Generating Realistic Faces: Leveraging VAE, GAN, and a Hybrid Approach

Chengming Wang, Jiahui Ren, Kexin Yuan, Liuzhenhan Song, Xiangyu Zeng, Xinrong Zhou, and Zhiang Yuan

Department of Statistics, The Chinese University of Hong Kong

Dec 10, 2024

Abstract

Recent advancements in generative models have significantly transformed the landscape of artificial intelligence, particularly in image synthesis and representation learning. The rapid advancement of computing power has brought image processing to the forefront by being applied to various fields, there is a growing demand for artificial faces driven by privacy concerns. This project investigates how to generate novel artificial faces from established face datasets by applying three different generative model architectures: Variational Autoencoder (VAE), Generative Adversarial Network (GAN), and a hybrid model that integrates both frameworks. The performance and effectiveness of these models are thoroughly compared to evaluate their capabilities in face generation. Some interpretations and further facial interpolations are also discussed.

Contents

1 Introduction

Rapidly developing computing power brings image processing to center stage. In real-life applications such as character face generation in realistic games and psychological research on emotion recognition and interpersonal communication, increasing demand for artificial new faces is seen due to privacy concerns. This leads us to our goal: to simulate entirely new and realistic faces by training models using existing images.

In the field of image processing, autoencoders are the most basic machine learning models. However, the limitation of only being able to reconstruct faces without imposing any structure on the latent space makes it challenging to generate new samples by simply sampling from the latent space. To achieve our goal of generating new and realistic faces, we mainly focus on variational autoencoder(Rezende and Mohamed, 2015) and generative adversarial networks (Goodfellow et al., 2014) based models in this project. We are aware of other generative models, such as diffusion models (Sohl-Dickstein et al., 2015), normalizing flows(Rezende and Mohamed, 2015), and energy-based models (LeCun et al., 2006). But the sampling costs of diffusion models, normalizing flows and energy-based models are much higher (Murphy, 2023). Meanwhile, diffusion models often demonstrate superior generation quality than Energy-Based Models and Normalizing Flows (Cao et al., 2024). So we will focus on VAEs and GANs in this project. But due to the time constraint and computational cost, we may not be able to cover these models in this project. Nevertheless, VAE and GAN based models already give us a lot of room to explore. And we consider extending our project to these models, especially diffusion models, in the future.

The rest of the report is organized as follows. In Section 2, we will introduce our dataset and the notation used throughout this project. We review VAE, GAN, and a combined model of VAE and GAN in Section 3. In Section 4, the generating results and model evaluation will be presented. In Section 5, we shall discuss the latent arithmetic. And we conclude our project in Section 6.

Through these methods, we hope to effectively simulate entirely new and realistic faces, providing more possibilities and solutions for related applications.

2 Dataset and Notation

2.1 Dataset and Pre-processing

To train our generative models, we downloaded the raw data from Large-scale CelebFaces Attributes (CelebA) Dataset. Celeba Dataset is a facial dataset developed by researchers from The Chinese University of Hong Kong to help train and test computer vision tasks such as facial analysis, facial attribute recognition, facial detection, face synthesis and face editing.

The dataset has large diversities, large quantities, and rich annotations, consisting of 202,599 images and 10,177 identity labels, each with 40 binary annotations such as mustache, hair color, and the shape of face.

We downloaded the Align&Cropped Images version, in which the raw images were first roughly aligned using a similarity transformation according to the two eye locations, and then resized to $218 * 178$. In consideration of computational cost, we further center-cropped and

resized the images to $64 * 64$.

2.2 Preliminary Visualization

The images below were pre-processed and would be proceeded as training dataset.

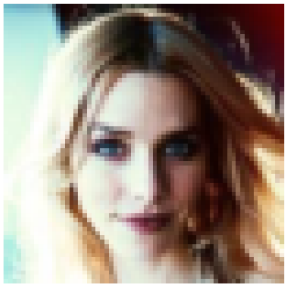

Figure 2.1

Figure 2.2

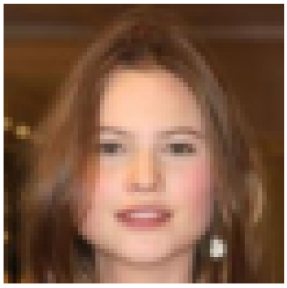

Figure 2.3

2.3 Notations

We introduce several general notations used throughout this project. Let N denote the sample size of the dataset, $\boldsymbol{x} = (x_1, \cdots, x_N)$ denote the input data, $\boldsymbol{z} = (z_1, \cdots, z_N)$ denote the latent variables, where $x_i \in \mathbb{R}^d$, $z_i \in \mathbb{R}^m$, $0 < m < d$. In our case, $d = 64 * 64$. And the loss function is denoted as $\mathscr{L}$.

