded to the next scale. Figure 2.1 presents an overview of the multi-scale architecture adopted by Kingma and Dhariwal (2018).

Figure 2.1: The multi-scale architecture defined in (DINH; SOHL-DICKSTEIN; BENGIO, 2016).



Adapted from (KINGMA; DHARIWAL, 2018)

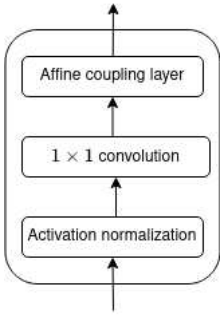
2.4.1 Squeeze operation

The squeeze operation performs an augmentation of the number of channels in an exchange for spatial dimensions. Its input is a tensor of shape $s \times s \times c$, where s is the spatial dimension and c is the number of channels. The output is a tensor of shape $\frac{s}{a} \times \frac{s}{a} \times 2a \cdot c$, where a is the exchange factor. The exchange factor used here is set to 2 as in the Glow model. The squeezing operation enables the model to capture local inter-channel correlations in subsequent layers.

2.4.2 Flow step

The flow step proposed by Kingma and Dhariwal (2018) is comprised of three transforms: an activation normalization, a 1×1 learnable convolution, and an affine coupling layer. Figure 2.2 presents the overall structure of the step.

Figure 2.2: One step of the flow



Adapted from (KINGMA; DHARIWAL, 2018)

The goal of the activation normalization is to approximately centralize and further limit the scale of the activation of each input channel. This is done by initializing the parameters of the normalization using a single batch of input data. Each channel will be centralized to the mean and standard deviation of the activation of the first batch. Thus, instead of recalculating these parameters for each data batch, as in batch normalization, this is done only once. Such an approach reduces computational effort and the noise added by batch normalization to the activation, which is inversely proportional to batch size. For example, in batch normalization, a batch size of 1 will incur greater activation variations than a much larger batch size. The latter tends to smooth the activation's standard deviation.

As described in Section 2.3.2, the parameterized 1×1 convolution learns a permutation of the input vector, such that the learned correlations better fit our maximum likelihood method and do not depend on fixed, manually devised vector partitioning. The resulting partitions are fed to the coupling layer. The coupling layer is what provides us with the functional form that will determine the generalization capacity of our model as well as the ability to efficiently compute the transform between the spaces of the distributions $p_x(\boldsymbol{x})$ and $p_u(\boldsymbol{u})$ (Section 2.3.1).

2.5 Normalizing Flow Coherence

Given the nature of this work, it is necessary to lay down, even if informally, two very important properties of what this work deems a coherent Normalizing Flow:

1. Given a high probability region of the embedded data distribution, the NF transform must recover coherently structured sample data.
2. Given a high probability sub-region of the base distribution in the latent space, the NF transform must recover coherently structured sample data.

The first property implies that the embedding conceals the data into sub-regions of the latent space where the embedding is itself well defined. Whereas the second property defines a much stronger condition and is the equivalent of saying that the base distribution, by means of the NF transform, model a region of high probability in the data space. If we can guarantee both these properties (or bounds on them) we have a coherent (partially coherent) NF model: the base distribution, the NF transform and the modeled data distribution are coherently related (within the given bounds). This could enable future works to use the base distribution as a proxy for the data distribution in the latent space.

3 MODELING PROCESS

This chapter presents the necessary information about the domain being considered and the final models used throughout this document. Section 3.1 presents an overview of the data set used and the pre-processing adopted in this work. The models' architectures are delineated in Section 3.2. The training procedure and immediate results are then discussed in Sections 3.3 and 3.4, respectively.

3.1 Data set

CelebA (LIU et al., 2015) is a data set of annotated celebrities' facial pictures. It has become a de-facto standard for NF-based models facial image generation capabilities and has been consistently used in studies that aim to model and manipulate such images. The Liu et al. (2015) report the following numbers:

- 10,177 number of identities.
- 202,599 number of face images.
- 5 landmark locations.
- 40 binary attributes annotations per image.

Sample images are presented in Figure 3.1. The proportions of attributes to data used in this work are depicted in Figures 3.3 and 3.4. This and other information pertaining to the model's training can be found in Section 3.3.

Figure 3.1: CelebA overview

Source: Liu et al. (2015)

3.1.1 Data pre-processing

The pre-processing is similar to the one described by Dinh, Sohl-Dickstein and Bengio (2016). The aligned and cropped data from the CelebA data set is used. The images have a resolution of 178×218 pixels, from which a central square area of size 148×148 pixels are cut. A resizing is then applied to result in 128×128 images, in opposition to the higher resolution 256×256 images from the CelebA HQ data set (KARRAS et al., 2017) used in the original work of Glow (KINGMA; DHARIWAL, 2018). This choice has one purpose and a consequence. Its purpose is to reduce the resource requirements for learning the models. The consequence is that this required one less layer in the model (Section 3.2).

3.1.1.1 Data dequantization

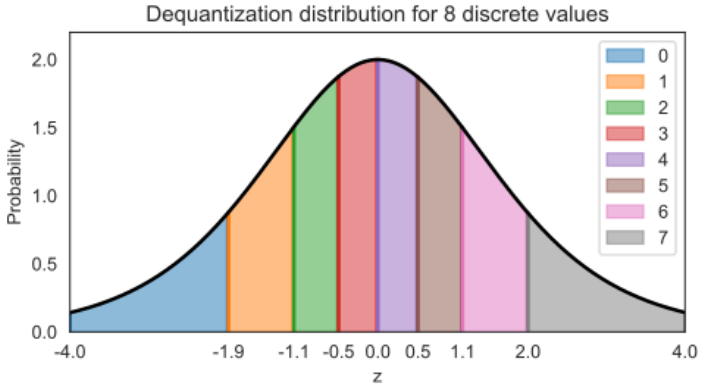
Since values of each of the images' channels are discrete, a uniform dequantization adapted from a lecture of a series authored by Lippe (2021) is used. The dequantization operation adds to the discrete values a uniform random noise in the interval $[0, 1)$. We

then have likelihoods over contiguous intervals given by

$$p(\mathbf{x}) = \mathbb{E}_{\mathbf{u} \sim U(0,1)^M} \left[p(\mathbf{x} + \mathbf{u}) \right] \tag{3.1}$$

These are then scaled according to the number of quanta, in our case 256. The resulting values, in the interval $[0, 1)$, are passed to a continuous and monotonically increasing function defined in the open interval $(0, 1)$, the logit function. The logit function effectively encodes each of the quantum's intervals to another corresponding interval in its codomain. If we invert the logit function we end up with a sigmoid, which we can interpret as a cumulative distribution function (CDF). By derivation of the sigmoid, we end up with a bell-shaped function, where domain intervals correspond to single quanta with probability masses encoded by each of their corresponding areas below the curve. Figure 3.2 (LIPPE, 2021) depicts this idea for 8 quanta. Each colored area corresponds to a single quantum's probability mass. Since the quanta are uniformly distributed, all the areas are equal.

Figure 3.2: Dequantization result for 8 quanta



Source: (LIPPE, 2021)

3.2 Model specification

Two models that follow the Glow architecture were used. Both were implemented using the same parameters, but with distinct base distributions. For ease of reference, let $\mathcal{M}_{\text{std}}$ and $\mathcal{M}_{\text{diag}}$ denote the models. Model $\mathcal{M}_{\text{std}}$ uses a fixed standard multivariate Gaussian, whereas model $\mathcal{M}_{\text{diag}}$ uses a parameterized multivariate diagonal Gaussian that learns to better fit the embedded data distribution. Both models were implemented using nflows (DURKAN et al., 2020), a PyTorch library with many ready-to-use functionalities

for the definition, training, and inference with NFs.

Since the data that is being input to the model has dimensions $128 \times 128 \times 3$, instead of $256 \times 256 \times 3$ as in the original Glow publication, the number of scales was reduced from six to five. Similarly, the number of neurons of the convolutional DNN that takes part in the affine coupling layer was reduced from 512 to 256.

3.3 Training procedure

For the training, $30,000$ samples from the CelebA training partition were used. These were mirrored, resulting in a total of $60,000$ images. Figures 3.3 and 3.4 respectively depict the distribution of attribute labels of the subsets of the train and the test data used in this work.

The models were trained for 42 epochs by minimizing the training data negative log-likelihood. The transform parameters are regularized following an L2 regularization using decoupled weight regularization (LOSHCHILOV; HUTTER, 2017) by means of its PyTorch implementation AdamW, with weight decay parameter set to 0.1. Batches of 16 samples were used. The initial learning rate was set to 0.0005 and was reduced by a factor of 0.8 every N_c checkpoint with no further decrease in the loss function. A checkpoint occurs every 8192 samples or 512 batches. N_c is initially set to 24 and is increased by a factor of 1.3 every time the learning rate is decreased.

Figure 3.3: Attribute distribution of the train data subset

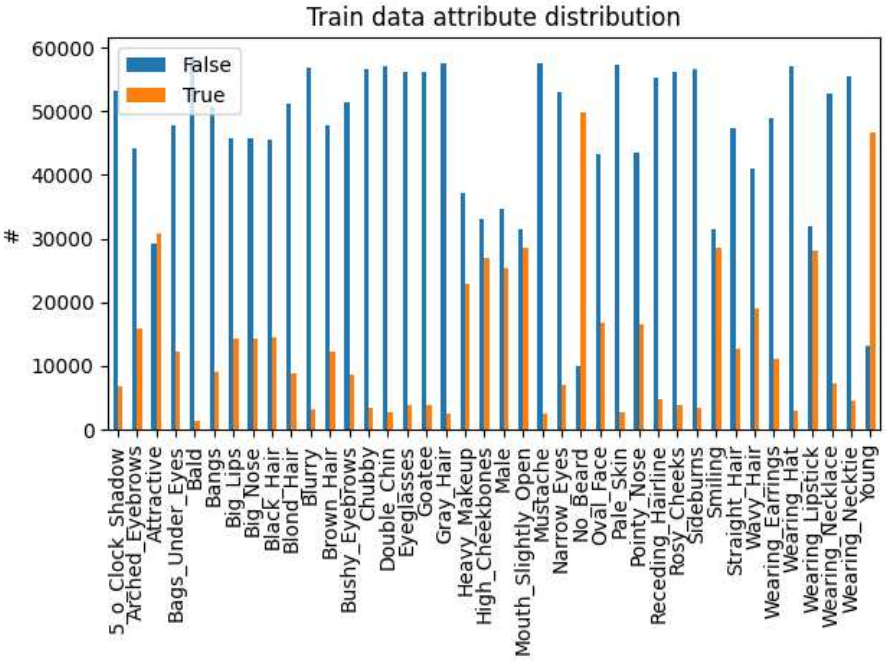
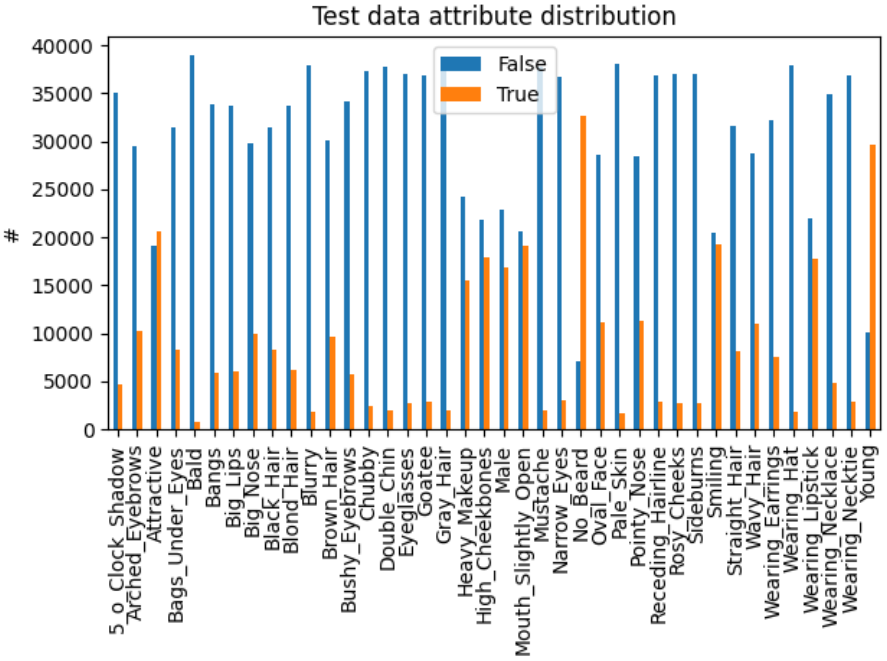


