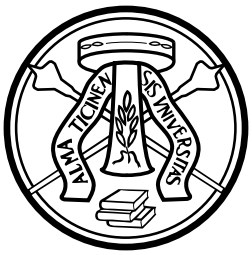


UNIVERSITÀ
DI PAVIA

UNIVERSITY OF PAVIA
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL, COMPUTER AND BIOMEDICAL ENGINEERING

MASTER'S DEGREE IN COMPUTER ENGINEERING

MASTER THESIS

Beyond Gaussian Priors: A Poisson Latent Space for Variational Autoencoders

Oltre le Distribuzioni a Priori Gaussiane: Uno Spazio Latente di Poisson per gli Autoencoder Variazionali

Candidate: Schifano Roberto

Supervisor: Prof. Cusano Claudio
Co-Supervisor: Dr. Cabassa Greta

A.Y. 2025/2026

Abstract

Variational Autoencoders conventionally rely on a continuous Gaussian prior to structure the latent space, an approach largely motivated by analytical simplicity. This thesis investigates an alternative formulation in which the Gaussian prior is replaced by a discrete Poisson distribution, parameterized by a vector of non-negative intensity rates. The adoption of a discrete latent space introduces a fundamental challenge: the Poisson sampling operator is non-differentiable and does not admit a reparametrization, ruling out standard backpropagation through the latent bottleneck. To address this, two gradient estimation strategies are implemented and evaluated: a Straight-Through Estimator based on a Gaussian surrogate gradient, and a Score Function Estimator derived from reinforcement learning principles. Both are compared against a Gaussian reparameterization baseline on the CelebA face dataset. Experimental results demonstrate that the Poisson latent space supports structured and semantically meaningful representations, as evidenced by latent traversals, UMAP clustering and attribute direction manipulations. The straight-through estimator achieves competitive generation quality relative to the Gaussian baseline, while the score function estimator suffers from high gradient variance and slow convergence. An architectural scaling analysis further reveals that generation quality in Poisson VAEs is bottlenecked by encoder capacity rather than total parameter count. Finally, the Poisson latent space exhibits distinctive geometric properties, including irregular cluster structures absent in the Gaussian baseline, suggesting fundamental differences in the organization of the two latent spaces.

Sommario

I Variational Autoencoder si basano convenzionalmente su una distribuzione a priori Gaussiana per strutturare lo spazio latente, una scelta dettata principalmente dalla semplicità analitica. Questa tesi investiga una formulazione alternativa in cui la distribuzione a priori Gaussiana viene sostituita da una distribuzione di Poisson discreta, parametrizzata da un vettore di tassi di intensità non negativi. L'adozione di uno spazio latente discreto introduce una sfida fondamentale: l'operatore di campionamento di Poisson non è differenziabile e non essendo continuo non ammette riparametrizzazione, precludendo la backpropagation standard attraverso il collo di bottiglia latente. Per affrontare questo problema, vengono implementate e valutate due strategie di stima del gradiente: uno Straight-Through Estimator basato su un gradiente surrogato Gaussiano, e uno Score Function Estimator derivato da principi di reinforcement learning. Entrambi vengono confrontati con una baseline di riparametrizzazione Gaussiana sul dataset di volti di celebrità CelebA. I risultati sperimentali dimostrano che lo spazio latente di Poisson supporta rappresentazioni strutturate e semanticamente significative, come evidenziato dal traversal dello spazio latente, dal clustering UMAP e dalle manipolazioni delle direzioni degli attributi. Lo stimatore straight-through raggiunge una qualità generativa competitiva rispetto alla baseline Gaussiana, mentre lo stimatore della score function soffre di alta varianza del gradiente e lenta convergenza. Un'analisi della scalabilità architetturale rivela inoltre che la qualità generativa nei VAE Poissoniani è limitata dalla capacità dell'encoder piuttosto che dal numero totale di parametri. Infine, lo spazio latente di Poisson presenta proprietà geometriche distintive, tra cui strutture di cluster irregolari assenti nella baseline Gaussiana, suggerendo differenze fondamentali nell'organizzazione dei due spazi latenti.

Contents

Chapter 1

Introduction

Generative modeling has undergone a profound transformation over the past decade, driven by the development of deep latent variable models capable of learning compact and semantically meaningful representations of high-dimensional observations. Among these, the Variational Autoencoder (VAE) [1] has established itself as a foundational framework, combining the expressive power of deep neural networks with a probabilistic formulation embedded in variational inference. By encoding observations into a structured latent space and learning to reconstruct them through a generative decoder, VAEs enable a wide range of subsequent tasks including image synthesis and attribute manipulation.

Despite their theoretical elegance, VAEs are susceptible to a well-documented failure known as *posterior collapse* [2], in which the encoder learns to ignore the input and the latent representations collapse to the prior distribution. This phenomenon is particularly pronounced when powerful decoder architectures are paired with expressive priors, causing the model to ignore the latent bottleneck entirely and reduce the informativeness of the learned representations.

A complementary limitation of the standard VAE formulation concerns the choice of the latent prior. The conventional isotropic Gaussian prior, while analytically convenient, imposes a continuous and symmetric structure on the latent space that may not align with the discrete nature of the underlying generative factors. In physical imaging systems, photon arrival processes are more naturally modeled as counting processes governed by Poisson statistics [3], suggesting that a discrete Poisson prior may offer a more grounded initial assumption for certain visual domains.

This thesis investigates the design and training of a Variational Autoen-

coder operating with a discrete Poisson latent space, replacing the standard Gaussian prior with a Poisson distribution parameterized by a vector of intensity rates λ.

The adoption of a discrete latent distribution introduces a fundamental technical challenge: unlike continuous distributions, the Poisson sampling operator does not admit a reparameterization, blocking standard backpropagation through the latent bottleneck. To address this, two alternative gradient estimation strategies are implemented and evaluated: a Straight-Through Estimator based on a Gaussian surrogate gradient (PGA), and a Score Function Estimator derived from reinforcement learning principles (RLT). A standard Gaussian reparameterization baseline (GRT) is included for reference. Beyond the gradient estimation problem, this work explores the architectural and optimization choices required to train a discrete Poisson VAE effectively on a real-world face dataset.

An asymmetric encoder-decoder architecture is proposed, combining an EfficientNet-inspired [4] inference network with an SRGAN-inspired [5] generative decoder equipped with AdaIN-based style modulation [6]. Training stability is ensured through a sigmoidal KL annealing [2] schedule and a free bits [7] regularization strategy. The experimental evaluation is conducted on the CelebA face dataset [8] at 64×64 resolution, assessing generation quality via the Fréchet Inception Distance (FID) [9] and complementing quantitative results with an extensive latent space analysis including traversals, UMAP clustering, attribute direction manipulation and attribute arithmetic.

The remainder of this thesis is organized as follows. Chapter 2 reviews the relevant literature on variational autoencoders, discrete latent variable models and gradient estimation strategies. Chapter 3 describes the proposed architecture and training mechanisms in detail. Chapter 4 presents the experimental results and analysis. Chapter 5 summarizes the findings and outlines directions for future work.

Chapter 2

Related work

This chapter provides the necessary theoretical background to understand the contribution of this thesis. We begin by giving a general introduction about Autoencoders and Variational Autoencoders (VAE), the core generative models on which this work is based. We then provide an overview of the main challenges in VAE training, namely posterior collapse and the techniques used to mitigate it. We also describe the architectures of the EfficientNet and SRGAN models, which inspired the structure of the proposed model. Next, we explore the major differences in using a different probability distribution, namely Poisson, in the VAE model. Finally, we describe an alternative training process that takes advantage of the policy gradient theorem of reinforcement learning, as well as the FID metric (Fréchet Inception Distance) used to evaluate the quality of generated images.

2.1 Autoencoders

An **Autoencoder** [10] is a neural network architecture that includes two main components: an encoder and a decoder.

The former compresses the input into a smaller representation, often called the *latent space* or *bottleneck*. The latter takes the compact representation and tries to reconstruct the original input. In other words, the encoder learns a compressed and meaningful representation of the data; the decoder learns to generate data from a latent vector. They are trained in a *self-supervised* way, the objective is to minimize the *reconstruction loss* (Figure 2.1), that is, the difference between the reconstructed sample and the original counterpart. By focusing on one part of the model at a time, its property can be exploited for a specific application. In particular, the

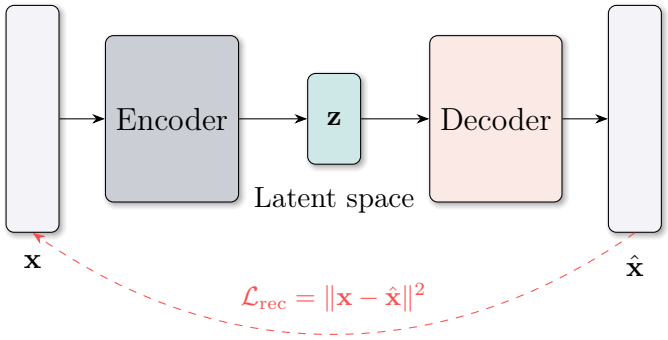


Figure 2.1: Autoencoder architecture. The encoder maps the input $\mathbf{x}$ to a latent representation $\mathbf{z}$; the decoder reconstructs $\hat{\mathbf{x}}$.

encoder can be used for dimensionality reduction, just like PCA, but with the advantage of capturing non-linear relationships in the data. In addition, the decoder can be used to generate samples. However, since the latent space is unstructured, we have no control over the generation. This happens because no constraints are applied to the latent space. Therefore, sampling a random latent vector does not guarantee that the decoder will be able to generate a meaningful output. This is the main limitation in autoencoders and is also the reason why **Variational Autoencoders** have been introduced.

2.2 Variational Autoencoders

To overcome this, VAEs [1] propose a probabilistic structure in the latent space, in particular a standard Gaussian. This means that the encoder, instead of mapping the input to a single point, maps it to a probability distribution, providing its μ and σ parameters. These are then used to sample the latent vector, which is fed through the decoder like in a standard autoencoder. As a result, being the latent space structured, close points correspond to similar outputs, and so sampling in the latent space will is likely to generate a meaningful output. However, since sampling is a random operation and, therefore, not differentiable, it requires a special trick to work.

2.2.1 Probabilistic formulation and ELBO

Formally, given a set of observations $\mathbf{x} \in X$, the VAE assumes that each sample is generated from a latent variable $\mathbf{z} \in \mathbb{R}^d$ through the generative

process:

$$p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x} \mid \mathbf{z})\, p_\theta(\mathbf{z}),$$

where $p_\theta(\mathbf{z})$ is the *prior* on the latent variable (typically a standard Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$) and $p_\theta(\mathbf{x} \mid \mathbf{z})$ is the *likelihood*, parameterized by a neural network called decoder. The goal is to maximize the marginal log-likelihood $\log p_\theta(\mathbf{x}) = \log \int p_\theta(\mathbf{x} \mid \mathbf{z})\, p_\theta(\mathbf{z})\, d\mathbf{z}$ which, however, is intractable for high-dimensional latent spaces. To get around the integral, VAEs introduce an approximated distribution $q_\phi(\mathbf{z} \mid \mathbf{x})$, parametrized by another neural network called the encoder whose purpose is to be as similar as possible to $p_\theta(\mathbf{z} \mid \mathbf{x})$, called the *posterior*. The **Kullback-Leibler** divergence can be used to minimize their distance [11]:

$$\begin{aligned}
&\mathcal{D}_{KL}[q_\phi(\mathbf{z} \mid \mathbf{x}) \parallel p_\theta(\mathbf{z} \mid \mathbf{x})] = \\
&= \int q_\phi(\mathbf{z} \mid \mathbf{x}) \log \frac{q_\phi(\mathbf{z} \mid \mathbf{x})}{p_\theta(\mathbf{z} \mid \mathbf{x})}\, d\mathbf{z} \\
&= \int q_\phi(\mathbf{z} \mid \mathbf{x}) \log \frac{q_\phi(\mathbf{z} \mid \mathbf{x})\, p_\theta(\mathbf{x})}{p_\theta(\mathbf{z}, \mathbf{x})}\, d\mathbf{z} \qquad (1) \\
&= \int q_\phi(\mathbf{z} \mid \mathbf{x}) \left(\log p_\theta(\mathbf{x}) + \log \frac{q_\phi(\mathbf{z} \mid \mathbf{x})}{p_\theta(\mathbf{z}, \mathbf{x})} \right) d\mathbf{z} \\
&= \log p_\theta(\mathbf{x}) + \int q_\phi(\mathbf{z} \mid \mathbf{x}) \log \frac{q_\phi(\mathbf{z} \mid \mathbf{x})}{p_\theta(\mathbf{z}, \mathbf{x})}\, d\mathbf{z} \qquad (2) \\
&= \log p_\theta(\mathbf{x}) + \int q_\phi(\mathbf{z} \mid \mathbf{x}) \log \frac{q_\phi(\mathbf{z} \mid \mathbf{x})}{p_\theta(\mathbf{x} \mid \mathbf{z})\, p_\theta(\mathbf{z})}\, d\mathbf{z} \qquad (3) \\
&= \log p_\theta(\mathbf{x}) + \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z} \mid \mathbf{x})} \left[\log \frac{q_\phi(\mathbf{z} \mid \mathbf{x})}{p_\theta(\mathbf{z})} - \log p_\theta(\mathbf{x} \mid \mathbf{z}) \right] \\
&= \log p_\theta(\mathbf{x}) + \mathcal{D}_{KL}[q_\phi(\mathbf{z} \mid \mathbf{x}) \parallel p_\theta(\mathbf{z})] - \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z} \mid \mathbf{x})}[\log p_\theta(\mathbf{x} \mid \mathbf{z})].
\end{aligned}$$