3 Methodology

3.1 VAE

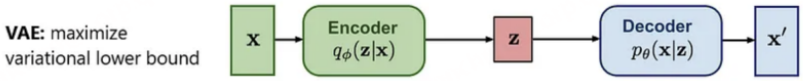

Figure 3.1: Illustration for VAE (Source).

In brief, VAE includes two main parts. First is mapping the dataset to a latent space by the encoder, and then do the opposite, which is mapping the latent space back to the input space. We have the following model assumptions:

$$\text{prior: } \boldsymbol{z} \sim p_{\boldsymbol{\theta}}(\boldsymbol{z}), \tag{3.1}$$

$$\text{likelihood: } \boldsymbol{x}|\boldsymbol{z} \sim p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}), \tag{3.2}$$

$$\text{posterior: } \boldsymbol{z}|\boldsymbol{x} \sim p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}), \tag{3.3}$$

$$\text{approximated posterior: } \boldsymbol{z}|\boldsymbol{x} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}), \tag{3.4}$$

where the parameters of $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ are computed by the neural network decoder $d_{\boldsymbol{\theta}}(\boldsymbol{z})$, and the parameters of $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ are computed by the neural network encoder $e_{\boldsymbol{\phi}}(\boldsymbol{z})$. The approximated posterior is used since the true posterior distribution is sometimes intractable.

To make sure $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ are as close as possible to $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$, we need the KL divergence:

$$\begin{aligned} D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})||p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})) &= \int q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})} \\ &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \left\{ \log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})} \right\} \\ &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \left\{ \log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) p_{\boldsymbol{\theta}}(\boldsymbol{x})}{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})} \right\} \\ &= \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \left\{ \log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})} \right\}, \end{aligned} \tag{3.5}$$

the evidence lower bound(ELBO):

$$\begin{aligned} L_{\boldsymbol{\theta},\boldsymbol{\phi}} :&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \left\{ \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \right\} \\ &= \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) - D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})||p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})), \end{aligned} \tag{3.6}$$

and the goal of VAE is to maximize the ELBO and minimize $D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})||p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}))$.

According to Murphy (2023), by defining the empirical distribution $p_{\mathscr{D}(\boldsymbol{x})} = \frac{1}{N}\sum_{n=1}^{N} \delta(\boldsymbol{x}_n - \boldsymbol{x})$, the aggregated posterior $q_{\mathscr{D},\boldsymbol{\phi}(\boldsymbol{z})} = \int_{\boldsymbol{x}} q_{\mathscr{D},\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{x}$,and the inference likelihood $q_{\mathscr{D},\boldsymbol{\phi}(\boldsymbol{x}|\boldsymbol{z})} = q_{\mathscr{D},\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z}) / q_{\mathscr{D},\boldsymbol{\phi}(\boldsymbol{z})}$, the ELBO can be rewritten as

$$L(\boldsymbol{\theta}, \boldsymbol{\phi}|\mathscr{D}) = -\underbrace{D_{\mathbb{KL}}(q_{\mathscr{D},\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z})||p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}))}_{\text{KL divergence}} + \underbrace{\mathbb{E}_{p_{\mathscr{D}}(\boldsymbol{x})}[\log p_{\mathscr{D}}(\boldsymbol{x})]}_{\text{Reconstruct the input data}} \tag{3.7}$$

$$\stackrel{c}{=} -D_{\mathbb{KL}}(q_{\mathscr{D},\boldsymbol{\phi}}(\boldsymbol{z})||p_{\boldsymbol{\theta}}(\boldsymbol{z})) - \mathbb{E}_{q_{\mathscr{D},\boldsymbol{\phi}}(\boldsymbol{z})}[D_{\mathbb{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z})||p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}))], \tag{3.8}$$

where $\stackrel{c}{=}$ denotes to mean equal up to additive constants.

According to 3.7, the loss function of VAE includes two parts: the reconstruction loss and the KL divergence, which measures the difference between input and output, and the difference between the assumed distribution and learned distribution. In other words, the loss function is a sum of the mean square error and KL divergence.

We apply gradient ascent to find $\theta^*, \phi^* = \arg\max_{\theta,\phi} = L_{\boldsymbol{\theta},\boldsymbol{\phi}}$, which is equivalent to find

$$\nabla_{\theta} \mathbb{E}_{\boldsymbol{z} \sim q_{\phi}(\cdot|\boldsymbol{x})} \left\{ \log \frac{p_{\theta}(\boldsymbol{x}, \boldsymbol{z})}{q_{\phi}(\boldsymbol{z}|\boldsymbol{x})} \right\}, \tag{3.9}$$

$$\nabla_{\phi} \mathbb{E}_{\boldsymbol{z} \sim q_{\phi}(\cdot|\boldsymbol{x})} \left\{ \log \frac{p_{\theta}(\boldsymbol{x}, \boldsymbol{z})}{q_{\phi}(\boldsymbol{z}|\boldsymbol{x})} \right\}. \tag{3.10}$$

Stochastic back propagation can be applied to conduct reparameterization, see Rezende et al. (2014)

3.2 Generative Adversarial Networks

Generative adversarial networks (GANs) share similarities with VAEs, both of which are examples of probabilistic latent variable models. According to Goodfellow et al. (2014), the objective of a GAN is to train a generator, denoted as $\boldsymbol{G}$ that effectively captures the underlying data distribution, enabling it to generate samples that closely resemble the original

input data. To achieve this, the original and generated data are fed into a discriminator, $\boldsymbol{D}$, which estimates the probability that a given sample originates from the training dataset rather than from the generator.