Figure 3.4: Attribute distribution of the test data



The train and test mean negative log-likelihood losses of both models are depicted in Figures 3.5 and 3.6. It is important to note that the presented values were calculated using a single batch of data to reduce the computational effort of its evaluation. Furthermore, the test data used is not fixed. That is, at each evaluation of the test data loss a new batch of test data is selected.

Figure 3.5: $\mathcal{M}_{\text{std}}$ Training Mean Negative Log-likelihood

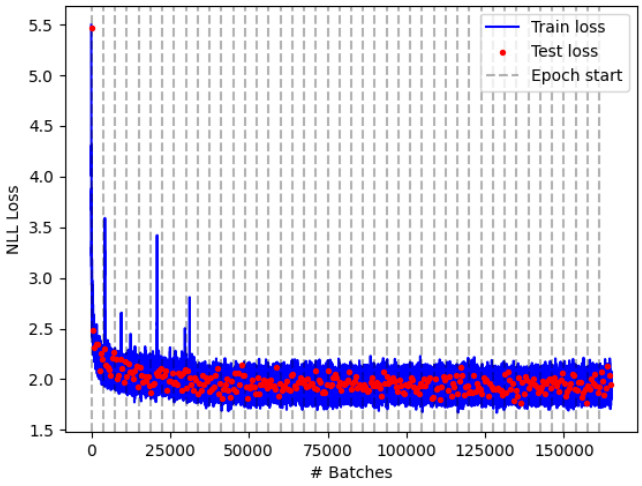
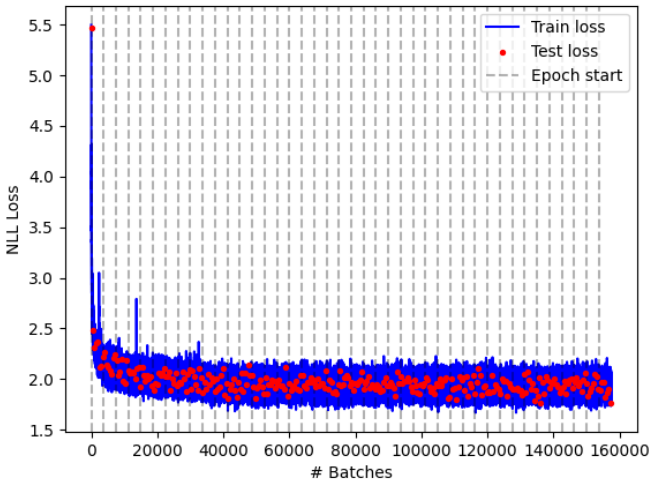


Figure 3.6: $\mathcal{M}_{\text{diag}}$ Training Mean Negative Log-likelihood



3.4 Mean images and models' samples

Let's discuss what we can readily get from the models. First, the mean images of each of the models are depicted in Figures 3.7 and 3.8. Below each image there are two pieces of information: $\ln p(x)$ refers to the log-likelihood of the data sample according to the learned data distribution, whereas $\ln p(u)$ refers to the log-likelihood in the latent space according to the base distribution. Notice that the parameterized diagonal base dis-

tribution of model $\mathcal{M}_{\text{diag}}$ attributes a higher probability to its mean than in the standard Gaussian case. Given that a Gaussian has a single mode, we can conclude that the base distribution of $\mathcal{M}_{\text{diag}}$ has lower variance values than the base distribution of $\mathcal{M}_{\text{std}}$ and, being parameterized, it has reduced its variance in an attempt to approximate the embedded data. Added to that, this seems to also reflect in the image likelihood in the data space. Notice also, that the $\mathcal{M}_{\text{std}}$ base distribution seems more biased towards women or longer hair than the $\mathcal{M}_{\text{diag}}$ model. Following the discussion presented with Figure 1.1, this can be merely artefactual. Even though it sounds reasonable to assume that higher probability data will be embedded towards the base distribution mode, there is little evidence that this is indeed the case.

These mean images' log probabilities will prove useful in our next attempts to comparatively infer how close or distant to the mode the samples of each model are.

Figure 3.7: $\mathcal{M}_{\text{std}}$ base distribution mean image

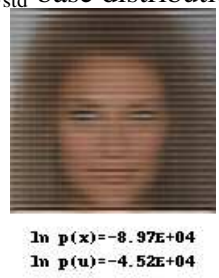


Figure 3.8: $\mathcal{M}_{\text{diag}}$ base distribution mean image

We might also sample images to inspect their structural qualities and try to relate them to the quality of the model. This also proves useful from the perspective of comparing the quality of sample generation with the original Glow publication. Figures 3.9 and 3.10 present ten generated samples for each of the models. Here, a tempered sampling was used with temperature $\tau = 0.7$, that is, the modified base distribution with only seventy percent of its standard deviation. Although the images present some overall coherent structure of a human face, it is uncertain why the quality of details is inferior to the

original. Figure 3.11 presents 24 samples of the Glow model. The observable diminished quality might happen for a number of reasons, such as the much-reduced training data set used, the reduced image resolution, the decrease in the number of scales, a less fine-tuned model and much possibly, a combination of these.

Figure 3.9: $\mathcal{M}_{\text{std}}$ Tempered Samples with $\tau = 0.7$

Figure 3.10: $\mathcal{M}_{\text{diag}}$ Tempered Samples with $\tau = 0.7$

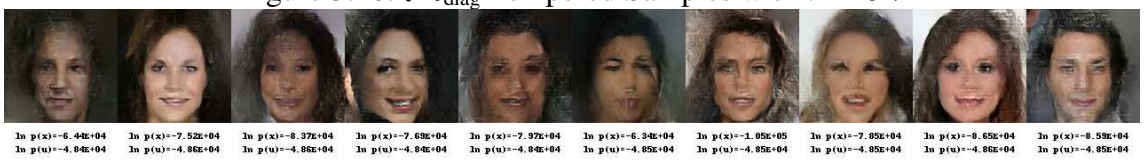
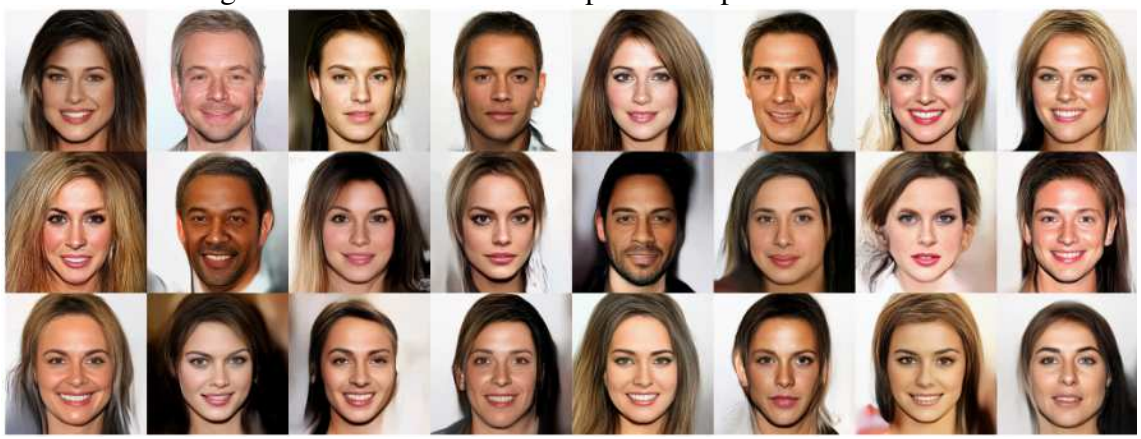


Figure 3.11: Glow Model Tempered Samples with $\tau = 0.7$



Source: (KINGMA; DHARIWAL, 2018)

4 EXPERIMENTS DESCRIPTION

The exploration of the latent space can take a number of forms. For instance, we could select subsets of samples according to some common attribute of interest and perform simple arithmetic operations using them. In this work, two experiments are defined. The first, described in Section 4.1, follows an informed vector manipulation by means of a Support Vector Machine (SVM) hyperplane as seen in the work by Shen et al. (2019) in the context of GANs. The second approach attempts to approximate multiple maximum a posteriori (MAP) parameterized diagonal Gaussians to the embedded data and is explained in Section 4.2. Experiments results can be found in Chapter 5.

4.1 Support Vector Machine classifier

In the context of GANs, the work of Shen et al. (2019) presents an approach based on learning directions of manifestation of attributes. To do so, for each attribute a linear SVM classifier is trained. In the binary case, a SVM classifier learns the hyperplane of the largest margin that separates one class from another. In this section, the approach adopted for single attribute manipulation is presented.

For each attribute, an SVM classifier is trained by means of the LinearSVC module from scikit-learn (PEDREGOSA et al., 2011) using the test data partition of CelebA. The classifier is used out of the box, and no parameter adjustments were applied. The vector that defines the learned hyperplane is then normalized. Embedded images are modified by means of a weighted addition of said vector. That is, given the latent space representation of an image $\mathbf{u}_{\text{orig}}$, the potential vector of manifestation of an attribute $\mathbf{v}_{\text{attr}}$ and a coefficient α, we have that the modified latent vector follows the equation

$$\mathbf{u}_{\text{mod}} = \mathbf{u}_{\text{orig}} + \alpha \frac{\mathbf{v}_{\text{attr}}}{\|\mathbf{v}_{\text{attr}}\|} \tag{4.1}$$

If α is negative, we should observe less of the attribute in the recovered image. Conversely, if α is positive, we should see more of the same attribute in the resulting image. We can use this method in an attempt to informedly explore the latent space and gather information about the embedding and the base distribution. We should be able to recover coherent images from the high probability regions of the base distribution. The results of these experiments are laid out in Section 5.1.