Then, we define

$$\mathcal{L}_{VAE}(\theta, \phi; \mathbf{x}) := -\log p_\theta(\mathbf{x}) + \mathcal{D}_{KL}[q_\phi(\mathbf{z} \mid \mathbf{x}) \parallel p_\theta(\mathbf{z} \mid \mathbf{x})].$$

Notes on steps:

(1) Application of Bayes' theorem rewritten as $p(\mathbf{z} \mid \mathbf{x}) = \frac{p(\mathbf{z}, \mathbf{x})}{p(\mathbf{x})}$.

(2) The term $\log p_\theta(\mathbf{x})$ does not depend on $\mathbf{z}$, thus it moves outside the integral. We then apply the probability density property where $\int q_\phi(\mathbf{z} \mid \mathbf{x})\, d\mathbf{z} = 1$.

(3) Decomposition of the joint probability using the chain rule: $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x} \mid \mathbf{z})\, p(\mathbf{z})$.

Finally,

$$\mathcal{L}_{VAE}(\theta, \phi; \mathbf{x}) = \mathcal{D}_{KL}[q_\phi(\mathbf{z} \mid \mathbf{x}) \parallel p_\theta(\mathbf{z})] - \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x} \mid \mathbf{z})]. \quad (2.1)$$

The first term is the Kullback-Leibler divergence between the approximated posterior and the prior, which acts as a regularizer forcing the latent distribution to align with the chosen prior. The second is the reconstruction error, which forces the decoder to faithfully reproduce the input data.

By flipping the signs of Equation 2.1, we obtain the **Evidence Lower Bound** (ELBO):

$$\text{ELBO}(\theta, \phi; \mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x} \mid \mathbf{z})] - \mathcal{D}_{KL}[q_\phi(\mathbf{z} \mid \mathbf{x}) \parallel p_\theta(\mathbf{z})]. \quad (2.2)$$

The name stems from the fact that the KL divergence is strictly non-negative; thus, the ELBO establishes a lower bound on the marginal log-likelihood, meaning $\log p_\theta(\mathbf{x}) \geq \text{ELBO}$. Consequently, minimizing the objective function $\mathcal{L}_{\text{VAE}}$ is equivalent to maximizing the lower bound of the probability of generating real data samples.

2.2.2 Reparameterization trick

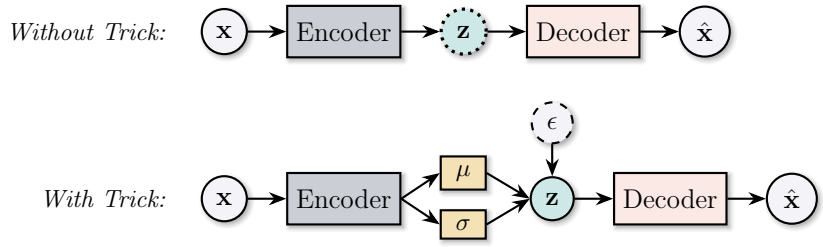


Figure 2.2: Difference in VAE configuration using the reparameterization. In the top configuration, direct stochastic sampling creates a barrier for backpropagation. In the reparameterized bottom configuration, stochasticity is isolated within the noise vector ϵ, establishing a continuous path for gradients of μ and σ.

The expectation term in equation 2.2 calls for sampling the latent variable $\mathbf{z}$ from the distribution $q_\phi(\mathbf{z} \mid \mathbf{x})$. Sampling is inherently a stochastic procedure; therefore, the gradients cannot flow through this operation during backpropagation. To ensure proper training, Kingma and Welling [1] introduced the **reparameterization trick**. As shown in figure 2.2, this technique isolates the stochasticity by expressing the latent variable as a deterministic function of an independent random noise vector:

$$\mathbf{z} = \mu + \epsilon \odot \sigma, \quad \text{where} \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

By shifting the stochastic component to ϵ, which is independent of the model parameters, the gradients can flow freely through $\mathbf{z}$. As a result, the optimization relies on deterministic variables, through which the network will be trained via standard gradient descent.

2.3 Posterior collapse

Despite their effectiveness, VAEs frequently suffer from a severe problem during training, called *posterior collapse* [2]. This phenomenon occurs when the encoder fails to learn a meaningful representation of the data, causing the decoder to reconstruct the input without relying on the information encoded within the latent variables $\mathbf{z}$. Consequently, the latent codes become entirely uninformative, and the decoder behaves essentially as a deterministic autoencoder or an isolated autoregressive model, relying solely on its own capacity to capture data distributions.

The root cause of posterior collapse lies in the balance between the two terms in the ELBO loss. In particular, during the initial phases of training, the reconstruction error pushes the decoder to use $\mathbf{z}$. However, if the decoder is powerful enough, it models $p_\theta(\mathbf{x})$ without $\mathbf{z}$ and at that point zeroing the KL divergence term is the easiest solution for the optimizer. To do so, it forces $q_\phi(\mathbf{z} \mid \mathbf{x}) = p_\theta(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, once the KL term reaches zero, the encoder stops receiving gradients and becomes stuck in this local minimum. To prevent the latent space from becoming uninformative, several interventions have been proposed in the literature. The most prominent methodologies focus on altering the optimization schedule or regularizing the effect of the KL term:

KL annealing: This approach introduces a scheduling hyperparameter [2], denoted as β, to scale the KL divergence term during training.

$$\text{ELBO}(\theta, \phi; \mathbf{x}) = \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x} \mid \mathbf{z})] - \beta_t \cdot \mathcal{D}_{KL}[q_\phi(\mathbf{z} \mid \mathbf{x}) \parallel p_\theta(\mathbf{z})].$$

By varying it dynamically from zero to one according to a predefined schedule (e.g., linear or sigmoidal), the model is allowed to focus exclusively on reconstructing data details during the initial epochs.

However, it requires thorough hyperparameter tuning to find an optimal annealing schedule.

Free bits (KL thresholding): Instead of minimizing the divergence across all latent dimensions indiscriminately, the free bits [7] technique enforces a minimum capacity threshold on individual dimensions.

$$\mathcal{D}_{KL}^{\text{free}} = \sum_i \max(\mathcal{D}_{KL,i}, \lambda).$$

By replacing the standard KL term, the penalty is active only if the divergence exceeds a certain value. While effective at preserving active latent dimensions, it can sometimes lead to under-regularized spaces if the threshold is set too high.

2.4 Beyond the Gaussian assumption

Although the Gaussian distribution is the standard choice for the latent space structure in VAEs, it is not always the most natural option for the data or for the representation at hand. When dealing with digital images, while the choice of a normal distribution is mathematically convenient, it often fails to mirror the true underlying physical process of digital image acquisition. Physically, camera sensors capture images by converting incoming light into electrical signals, a process that fundamentally relies on counting the number of discrete photons that reach the sensor grid. This counting process is subject to quantum fluctuations, a phenomenon known as photon shot noise, which is governed precisely by a Poisson distribution [3]. Shifting the generative hypothesis from a Gaussian implementation to a Poisson scheme alters both the probabilistic formulation of the model and the structural design of the encoder network. In a conventional VAE, the encoder must parametrize a continuous distribution by providing two distinct vectors for each pixel: the mean μ and the variance σ^2. On the other hand, the Poisson distribution is completely defined by a single parameter, the intensity rate λ, which regulates both mean and variance of the distribution.

$$P(x) = \frac{e^{-\lambda}\lambda^x}{x!}.$$

Consequently, the introduction of this distribution induces two criti-

cal modifications at the architectural level of the network. First, due to this single-parameter nature, the output layer of the encoder is simplified. Instead of generating two different parameters to represent the mean and variance, the encoder outputs a single parameter vector. This effectively halves the required data flows of the output layer compared to a standard Gaussian implementation. Second, the intensity rate of a Poisson process must be strictly positive ($\lambda > 0$) to represent a valid physical count of photons. Standard linear activation functions in the final layer are therefore unsuitable. To enforce this constraint, the final activation layer of the encoder must utilize bounded functions, such as the *Softplus* activation ($\text{Softplus}(x) = \log(1 + e^x)$) or an *Exponential* activation ($\text{Exponential}(x) = \exp(x)$) to ensure that the network outputs mathematically valid intensity rates across all latent dimensions.

2.4.1 Gradient estimation for discrete sampling

Additionally, the use of a Poisson prior introduces a further challenge during training. The standard VAE scheme relies on the reparameterization trick to enable the flow of gradients during backpropagation. This trick requires the distribution to be continuous and smoothly differentiable. Because the Poisson distribution is inherently discrete, the sampling operation is non-differentiable. Consequently, standard backpropagation fails, as gradients cannot flow from the decoder back to the encoder. When dealing with discrete operations, the literature provides two main methodological patterns:

Straight-through estimators: This paradigm [12] bypasses non-differentiable operations by decoupling the forward pass execution from the backward gradient computation. In particular, it uses the true discrete distribution during the forward process, while substituting a differentiable surrogate function during the backward pass. In the context of a Poisson framework, literature demonstrates that the discrete distribution can be approximated asymptotically by a Gaussian with mean and variance λ ($\mathcal{N}(\lambda, \lambda)$) due to the central limit theorem.

Score function estimators: The second paradigm encompasses the non-differentiable sampling by treating it as a stochastic process, optimizing the model via gradient estimation methods borrowed from reinforcement learning. By applying the *Policy Gradient Theorem*

through the REINFORCE estimator [13], it is possible to compute unbiased gradient estimates of the expected loss without requiring the sampling operation itself to be differentiable. However, a notorious limitation of score function estimators is their exceptionally high variance, which can destabilize the optimization problem. To mitigate this issue, the literature relies on variance reduction techniques, most notably the implementation of a baseline function. That is, an element subtracted from the reward to center the gradient updates without introducing bias.

2.5 Relevant architectures

The structural configuration of deep convolutional networks dictates their capacity for feature extraction and image synthesis. In generative modeling, traditional architectures relied on perfectly symmetrical convolutional layers for both the extraction and the generation phases. However, as deep learning evolved, a new paradigm emerged, favoring asymmetric architectures. This change recognizes that feature extraction and image reconstruction require different architectural patterns. Modern architectures frequently adapt components from specialized state-of-the-art models, most notably *EfficientNet* [4], *SRGAN* [5] and *StyleGAN* [14].

2.5.1 EfficientNet

The primary objective of the encoder is to compress high-dimensional inputs into an abstract, lower-dimensional representation without losing critical information. For this task, classification models like EfficientNet [4] stand out in the literature by introducing the concept of *compound scaling*, a methodology that balances network depth, width and resolution using a single compound coefficient. Driven by Mobile Inverted Bottleneck Convolutions (MBConv) and depthwise convolutions, EfficientNet captures image features with a fraction of the parameters required by traditional networks.