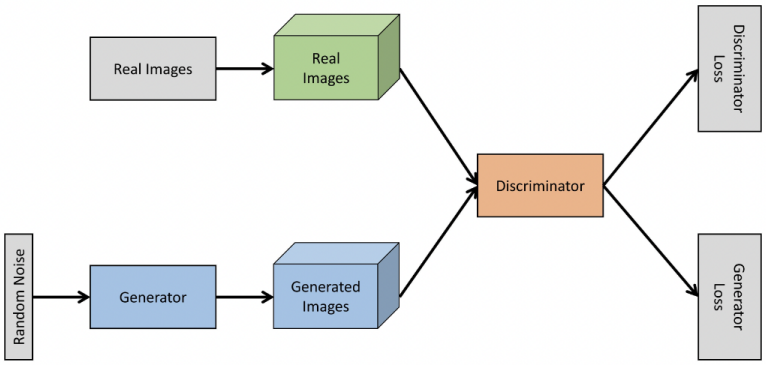

Figure 3.2: Illustration for GAN (Source).

We assume that $\boldsymbol{x} \sim p_{data}$ and $\boldsymbol{z} \sim p_g$. The mapping function $\boldsymbol{G}$ represents the procedure of mapping from $\mathbb{R}^m$ to $\mathbb{R}^d$. The discriminator's veracity $\boldsymbol{D}(\boldsymbol{x})$ represents the accuracy of the discriminator's judgment. And the generating process is as follows according to Murphy (2023):

$$
\begin{aligned}
&\boldsymbol{x} = G(\boldsymbol{z}^{'}),\\
&\boldsymbol{z}^{'} \sim p_g(\boldsymbol{z}),\\
&p_g = \frac{\partial}{\partial x^1} \cdots \frac{\partial}{\partial x^d} \int_{\{G(\boldsymbol{z})\}} p_g(\boldsymbol{z}) d\boldsymbol{z}.
\end{aligned}
$$

The training process for $\boldsymbol{D}$ is to minimize the probability of $\boldsymbol{D}$ making a mistake. By definition, it can be written as:

$$\max_D \boldsymbol{D}(\boldsymbol{x}), \tag{3.11}$$

$$\min_D \boldsymbol{D}(\boldsymbol{G}(\boldsymbol{z})), \tag{3.12}$$

which are equivalent to $\max_D \{\boldsymbol{D}(\boldsymbol{x}) + [1 - \boldsymbol{D}(\boldsymbol{G}(\boldsymbol{z}))]\}$.

Similarly, the training process for $\boldsymbol{G}$ is to maximize the probability of $\boldsymbol{D}$ making a mistake. By definition, it can be written as:

$$\max_G(\boldsymbol{D}(\boldsymbol{G}(\boldsymbol{z}))) = \min_G(1 - \boldsymbol{D}(\boldsymbol{G}(\boldsymbol{z}))), \tag{3.13}$$

Applying to all data, we have discriminator loss and generator loss

$$\mathscr{L}^{\mathrm{D}} = -\frac{1}{N}\sum_{i=1}^{N}\{\boldsymbol{D}(\boldsymbol{x}) + [1 - \boldsymbol{D}(\boldsymbol{G}(\boldsymbol{z}))]\}, \tag{3.14}$$

$$\mathscr{L}^{\mathrm{G}} = -\frac{1}{N}\sum_{i=1}^{N}\{1 - \boldsymbol{D}(\boldsymbol{G}(\boldsymbol{z}))\}. \tag{3.15}$$

Thus, our goal can be achieved by

$$\mathscr{L}^{\mathrm{GAN}} = \min_{\boldsymbol{G}}[\max_{\boldsymbol{D}}\{\mathbb{E}_{\boldsymbol{x}\sim p_{data}}[log\boldsymbol{D}(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim p_g}[log(1 - \boldsymbol{D}(\boldsymbol{x}))]\}]. \tag{3.16}$$

In an ideal scenario, the optimal solution is achieved when Discriminator loss equals Generator loss. However, this balance is not commonly attained due to the inherent adversarial nature of GAN. To assess the performance of the model, it is essential to visualize the loss functions. If the loss plots appear to be overly stable, it may mean that Discriminator and Generator have ceased to learn meaningful features. Conversely, if Discriminator loss and Generator loss fluctuate within a certain range, it suggests that they are both robustly competing against one another. Therefore, careful consideration of both the loss plots and the quality of the generated images is crucial before concluding the training process. This dual assessment ensures that the models are effectively learning and improving throughout training.