4.2 Maximum a posteriori diagonal Gaussian parameterization

It is natural to consider what the embedded data distribution looks like. The first problem that arises with current NF-based models is that there is actually no guarantee that the embedded data distribution can be adequately modeled by any analytically tractable probability distribution. As a first approach, this work proposes approximating the embedded observations of each of the attribute labels using a multivariate diagonal Gaussian distribution parameterized by learned MAP parameters. Having a distribution model for each positive and negative attribute should provide us with interesting information about the embedded data overall distribution, and might offer us a means of comparison and future label prediction if the approximations are adequate and the embedded data conveys enough structure to do so. The mathematical description of this approach is illustrated below. Experiments results are presented and discussed in Section 5.2.

4.2.1 Mathematical description

A multivariate diagonal Gaussian distribution, denoted $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, is defined by a mean parameter vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_M)$ and a diagonal covariance matrix $\boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, \ldots, \sigma_M^2)$. Our goal is to use the embedded data to determine the parameters of the Gaussian that best fit its distribution. A characteristic that makes the diagonal Gaussian interesting is the ease with which we can work with its closed form, making it a good first candidate approximate distribution for analysis.

Multivariate Gaussian distributions with diagonal covariance matrix are equivalent to multiple independent 1-D Gaussians. Given a vector $\mathbf{x} = (x_1, \ldots, x_M)$, its likelihood probability $p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ of being generated by the random process the diagonal Gaussian describes is

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathbb{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{m=1}^{M} \mathbb{N}(x_m|\mu_m, \sigma_m^2) \tag{4.2}$$

Let $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_M)$, where $\lambda_i = \sigma_i^{-2}$, for $i = 1, \ldots, M$. $\boldsymbol{\Lambda}$ is also known as the precision matrix. By means of the Bayes' Theorem and of conjugate priors (BISHOP, 2006), given a vector $\mathbf{x}$ we can calculate the posterior probability of the parameters in

closed form as

$$\begin{aligned} p(\boldsymbol{\mu}, \boldsymbol{\Lambda}|\mathbf{x}) &= p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda})p(\boldsymbol{\mu}|\boldsymbol{\Lambda})p(\boldsymbol{\Lambda}) \\ &= \prod_{m=1}^{M} p(x_m|\mu_m, \lambda_m)p(\mu_m|\lambda_m)p(\lambda_m) \\ &= \prod_{m=1}^{M} \mathbb{N}(x_m|\mu_m, \lambda_m^{-1})\mathbb{N}(\mu|\mu'_0, (\lambda'_0\lambda_m)^{-1})\text{Gam}(\lambda_m|a_0, b_0) \\ &= \prod_{M=1}^{M} \mathbb{N}(x_m|\mu_m, \lambda_m^{-1})\text{GaussianGamma}(\mu_m, \lambda_m|\mu'_{m0}, \lambda'_{m0}, a_{m0}, b_{m0}) \end{aligned} \tag{4.3}$$

where $\text{Gam}(\cdot)$ denotes the Gamma distribution. Notice that the primes $\cdot'$ are used to distinguish between parameters of the Gaussian-Gamma and the parameters of the Gaussian likelihood. Given the Gaussian-Gamma distribution's functional form, the parameters can be updated in an incremental way (BISHOP, 2006). After D observations, the parameters of the Gaussian-Gamma distribution will have the form

$$\mu'_D = \frac{\lambda'_0\mu'_0 + D\mu_{m_{MLE}}}{\lambda'_0 + D} \tag{4.4}$$

$$\lambda'_D = \lambda'_0 + D \tag{4.5}$$

$$a_D = a_0 + \frac{D}{2} \tag{4.6}$$

$$b_D = b_0 + \frac{1}{2}\left(D\sigma^2_{m_{MLE}} + \frac{\lambda'_0 D\left(\mu_{m_{MLE}} - \mu'_0\right)^2}{\lambda'_0 + D}\right) \tag{4.7}$$

where $\mu_{m_{\text{MLE}}}$ and $\sigma^2_{m_{\text{MLE}}}$ are the mean and variance parameters of a single Gaussian approximate that maximize the likelihood probability of the elements $\{x_{dm}\}_{d=1}^{D}$ of the vector data, and are given by

$$\begin{aligned} \mu_{m_{\text{MLE}}} &= \frac{1}{D}\sum_{d=1}^{D} x_{dm} \\ \sigma^2_{m_{\text{MLE}}} &= \frac{1}{D}\sum_{d=1}^{D}(x_{dm} - \mu_{m_{\text{MLE}}})^2 \end{aligned} \tag{4.8}$$

Taking the modes of the posterior distribution for $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$, we have

$$\boldsymbol{\mu}_{\text{MAP}} = \boldsymbol{\mu}'_D \tag{4.9}$$

$$\boldsymbol{\Lambda}_{\text{MAP}} = \text{diag}\left(\frac{\alpha_{D1} - \frac{1}{2}}{\beta_{D1}^{-1}}, \ldots, \frac{\alpha_{DM} - \frac{1}{2}}{\beta_{DM}^{-1}}\right) \tag{4.10}$$

The above parameters effectively maximize the posterior probability $p(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathbf{X})$

(thus the name) with respect to a set of observed data $\mathbf{X} = \{\mathbf{x}_d\}_{d=1}^{D}$. The updated diagonal Gaussian distribution that approximates $\mathbf{X}$ is then $\mathcal{N}(\boldsymbol{\mu}_{\text{MAP}}, \boldsymbol{\Lambda}_{\text{MAP}}^{-1})$. The likelihood probability of new data points can then be evaluated according to this learned approximate distribution.

5 RESULTS

This chapter is dedicated to presenting and discussing the results of the experiments described in Chapter 4. Sections 5.1 and 5.2 contain the results from the SVM and the MAP-parameterized Gaussian experiments, respectively.

5.1 Support Vector Machine experiments results

The SVMs are trained using a subset of the test data partition of the CelebA data set. The evaluations that follow use a subset of the evaluation partition of the said data set. The attribute distribution of both subsets are presented in Figures 5.1 and 5.2, respectively.

Figure 5.1: Attribute distribution of the test data subset

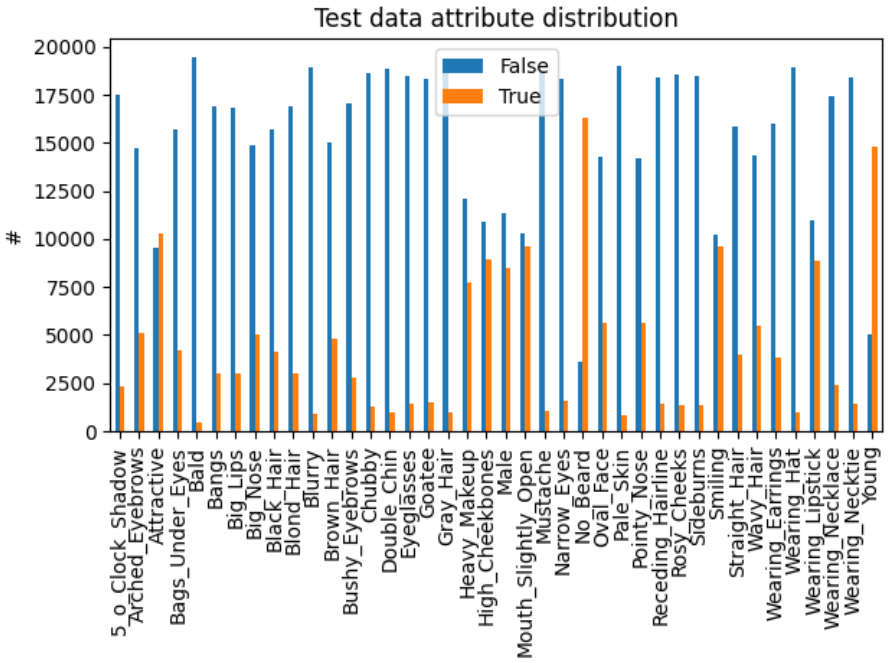
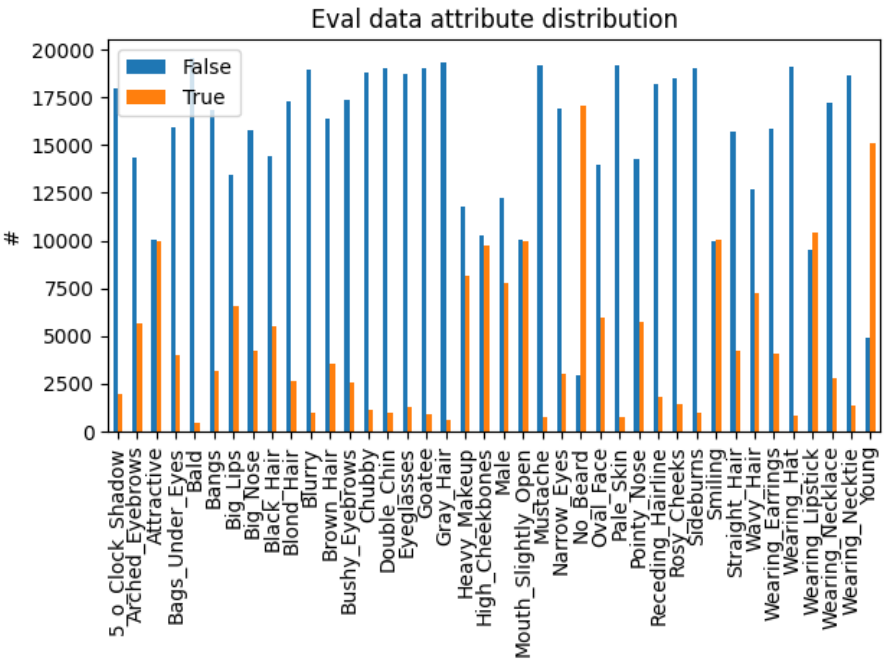


Figure 5.2: Attribute distribution of the evaluation data subset



The accuracies of the learned SVMs using each of the models $\mathcal{M}_{\text{std}}$ and $\mathcal{M}_{\text{diag}}$ are depicted in Figures 5.3 and 5.4, respectively. Remember that one classifier is trained for each attribute. In both cases we see that some classifiers are barely not random, whereas others have learned what appears to be quite good decision hyperplanes. Still, these accuracies might be biased by the data itself. Given that we have, e.g., much less bald images than non-bald ones, the SVM accuracy can be much higher by simply always predicting the same non-bald class. If we look at the support vector numbers, which will not be explicitly presented here because of technicalities involving the used library, we have that all SVM classifiers that use the embedded space of $\mathcal{M}_{\text{std}}$ have more than $13,000$ support vectors. Whereas the number of support vectors of the SVMs that used the $\mathcal{M}_{\text{diag}}$ embedded data vary from approximately $9,000$ to $16,000$. Provided that the test data partition has $19,868$ observations, we can conclude that the embedded data of both models do not convey much structure when it comes to separating the attributes' binary classes, or that the very high dimension of the problem imposes such small scales that our ability to explore and interpret the space with this method is hindered.