2.5.2 SRGAN

Conversely, the decoder must solve an inverse mapping problem, transforming compressed latent coordinates back into pixel space. A persistent challenge in variational autoencoders is the production of blurry or averaged

outputs, an artifact of transpose-convolutional layers. To mitigate this, the literature offers architectural innovations such as in Super-Resolution Generative Adversarial Networks (SRGAN) [5]. The SRGAN generator pattern is designed specifically to recover fine and complex detail from compressed inputs. By utilizing *Residual Blocks* with deep skip-connections [15], the architecture enables gradients to flow freely during optimization. Furthermore, the integration of pixel shufflers, which perform sub-pixel convolutions [16] to upsample feature maps, successfully mitigates the introductions of checkerboard artifacts.

2.5.3 Style modulation

In traditional symmetric configurations, the latent vector is supplied solely as an input to the initial layer of the decoder, forcing the network to preserve abstract concepts through an increasingly deep stack of deterministic convolutions. To overcome this bottleneck, contemporary designs incorporate conditioning techniques derived from style-based generative frameworks, most notably the StyleGAN architecture [14]. Rather than relying on a single injection point, a dedicated multi-layer mapping network is introduced to transform the input latent code into an intermediate space. These decoupled features are then continuously distributed across every resolution scale of the synthesis network using Adaptive Instance Normalization (*AdaIN*) [6]. By modulating feature maps dynamically at each convolutional block, the decoder can efficiently partition tasks, meaning that deeper layers supervise the global structural and geometric layout, while shallower layers specialize in fine detail texture synthesis.

2.6 FID as an evaluation metric

Evaluation of generative models' performance represents a complex challenge in computer vision. Traditional pixel-level metrics, such as the Mean Squared Error (MSE), fail to capture human perceptual judgment. For instance, a simple pixel-wise translation or rotation can yield a massive numerical error despite the image preserving impeccable visual quality. At the same time, a blurry image can present a low pixel error, while being inadequate for a human. To address this limitation, the literature introduced the *Fréchet Inception Distance* (FID) [9], which has become the standard benchmark for quantifying the realism of synthesized images. The FID

metric operates within a high-level feature space extracted from a state-of-the-art deep classification network, specifically *Inception-v3* [17] pre-trained on ImageNet. Images from both the target (real) and the generated (synthetic) distributions are fed through the network and their activations are captured at an intermediate layer. This method simplifies complex spatial variations into low-dimensional features that capture abstract semantic concepts, including textures, geometric shapes and structural elements. The vectors extracted from the two distributions are modeled as continuous multivariate Gaussian distributions. The FID is then computed by calculating the Fréchet distance between them, using their respective empirical means (μ_r, μ_g) and covariance matrices (Σ_r, Σ_g):

$$\mathrm{FID}(r, g) = \|\mu_r - \mu_g\|_2^2 + \mathrm{Tr}\left(\Sigma_r + \Sigma_g - 2\left(\Sigma_r \Sigma_g\right)^{1/2}\right).$$

A lower FID score indicates that the distance between the two multi-dimensional Gaussians is minimized, implying that the synthetic images are statistically indistinguishable from the real target in terms of both quality and diversity of the generated image distribution.

Chapter 3

Method

This chapter describes the concrete architectural design and optimization strategies implemented in this thesis to implement a Variational Autoencoder operating with a discrete Poisson latent space. By substituting the conventional continuous Gaussian formulation with a discrete Poisson distribution, the model inherently reduces parameter complexity, resulting in a simplified and more mathematically concise structure. In contrast with standard symmetric designs, the proposed framework introduces an asymmetric configuration that focuses on both high-level feature compression and high-fidelity face synthesis from the CelebA dataset. We begin by describing the structural components of the generative scheme, detailing the Inverted Bottleneck layers of the inference network (Encoder), the discrete stochastic bottleneck and the texture-preserving residual blocks of the generation network (Decoder). We then formalize the framework's optimization objective, presenting the formulation in its two components, the KL divergence and the L_1 reconstruction penalty. Next, we document the two alternative gradient estimation strategies implemented to enable backpropagation through the non-differentiable sampling, namely, the straight-through estimator with an approximated Gaussian and the score function estimator derived from reinforcement learning. Finally, we analyze the architectural trade-off across different parameter count, discussing the characteristics of the evaluated 60M, 53M and 36M configurations.

3.1 Encoder architecture

The primary task of the encoder within the proposed Poisson VAE is to compress a high-dimensional color image $\mathbf{x} \in \mathbb{R}^{3\times H\times W}$ into a vector of strictly

positive rates $\lambda \in \mathbb{R}^D$, which parameterizes the latent Poisson distribution. To achieve high parameter efficiency while still maintaining robust feature extraction, the inference network implements a custom convolutional foundation inspired by the EfficientNet topology [4]. The overall structure of the encoder is organized into two main stages, a deep downsampling pipeline followed by a sequence of inverted residual bottleneck blocks.

Downsample blocks: Each of these blocks consists of a standard 2D convolutional layer with a kernel size of 3×3, a padding of 1 and a stride of 2. Additionally, the convolution is normalized via an Instance Normalization (InstanceNorm2d) layer with learnable affine parameters. Non-linear activation is provided by a Parametric ReLU (PReLU), which introduces a learnable slope for negative inputs to mitigate the dying ReLU phenomenon during the early phases of training.

Mobile Inverted Bottleneck Convolution (MBConv) blocks: The main building block of the network is the MBConv block module, implementing depthwise convolutions alongside Squeeze-and-Excitations (SE) mechanism. The block first expands the input channel's dimensionality by an expansion ratio $\alpha = 6$ using a 1×1 point-wise convolution to project features into a higher-dimensional space. A 3×3 depthwise convolution (with the number of groups equal to the channel size) then processes spatial dependencies independently per channel. To capture global context, an adaptive average pooling layer compresses the feature maps into a 1×1 representation, which is passed through a two-layer reduction network (with a reduction ratio of $r = 4$) and a Sigmoid activation. Finally, a point-wise projection convolution maps the features back to the target channel size. If input and output dimensionalities match, a residual skip-connection adds the original input to the calculated features.

The principal implementation of the inference network is the *encoder_ 60M* parameter model shown in Figure 3.1. Given an input image with height H and width W, the network execution flows deterministically through the following pipeline:

1. **Initial feature mapping**: the raw 3-channel input is mapped in a 16-channel output using a 3×3 convolution layer.

2. **Downsampling**: The tensor passes through a pipeline of Downsample block modules, progressively reducing the spatial dimension by a

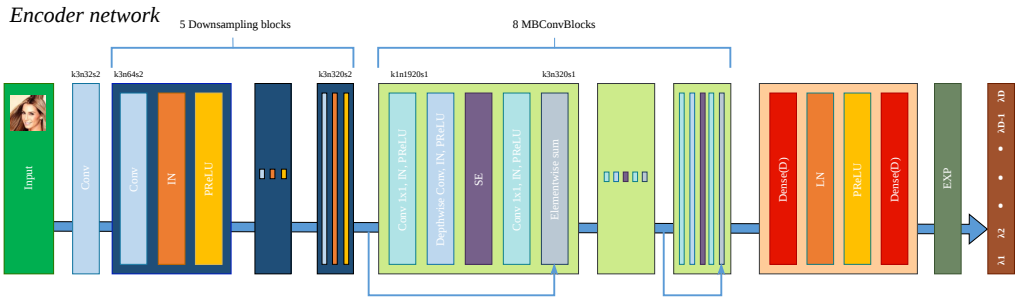


Figure 3.1: Encoder architecture with corresponding kernel size (k), number of feature maps (n) and stride (s) indicated for each convolutional layer.

total factor of $2^5 = 32$, while scaling the channel depth from 16 to 32, 64, 128, 256 and ultimately 320.

3. **Feature extraction**: The 320-channel feature map is fed into a deep sequential stack of 8 consecutive MBConv block modules, refining facial feature representations and long-range spatial dependencies.

4. **Final projection layer**: The final tensor is flattened into a one-dimensional vector of shape $(H/32) \times (W/32) \times 320$. This is then processed by a multi-layer perceptron consisting of a linear layer, a normalization block, a PReLU activation and a final linear projection layer to output a raw feature representation of size equal to the target latent space dimension D.

To satisfy the non-negativity constraint mandated by the Poisson distribution ($\lambda > 0$), the raw output of the fully connected layer is transformed via an exponential activation function. This exponential mapping forces the encoder to parameterize valid intensity rates across all latent dimensions, ensuring valid rate parameters for sampling from the discrete distribution.

3.2 Latent space sampling

Once the encoder processes the input image $\mathbf{x}$, it outputs the vector of intensity rates $\lambda \in \mathbb{R}^D$ through the exponential activation function. The rate vector is then used to draw a discrete sample $\mathbf{z}$ directly from a multivariate Poisson distribution:

$$\mathbf{z} \sim \mathcal{P}(\lambda) \in \mathbb{N}^D.$$

In practical implementation, this operation is handled during the forward pass. Although this sampling procedure provides the exact discrete prop-

erties required by the framework, it introduces a fundamental challenge. The sampling operation is non-differentiable, blocking gradients flow during backpropagation. The integer vector sampled $\mathbf{z}$ is then directly forwarded as the input for the subsequent generative network.

3.3 Decoder architecture

The generation network takes the discrete Poisson latent vector $\mathbf{z} \in \mathbb{N}^D$ and reconstructs a high quality color image $\mathbf{x} \in \mathbb{R}^{3\times H\times W}$. In order to preserve high frequency textures (such as hair and skin boundaries) without introducing blurriness, the decoder avoids standard transposed convolutions. Instead, it combines an AdaIN-based [6] modulation mechanism inspired by StyleGAN [14] with a convolution upsample pipeline inspired by super-resolution architectures [5]. The architecture is divided into three key functional stacks: a multi-layer non-linear mapping network, a sequence of style-modulated residual inverted bottlenecks and a cascade of pixel-shuffling operations.

Style modulation layer: To condition the generation process on the latent code, a custom StyleModulation layer implements a form of Adaptive Instance Normalization (*AdaIN* [6]). Given the latent representation $\mathbf{z}$, a fully connected layer projects it into a modulation space of size $2\times C$, where C is the channel depth. This vector is split into scaling (γ) and shifting (β) parameters. To guarantee training stability, the projection weights are initialized to zero, while the baseline scaling biases are set to one and shifting biases to zero. The input tensor is normalized via an instance normalization operator and subsequently transformed as shown in Equation 3.1.

$$\text{StyleMod}(\mathbf{x}, \mathbf{z}) = \left(\frac{\mathbf{x} - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\right) \cdot \gamma(\mathbf{z}) + \beta(\mathbf{z}). \tag{3.1}$$

Style-modulated MBConv blocks: These blocks replicate the expansion and depthwise convolutional mechanics of the encoder, including the Squeeze-and-Excitation attention mechanism. However, the standard *InstanceNorm2d* layer following the final point-wise projection is replaced by the conditional *StyleModulation* layer, allowing the discrete vector $\mathbf{z}$ to directly alter feature statistics dynamically based on the latent code.

Pixel-shuffle blocks: Traditional upsampling operations frequently introduce checkerboard-like patterns. To circumvent this, the network uses sub-pixel convolution layers via the *PixelShuffleBlock*. This module projects C_{in} channels into a higher channel space of size $C_{\text{out}} \times \text{upscale\_factor}^2$ using a 3×3 convolution. A pixel-shuffle operator then rearranges these channels into spatial dimensions, increasing the spatial resolution by a factor of 2 while smoothly reducing channel depth.

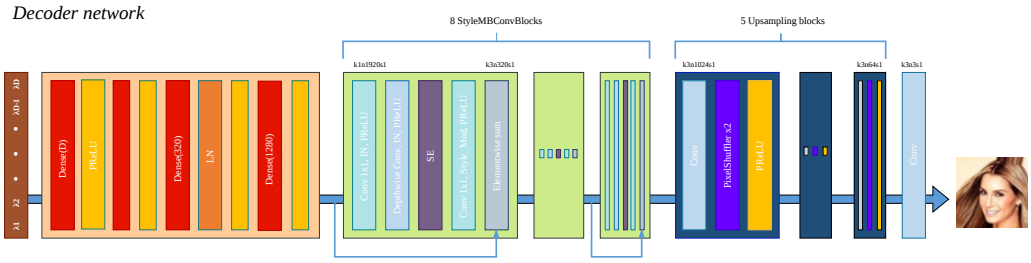


Figure 3.2: Decoder architecture with corresponding kernel size (k), number of feature maps (n) and stride (s) indicated for each convolutional layer.