3.3 VAE+GAN

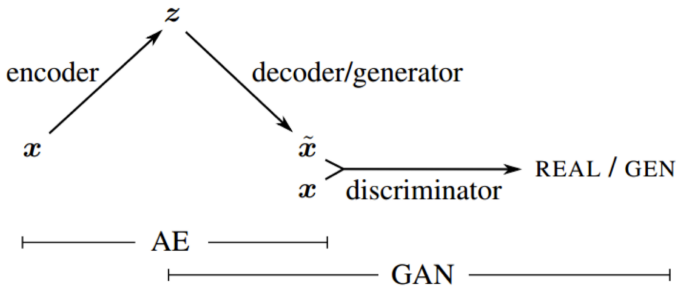

Figure 3.3: The model structure of VAE+GAN (Larsen et al., 2016).

From (3.7), the loss function of VAE can be decomposed as the reconstruction error term and the KL divergence term. The reconstruction term is based on the mean squared error, which is indeed calculated pixel-wisely. But this pixel-wise loss may not conform to human visual intuition. To address this issue, Larsen et al. (2016) proposed VAE+GAN, which combines the VAE and GAN models, and instead of using this reconstruction error term, a GAN discriminitor is used to identify whether the reconstructed images look like real ones.

Assume a *Gaussian observation model* $\boldsymbol{D}(\boldsymbol{x}) \mid \boldsymbol{z} \sim \mathrm{N}(\boldsymbol{D}(\tilde{\boldsymbol{x}}), \mathbf{I})$, where $\tilde{\boldsymbol{x}} \sim \boldsymbol{D}(\boldsymbol{z})$ is the sample from the decoder of $\boldsymbol{x}$, $\mathbf{I}$ is the identity matrix. The reconstruction error term is

$$\mathscr{L}^{\text{Discriminator}}_{\text{Similarity}} = -\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log p\left(\boldsymbol{D}(\boldsymbol{x}) \mid \boldsymbol{z}\right)\right],$$

and the total loss function of VAE+GAN can be represented as

$$\mathscr{L}^{\text{Total}} = \mathscr{L}^{\text{KL}} + \mathscr{L}^{\text{Discriminator}}_{\text{Similarity}} + \mathscr{L}^{\text{GAN}}, \tag{3.17}$$

where $\mathscr{L}^{\text{KL}}$ is the same as the KL divergence term in (3.7), $\mathscr{L}^{\text{GAN}}$ is the loss function of GAN as in (3.16).

The training process of VAE+GAN is summarized in Algorithm 1 of Larsen et al. (2016). We want to highlight that, the encoder is updated based on $\mathscr{L}^{\text{KL}}$ and $\mathscr{L}^{\text{Discriminator}}_{\text{Similarity}}$, the decoder is updated based on $\mathscr{L}^{\text{GAN}}$ and $\mathscr{L}^{\text{Discriminator}}_{\text{Similarity}}$, the discriminator is updated based on $\mathscr{L}^{\text{GAN}}$. Notice that, for updating decoder, $\mathscr{L}^{\text{GAN}}$ and $\mathscr{L}^{\text{Discriminator}}_{\text{Similarity}}$ are used, but one may dominate another in real data. So we introduce another hyperparameter γ in front of $\mathscr{L}^{\text{Discriminator}}_{\text{Similarity}}$. When we are updating the decoder, we are actually updating based on $\gamma\mathscr{L}^{\text{Discriminator}}_{\text{Similarity}}$ and $\mathscr{L}^{\text{GAN}}$, hence the ability of reconstruction and fooling the discriminator can be balanced. But the additional computational cost of including γ is not ignorable.

4 Result

4.1 Generated Images and Loss Plots

4.1.1 VAE

The loss plot of the VAE model indicates that VAE achieved convergence at around the 20th epoch, indicating that the model has reached a stable state in its learning process. To reduce the risk of overfitting and to ensure optimal output quality, training was terminated at this point.

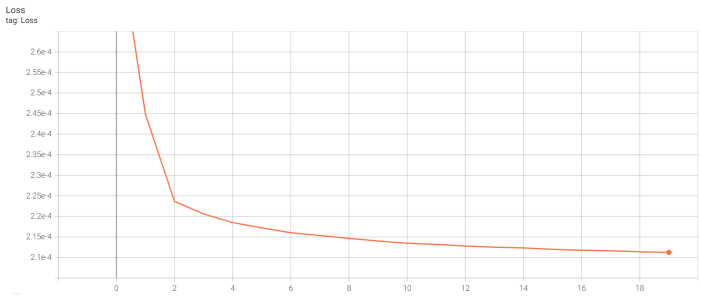

Figure 4.1: Loss plot of VAE.