Figure 5.3: $\mathcal{M}_{\text{std}}$ model - Individual attributes SVM classifier accuracy

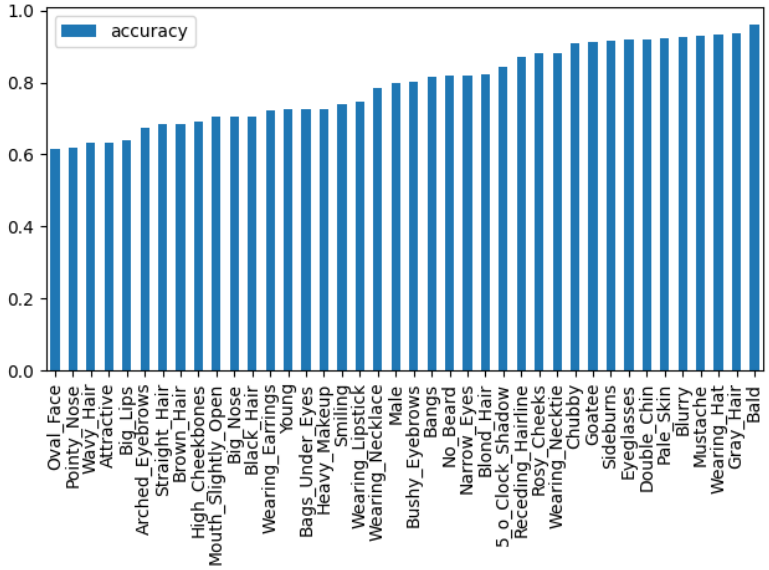
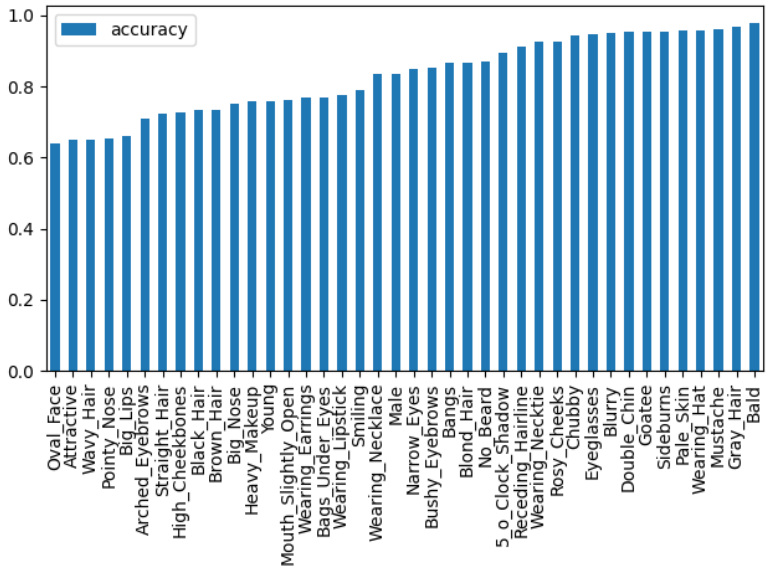


Figure 5.4: $\mathcal{M}_{\text{diag}}$ model - Individual attributes SVM classifier accuracy



5.1.1 Informed manipulations

Even though the previous analysis suggests negative results regarding the structuring of the embedded data, we can try and use the directions captured to explore if our regions of high probability of the latent space really result in coherently structured images in the data space. To this end, let's select some easy to identify attributes, like smiling and no beard. The value of α is varied linearly from -20 to 20 for 20 iterations. Figures 5.5 and 5.6 present the results of manipulating the same 30 facial images using the embedded

spaces of $\mathcal{M}_{\text{std}}$ and $\mathcal{M}_{\text{diag}}$ with the smiling hyperplane, respectively. Similarly, Figures 5.7 and 5.8 present the results for the no beard attribute. In each figure, the original images are the ones in the leftmost column. The first thing to notice is that all of them have relatively low probabilities when compared with the average images, presented in Section 3.4. Similarly, if we compare the probabilities of the originals with the sample images from said section, we see that all of them present non-negligible lower probabilities in the base distribution, even though they are originals and present much better quality. This might be an indication that the embedded data is sparsely organized in the base distribution space. Furthermore, this might be itself an effect of using too little data for the training of the models considering the very high data and latent spaces dimensions.