The principal implementation of the synthesis network is the *decoder_ 60M* parameter model shown in Figure 3.2. Given a latent code $\mathbf{z} \in \mathbb{N}^D$, the network execution flows deterministically through the following pipeline:

1. **Latent mapping and feature projection:** The incoming vector $\mathbf{z}$ passes through a multi-layer perceptron (MLP) that acts as a mapping network. It consists of two consecutive blocks of a dense layer and a PReLU activation to introduce non-linear feature mixing before modulation. A third linear layer maps the features to a dimension of 320, stabilized by a Layer Normalization block and a PReLU activation. A final linear layer expands the representation to match the channel dimension $(H/32) \times (W/32) \times 320$, which is subsequently reshaped into a low-resolution feature tensor $\mathbf{x} \in \mathbb{R}^{320 \times \frac{H}{32} \times \frac{W}{32}}$.

2. **Latent modulation:** This low-resolution tensor is processed by a sequential stack of $N_{\text{res}} = 8$ consecutive *StyleMBConvBlock* layers. Each block receives both the current feature map and the original latent vector $\mathbf{z}$, guiding the global stylistic layout of the synthesis.

3. **Sub-pixel resolution upsampling:** Features are fed through a sequential pipeline of five *PixelShuffleBlock* modules. This progressively

increases spatial dimensions by a cumulative factor of $2^5 = 32$, effectively recovering the original input image resolution ($H \times W$), while compressing the channel maps from $320 \to 256 \to 128 \to 64 \to 32 \to 16$.

4. **Output layer:** The final 16-channel tensor is mapped into the 3-channel color space via a 3×3 convolutional layer. A *Tanh* activation function constraints the output pixels values to the bounded continuous range $[-1, 1]$.

3.4 Objective Function

To train the Poisson Variational Autoencoder, the optimization objective must balance two constraints: minimizing the reconstruction error between the input and generated images, and regularizing the predicted intensity fields against a prior distribution. Following the variational inference framework, the total objective is formulated as a weighted loss function.

$$\mathcal{L}_{\text{total}} = \beta \cdot \mathcal{L}_{\text{KL}}(\lambda \parallel \lambda_0) + \mathcal{L}_{\text{rec}}(\mathbf{x}, \hat{\mathbf{x}}), \tag{3.2}$$

where in Equation 3.2 β represents a scaling factor utilized to modulate the regularization strength and to prevent posterior collapse ([2]). Unlike standard Gaussian VAEs that regularize the latent space toward a normal distribution $\mathcal{N}(0, \mathbf{I})$, our framework requires a discrete, non-negative probability distribution. We define a parametric prior distribution in which each latent dimension is modeled as an independent Poisson distribution with a fixed target intensity rate λ_0. Given the encoder's predicted intensity vector $\lambda \in \mathbb{R}^D$ and the prior intensity $\lambda_0 \in \mathbb{R}^D$, the regularization term is formalized by calculating the exact analytical Kullback-Leibler divergence between two multivariate independent Poisson distributions:

$$\mathcal{L}_{\text{KL}}(\lambda \parallel \lambda_{\mathbf{0}}) = \lambda \left(\log \frac{\lambda}{\lambda_{\mathbf{0}}} \right) - \lambda + \lambda_{\mathbf{0}}. \tag{3.3}$$

Minimizing Equation 3.3 forces the encoder to distribute semantic characteristics across the latent channels, preventing individual features from growing infinitely or collapsing into inactive states.

The soundness of the reconstruction between the synthesized output $\hat{\mathbf{x}}$ and the source image $\mathbf{x}$ is obtained using the Mean Absolute Error (MAE),

equivalently known as the L_1 loss. The L_1 norm is preferred over the standard Mean Squared Error (L_2 loss) due to its structural robustness against outliers and its tendency to preserve sharper details. The reconstruction error is formalized as follows:

$$\mathcal{L}_{\text{rec}}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{3 \times H \times W} \sum_{c=1}^{3} \sum_{i=1}^{H} \sum_{j=1}^{W} |x_{c,i,j} - \hat{x}_{c,i,j}| \,. \tag{3.4}$$

By optimizing Equation 3.4, the decoder is forced to match image details with the original face domain, while the L_1 penalty prevents the blurry artifacts typically introduced by squared Euclidean distance approximations.

3.5 Training mechanisms

The fundamental challenge of optimizing a Variational Autoencoder with a discrete Poisson latent space lies in the non-differentiable nature of the sampling operator $\mathbf{z} \sim \mathcal{P}(\lambda)$. Because the Poisson distribution generates discrete integers, the derivative of the forward sampling step with respect to the parameters λ is undefined. Consequently, standard backpropagation through the computational graph is blocked. To overcome this limitation, this thesis implements two alternative gradient estimation strategies: a Straight-Through Estimator variant and a Policy Gradient method rooted in Reinforcement Learning.

3.5.1 Straight-through estimator

The first approach relies on a surrogate gradient framework. During the forward pass, the model draws discrete samples using the Poisson operator:

$$\mathbf{z}_{\text{forward}} = \text{Poisson}(\lambda).$$

During the backward pass, the framework must evaluate the gradient of the total loss with respect to the intensity vector λ, denoted as $\frac{\partial \mathcal{L}}{\partial \lambda}$.

$$\frac{\partial \mathcal{L}}{\partial \lambda} = \frac{\partial \mathcal{L}}{\partial \mathbf{z}} \cdot \frac{\partial \mathbf{z}}{\partial \lambda}. \tag{3.5}$$

By applying the chain rule, this can be decomposed as in Equation 3.5, where $\frac{\partial \mathcal{L}}{\partial \mathbf{z}}$ represents the incoming gradient from the decoder layers. Because the Poisson distribution is inherently discrete, the true analytical

derivative $\frac{\partial \mathbf{z}}{\partial \lambda}$ is not well defined, which disrupts backpropagation. To overcome this, we introduce a proxy approximation. The discrete sampling procedure is modeled as a continuous multivariate Gaussian distribution that shares the same mean and variance as the underlying Poisson process, such that $\mathcal{P}(\lambda) \approx \mathcal{N}(\lambda, \lambda)$. Under this assumption, the sampled vector $\mathbf{z}$ can be expressed using the reparameterization trick:

$$\mathbf{z} = \lambda + \sqrt{\lambda} \odot \epsilon, \quad \text{with } \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

Differentiating this continuous reparametrized representation with respect to the intensity rates λ yields the following algebraic derivation:

$$\frac{\partial \mathbf{z}}{\partial \lambda} = \mathbf{1} + \frac{\epsilon}{2\sqrt{\lambda}} = \mathbf{1} + \frac{\mathbf{z} - \lambda}{2\lambda} = \frac{\mathbf{1}}{2} + \frac{\mathbf{z}}{2\lambda}. \tag{3.6}$$

By substituting the original reparametrized expression of $\mathbf{z}$ back into the numerator of Equation 3.6 and defining all vector divisions as element-wise operations, the formulation simplifies into the surrogate gradient:

$$\frac{\partial \mathbf{z}}{\partial \lambda} \approx \frac{\mathbf{1}}{2} + \frac{\mathbf{z}}{2\lambda},$$

which provides a non-zero surrogate gradient during the backward pass, enabling stable end-to-end optimization. From an implementation perspective, this surrogate gradient mechanism is embedded directly into the computational graph via a custom autograd function.

3.5.2 Score function estimator

The second training paradigm approaches the non-differentiable bottleneck from a reinforcement learning perspective, formalizing the generation of discrete latents as an action selection policy. This optimization takes advantage of the classical REINFORCE [10] algorithm. Under this paradigm, the objective is to maximize the expected reward across the stochastic latent space. Formally, given a reward function $f(\mathbf{z})$ and the conditional Poisson probability distribution $p(\mathbf{z} \mid \lambda)$, the gradient of the expected reward with respect to the distribution parameters λ is formulated as:

$$\nabla_\lambda \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z}|\lambda)}[f(\mathbf{z})] = \nabla_\lambda \sum_{\mathbf{z}} p(\mathbf{z} \mid \lambda) f(\mathbf{z}) = \sum_{\mathbf{z}} \nabla_\lambda p(\mathbf{z} \mid \lambda) f(\mathbf{z}). \tag{3.7}$$

By applying the log-derivative trick, we exploit the identity $\nabla_\lambda p(\mathbf{z} \mid \lambda) = p(\mathbf{z} \mid \lambda)\nabla_\lambda \log p(\mathbf{z} \mid \lambda)$. Substituting this into Equation 3.7 transforms the analytical gradient into an expectation that can be approximated via Monte Carlo sampling:

$$\sum_{\mathbf{z}} p(\mathbf{z} \mid \lambda)\nabla_\lambda \log p(\mathbf{z} \mid \lambda) f(\mathbf{z}) = \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z}|\lambda)} \left[f(\mathbf{z})\nabla_\lambda \log p(\mathbf{z} \mid \lambda) \right].$$

To mitigate the typically high variance associated with stochastic score functions, the method draws K independent latent samples $\mathbf{z}_k$ for each input, evaluating a corresponding stack of reconstructed images $\hat{\mathbf{x}}_k$. In this framework, the mean absolute reconstruction error acts as the reward signal. For each sample k, the reward f_k is defined as:

$$f_k = \mathcal{L}_{\text{rec}}(\mathbf{x},\, \hat{\mathbf{x}}_k) = \frac{1}{3 \times H \times W} \sum_{c,i,j} \left| x_{c,i,j} - \hat{x}_{k,c,i,j} \right|.$$

To backpropagate this reward through the discrete layer, the backward pass tracks the analytical score function of the Poisson distribution:

$$\nabla_\lambda \log P(\mathbf{z} \mid \lambda) = \frac{\mathbf{z} - \lambda}{\lambda}.$$

To compute stable optimization updates, every individual sample must be evaluated against a baseline to determine its relative advantage. This work implements a *Leave-One-Out* [18] multi-sample baseline. For any given sample k within the ensemble, its baseline b_k is computed as the average performance realized by all other parallel trajectories in the batch:

$$b_k = \frac{1}{K-1} \sum_{j \neq k}^{K} f_j.$$

By subtracting this baseline from the individual sample's reward, the advantage signal $\mathbb{A}_k = f_k - b_k$ is effectively centered. The final gradient update for the encoder parameters is then formalized by weighting the score function with the centered advantage, averaged across the K iterations:

$$\nabla_\lambda \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z}|\lambda)}[f(\mathbf{z})] = \frac{1}{K} \sum_{k=1}^{K} \left[(f_k - b_k) \cdot \left(\frac{\mathbf{z}_k - \lambda}{\lambda} \right) \right].$$

This approach completely eliminates the need for an external critic or a parametrized baseline network, achieving variance reduction while remaining analytically consistent with the discrete Poisson setting.

Free Bits

To prevent early optimization collapse, where the network satisfies regularization constraints by instantly forcing all intensities to identical baseline levels before learning structural variations, this loss incorporates a regularization technique known as *Free Bits* [7]:

$$\mathcal{L}_{\text{KL\_free\_bits}} = \frac{1}{D} \sum_{i=1}^{D} \max\left(\tau,\, \mathcal{L}_{\text{KL}}(\lambda_i \parallel \lambda_{0,i})\right),$$

where the threshold parameter is set empirically to $\tau = 0.1$. By clamping the lower bound of the Kullback-Leibler penalty, the network is allowed to freely utilize a minimum informational capacity of 0.1 nats per latent dimension without penalty, protecting early spatial representations.