The three plots of 25 faces below were generated by the VAE model after training for 1, 10, 20 epochs respectively. The VAE model converged fast. The outputs from epochs 10 and 20 demonstrate significant improvement than epoch 1, with clearer representations of facial expressions, more defined hair outlines and reduced artifacts.

The face images generated by the VAE after 20 epochs show good diversity, with distinct facial and character features. However, it is worth noting that these generated faces exhibit noticeable blurriness. This may result from the fact that VAEs usually uses a small latent dimension; in our case, the latent dimension was 128. The input information could not pass through this bottleneck efficiently, meanwhile, the construction loss used by VAEs aims to minimize the Euclidean distance by averaging all plausible outputs. Therefore, VAEs tend to produce blurry results.

Figure 4.2: VAE with 1 epoch.

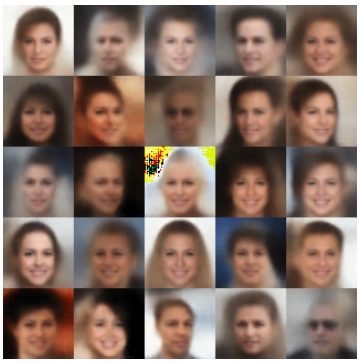

Figure 4.3: VAE with 10 epoch.

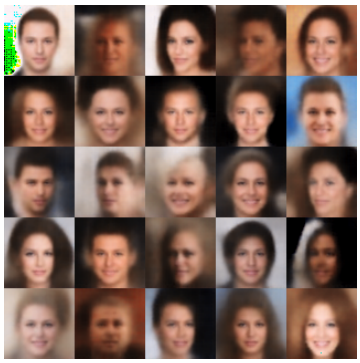

Figure 4.4: VAE with 20 epochs.

4.1.2 GAN

In the context of Generative Adversarial Networks (GANs), the loss curves exhibits significant fluctuations, mainly attributed to the inherent adversarial dynamics between the generator and the discriminator. This dynamic interplay suggests that each component is effectively adapting to the other's performance, which is a hallmark of a well-functioning adversarial training process.

Notably, after approximately 300 epochs, the discriminator loss $\mathscr{L}^{\mathrm{D}}$ and generator loss $\mathscr{L}^{\mathrm{G}}$ reached a relative stabilization, resulting in more consistent fluctuations. This stabilization indicates that the model has reached a relatively balanced state, where the generator and discriminator effectively optimized their respective goals without extreme oscillations. This behavior suggests that the generator was producing more realistic face images, while the discriminator was still able to effectively distinguish between fake generated samples and real faces. The training process converged to a point of equilibrium, allowing for the generation of more reliable and realistic faces.

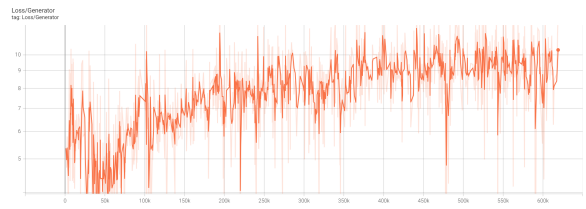

Figure 4.5: Generator Loss Plot of GAN.

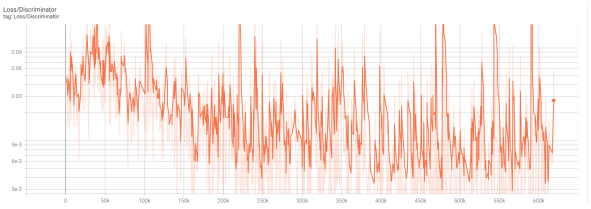

Figure 4.6: Discriminator Loss Plot of GAN.

We compared the images generated by the GAN model trained for 5, 300, and 500 epochs. The results show that the image quality improved significantly from the 5th to the 300th epoch, while the subsequent epochs, from the 300th to the 500th, demonstrated only marginal improvements in quality. This observation suggests the GAN model achieved relative stability after training for 300 epochs, a notably long training period compared to the VAE. Such extended training duration underscores the time-intensive nature of GANs and their inherent instability during earlier stages of training.

Furthermore, when evaluating the images generated by the GAN model at the 300th and 500th epochs against those generated by the VAE, the output of the GAN exhibits significantly superior properties. Specifically, the faces generated by the GAN model exhibit well-defined boundaries and precise features closer to a real face's representation.

This improvement in image quality can be attributed to the fact that VAEs primarily focus on minimizing the mean squared error between the generated images and the real images. Consequently, the ideal output tends to represent the average image across all plausible variations. In contrast, GANs prioritize the plausibility of each output, emphasizing the generation of realistic samples rather than requiring an exact local match to the training data.