Figure 5.5: $\mathcal{M}_{\text{std}}$ model - Smiling manipulation

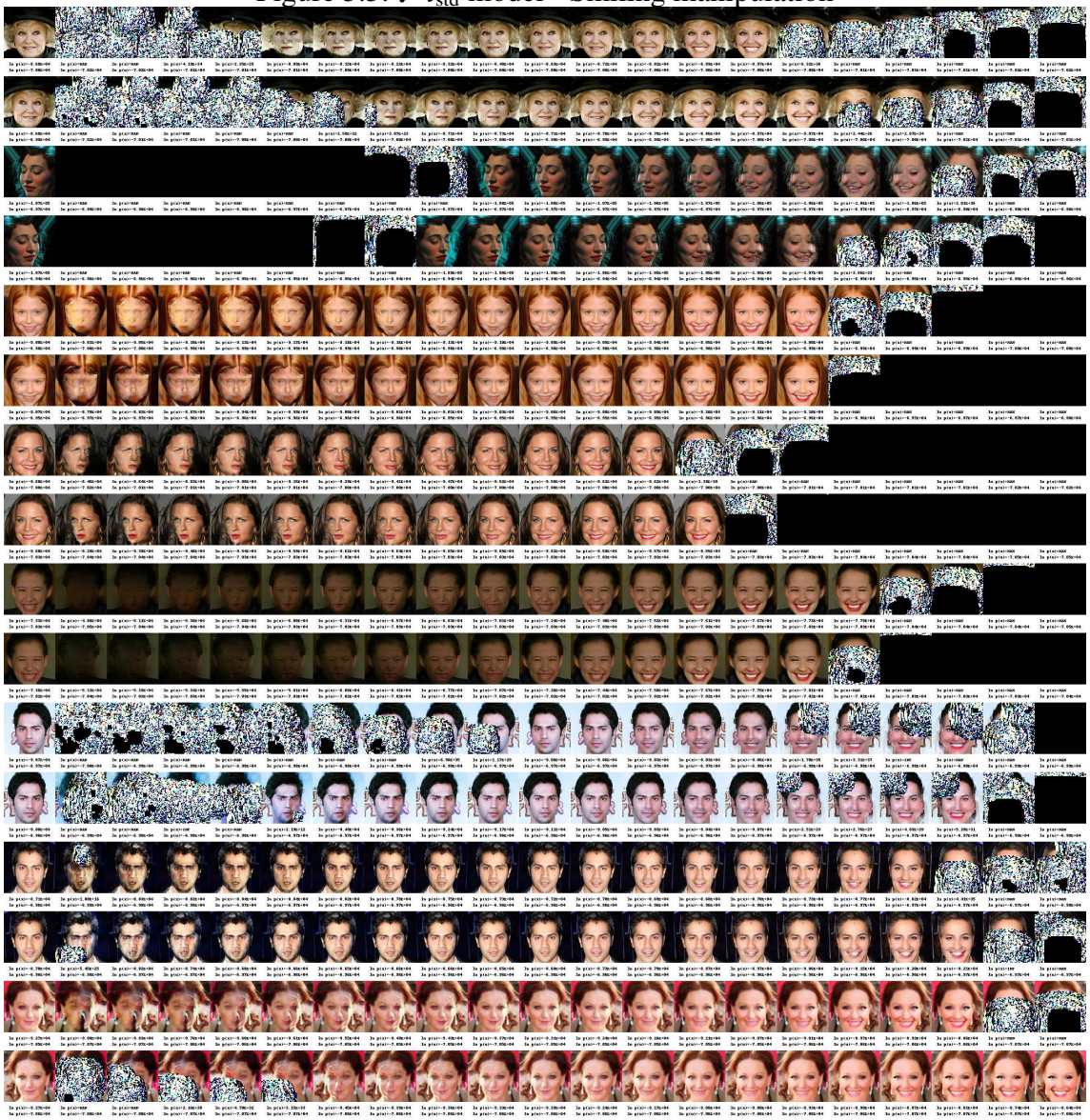


Figure 5.6: $\mathcal{M}_{\text{diag}}$ model - Smiling manipulation

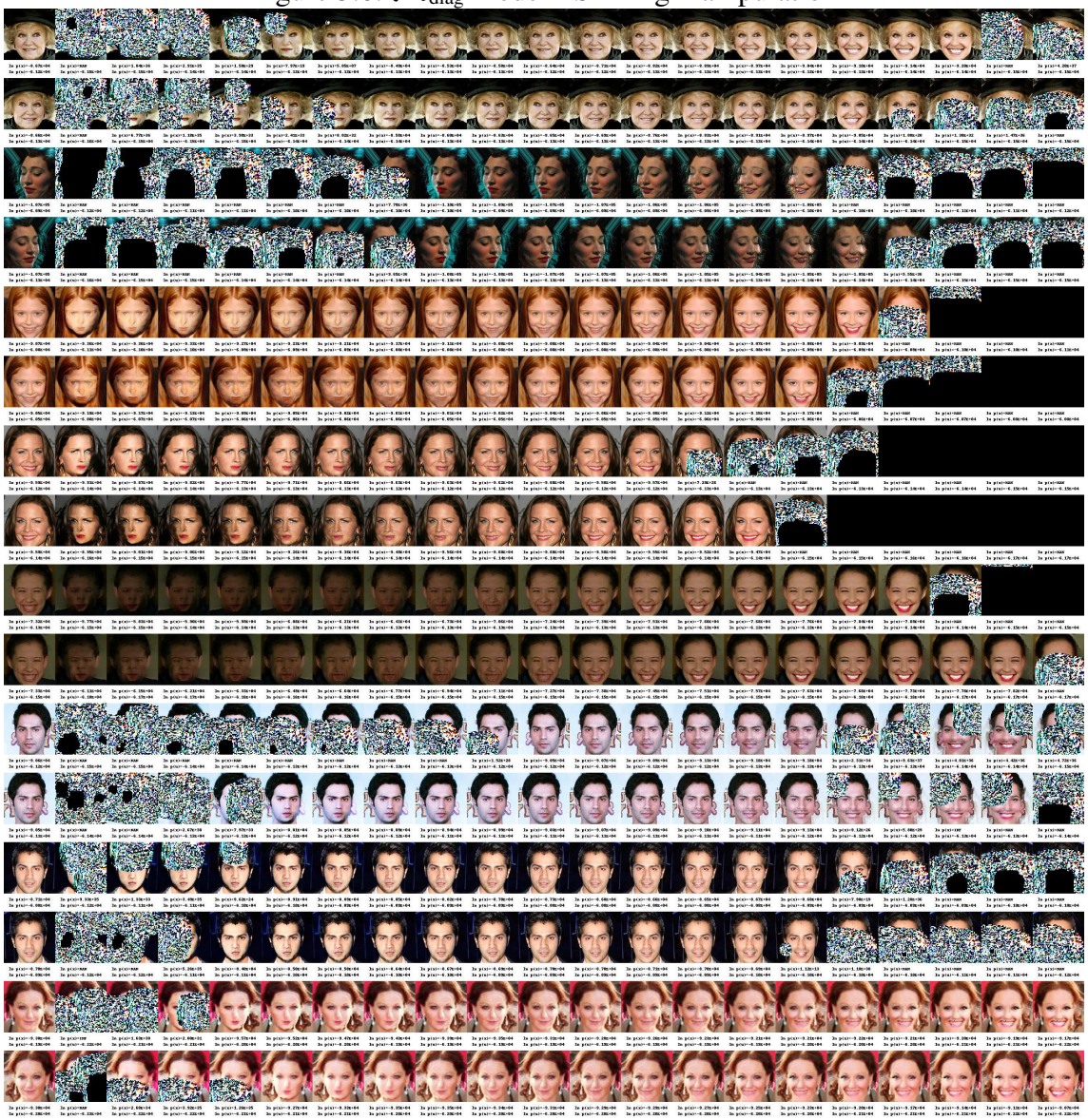


Figure 5.7: $\mathcal{M}_{\text{std}}$ model - No Beard manipulation

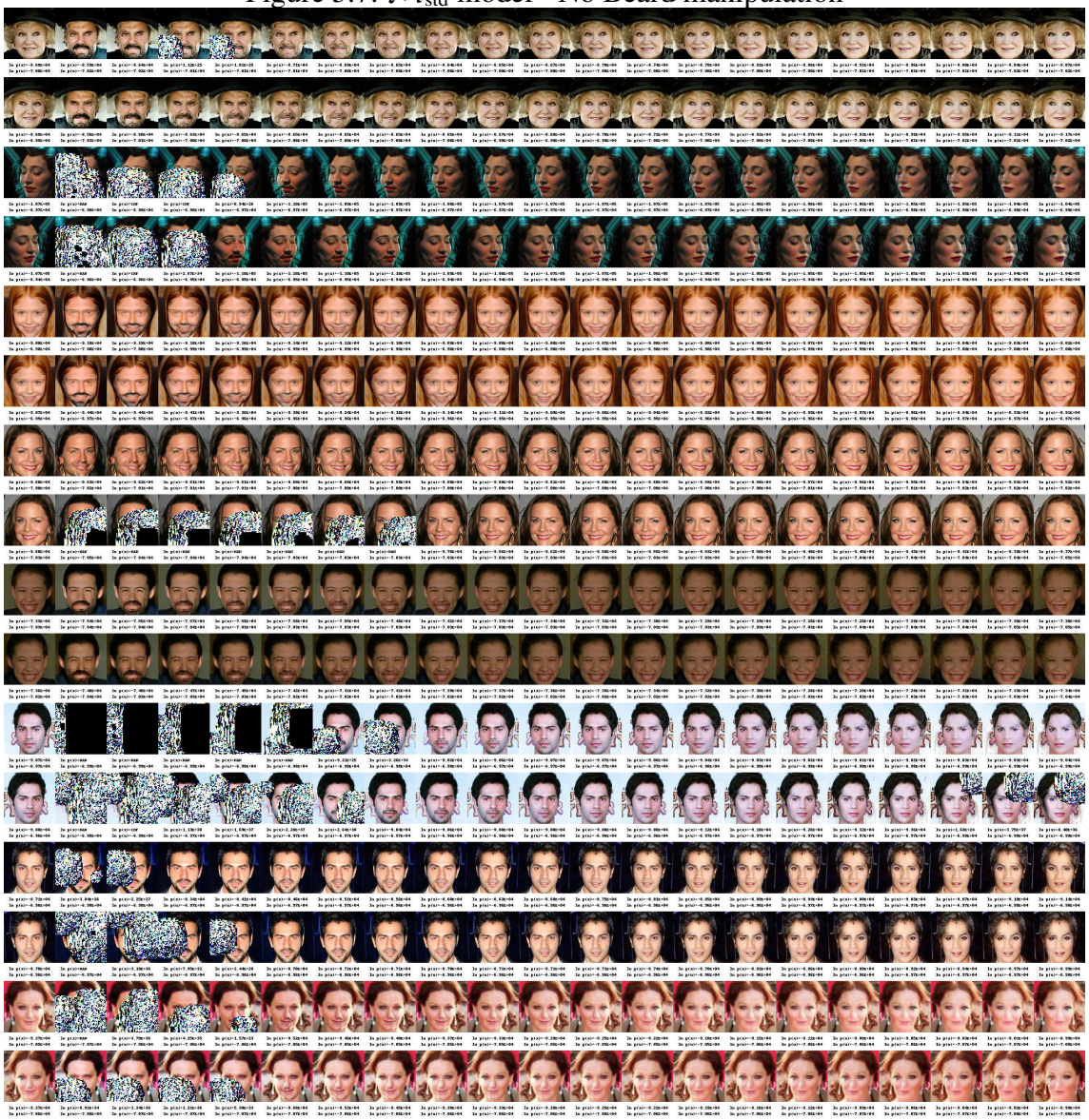
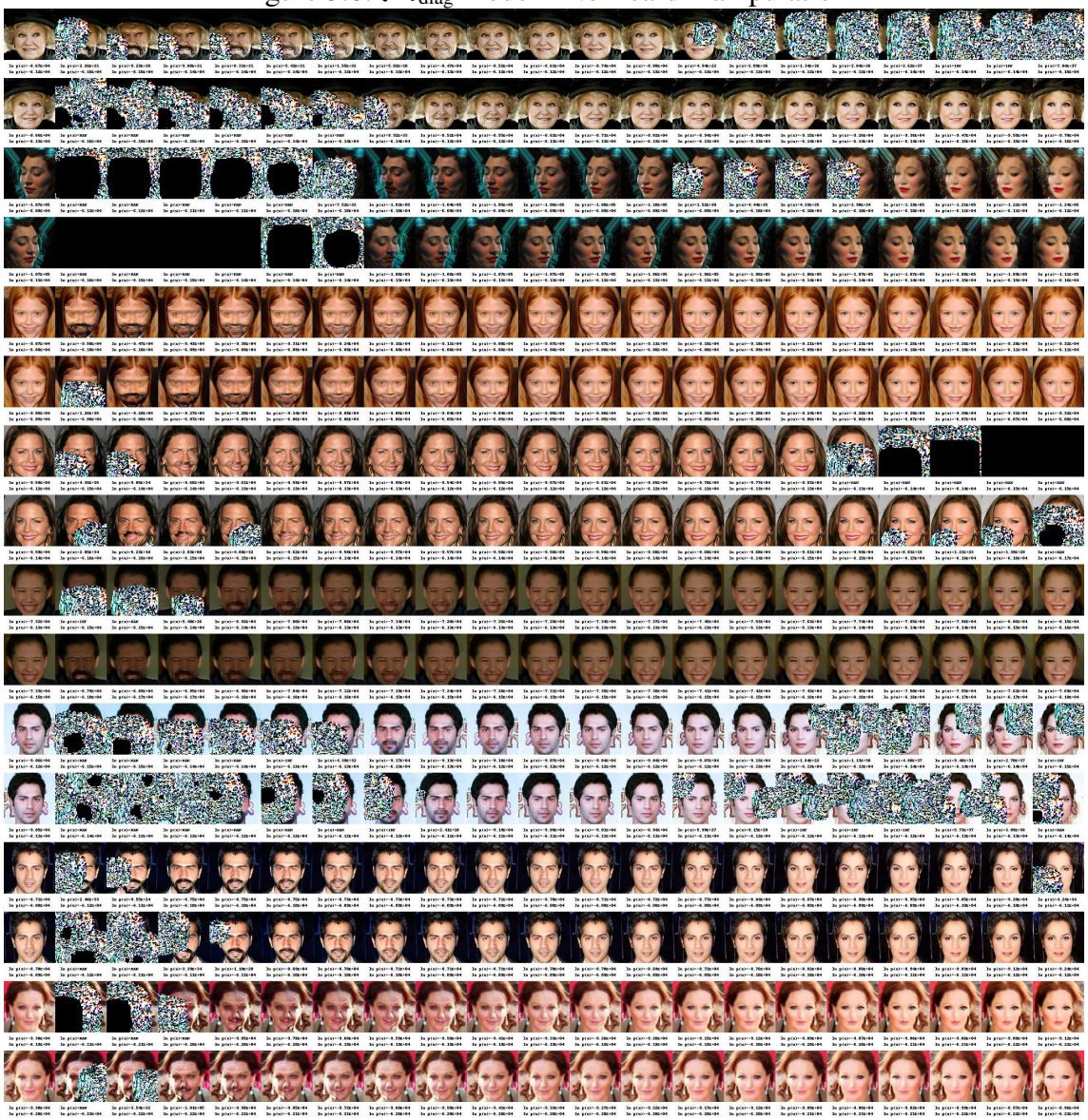


Figure 5.8: $\mathcal{M}_{\text{diag}}$ model - No Beard manipulation



Given the previous comparisons, it is difficult to say if the data presented in the above images should or not be considered to be in a high probability region of the base distribution. This is actually not defined in the context of NFs. We might, for instance, take a look at the log probabilities of points located at two standard deviations from the mean of each of the models' base distributions:

$$\begin{aligned} \ln p_{\mathcal{M}_{\text{std}}}(\boldsymbol{\mu} + 2\boldsymbol{\Sigma}^{\frac{1}{2}}\mathbf{1}) &= -143471.671875 \\ \ln p_{\mathcal{M}_{\text{diag}}}(\boldsymbol{\mu} + 2\boldsymbol{\Sigma}^{\frac{1}{2}}\mathbf{1}) &= -134741.21875 \end{aligned} \tag{5.1}$$

where the overloaded mean and covariance symbols should be taken to be the mean and covariance of the respective base distributions, which are diagonal, and $\mathbf{1}$ is a vector of ones. If we consider that the region comprised by two standard deviations is to be deemed

of high probability, then all of the presented images, even the incoherent ones, are well within this boundary. Thus, we clearly have a violation of NF coherence: the recovered images do not present the structure of the data we want to model when we transform back latent vectors in high probability regions of the base distribution. Before we continue, we should not consider this a good bound. Such a bound may be too broad and was chosen for the sake of the argument. A sample at two standard deviations in such a high dimensional space might be a very improbable one. Back to the results, notice in the first row of Figure 5.7 how there is a gap in the domain of the NF inverse, where it passes through an undefined subregion of the latent space to then again reach a well-defined subregion. This could be an indication of overfitting, or it could be an artifact common to NFs when we start to distance the sampling too much from the mode.

It is still uncertain why we have these ill-defined NF transform codomains. They might have manifested for a number of reasons. For instance, this could be because the transform model of the adapted Glow is too limited, or too little data was used for training considering the very high dimensionality of the problem and the data embedding is too sparse and its distribution even multimodal. In the last case, metrics and validation methods could direct us toward further results regarding how the unimodality of the embedded data distribution relates to the quality of sample generation and how to achieve it. Section 6.1 presents a discussion in this regard. Going back to the results at hand, and given all the considerations above, we cannot conclude much at this point. All we know is that the embedding of the models here presented is not coherent with the data of interest and the base distributions given the bound of two standard deviations.

5.2 MAP-parameterized diagonal Gaussian results

This section presents an analysis of the results of the learned attribute class approximate distributions. For each attribute, we can define two distributions: the distributions of the positive and the negative classes. As in the SVM training, the test data partition of the CelebA data set is used to learn the parameters of each of the diagonal Gaussian approximates, whereas the evaluation partition is used to assess them. Figures 5.1 and 5.2 depict the distribution of the subsets of data used. Section 5.2.1 presents an initial discussion about the learned distributions. The learned approximate distributions for two selected attributes are discussed more in depth in Section 5.2.2. Lastly, Section 5.2.3 shows results of the attempted sampling of these distributions.

5.2.1 Distributions' overview

Let's start with an overview of the distributions learned. To this end, Figures 5.9 and 5.10 present the log probabilities of the average vectors of each of the positive attributes' distributions according to the learned distribution itself, the base distribution and the data distribution. The top row shows the log probabilities calculated with the learned approximate distributions. The middle row presents the base distribution probabilities assigned to the mentioned mean vectors, and in the bottom row are the assigned probabilities according to the data distribution. Notice that the x-axis scales are distinct.

First, notice how close all of these log probabilities are to the mean probabilities of the base distribution (Figures 3.7 and 3.8) in both cases. This might mean that they have very similar precisions and, as a consequence, might not be capturing the data clusters adequately. If this is the case, the embedded images of each attribute class are scattered across the high dimensional latent space and the Gaussians offer poor approximations. In another words, the attribute class embedding does not result in a single cluster region. If we compare the base distribution likelihoods of the average vectors (second row) with the likelihoods of the models' samples from Figures 3.9 and 3.10, we see that the latter are much more distanced from the mode of the base distribution than any of the average vectors of the attributes' distributions. In Section 6.1 a discussion about how we might get a better approximation by means of a Student's t-distribution is presented. For now, let's try and take a look if we can use the approximate Gaussians for anything else.

Figure 5.9: $\mathcal{M}_{\text{std}}$ mean positive attribute log probabilities

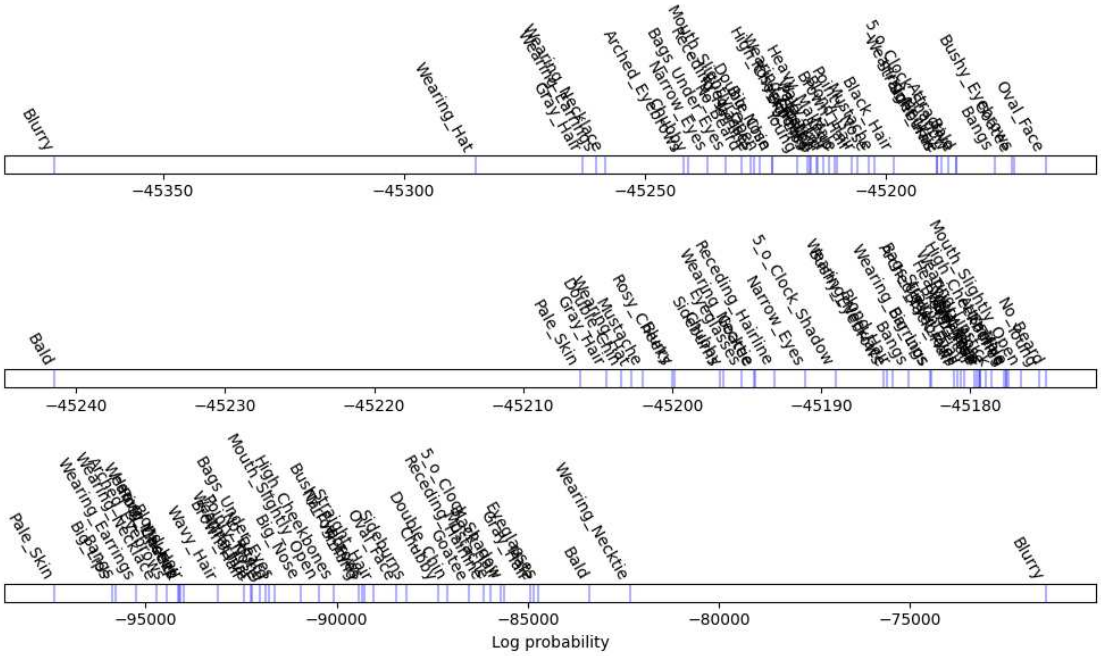
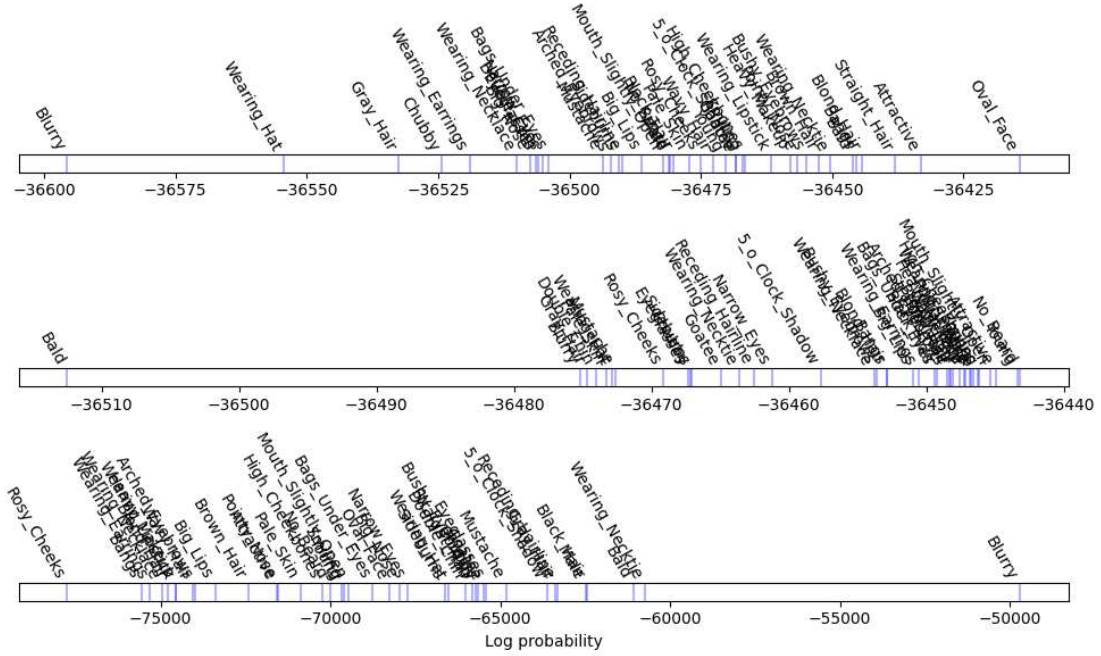


Figure 5.10: $\mathcal{M}_{\text{diag}}$ mean positive attribute log probabilities



Before we proceed to the next topic, let's select two of the attribute distributions and check its average image. As with the previous analysis using SVMs, let's use the smiling and no beard attributes. But now, instead of the no beard attribute, let's look at beard. Figures 5.11 and 5.12 present the means images for smiling using both models, $\mathcal{M}_{\text{std}}$ and $\mathcal{M}_{\text{diag}}$. Analogously, Figures 5.13 and 5.14 present the mean image to the beard attribute. We can see an overall coherent image structure, even if in low quality, that does

resemble the attribute it encodes.

Figure 5.11: $\mathcal{M}_{\text{std}}$ mean image for Smiling

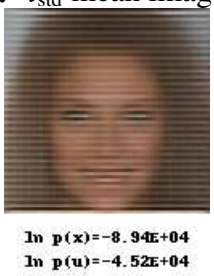


Figure 5.12: $\mathcal{M}_{\text{diag}}$ mean image for Smiling

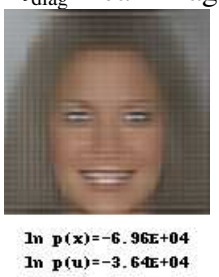
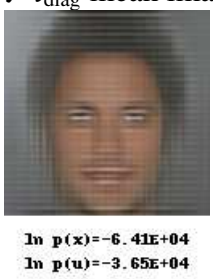


Figure 5.13: $\mathcal{M}_{\text{std}}$ mean image for Beard

Figure 5.14: $\mathcal{M}_{\text{diag}}$ mean image for Beard



5.2.2 Log probability distributions

As we have seen, the mean images indeed do provide some local characteristics that resemble the attributes of interest. Still, their probability values indicate that they have a very similar variance to the base distribution. This raises questions about their

usefulness. Figures 5.15, 5.16, 5.17 and 5.18 depict histograms of log probabilities attributed to the evaluation data partition using the learned MAP-parameterized diagonal Gaussians. The positive label distribution indicates the Gaussian that approximates the positively labeled data of a given attribute. Similarly, the negative label distributions are the Gaussian that approximate the negatively labeled data. In these images, true positives and true negatives indicate the data attribute value with respect to the data, i.e., true positive indicates that a data sample is marked positively for determining attribute and true negative indicates that the attribute is labeled as absent. We see that the distributions of log-likelihoods in all histograms, excluded the magnitude differences caused by the number of data samples, are very similar.

Figure 5.15: $\mathcal{M}_{\text{std}}$ log probabilities histograms for Smiling

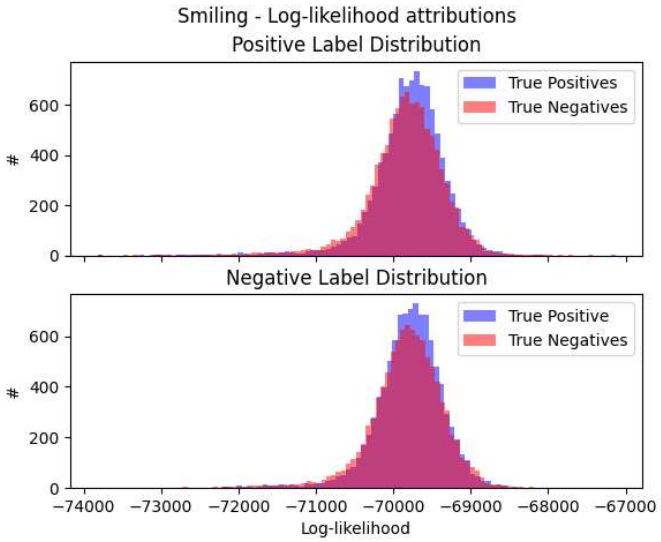


Figure 5.16: $\mathcal{M}_{\text{diag}}$ log probabilities histograms for Smiling

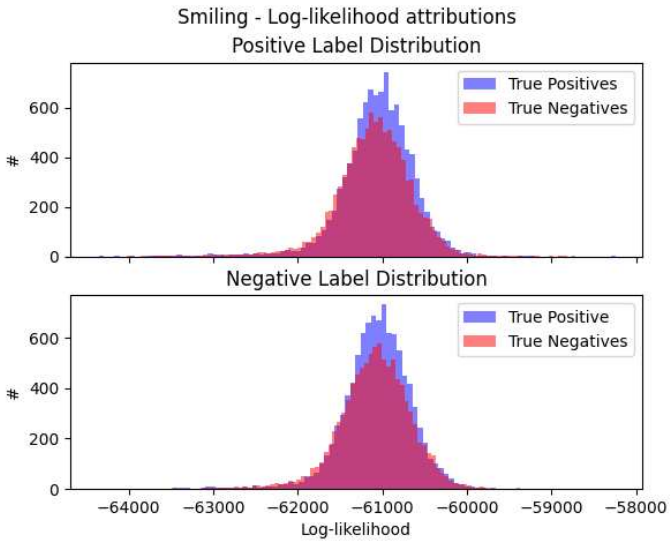


Figure 5.17: $\mathcal{M}_{\text{std}}$ log probabilities histograms for No Beard

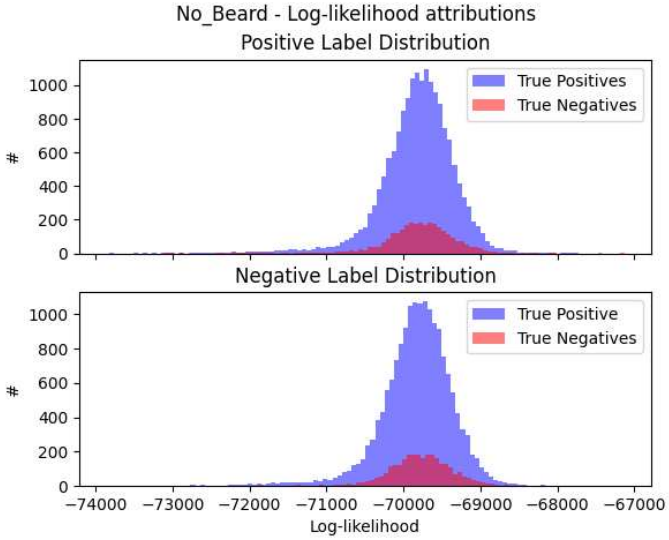
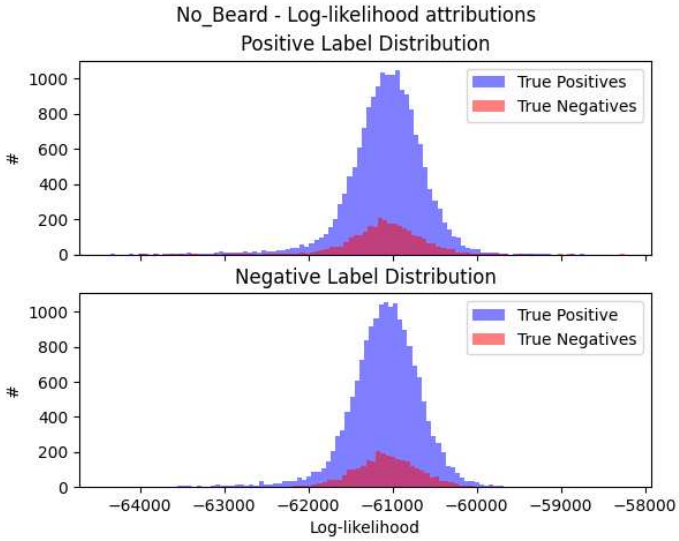


Figure 5.18: $\mathcal{M}_{\text{diag}}$ log probabilities histograms for No Beard



For the sake of completeness, let's take only the smiling attribute and try determining its class, smiling or not smiling, using the positive and negative attributes' Gaussian approximates. Let's consider a simple decision criterion: the distribution that assigns the highest log probability to the sample image determines its class. Figures 5.19 and 5.20 present the confusion matrix that results from this experiment. As we can see, the results are not too bad to the point of being random, with approximately 20% of error. Furthermore, there seems to be a bit of a better generalization capability for the Gaussians that used the $\mathcal{M}_{\text{diag}}$ embedded data for training. This might be because the higher precision of the diagonal Gaussian base distribution induces greater penalties to the NF, giving it more space to improve. Still, depending on the application and its requirements, these models together with this simple decision criterion might prove to be not enough. Section 6.1

presents a discussion on why and when a Student's t-distribution might provide us with better distribution approximates.

Figure 5.19: $\mathcal{M}_{\text{std}}$ Smiling confusion matrix

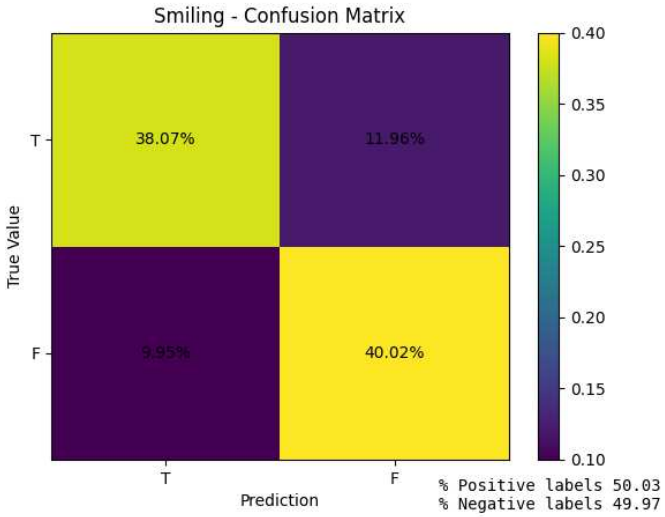
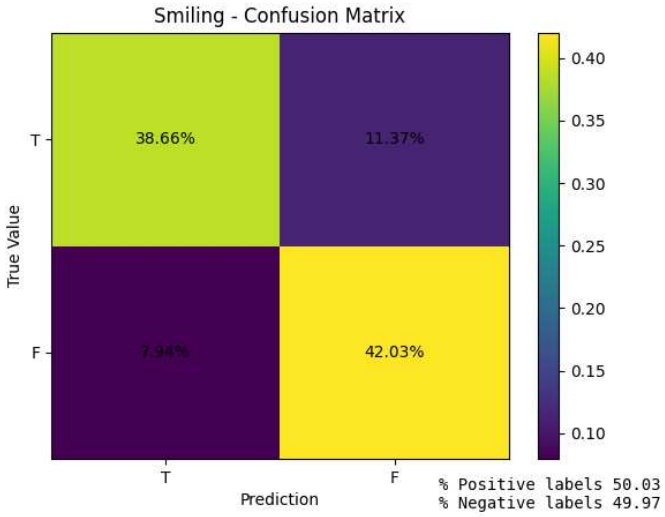


Figure 5.20: $\mathcal{M}_{\text{diag}}$ Smiling confusion matrix



5.2.3 Sampling the distributions

Even though using the distributions above for classification tasks is less than ideal, they might prove useful when it comes to sampling from specific attributes. Figures 5.21, 5.22, 5.23 and 5.24 present ten samples each for each of our attribute class models. Remember that τ stands for the temperature parameter in a tempered sampling. Here, seventy percent of the standard deviation of the learned distributions are used. Below each image are their assigned data and base distributions probabilities, as well as third values

that indicate the probability of the image according to the learned attribute distribution, denoted $\ln p(c)$. In all cases we see very close values between the base and attribute distributions assigned probabilities. As discussed in previous sections, this suggests that both distributions' variances are similar. Still, even though this similarity could impair the use of the learned attributes' distributions for sampling purposes, we observe that it does bias the sampling process towards the attribute of interest.

Figure 5.21: $\mathcal{M}_{\text{std}}$ Smiling sample with $\tau = 0.7$

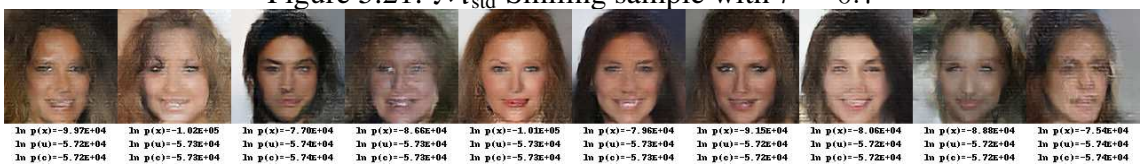


Figure 5.22: $\mathcal{M}_{\text{diag}}$ Smiling sample with $\tau = 0.7$

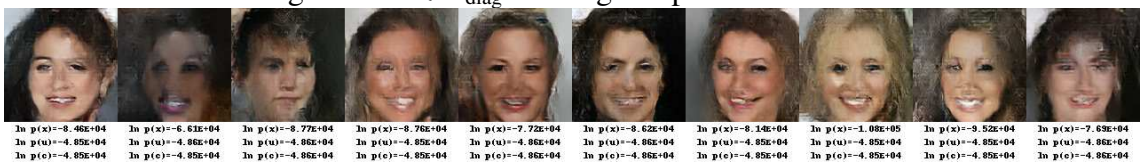


Figure 5.23: $\mathcal{M}_{\text{std}}$ Beard sample with $\tau = 0.7$

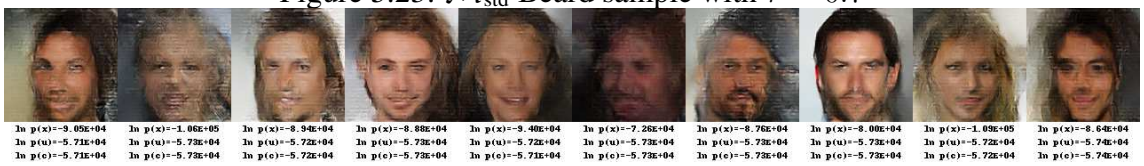
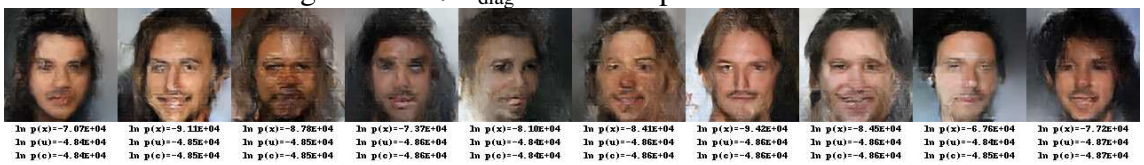


Figure 5.24: $\mathcal{M}_{\text{diag}}$ Beard sample with $\tau = 0.7$



6 DISCUSSION AND FUTURE WORK

We have seen that the models this work uses do not comply with the second NF coherence property. Still, it is difficult to determine the reasons this might be happening. We do not have enough tooling to assess the embedded data distribution properties and ways to relate these to the base distribution. Section 6.1 presents a discussion about how we could gather more information about the embedded data distribution by means of a non-unimodality metric and how could we compose such a metric. Open questions with respect to NFs and their coherence are laid out in Section 6.2. Other questions on tangent subjects are presented last in Section 6.3.

6.1 How to detect multimodality?

In the presented Bayesian approach, the embedded data was modeled by a Gaussian distribution with fixed MAP mean and precision parameters. This can be a poor approximation if the data present multiple modes, as both will be scattered across the space. The Student's t-distribution might offer a better approximation. The Student's t-distribution integrates the precision parameter by summing an infinite number of Gaussian distributions of distinct precision. This mitigates the effects of outliers and small clusters in our approximate distribution. One potential problem of this approach might still be that, if the distribution is indeed multimodal, the Student's t-distribution's approximation is less than ideal.

Having more insight on the reasons, why the embedded data distribution Gaussian approximate might fail, let us try and relate this with the properties of NF coherence and present how we might be able to assess such properties. Going back to the first property, which relates the codomain of the NF transform with the embedded data distribution, we might become more confident of its holding if somehow the latter was simpler and more readily assessable. Given that the Student's t-distribution contains each single Gaussian precision case, it also accounts for the specific case of the Gaussian with MAP precision. Using this, future work might be able to determine the degree of multimodality of the embedded data distribution to some extent by means of a divergence metric between the Student's t and the Gaussian distributions modeling the embedded data. This might prove useful in future models, for instance, by guiding the embedding codomain into a single convex region of high probability.

We have discussed the concept of a metric of non-unimodality and how reducing the divergence of an MAP-parameterized Gaussian and a Student's t-distribution could help reduce the potential multimodal characteristics of the embedded data distribution. But what can it say about the second property? The second property basically says that we want the base distribution of the NF model to actually model the data by means of the NF transform. Even if the embedded data is adequately approximated by both MAP-parameterized Gaussian and the Student's t-distributions, we might still end up with a base distribution that is not an adequate model of the embedded data distribution. Although uncertain, the solution might prove to be simple: use one of the approximate distributions as the base distribution itself. After the model's training, if both are sufficiently similar, we might even exchange them for working with distinct functional forms.

6.2 NF Coherence open questions

This work suggests that the adapted Glow models present regions of high probability where the embedding fails to recover a valid image. This clearly does not comply with the second property of NF coherence. Still, the results presented are themselves very limited and actually raise more questions than answer them:

- Currently, the NFs' literature does not cover any real form of validation of the embedding with respect to the base distribution model. This greatly hinders the advance of studies regarding the usage of the base distribution as a proxy for the data distribution in the latent space. In Section 6.1, a potential way of quantifying the degree of unimodality of the embedded data is presented. Can we derive it further with the goal of validating the base distribution approximation to the embedded data distribution? Also, what should we consider to be the "high probability region" of the base distribution when it comes to NFs? Can we find ways to guarantee NF coherence within better-defined bounds?
- A more thorough and well-grounded assessment should be made regarding the feasibility of the second property of NF coherence. For example, is there a relation between the dimension of the latent space and the required number of samples, number of transforms, or the functional form of the transforms?
- Can we determine a schedule for the precision of the base distribution, aiming at its effect in the NF transform? For example, increasing the precision of the base

distribution in hopes of making the transform contract and expand regions of the latent space to better cover the determined high probability region.

- There is still little evidence suggesting that a coherent NF offers any real improvement. For instance, the quality of generated samples or other applications such as attribute manipulation or OOD data detection. Are there any immediately notable improvements of a coherent NF?

6.3 Other questions

This section presents other questions that have surfaced while this work was being produced. These are presented below:

- The very high dimension of the image space offers possibilities for highly multimodal distributions. Added to that, the values of the loss tend to get very small for individual vector elements of an individual datum, limiting its influence on the learning procedure. Could we find ways to mitigate this effect?
- We could try and approximate the posterior probability of the NF transform parameters to a Gaussian distribution (AZEVEDO-FILHO; SHACHTER, 1994), thus having a metric of uncertainty when the embedding is confronted with new samples. Even if theoretically possible, the approximation requires calculating the second derivatives of the DNN's parameters and can prove intractable, often relying on approximations. Is such a method viable in the context of NFs, and can we find a use for it? For example, can we use it to validate generated data? Could we use it to define a sampling policy in the latent space? Or maybe use it to bias the model against data we want to protect? What about removing data not belonging to the distribution of interest from the high probability regions encoded by the model?
- If we can determine that quality of fit is given by the degree of unimodality of the embedded data distribution of an NF model, we might have something with which to guide our models' learning, or even change its structures. Given that, would a non-unimodality metric prove useful for meta-learning? If this is the case, and if we are indeed able to use the latent data representations by means of the base distribution to build efficient relations between random variables from distinct NF embeddings, where can it take us?

7 CONCLUSION

In this work, the concept of Normalizing Flow (NF) coherence and an assessment of two models from the perspective of its properties are presented. Results suggest that their embedding codomain is not coherently defined if we factor in the data and base distributions when high probability regions of the latter are considered. This renders the base distribution a poor approximation for the embedded data, hindering its use as a proxy for the data space in the latent space by models that build upon NFs. Still, what should be considered a high probability region of the base distribution is itself ill-defined in this context. In this thesis, two standard deviations were considered, but this might prove to be too broad a bound for the models at hand, the cardinality of the data set used for their training, and the very high dimensionality imposed by the image data domain.

The analysis through Gaussian approximates of individual data attributes suggests that the attributes data are scattered throughout the base distribution space. Even though this could prevent the usage of these approximates, their sampling was shown to be biased toward the attributes of interest. From the results, a discussion is presented in which it is suggested that we might be able to define a non-unimodality metric and further guide the embedded data into a single region of high density. This could be useful from the point of view of NF coherence, but a number of things are still left uncertain. For instance, we don't know if guaranteeing NF coherence brings any improvement in data generation, out-of-distribution data detection, or latent space usage and interpretability. These uncertainties and other open questions suggest that there is a lot of potential in NFs future work, and that the perspective of its coherence might offer a direction for its future research.

REFERENCES

AZEVEDO-FILHO, A.; SHACHTER, R. D. Laplace's method approximations for probabilistic inferencein belief networks with continuous variables. In: **Proceedings of the Tenth International Conference on Uncertainty in Artificial Intelligence**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994. (UAI'94), p. 28–36. ISBN 1558603328.

BISHOP, C. M. **Pattern Recognition and Machine Learning (Information Science and Statistics)**. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN 0387310738.

DINH, L.; KRUEGER, D.; BENGIO, Y. NICE: non-linear independent components estimation. In: BENGIO, Y.; LECUN, Y. (Ed.). **3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings**. [s.n.], 2015. Available from Internet: <http://arxiv.org/abs/1410.8516>.

DINH, L.; SOHL-DICKSTEIN, J.; BENGIO, S. Density estimation using real NVP. **CoRR**, abs/1605.08803, 2016. Available from Internet: <http://arxiv.org/abs/1605.08803>.

DURKAN, C. et al. **nflows: normalizing flows in PyTorch**. Zenodo, 2020. Available from Internet: <https://doi.org/10.5281/zenodo.4296287>.

FUNES, E. G. **Understanding latent vector arithmetic for attribute manipulation in normalizing flows**. 2021. Available from Internet: <http://hdl.handle.net/10230/49203>.

GOODFELLOW, I. J. et al. **Generative Adversarial Networks**. arXiv, 2014. Available from Internet: <https://arxiv.org/abs/1406.2661>.

KARRAS, T. et al. **Progressive Growing of GANs for Improved Quality, Stability, and Variation**. arXiv, 2017. Available from Internet: <https://arxiv.org/abs/1710.10196>.

KINGMA, D. P.; DHARIWAL, P. **Glow: Generative Flow with Invertible 1x1 Convolutions**. arXiv, 2018. Available from Internet: <https://arxiv.org/abs/1807.03039>.

KINGMA, D. P.; WELLING, M. **Auto-Encoding Variational Bayes**. arXiv, 2013. Available from Internet: <https://arxiv.org/abs/1312.6114>.

KIRICHENKO, P.; IZMAILOV, P.; WILSON, A. G. **Why Normalizing Flows Fail to Detect Out-of-Distribution Data**. arXiv, 2020. Available from Internet: <https://arxiv.org/abs/2006.08545>.

KOBYZEV, I.; PRINCE, S. J.; BRUBAKER, M. A. Normalizing flows: An introduction and review of current methods. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Institute of Electrical and Electronics Engineers (IEEE), v. 43, n. 11, p. 3964–3979, nov 2021. Available from Internet: <https://doi.org/10.1109%2Ftpami.2020.2992934>.

KOLLER, D.; FRIEDMAN, N. **Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning**. [S.l.]: The MIT Press, 2009. ISBN 0262013193.

LIPPE, P. **UvA Deep Learning Tutorials**. University of Amsterdam, 2021. Available from Internet: <https://uvadlc.github.io/>.

LIU, Z. et al. Deep learning face attributes in the wild. In: **Proceedings of International Conference on Computer Vision (ICCV)**. [S.l.: s.n.], 2015.

LOSHCHILOV, I.; HUTTER, F. **Decoupled Weight Decay Regularization**. arXiv, 2017. Available from Internet: <https://arxiv.org/abs/1711.05101>.

PAPAMAKARIOS, G. et al. Normalizing flows for probabilistic modeling and inference. arXiv, 2019. Available from Internet: <https://arxiv.org/abs/1912.02762>.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

REZENDE, D. J.; MOHAMED, S. **Variational Inference with Normalizing Flows**. arXiv, 2015. Available from Internet: <https://arxiv.org/abs/1505.05770>.

SHEN, Y. et al. Interpreting the latent space of gans for semantic face editing. **CoRR**, abs/1907.10786, 2019. Available from Internet: <http://arxiv.org/abs/1907.10786>.

TABAK, E. G.; TURNER, C. V. A family of nonparametric density estimation algorithms. **Communications on Pure and Applied Mathematics**, v. 66, n. 2, p. 145–164, 2013. Available from Internet: <https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.21423>.

VALENZUELA, A. et al. Expression transfer using flow-based generative models. In: **2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)**. [S.l.: s.n.], 2021. p. 1023–1031.