3.5.3 KL Divergence annealing schedule

Even with dedicated gradient estimators and the asymmetric encoder-decoder design, optimizing a discrete latent space remains highly susceptible to posterior collapse during the initial epochs of training. When the optimization process begins, the decoder is unoptimized and generates high reconstruction errors. If the full penalty of the Kullback-Leibler divergence (Equation 3.3) is applied immediately, the network often finds a trivial local minimum: it drives the intensity rates λ directly to the uniform prior target λ_0 to minimize the regularization loss to zero, effectively causing the encoder to ignore the input image $\mathbf{x}$ and trapping the latent space in an uninformative state. To mitigate this behavior across both the Straight-Through and Reinforcement Learning training variants, we implement a *Sigmoidal KL annealing* schedule. This setup dynamically modulates the regularization weight β at each epoch e using a logistic function to smooth the gradient updates. Let E_{cycle} represent the total length of the training cycle, and let $E_{\text{warm}} = 0.6 \cdot E_{\text{cycle}}$ denote the warm-up window covering the first 60% of the schedule. To centralize the sigmoid activation a modulation point e_{center}

and a normalized coordinate $x(e)$ are formulated as follows:

$$e_{\text{center}} = \frac{E_{\text{warm}}}{2}, \quad x(e) = \frac{e - e_{\text{center}}}{e_{\text{center}}}.$$

The regularization scaling factor $\beta(e)$ is then dynamically evaluated according to the following piece-wise function:

$$\beta(e) = \begin{cases} \dfrac{1}{1 + \exp\left(-10 \cdot x(e)\right)} & \text{if } e \leq E_{\text{warm}}, \\ 1.0 & \text{if } e > E_{\text{warm}}. \end{cases}$$

During the first stages ($e \to 0$, implying $x \to -1$), the term $\beta(e)$ is asymptotically suppressed close to zero. This configuration temporarily deactivates the KL term, allowing the VAE to operate similarly to a deterministic autoencoder. The encoder and decoder focus exclusively on minimizing the L_1 reconstruction loss. As training progresses toward the inflection point ($e \to e_{\text{center}}$, $x \to 0$), the growth rate increases rapidly but smoothly, introducing the Poisson regularizer. Finally, as the schedule approaches the boundary of the warm-up window ($e \to E_{\text{warm}}$, $x \to 1$), the sigmoidal derivative naturally decelerates, flattening asymptotically toward a stable ceiling of 1.0. This ensures smooth optimization without abrupt changes in the parameters λ, ensuring smooth convergence without sudden disruptions to the learned latent representations.

3.6 Architectural configurations

To evaluate the impact of parameter capacity on the synthesis quality of discrete Poisson VAEs, this work implements and compares three network configurations. Instead of altering the topology of the model, scaling is achieved exclusively by adjusting the number of feature maps across the internal convolutional layers of the framework.

The 60M parameter model: This implementation represents the primary model of this study, comprising approximately 60 million trainable parameters. The channel maps scale progressively from $16 \to 32 \to 64 \to 128 \to 256 \to 320$ within the downsampling stack and are progressively reduced through the sub-pixel upsampling stages of the decoder. This structure establishes an even balance between feature

extraction and image generation.

The 53M configuration: This variant introduces an aggressive parameter imbalance between the inference and generative components. Due to a reduction in the channel dimensions of the encoder, the inference network operates with a highly compact feature representation. Conversely, the synthesis network retains its full channel allocation, with the encoder accounting for approximately 7M parameters and the decoder retaining the remaining 46M.

The 36M configuration: The 36 million parameter version implements a uniform reduction of channel counts across all internal convolutional layers. This lightweight version serves as the baseline for ablation studies and hyperparameter tuning.

Chapter 4

Experiments

This chapter presents an empirical evaluation of the proposed Poisson Variational Autoencoder, validating the architectural designs, discrete sampling mechanisms and optimization strategies detailed in Chapter 3. Through a sequence of experiments conducted on the CelebA [8] face dataset, we assess whether replacing the conventional Gaussian latent space with a discrete Poisson distribution yields measurable benefits in terms of generation quality and latent space structure, while examining the practical costs introduced by non-differentiable sampling. The evaluation is structured around three primary directions. First, we establish a baseline comparison between the alternative gradient methodologies, contrasting the Poisson Gradient Approximation (PGA), the Score function estimator, implemented as the Reinforcement Learning Trick (RLT) and standard continuous reparameterization trick (GRT). Second, we analyze architectural scaling across our 36M, 53M and 60M parameter configurations, highlighting the performance difference using an asymmetric encoder-decoder configuration. Finally, we analyze the internal properties of the Poisson latent space through latent space traversals, attribute-driven UMAP clustering and linear direction manipulations.

4.1 Experimental setup

To ensure strict replicability and standardize the comparison across different training modalities and model capacities, a uniform experimental protocol was enforced throughout all training configurations.

4.1.1 Dataset

All experiments are conducted on the *CelebA* dataset [8], a large-scale collection of more than 200000 celebrity face images which exhibits a wide variety of facial poses, lighting configurations, backgrounds and diverse demographic settings, annotated across 40 binary attributes. The images are preprocessed with a center crop of 178×178 pixels, resized to a fixed resolution of 64×64 and normalized to the range $[-1, 1]$ to match the limits enforced by the final *tanh* activation layer of the decoder. The dataset is partitioned according to the official split, yielding approximately 162000 training images and 20000 validation images.

4.1.2 Evaluation metrics

Generation quality is assessed using the Fréchet Inception Distance (FID) [9], computed via the PyTorch implementation, between two sets of 10000 real and 10000 generated images. Lower FID scores indicate a higher perceptual similarity to the real data distribution. In addition to FID, training dynamics are monitored through the reconstruction loss (L_1) and the KL divergence term.

4.1.3 Training configuration

All models are optimized using the AdamW optimizer. The learning rate is set to 1×10^{-4}. The batch size is fixed to 128 across all experiments. Models are trained for a variable number of epochs depending on convergence, ranging from approximately 200 to 600 epochs, with checkpoints saved every 10 epochs. The latent space dimensionality is set to $D = 512$ for the majority of experiments. Configurations with $D = 256$ are also evaluated and $D = 128$ is explored exclusively within the 36M parameter models. The prior rate λ_0 is set to 4 as the primary configuration. The KL rescaling factor is the most extensively tuned hyperparameter, with values explored across multiple orders of magnitude to identify the optimal balance between reconstruction fidelity and regularization.

4.1.4 Implementation and hardware infrastructure

All experiments, model training phases and subsequent latent space evaluations are implemented in PyTorch [19] and executed on a dedicated worksta-

tion equipped with an NVIDIA Titan XP GPU, provided by the University of Pavia laboratory.

4.2 Baseline comparison

This section presents a comparative evaluation of the three gradient estimation strategies implemented in this work: the Poisson Gradient Approximation (PGA), the Score Function Estimator (RLT) and the Gaussian Reparameterization Trick (GRT). Rather than enforcing a strict iso-hyperparameter protocol, each method is evaluated under its individually optimized configuration, as preliminary tuning revealed that the optimal KL rescaling factor differs substantially across gradient estimation strategies. This choice reflects a realistic assessment of each method's best achievable performance, while the observed sensitivity to rescaling is itself treated as a relevant finding and discussed in Section 4.2.2.

4.2.1 Quantitative results

Table 4.1 reports the FID scores achieved by each method under both its best-FID configuration and the configuration that produced the most visually convincing results, which did not always coincide.

Method	**Config**	**Params**	**Rescale**	**Epochs**	**Lat. dim**	**FID**
GRT	Best FID	60M	1×10^{-6}	400	256	75.50
GRT	Best visual	60M	1×10^{-5}	400	256	98.25
PGA	Best FID	36M	1×10^{-2}	300	128	124.56
PGA	Best visual	60M	5×10^{-2}	600	512	152.99
RLT	Best LOO	60M	5×10^{-1}	100	512	385.34

Table 4.1: FID scores for the three gradient estimation strategies under best-FID and best-visual configurations. RLT is evaluated exclusively under the Leave-One-Out baseline as discussed in Chapter 3.

A notable discrepancy emerges between FID rankings and perceptual quality. The configurations achieving the lowest FID scores do not consistently correspond to those producing the most visually convincing face generations. This misalignment is particularly evident in the GRT method, where the best-FID model operates without KL annealing and converges to a regime that scores well statistically but produces less structured facial

features. This phenomenon is consistent with known limitations of FID as a perceptual metric [20].

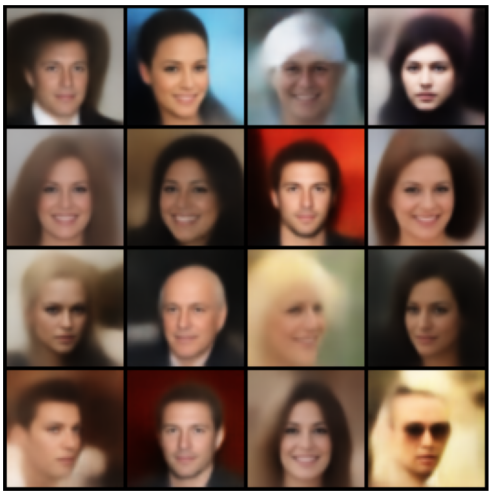


Figure 4.1: Faces generated by the 60M GRT model (best-FID configuration, rescale $= 1 \times 10^{-6}$, FID $= 75.70$).

Figures 4.1, 4.2 and 4.3 show representative faces generated by the best-FID configuration of each method. A visual comparison with the best-visual configurations presented in Section 4.5 further illustrates the discrepancy between FID rankings and perceptual quality.

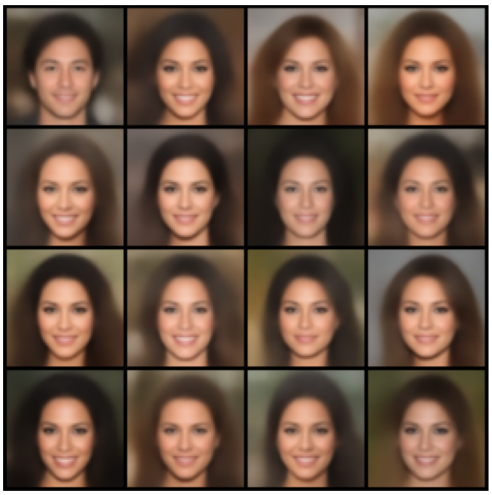


Figure 4.2: Faces generated by the 36M PGA model (best-FID configuration, rescale $= 1 \times 10^{-2}$, FID $= 124.56$).

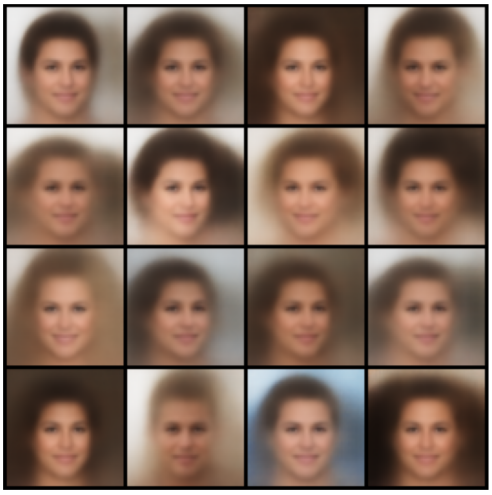


Figure 4.3: Faces generated by the 60M RLT model (best-LOO configuration, rescale $= 5 \times 10^{-1}$, FID $= 385.34$).

4.2.2 Rescaling sensitivity

A significant observation emerges from the hyperparameter tuning process. The GRT baseline achieves its lowest FID at a KL rescaling factor of 1×10^{-6}, whereas PGA performs optimally at 1×10^{-2}, a difference of four orders of magnitude. This suggests that operating in a discrete Poisson latent space alters the optimization context relative to the continuous Gaussian case. The Poisson KL divergence grows at a different rate than its Gaussian counterpart as the latent distribution deviates from the prior, requiring a larger rescaling coefficient to maintain a comparable balance between regularization and reconstruction. This sensitivity constitutes an implicit practical cost of adopting a discrete latent distribution.

A further consequence of the rescaling factor concerns the perceptual trade-off between sample diversity and facial detail quality, observed across PGA configurations. Higher rescaling values impose a stronger KL penalty, encouraging the encoder to produce rate vectors λ closer to the prior λ_0, which increases the diversity of generated faces but introduces a degree of blurriness in the facial details. Conversely, lower rescaling values relax the regularization pressure, allowing the encoder to encode more discriminative information into the latent representation and yielding sharper facial features at the cost of reduced generative diversity. Figure 4.4 illustrates this trade-off across PGA configurations.

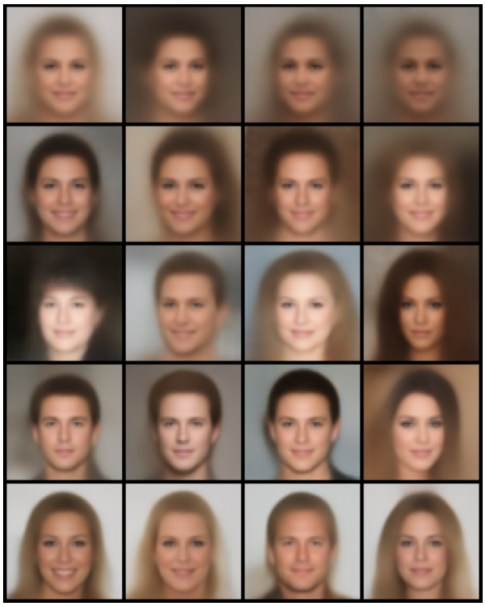


Figure 4.4: Generated faces across increasing KL rescaling values for the PGA model. Each row corresponds to a different rescaling configuration, in decreasing order: $\beta = 5, 1, 5 \times 10^{-1}, 1 \times 10^{-1}, 1 \times 10^{-2}$. All rows are decoded from the same latent vector $\mathbf{z}$, sampled once from $\mathcal{P}(\lambda_0 = 4)$. Higher rescale values produce more diverse but less detailed outputs, while lower values yield sharper faces with reduced diversity.

4.2.3 Score function estimator results

The RLT method consistently underperforms in relation to both PGA and GRT in all evaluated configurations, with FID scores ranging from **385.34** to **415.58** under the Leave-One-Out baseline. As Figure 4.3 shows, generated faces exhibit blurriness, reflecting the high variance inherent to the score function estimator. Despite the theoretical unbiasedness of the REINFORCE gradient and the variance reduction provided by the leave-one-out baseline, optimization remains unstable and convergence is substantially slower compared to the straight-through approach. These results confirm that, in the context of discrete Poisson VAEs, the practical trainability of the score function estimator remains a significant limiting factor.

4.3 Architectural scaling

This section examines the impact of parameter capacity on generation quality across the three architectural configurations introduced in Chapter 3: the 36M, 53M and 60M parameter models. All configurations are evalu-

ated using the PGA gradient estimator with $\lambda_0 = 4$, isolating the effect of architectural scaling from the choice of gradient estimation strategy.

4.3.1 Parameter distribution

Despite sharing the same topological structure, the three configurations exhibit substantially different distributions of parameters between the inference and generative networks, as summarized in Table 4.2.

Configuration	Encoder	Decoder	Ratio (Enc/Dec)
36M	∼6M	∼30M	1:5
53M	∼7M	∼46M	1:6.6
60M	∼30M	∼30M	1:1

Table 4.2: Approximate parameter distribution between encoder and decoder across the three architectural configurations, evaluated with $D = 512$

The 60M model achieves a balanced configuration, allocating approximately equal capacity to feature extraction and image synthesis. In contrast, the 53M configuration introduces a pronounced asymmetry, with the decoder holding approximately 86% of the total parameter budget. The 36M model presents an intermediate imbalance, consistent with a lightweight design optimized for efficiency.

4.3.2 FID evaluation

Table 4.3 reports the FID scores achieved by representative configurations of each model size.

Model	Rescale	Epochs	Lat. dim	FID
36M	1×10^{-2}	300	128	124.56
36M	1×10^{-2}	600	256	125.05
36M	1×10^{-2}	300	512	128.78
53M	1×10^{-2}	500	512	159.89
53M	5×10^{-2}	600	512	164.55
60M	1×10^{-1}	600	512	152.99
60M + annealing	1×10^{-1}	200	512	141.32

Table 4.3: FID scores for PGA models across architectural configurations.

4.3.3 Discussion

The results reveal that there is no direct relationship between total parameter count and generation quality. The 60M balanced configuration achieves the best FID among the larger models and produces the highest perceptual quality, confirming that symmetric encoder-decoder capacity is beneficial for discrete Poisson VAEs. The 36M lightweight model achieves surprisingly competitive FID scores, suggesting that within the PGA framework a compact architecture can be an effective alternative to larger configurations. The most notable finding concerns the 53M configuration. Despite holding more total parameters than the 36M model, it consistently underperforms in terms of FID. We attribute this behavior to the severe encoder-decoder imbalance: with only $\sim$7M parameters, the inference network lacks the representational capacity to produce sufficiently informative rate vectors λ, regardless of the decoder's generative power. This suggests that in a discrete Poisson VAE, the quality of the latent representation is bottlenecked by encoder capacity, and that scaling the decoder disproportionately yields diminishing returns. This finding motivates the balanced design adopted in the 60M primary configuration.

4.4 Latent space analysis

Beyond generation quality metrics, this section investigates the internal structure of the learned Poisson latent space. All analyses are conducted on the 60M parameter models, using the PGA visually preferred configuration as the primary reference unless otherwise specified.

4.4.1 Latent space traversal

To assess the informational content encoded across all latent dimensions, we perform a systematic traversal of the latent space. Starting from a base encoding $\lambda = \text{Enc}(\mathbf{x})$, each dimension $d \in \{1, \ldots, D\}$ is independently swept across a fixed range of values while all other dimensions are held constant. To preserve only the most semantically meaningful dimensions, a difference threshold $\tau = 0.1$ is applied; a dimension is considered active if the mean absolute pixel difference between the first and last decoded image in the sweep exceeds τ. A lower threshold of $\tau = 0.05$ was initially explored but yielded an excessive number of active dimensions, motivating the more conserva-

Figure 4.5: Latent space traversal across active dimensions ($\tau = 0.1$) for the 60M PGA model trained for 200 epochs with 512 latent dimensions, a prior $\lambda_0 = 4$, rescale set to 1×10^{-1} and KL annealing. Each row corresponds to an active latent dimension, swept from 0 to 100 across 10 uniformly spaced steps.

tive choice. Under this criterion, a substantial subset of latent dimensions passes the threshold in the 60M PGA model, confirming that the discrete Poisson bottleneck does not suffer from posterior collapse, since no dead dimensions are observed. As we can see in Figure 4.5, the active dimensions primarily encode low-level visual properties such as color tone, brightness and illumination direction. Higher-level semantic variations, including pose changes, hair color, gender appearance, are observed only in a minority of dimensions. This is consistent with the low spatial resolution of the dataset (64×64), which limits high-frequency semantic structures.

Figure 4.6 presents the analogous latent traversal for the GRT model. Since the Gaussian latent space is parameterized by a mean vector μ and a log-variance vector $\log \sigma^2$, the traversal is performed by independently varying each active dimension of μ while keeping $\log \sigma^2$ fixed. The sweep range spans $[-5, 5]$ across 11 uniformly spaced steps, allowing the exploration of both positive and negative deviations from the learned mean. In contrast to the Poisson traversal, several dimensions of the Gaussian latent space exhibit clear semantic symmetry around zero: traversing a dimension in the negative direction produces the opposite visual effect relative to the positive direction. Representative examples include head orientation (leftward vs. rightward pose), background color (cool vs. warm tones), hair color (dark vs. blonde) and illumination intensity (underexposed vs. overexposed).

This symmetric structure is a direct consequence of the zero-centered Gaussian prior, which encourages the latent space to organize generative factors around a natural origin.

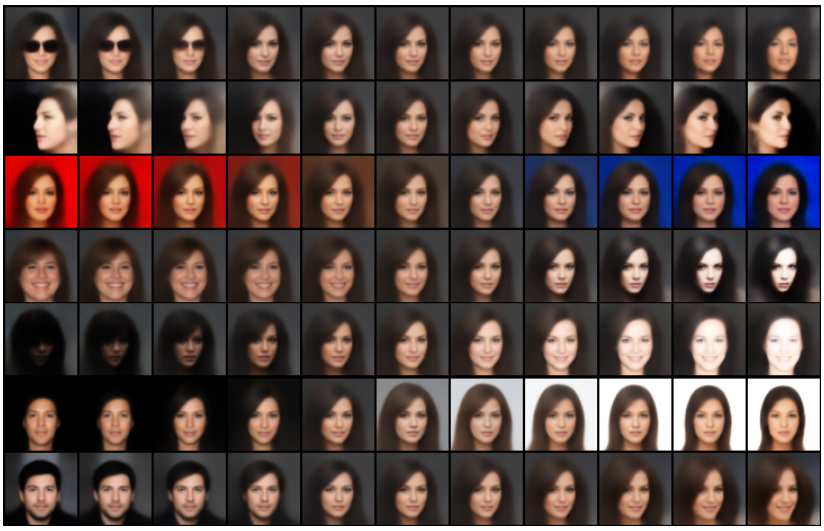


Figure 4.6: Latent space traversal across active dimensions ($\tau = 0.1$) for the 60M GRT model. Each row corresponds to an active latent dimension, swept from -5 to 5 across 11 uniformly spaced steps on the mean vector μ.

4.4.2 UMAP clustering

To evaluate the global geometric structure of the latent space, we apply UMAP dimensionality reduction [21] to 5000 encoded representations drawn from the CelebA validation set, using $n\_\text{neighbors} = 50$ and $\text{min}\_\text{dist} = 0.0$ to encourage tight cluster formation. The resulting two-dimensional embeddings are visualized across four separate plots, one per attribute (*Male*, *Smiling*, *Blond_Hair* and *Young*), where each point is colored according to the binary attribute value of the corresponding image. Figure 4.7 shows the UMAP projections for the GRT baseline, which produces the most geometrically coherent embeddings across all four attributes. The PGA model (Figure 4.8) yields more mixed results. Some attributes exhibit meaningful separation, while others produce geometrically irregular projections. We interpret this as a consequence of the discrete Poisson bottleneck, whose non-negative integer support induces a different latent geometry compared to the continuous Gaussian case. The RLT model (Figure 4.9) produces the least structured embeddings, with weak attribute separation across all four projections, consistent with its lower generation quality and higher optimization variance.

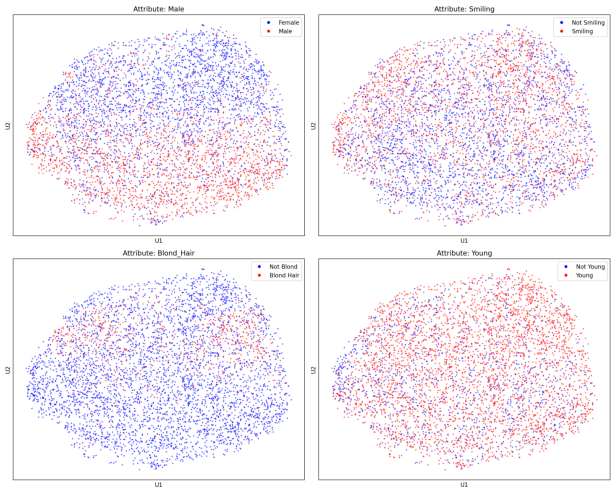


Figure 4.7: UMAP projections of 5000 validation encodings for the 60M GRT model trained for 100 epochs with 256 latent dimensions, rescale set to 1×10^{-5} and KL annealing, colored by binary attribute (Male, Smiling, Blond Hair, Young).

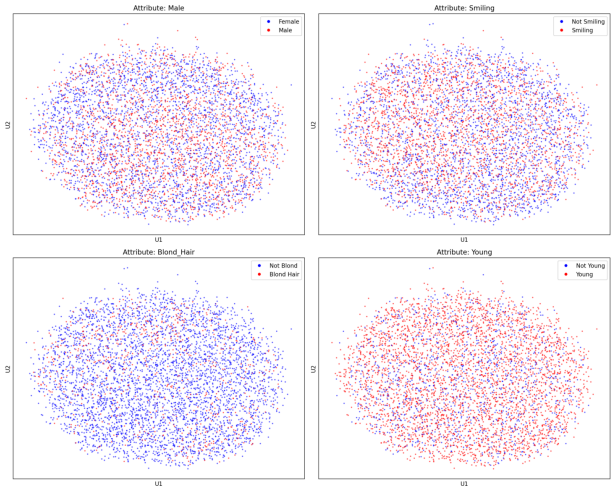


Figure 4.8: UMAP projections of 5000 validation encodings for the 60M PGA model trained for 200 epochs with 512 latent dimensions, a prior $\lambda_0 = 4$, rescale set to 5×10^{-2} and KL annealing. colored by binary attribute (Male, Smiling, Blond Hair, Young).

4.4.3 Attribute direction manipulation

To evaluate the disentanglement of the latent space, we compute linear attribute directions following the approach of [22]. For each binary attribute a, the direction vector $\mathbf{v}_a$ is computed as the difference between the mean encodings of attribute-positive and attribute-negative validation samples:

$$\mathbf{v}_a = \mathbb{E}[\lambda \mid a = 1] - \mathbb{E}[\lambda \mid a = 0].$$

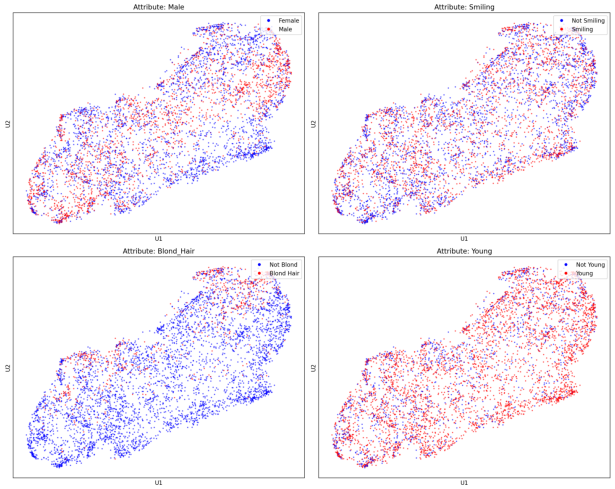


Figure 4.9: UMAP projections of 5000 validation encodings for the 60M RLT model trained for 100 epochs with 512 latent dimensions, a prior $\lambda_0 = 4$, rescale set to 1 and $K = 2$. colored by binary attribute (Male, Smiling, Blond Hair, Young).

Manipulated samples are then generated by displacing the base encoding along the attribute direction:

$$\lambda_{\text{manip}} = \lambda + \alpha \cdot \mathbf{v}_a, \quad \alpha \in [\alpha_{\min}, \alpha_{\max}]$$

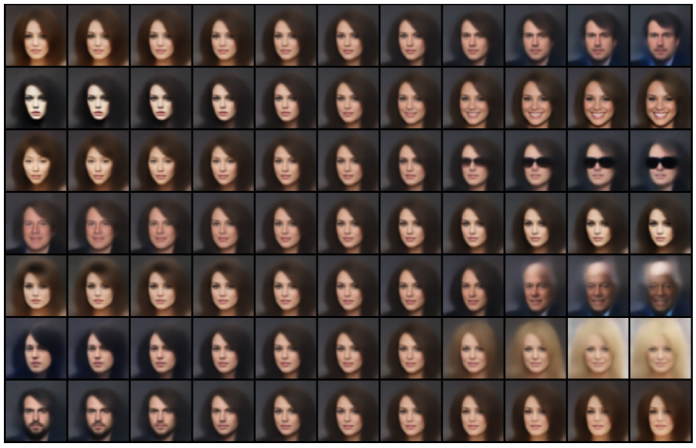


Figure 4.10: Attribute direction manipulations applied to a base encoding for the 60M PGA model trained for 200 epochs with 512 latent dimensions, a prior $\lambda_0 = 4$, rescale set to 5×10^{-2} and KL annealing. Each row corresponds to a different attribute, in order: *Male*, *Smiling*, *Eyeglasses*, *Young*, *Bald*, *Blond_Hair*, *No_Beard*. Each column display the base encoding plus $\alpha \cdot \mathbf{v}_a$, where α varies linearly from -5 to 5 from left to right.

Figure 4.10 demonstrates how the PGA model responds consistently to

attribute manipulations across all tested directions (*Male*, *Smiling*, *Eyeglasses*, *Young*, *Bald*, *Blond_Hair*, *No_Beard*), producing smooth and visually coherent transitions. The sole exception is the *Bald* attribute, which produces unconvincing manipulations likely due to the limited spatial resolution of the dataset making hair removal a particularly challenging transformation.

Figure 4.11 presents the attribute direction manipulations for the GRT model. The results are qualitatively comparable to those obtained with the PGA model, with coherent transitions observed across all tested attributes with the exception of *Bald*, which remains challenging across both models. However, the effective range of the scaling factor is substantially narrower: while the PGA model accommodates $\alpha \in [\alpha_{\min}, \alpha_{\max}]$, the Gaussian latent space saturates rapidly beyond $\alpha \in [-3, 3]$, producing extreme visual artifacts at larger displacements. This further corroborates the observation made in the traversal analysis: the Gaussian latent space is more sensitive to deviations from the learned mean than the discrete Poisson space, reflecting the different geometric properties of the two priors.

The RLT model produces less interpretable manipulations, as seen in Figure 4.12, consistent with its weaker latent structure.

Figure 4.11: Attribute direction manipulations applied to a base encoding for the 60M GRT model trained for 400 epochs with 256 latent dimensions, rescale set to 1×10^{-5} and KL annealing. Each row corresponds to a different attribute, in order: *Male*, *Smiling*, *Eyeglasses*, *Young*, *Bald*, *Blond_Hair*, *No_Beard*. Each column represents a uniformly spaced value of $\alpha \in [-3, 3]$.

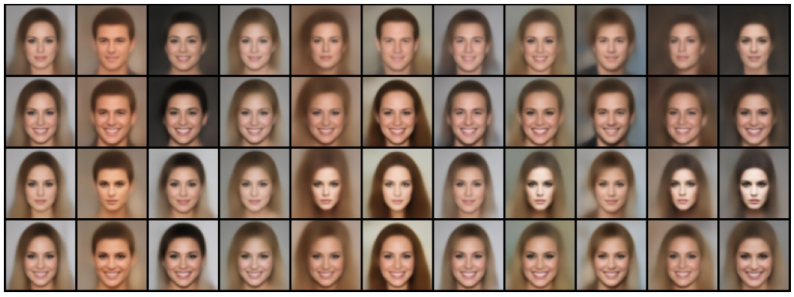
Figure 4.12: Attribute direction manipulations applied to a base encoding for the 60M RLT model trained for 100 epochs with 512 latent dimensions, rescale set to 1 KL annealing and free bits set at $\tau = 0.1$. Each row corresponds to a different attribute, in order: *Male*, *Smiling*, *Eyeglasses*, *Young*, *Bald*, *Blond_ Hair*, *No_ Beard*. Each column display the base encoding plus $\alpha \cdot \mathbf{v}_a$, where α varies linearly from -5 to 5 from left to right.

4.4.4 Attribute arithmetic

To further probe the compositional structure of the learned latent space, we evaluate whether multiple attribute directions can be combined additively to produce coherent multi-attribute manipulations. For each combination, we construct a four-row grid showing the base encoding, the effect of each attribute applied individually, and their linear combination:

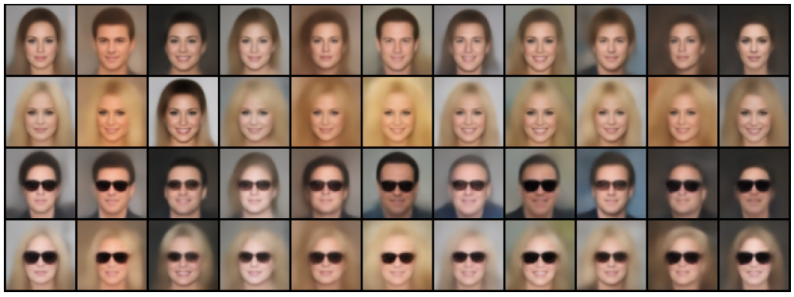
$$\lambda_{\text{combined}} = \lambda + \alpha \cdot \mathbf{v}a_1 + \alpha \cdot \mathbf{v}_{a_2}.$$

All experiments are conducted on the 60M PGA and GRT model variants with a fixed $\alpha =$ **[3]**. Figure 4.13 presents the results relative to the PGA model across three attribute combinations. The *Young + Smiling* combination produces visually convincing outputs, yielding faces that simultaneously exhibit both attributes. The *Male + No_ Beard* combination successfully maintains the male appearance while removing facial hair, with only marginal exceptions. The *Blond_ Hair + Eyeglasses* combination produces faces consistently wearing eyeglasses, with the hair color manipulation also applied. A consistent observation across all three combinations is that several attribute directions carry an implicit gender component, systematically shifting generated faces toward a female appearance when applied individually. This behavior is attributable to demographic correlations present



(a) Attribute combination: *Smiling* + *Young*

(b) Attribute combination: *Male* + *No_Beard*



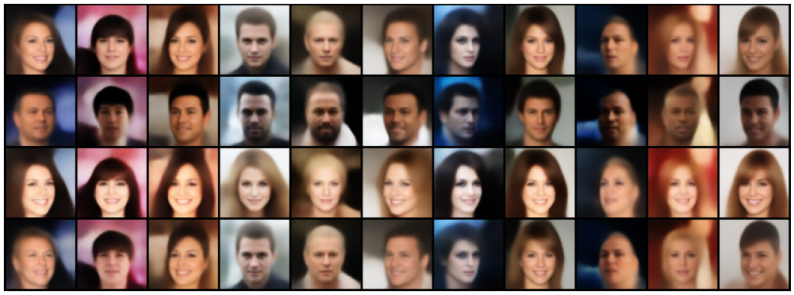
(c) Attribute combination: *Blond_Hair* + *Eyeglasses*

Figure 4.13: Attribute arithmetic on the 60M PGA model trained for 600 epochs with 512 latent dimensions, a prior $\lambda_0 = 4$, rescale set to 1×10^{-1} and without KL annealing. Each group of four rows shows: base encoding, first attribute applied individually, second attribute applied individually, and both attributes combined. Combinations tested: *Smiling* + *Young*, *Male* + *No_Beard*, *Blond_Hair* + *Eyeglasses*.

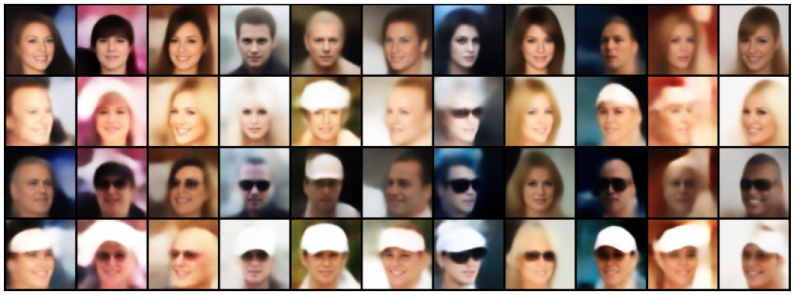
in the CelebA dataset, where attributes such as *Smiling*, *Blond_Hair* and *No_Beard* are disproportionately associated with female subjects in the training partition. The model has correctly learned these statistical dependencies, which manifest as entangled directions in the latent space. Despite this entanglement, the additive combinations produce semantically coherent outputs in all three cases, confirming that the discrete Poisson latent space

supports linear compositionality of semantic directions.

(a) Attribute combination: *Smiling* + *Young*



(b) Attribute combination: *Male* + *No_Beard*



(c) Attribute combination: *Blond_Hair* + *Eyeglasses*

Figure 4.14: Attribute arithmetic on the 60M GRT model trained for 400 epochs with 256 latent dimensions, rescale set to 1×10^{-5} with KL annealing. Each group of four rows shows: base encoding, first attribute applied individually, second attribute applied individually, and both attributes combined. Combinations tested: *Smiling* + *Young*, *Male* + *No_Beard*, *Blond_Hair* + *Eyeglasses*.

Figure 4.14 presents the attribute arithmetic results for the GRT model. The compositions produce less consistent outputs compared to the PGA model, with attribute entanglement more frequently disrupting the expected combinations. The *Smiling* + *Young* combination yields the most coherent result: the *Smiling* direction successfully applies to both male and female

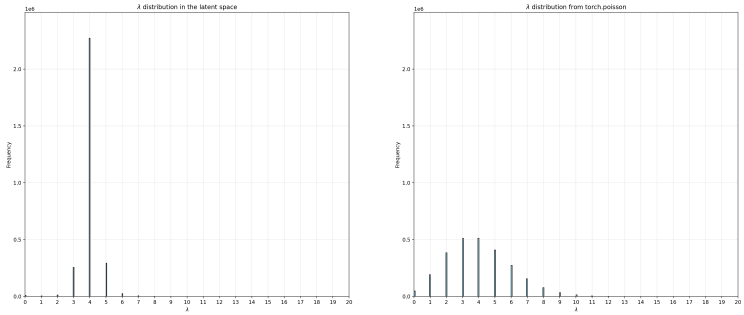
subjects independently, while *Young* exhibits a correlation with female appearance. Their combination consistently produces young female faces, with the gender bias introduced by *Young* dominating the composition. The *Male + No_Beard* combination shows partial success: the *Male* direction reliably shifts generated faces toward male appearance, while *No_Beard* is correlated with female subjects. The combined manipulation fails to consistently produce male faces, with a non-negligible fraction of outputs retaining female appearance, suggesting that the two directions partially interfere in the Gaussian latent space. The *Blond_Hair + Eyeglasses* combination exhibits the most pronounced limitations. The *Blond_Hair* direction introduces color saturation artifacts, producing female faces with visually overexposed hair. The *Eyeglasses* direction carries a mild male correlation, and the combined manipulation fails to consistently place eyeglasses on the generated faces. These results suggest that the Gaussian latent space encodes certain attribute directions with greater entanglement than the Poisson counterpart, particularly for attributes that are strongly correlated with gender in the CelebA training distribution.

4.4.5 Rate distribution analysis

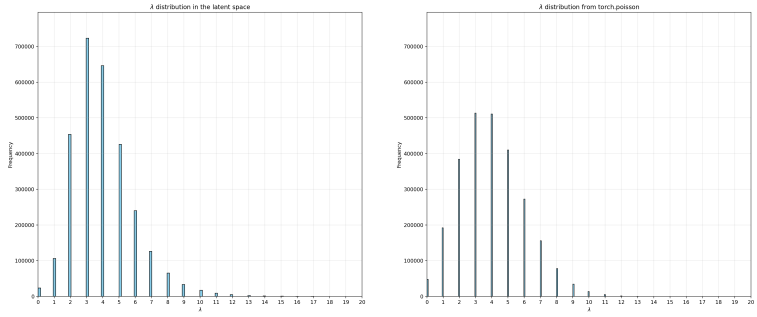
To verify that the learned prior aligns with the target distribution, we compare the empirical distribution of encoded rates against the reference Poisson distribution $\mathcal{P}(\lambda_0)$. Figure 4.15 shows the histogram of encoder outputs across 5000 validation images alongside the reference distribution. The PGA model trained with KL annealing produces a rate distribution that closely follows the expected Poisson shape, confirming that the annealing schedule successfully regularizes the latent space without collapsing it. In contrast, the RLT model produces a poorly calibrated rate distribution even when trained with KL annealing, further corroborating the optimization difficulties observed in the previous analyses.

4.5 Qualitative results

To complement the quantitative FID evaluation, this section presents a visual inspection of faces generated by the three gradient estimation strategies. Figures 4.16, 4.17 and 4.18 show representative samples generated by the 60M PGA, GRT and RLT models respectively. The GRT model exhibits the highest generative diversity, producing faces with a wide vari-



(a) Without KL annealing: the distribution collapses to $\lambda = 4$.



(b) With KL annealing: the distribution follows the reference Poisson prior $\mathcal{P}(\lambda_0 = 4)$.

Figure 4.15: Empirical rate distributions produced by the PGA encoder on the validation set (left), compared against the reference Poisson distribution (right). Top: model trained for 600 epochs with 512 latent dimensions, a prior $\lambda_0 = 4$, rescale set to 1 and without KL annealing. Bottom: model trained for 200 epochs with 512 latent dimensions, a prior $\lambda_0 = 4$, rescale set to 5×10^{-2} and KL annealing.

ety of identities, poses and lighting conditions. However, fine-grained facial details such as eye structure, hair texture and skin boundaries appear less sharp compared to the Poisson-based models, suggesting that the continuous Gaussian latent space favors broad coverage of the data distribution at the cost of local detail.

The PGA model produces faces with noticeably higher facial detail quality. This comes at the cost of a reduced perceptual diversity across samples, with generated faces appearing somewhat more homogeneous in terms of identity and pose. This trade off is consistent with the structural properties of the discrete Poisson latent space, which imposes a more constrained prior over the latent representations. The RLT model produces visibly blurry outputs across all samples, with poorly defined facial structure and absent

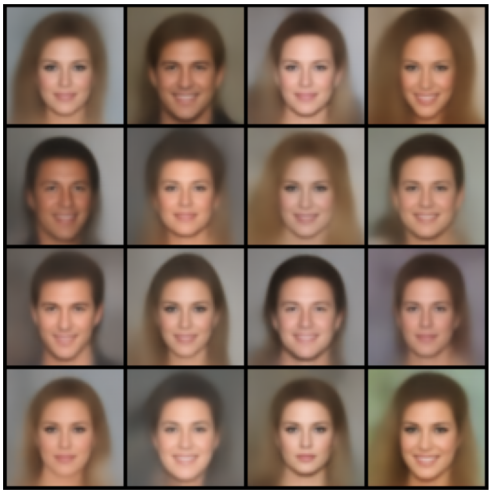


Figure 4.16: Faces generated by the 60M PGA model trained for 600 epochs with 512 latent dimensions, a prior $\lambda_0 = 4$, rescale set to 5×10^{-2} and without KL annealing.

fine details. This confirms the quantitative findings reported in Section 4.2, where RLT consistently achieved the highest FID scores, and is attributable to the high variance of the score function gradient estimator limiting the model's ability to learn sharp generative factors.

Figure 4.17: Faces generated by the 60M GRT model trained for 400 epochs with 256 latent dimensions, rescale set to 1×10^{-5} with KL annealing.

Figure 4.19 shows the pure reconstruction outputs of the RLT model, where the latent vector is obtained by directly encoding a validation image rather than sampling from the prior. The improved sharpness relative to the generated faces confirms that the decoder retains sufficient capacity for image synthesis, and that the quality degradation observed in generation is

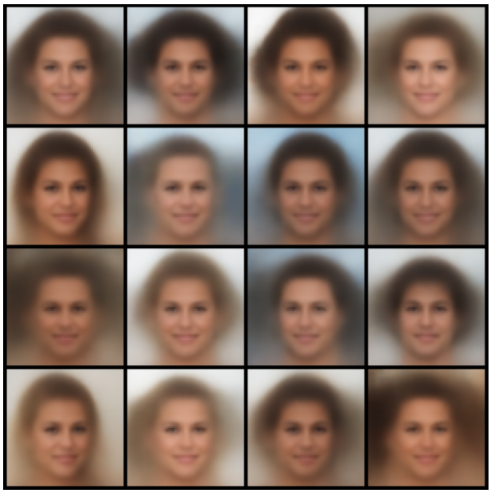


Figure 4.18: Faces generated by the 60M RLT model trained for 100 epochs with 512 latent dimensions, a prior $\lambda_0 = 4$, rescale set to 1, k set to 2 using a Leave-One-Out baseline with KL annealing and free bits set to 0.1.

attributable to the optimization difficulties of the score function estimator rather than to architectural limitations.

Figure 4.19: Pure reconstructions produced by the 60M RLT model trained for 100 epochs with 512 latent dimensions, rescale set to 1 with KL annealing and free bits set to 0.1.

Chapter 5

Conclusions

This thesis has investigated the feasibility and properties of a Variational Autoencoder operating with a discrete Poisson latent space, addressing the fundamental challenge of backpropagating through a non-differentiable sampling operator. The work has produced a set of concrete findings across gradient estimation, architectural design and latent space analysis.

5.1 Summary of contributions

The primary contribution of this thesis is the design and implementation of a Poisson VAE framework supporting two alternative gradient estimation strategies. The Poisson Gradient Approximation (PGA) approximates the discrete sampling backward pass via a Gaussian surrogate gradient, while the Score Function Estimator (RLT) approaches the problem from a reinforcement learning perspective using a REINFORCE estimator with a leave-one-out baseline. Both strategies are evaluated against a standard Gaussian reparameterization baseline (GRT) on the CelebA face dataset. On the architectural side, an asymmetric encoder-decoder design is proposed, combining an EfficientNet-inspired inference network with an SRGAN-inspired generative decoder equipped with AdaIN-based style modulation. Training stability is ensured through a sigmoidal KL annealing schedule and a free bits regularization strategy.

5.2 Key findings

The experimental evaluation produced several findings of note. The PGA gradient estimator achieves competitive generation quality relative to the

Gaussian baseline, producing faces with higher perceptual sharpness at the cost of reduced diversity. This demonstrates that a discrete Poisson prior can sustain a structured and semantically meaningful latent space, as confirmed by latent traversals, UMAP clustering and attribute direction manipulations. A structurally significant observation concerns the sensitivity of the KL rescaling factor across gradient estimation strategies. The GRT baseline achieves its best FID at a rescaling value of 1×10^{-6}, whereas PGA performs optimally at 1×10^{-2}, a difference of four orders of magnitude. This divergence suggests that the discrete Poisson latent space induces a fundamentally different optimization context relative to its continuous Gaussian counterpart. The architectural scaling experiments revealed a non-monotonic relationship between total parameter count and generation quality. The 53M configuration, despite holding more parameters than the 36M model, consistently underperforms due to a severe encoder-decoder imbalance. With only $\sim 7M$ parameters allocated to the inference network, the encoder lacks the representational capacity to produce sufficiently informative rate vectors λ, regardless of the decoder's generative power. This finding confirms that in discrete Poisson VAEs, generation quality is bottlenecked by encoder capacity rather than total parameter count. The UMAP analysis further revealed that the Poisson latent space exhibits geometrically irregular cluster structures, absent in the Gaussian baseline. This constitutes an open and interesting property of discrete non-negative latent spaces that justifies further investigation.

5.3 Limitations

The primary limitation of this work concerns the Score Function Estimator. The RLT method was evaluated exclusively with $K = 2$ parallel samples due to the substantial computational overhead introduced by multi-sample REINFORCE training. Higher values of K are expected to reduce gradient variance and improve convergence, but were not feasible within the available training budget on the NVIDIA Titan XP GPU. As a consequence, the RLT results reported in this thesis likely underestimate the method's potential. A secondary limitation concerns the evaluation metric. FID provides a useful aggregate measure of generation quality but has known limitations at low spatial resolutions and does not always correlate with perceptual quality, as observed in our experiments. This misalignment was particularly evident

in the GRT models, where the best-FID configuration did not correspond to the most visually convincing outputs.

5.4 Future work

Several directions are identified for future investigation. First, training the RLT estimator with larger ensemble sizes ($K > 2$) on more powerful hardware would provide a more faithful assessment of the score function approach and potentially close the performance gap with PGA. Second, the adoption of a perceptual loss function tailored to facial structure, such as a face recognition-based metric, could better align the optimization objective with the visual criteria relevant to face synthesis, addressing the FID misalignment observed in this work. Third, the irregular geometric structures observed in the Poisson latent space, represent an open theoretical question. Understanding whether these structures are an intrinsic property of discrete non-negative priors or an artifact of the gradient approximation could provide valuable insight into the geometry of discrete latent spaces. A further direction concerns the choice of the prior rate λ_0. All experiments in this thesis use $\lambda_0 = 4$, a relatively low value. Increasing λ_0 would reduce the relative variance of the Poisson distribution, potentially improving the quality of the Gaussian surrogate approximation used in the PGA estimator, since the Poisson distribution becomes increasingly symmetric and bell-shaped as λ_0 grows. Investigating how higher prior rates affect both the approximation quality and the generative diversity of the learned latent space represents a natural extension of this work. Finally, scaling the framework to higher image resolutions (128×128 or 256×256) would allow a more thorough assessment of the Poisson prior's ability to capture fine-grained facial attributes, which are unresolvable at 64×64.

Bibliography

[1] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

[2] Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL conference on computational natural language learning*, pages 10–21, 2016.

[3] Samuel W. Hasinoff. *Photon, Poisson Noise*, pages 608–610. Springer US, Boston, MA, 2014.

[4] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[5] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[6] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017.

[7] Diederik P. Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems (NIPS)*, volume 29, 2016.

[8] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[9] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[10] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, 1985.

[11] Lilian Weng. From autoencoder to beta-vae. Lil'Log, August 2018.

[12] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[13] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

[14] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[16] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016.

[17] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[18] Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 reinforce samples, get a baseline for free! 2019.

[19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035, 2019.

[20] Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: Towards a better evaluation metric for image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9307–9315, 2024.

[21] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861, 2018.

[22] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2016.