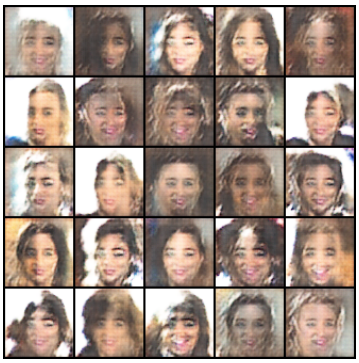

Figure 4.7: GAN with 5 epochs.

Figure 4.8: GAN with 300 epochs.

Figure 4.9: GAN with 500 epochs.

4.1.3 VAE+GAN

For the VAE+GAN model, the loss plot exhibits similar fluctuations to those observed in the previous GAN model, which is attributed to the adversarial dynamics between the generator and the discriminator. We ran 200 epochs under the conditions of $\gamma = 1.5/15$, respectively. From the generated images, we found that the VAE+GAN model reached a relatively stable state at around the 150th epoch, $\gamma = 15$, which we defined as the optimal state. The loss plots of the VAE+GAN model when $\gamma = 15$ are shown in Figure 4.10, 4.11, 4.12, and 4.13.

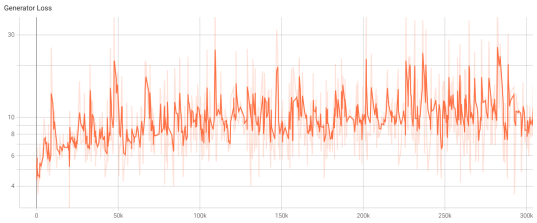

Figure 4.10: Generator Loss of VAE+GAN.

Figure 4.11: Discriminitor Loss of VAE+GAN.

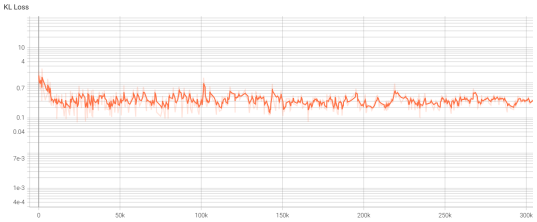

Figure 4.12: KL Loss of VAE+GAN.

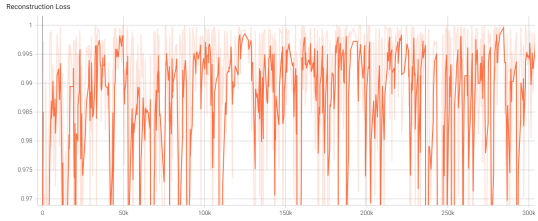

Figure 4.13: Reconstruction Loss of VAE+GAN.

We show a series of facial images generated by the VAE+GAN model with $\gamma = 15$ throughout the generation process. It can be observed that the generated images after 100 epochs tend to be sharper than the ones generated by the VAE, which can be attributed to the incorporation of the GAN component. Generally, the VAE+GAN model combines the advantages

of rapid convergence and enhanced output clarity from both the VAE and GAN architectures.

Although the final images generated are apparent, there is one noteworthy phenomenon: Eigenfaces. The facial features in the 25 images generated by the VAE+GAN model generated by the VAE+GAN are quite similar, suggesting that the VAE+GAN may converge on a limited set of facial features. This may be because the VAE imposes a regular structure on the latent space through its encoding process. While this can help in generating coherent images, it may also limit the diversity of the generated outputs. The latent space representation learned by the VAE may not capture the full variability present in the data, especially if it is overly regularized. The second reason could be the interactions among the loss functions of the VAE and GAN. If the VAE's reconstruction loss dominated, the generator may prioritize reproducing average features and sacrifice diversity.

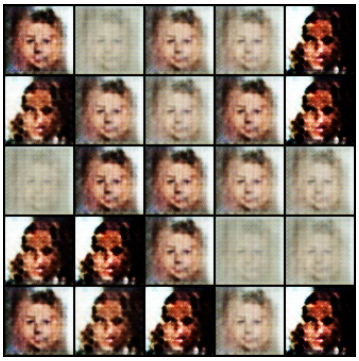

Figure 4.14: VAE + GAN with 5 epochs(γ =15).

Figure 4.15: VAE + GAN with 100 epochs(γ =15).

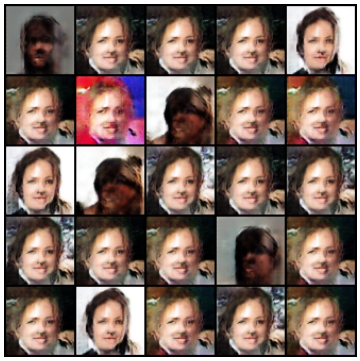

Figure 4.16: VAE + GAN with 150 epochs(γ =15).

4.2 Model Evaluations

Based on the images generated by the three models VAE, GAN, and VAE+GAN as well as their corresponding loss plots, we conducted a comprehensive evaluation of these models. This evaluation included quantitative methods to assess the quality of the generated images alongside the performance of the models themselves.

4.2.1 Generated Images

We implemented the Inception score(Murphy, 2023) and Laplacian Variance(Canny, 1986) to evaluate the generated images based on the uniqueness of facial features, the diversity and visual credibility of generated faces, the preservation of facial features after changing features, and image clarity. The mathematical formulations for the Inception Score (IS) and Laplacian Variance are presented below:

$$IS = \exp[\mathbb{E}_{p_{\boldsymbol{\theta}}(x)} D_{\mathbb{KL}}(p_{disc}(Y|\boldsymbol{x})||p_{\boldsymbol{\theta}}(Y))] \tag{4.1}$$

The Inception Score (IS) is a prominent metric for assessing the quality of images produced by generative models. It utilizes a pre-trained Inception network to evaluate the generated samples' clarity and diversity. The score is derived from the distribution of predicted class labels for these images, indicating how well they capture various categories (Murphy, 2023). A higher Inception Score indicates that the images exhibit more distinct features and greater

diversity, reflecting the model's ability to generate high-quality, visually appealing outputs.

For clarity of the picture, Laplacian Variance is implemented, calculating the variance of the output of a Laplace filter. Laplace variance is calculated as follows (Canny, 1986):

$$\text{Laplacian Variance} = \text{Var}(\nabla^2 I) \tag{4.2}$$

Similar to IS, larger Laplace variance values indicate greater clarity and detail, making it an effective indicator of the visual quality of generated output.

	VAE	GAN	VAE+GAN	Real
Inception Score	1.704	2.650	1.231	4.239
Laplacian Variance	251.79	2506.18	1360.62	1073.71

Table 4.1: Comparison of generated images.

Table 4.1 calculates the Inception Score and Laplacian Variance of the three models. The GAN model performs better than the VAE and VAE+GAN models in both aspects, indicating that the GAN performs better than the other two models in generating images. Generally, the VAE+GAN model did not perform well based on these two measurements of generated images. This result is consistent with our previous observations regarding eigenfaces issues of the VAE+GAN.

Additionally, the images generated by the GAN and VAE+GAN have higher Laplace variance than the original images. This may be because GAN-generated images usually produce richer details and textures, which can improve the Laplace variance since this metric examines how the edges and textures in an image change. On the other hand, GAN/VAE+GAN may overfit the training data in our training porcess, thus the generated images may show higher contrast and more details, affecting the Laplace variance.

After all, data in Table 4.1 is only a reference for assessing the clarity and diversity of the images. Both Laplace variance and IS have limitations since they are affected by many factors when evaluating images, such as overfitting. Thus, we added the model evaluation to supplement our assessment of the model.

4.2.2 Model Comparison

Table 4.2 analyzes the advantages and disadvantages of the three models:

Model	**Epochs to Optimal**	**Advantages**	**Disadvantages**
VAE	20 epochs	Fast Convergence	Poor Image Quality
GAN	300 epochs	High Image Quality	Unstable & Time-Consuming
VAE + GAN	150 epochs	Hybrid Benefits	Eigenfaces

Table 4.2: Model Comparison.

In summary, each model exhibits unique advantages and limitations, with no single model outperforming the others across all metrics. In practical applications, selecting an appropriate model should be informed by specific requirements and objectives to ensure alignment with the desired outcomes. In our practice, while careful combination of two existing models

may yield beneficial results, it can also introduce new challenges, such as the complexity of selecting optimal hyperparameters. Furthermore, the combination may not fully capitalize on the strengths of each model, potentially resulting in compromises and sacrifices.

5 Discussion

Our research investigated latent space arithmetic within generative models, specifically focusing on the VAE framework. Latent space arithmetic enables controlled manipulation of generated samples through mathematical operations in the latent space, enhancing model flexibility while providing insights into the interpolation of facial characteristics.

5.1 Interpretability of Latent Space

The interpretable nature of latent space arithmetic is a significant strength of our approach. The graph shown below is the visualization of the latent space of the trained optimal VAE model.

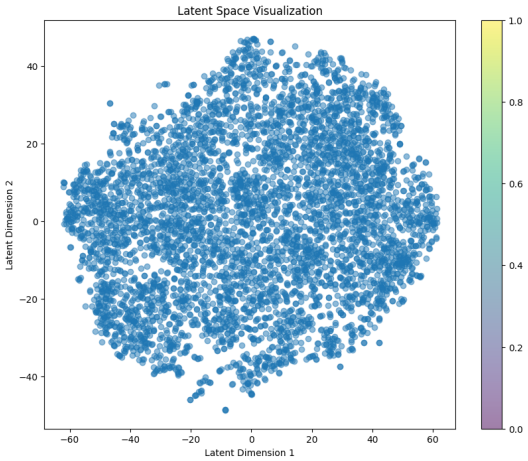

Figure 5.1: Visualization of Latent Space of VAE.

As we can see from the scatter plot, the data points exhibit relatively uniform spread, suggesting that the model has learned to represent the data distribution regularly. Within this uniform distribution, many distinct clusters are observable. This clustering reflects the model's ability to encode meaningful facial features such as smiling, sunglasses, hair styles and so forth.

By performing vector operations in latent space, we can achieve precise control over these facial features. Our experiments showcased this capability by manipulating specific facial features. We computed the difference between the mean latent vectors of images with and without certain attributes (such as smiling expressions), allowing for targeted modifications while maintaining other characteristics. This approach not only validates the linear separability hypothesis of latent space but also demonstrates the potential for controlled image generation.

Furthermore, integrating latent space arithmetic with Conditional Variational Autoencoders (CVAE) can enhance controllability. By combining conditional variables with latent vector operations, we can generate images under specific conditions while precisely adjusting particular attributes, thereby enabling more complex generative tasks.

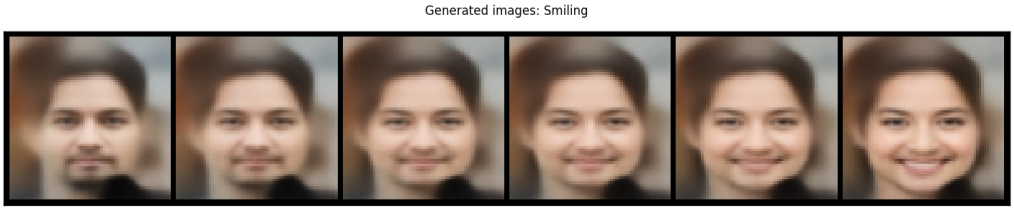

Figure 5.2: Interpolation: unsmiling to smiling (Source).

5.2 Limitations of Arithmetic Operations

Our investigation revealed several constraints in latent space arithmetic applications. Linear interpolation techniques sometimes struggled to represent complex feature transitions, particularly when fine control was required. The effectiveness of arithmetic operations showed strong dependence on latent space parameters, including dimensionality and structure. Determining optimal latent space dimensions for specific applications remains challenging and requires further investigation.

5.3 Future Research Directions

Several promising research paths emerge from our findings. Investigating latent space arithmetic in newer architectures, such as diffusion models, could yield valuable insights. Integration with advanced neural network components might offer novel approaches to latent space manipulation. These developments could expand the practical applications of generative models while improving their performance characteristics.

6 Conclusion

The objective of this project is to generate new data samples, namely artificial faces, from an existing facial dataset through the utilization of two generative model architectures: Variational Autoencoder (VAE) and Generative Adversarial Network (GAN). We also experimented with a hybrid approach that integrates both architectures to explore its potential to enhance the generation outcomes.

We start from a VAE model to learn a structured and continuous distribution of data and generate new faces from it. This model converged quickly but produced outputs with low sharpness. Next, we employed a GAN model to make the generator compete with the discriminator, resulting in new samples that were highly sharp, authentic, and diverse, although the model converged slowly and was unstable. Finally, we implemented a VAE+GAN model by substituting the GAN's generator with the VAE, enabling it to compete against the GAN's discriminator. This approach led to a relatively rapidly converging model that can generate new faces with high sharpness but low diversity. What's more, we also investigated the latent space of the VAE model, allowing to exert control over the latent representations.

The results presented in the formal sections of this report indicate that the outcomes from the GAN may be the best among the three models. Meanwhile, the VAE offers the advantage of learning a continuous latent space representation, enabling the interpolation of facial features. However, the limitations of the GAN stem from insufficient computing power

and the structure of the code, which contribute to its instability and prolonged convergence. With additional computing power and more efficient code, these limitations can be mitigated. In real-life applications, the generated realistic faces can be used for virtual avatar generation in realistic games, as subjects for psychological research on emotion recognition and interpersonal communication, etc. Additionally, the privacy of facial data providers and users can be well protected.

References

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698.

Cao, H., Tan, C., Gao, Z., Xu, Y., Chen, G., Heng, P.-A., and Li, S. Z. (2024). A survey on generative diffusion models. *IEEE Transactions on Knowledge and Data Engineering*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

Larsen, A. B. L., Sønderby, S. K., Larochelle, H., and Winther, O. (2016). Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning*, pages 1558–1566. PMLR.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F., et al. (2006). A tutorial on energy-based learning. *Predicting structured data*, 1(0).

Murphy, K. P. (2023). *Probabilistic machine learning: Advanced topics*. MIT press.

Rezende, D. and Mohamed, S. (2015). Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In Xing, E. P. and Jebara, T., editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China. PMLR.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR.