Data Generation and Latent Space Based Feature Transfer Using ED-VAEGAN, an Improved Encoder and Decoder Loss VAEGAN Network

Jiatong Li[1,2(✉)]

[1] The Grainger College of Engineering, Electrical and Computer Engineering, Zhejiang University, Hangzhou, China
jl180@illinois.edu
[2] University of Illinois at Urbana Champaign, Illinois, USA

Abstract. To combine the advantages of VAEs and GANs to generate both diverse and high-quality samples, this paper proposes ED-VAEGAN which improves encoder and decoder loss of traditional feature-wise VAEGAN [4]. More precisely, a reconstruction score term is added to encoder loss function, which accelerates the training of the whole model. The decoder loss was similar to traditional definition, but discarded an irrelevant term to decoder. This paper applied this new model to face datasets and compares the generations with other models when the models are fully trained and when trained for the same iterations. And the latent space expedition was done by first encode the images and then do the latent code walk between two images. As a result, ED-VAEGAN outperformed traditional VAEGAN on training speed, and its latent space expedition result indicates better continuity comparing to other pixel-wise models. In the end, this paper applied simple data augmentation method to solve the brightness problem that happened when training iterations increase.

Keywords: ED-VAEGAN · Feature-wise Reconstruction loss · latent space expedition

1 Introduction

Think about the problem there is a bunch of input training images, and they need to be encoded into a lower dimensional latent code z, which can also be treated as a latent data from complete data (X, Z), where each x_i has its corresponding latent data z_i, which was randomly sampled from distribution P(Z). Different latent code z_i can be obtained, and they provide some key information for generating process, for example, in face generating training, it can contain information that leads to black hair or yellow hair, big nose or small nose, smiling or not smiling.

To achieve unsupervised learning and to learn underlying data distribution of unlabeled data and generate new data from it, Variational Autoencoders (VAE) [1], Generative Adversarial Networks (GAN) [2], or Deep Convolutional Generative Adversarial

K. Hemachandran et al. (Eds.): ICAID 2023, AHIS 9, pp. 123–135, 2023.
https://doi.org/10.2991/978-94-6463-222-4_12

Networks (DCGAN [3]), and their combination in the form of Variational Autoencoders Generative Adversarial Networks (VAEGAN) are designed [4]. They enable much more effective data generation, feature learning, and representation learning than traditional machine learning models. VAEs use the idea of probabilistic inference and reparameterization trick to get various latent code z thus are used in tasks such as image generation and natural language processing. GANs' creative idea of discriminator and generator and their ability to generate realistic data, allowing for realistic image generation, image inpainting, image synthesis, and image super-resolution made it one of the most popular studies interest these years. VAEGAN is a combination of VAEs and GANs which combines the advantages of VAEs and GANs while avoiding their separate disadvantages.

In Autoencoding beyond pixels using a learned similarity metric, they firstly redefined the reconstruction loss of Encoded and then regenerated images. But it seems that before they propose beyond pixels reconstruction loss, the most popular approach is still mean squared error [1], they used mean square error between original images and reconstructed images. However, faces generated by VAEs have the problem of lack of details such as blur hair, lack of face texture. That is because encoder only uses simple element-wise error, which is quite different from the human-beings judges since people see the image features on a higher level but not from element-wise error from an original image. From this point of view, GANs including DCGANs did a good job by using the Discriminator to judge the similarity, for the Discriminator use Convolution to save much higher level of information in images. That's why GANs can capture fine details and generate sharp images, but they struggle to be trained steadily because the Generator in GANs learn from a completely random z distribution and the loss function depends on discriminators. When the generated images fool the discriminator and that the possibility of judging it as correct goes to nearly 50 percent, the model might start to accept some strange generation results and the new generated quality will become very bad. To compare with, VAEGAN additionally train the generator using a more meaningful latent code z, which contributes to the discriminator loss, thus makes the model training steadier. Moreover, there seems to be a research gap on pixel-wise VAEGANs since the feature-wise VAEGAN perfectly achieved the detail generation job. But pixel-wise error will not be the only judge of encoder's performance, the discriminator loss also affects encoder, since Dis(D(E(X))) is all about convolution of the reconstructed image, which is exactly feature-wise consideration but not pixel-wise. So this paper also propose to train a pixel-wise VAEGAN to see if it achieves any improvements on detail control comparing to mere VAEs. Back to the topic of this pioneer paper [4], their method to solve the concern on detail generation is to consider a new Gaussian observation distribution described by the intermediate layers of discriminator, which in this paper defined to be the last but one layer output, and the reconstruction loss is instead defined as the squared error between reconstructed images' output and the real images' output.

In other VAEGAN studies, researchers did not attempt to change the feature-wise VAEGAN loss function or used pixel-wise error. In AC-VAEGAN [5], an auxiliary classifier was applied to the last but one layer of discriminator so that it not only outputs the possibility of input image being a real image, but also outputs the classification of input image. But the loss function was still based on mean squared error between

reconstructed images and original images, and their datasets are not face datasets. In lifelong VAEGAN [6], the L-VAEGAN was designed to be a lifelong learning system, meaning that it can continuously learn from new data and transfer knowledge from previous tasks. While the loss function for Lifelong VAEGAN is composed of four parts: a reconstruction loss, an adversarial loss, a KL divergence loss, and an entropy regularization loss. The reconstruction loss encourages the VAE to accurately reconstruct the input image, which is done by minimizing the mean squared error between the input image and the image reconstructed by the VAE. In Hierarchical Patch VAE-GAN: Generating Diverse Videos from a Single Sample [7], they used patches of VAE and patches of GANs, yet reconstruction loss is still defined as squared error between images.

In this paper, VAE, DCGAN, p-w VAEGAN (pixel-wise reconstruction error), f-w VAEGAN (feature-wise reconstruction error which was first proposed by [4]), and ED-VAEGAN (Encoder and Decoder loss improved) are trained and compared. ED-VAEGAN is proposed with the BCE loss between reconstruction images' discriminator score and real labels joining the encoder loss to make the training process faster and more stable, and decoder loss no longer use the old way of minus the discriminator loss, instead remove one of the three components of discriminator loss and change the rest of the two to a more understandable representation. The result section will include the generation results, the continuity of encoded space Z measured by latent walk method.

2 Methods

In GMM, Gaussian mixed models, when dealing with real life images, the latent data z will be a continuous, high-dimensional random variable and it can be assumed that data distribution obeys an unobserved Gaussian Distribution. Using a formula to measure this scenario, maximizing the probability of generating something that is close to the training dataset X is tried. $\boldsymbol{P}(\boldsymbol{X}) = \int \boldsymbol{P}(\boldsymbol{X}|z, \boldsymbol{\Theta})\boldsymbol{P}(z)\boldsymbol{dz}$, where $\boldsymbol{P}(z)$ indicates the probability of sampling out z from distribution Z, $\boldsymbol{P}(\boldsymbol{X}|z, \boldsymbol{\Theta})$ indicates a mapping f from the sampled z and parameter $\boldsymbol{\Theta}$ to X, and if this mapping describes a distribution that is widely used in VAEs, which is Gaussian distribution, then it can be wrote in another way: $\boldsymbol{P}(\boldsymbol{X}|z, \boldsymbol{\Theta}) = \boldsymbol{P}(\boldsymbol{X}|\boldsymbol{f}(z|\boldsymbol{\Theta}), \boldsymbol{\sigma}^2 * \boldsymbol{I})$. If z is sampled from a normal Gaussian distribution, z would hardly describe correct guidance to generate images, in another word, $\boldsymbol{P}(\boldsymbol{X}|z, \boldsymbol{\Theta})$ will be mostly zero. This gives us the idea to find a prior distribution $\boldsymbol{Q}(z|\boldsymbol{X})$ which aims at sample better latent code z. This was encoder's obligation. Now it is not necessary to compute $\boldsymbol{P}(z)$ and then $\boldsymbol{P}(\boldsymbol{X}|z, \boldsymbol{\Theta})$, which most of the time in GANs defined as $\boldsymbol{N}(0, \boldsymbol{I})$, instead consider $\boldsymbol{E}_{z\sim\boldsymbol{Q}}(\boldsymbol{P}(z))$. To connect a bridge between $\boldsymbol{P}(\boldsymbol{X}|z)$ and $\boldsymbol{E}_{z\sim\boldsymbol{Q}}(\boldsymbol{P}(z))$, first use KL-divergence (Kullback-Leibler divergence) to measure the similarity of two distributions $\boldsymbol{P}(\boldsymbol{X}|z)$ and $\boldsymbol{Q}(z)$, $\boldsymbol{D}[\boldsymbol{Q}(z)||\boldsymbol{P}(z|\boldsymbol{X})] = \boldsymbol{E}_{z\sim\boldsymbol{Q}}[\boldsymbol{logQ}(z) - \boldsymbol{logP}(z|\boldsymbol{X})]$, apply Bayes Rule further derives it to: $\boldsymbol{D}[\boldsymbol{Q}(z)||\boldsymbol{P}(z|\boldsymbol{X})] = \boldsymbol{E}_{z\sim\boldsymbol{Q}}[\boldsymbol{logQ}(z) - \boldsymbol{logP}(\boldsymbol{X}|z) - \boldsymbol{logP}(z)] + \boldsymbol{logP}(\boldsymbol{X})$. Here, $\boldsymbol{log}(\boldsymbol{P}(\boldsymbol{X}))$ has nothing to do with z so it can be written outside the expectation. Next, use the definition of continuous P,Q KL-divergence, which is $\boldsymbol{D}(\boldsymbol{P}||\boldsymbol{Q}) = \boldsymbol{\Sigma}_z\boldsymbol{log}(\boldsymbol{P}(z)) - \boldsymbol{log}(\boldsymbol{Q}(z))\boldsymbol{P}(z)$ the equation becomes $\log \boldsymbol{P}(\boldsymbol{X}) - \boldsymbol{D}[\boldsymbol{Q}(z)||\boldsymbol{P}(z|\boldsymbol{X})] = \boldsymbol{E}_{z\sim\boldsymbol{Q}}[\boldsymbol{logP}(\boldsymbol{X}|z)] + \boldsymbol{D}(\boldsymbol{Q}(z)||\boldsymbol{P}(z))$, note that in the work, the distribution Q is required to depend on dataset X, so write $\boldsymbol{Q}(z)$ as $\boldsymbol{Q}(z|\boldsymbol{X})$:

$$\boldsymbol{logP}(\boldsymbol{X}) - \boldsymbol{D}[\boldsymbol{Q}(z|\boldsymbol{X})||\boldsymbol{P}(z|\boldsymbol{X})] = \boldsymbol{E}\_\{z \sim \boldsymbol{Q}\}[\boldsymbol{logP}(\boldsymbol{X}|z)] + \boldsymbol{D}(\boldsymbol{Q}(z|\boldsymbol{X})||\boldsymbol{P}(z)) \quad (1)$$

Since KL-divergence must be bigger than 0, predecessors named the right-hand side of Eq. 1 the lower bound. The target is to maximize $logP(X)$, which can be done optimize the right side and also minimize the KL-div, meaning that this training process should encode X to z without much loss to normal gaussian distribution. To be more specific, usually it is assigned like:$Q(z|X) = N(z|\mu(X; \Theta), \Sigma(\mathrm{X}; \Theta))$. Here Θ is the encoder network parameter, with X as input. Notice that using the property that both distributions are multi-variate Gaussian distribution, rewrite KL-divergence

$D(Q(z|X)||P(z))$:

$$D[N(\mu(X; \Theta), \Sigma(X; \Theta))||N(0, I)] = \frac{1}{2}(tr(\Sigma(X; \Theta)) + \mu(X; \Theta)^T \mu(X; \Theta) - k - log(det(\Sigma(X)))) \quad (2)$$

2.1 VAE'S Reparameterization Trick

For the right hand side of Eq. 1, $\boldsymbol{E}_{\boldsymbol{z}\sim\boldsymbol{Q}}[logP(X|z)]$ was too computation consuming when sample many z and then take the average of it, instead it would be reasonable to only sample one z and let $\boldsymbol{E}_{\boldsymbol{z}\sim\boldsymbol{Q}}[logP(X|z)] = logP(X|z)$. Effort is being made to optimize $\log \boldsymbol{P}_{\boldsymbol{z}\sim\boldsymbol{Q}}(X|z) + D(Q(z|X)||P(z))$ and then minimize $D[Q(z|X)||P(z|X)]$ to maximize $log(P(X))$. However, when trying to backpropagate through encoder, which is Q, there mustn't be totally random. But it needs to be a multi-variate Gaussian distribution. The solution is to separately sample a ε ~ N(0,I), and the encoder encode X to mean μ (X) and covariance Σ(X), then produce $z = \mu(X) + \varepsilon * \Sigma^2(X)$. Finally, write the right hand side ELBO as $E_{\varepsilon\sim N(0,I)}[logP(X|z = \mu(X) + \varepsilon * \Sigma^2(X))] - D[Q(z|X)||P(z)]$, which is able for us to compute its gradient [8].

2.2 Training Theory of GANs

In GAN, it has a generative network G and a discriminator D. D distinguish the input image as real or fake, while the generator tries to fool the discriminator by producing better images. This study use *theta* to map the Gaussian Distribution z to another distribution and use G as generator to represent this process, and for a data (real or fake), it was feed into into a discriminator D. So when the G is fixed, the optimal discriminator should have $D_{G*}(x) = \frac{p_{data}(x)}{p_{data}(x)+p_G(x)}$, which roughly means it can be said that the correct probability to say this data is real to the proportion that this data comes from sampled real data [8].

The loss function regarding discriminator D is

$J(D) = \boldsymbol{E}_{\boldsymbol{x}\sim realdata}[-log(D(x))] + \boldsymbol{E}_{\boldsymbol{z}\sim N(0,I)}[-log(1 - D(G(z))]$, which should be close to 0 + plus 0 + when the discriminator correctly labeled every true data as 1 and every generated fake data as 0. For generator loss, J(G)= $\boldsymbol{E}_{\boldsymbol{z}\sim N(0,I)}$ [−log(1 − D(G(z)))], which should also be 0 + when the generator successfully generate nearly true image

from random noise sampled z, so that D(G(z)) gets close to 1 and expectation (also can be written as summation since deep neural network almost makes the continuous problem to discrete problem) goes close to 0 from positive end. By iteratively update both discriminator and generator, model will converge [8].

2.3 Loss Functions VAE

Update encoder and decoder using the combined loss of MSE and KL-div, same as part of VAEGAN, here is the previous mentioned loss function:

$$E_{\varepsilon\sim N(0,I)}[\text{logP(X|z} = \mu(\text{X}) + \varepsilon * \Sigma^2(\text{X}))] - \text{D[Q(z|X)||P(z)]} \tag{3}$$

$$-D_{KL}((q_\varphi(z)||p_\theta(z)) = \int q_\theta(z)(logp_\theta(z) - \log q_\theta(z))dz$$

$$= \frac{1}{2}\sum_{j=1}^{J}(1 + \log\left(\sigma_j\right)^2 - \left(\mu_j\right)^2 - \left(\sigma_j\right)^2) \tag{4}$$

The derivation of KL-divergence [1].

Using this formula, the KL divergence can be wrote easily in python code. For the first part expectation computation of $logP(X|z)$, use the mean squared error between the reconstructed image $D(E(X))$ and original image X.

2.4 Loss Functions DCGAN

Discriminator loss of DCGAN is the sum of: 1. BCEloss between (x: random z generation on Discriminator score's, y: fake labels); 2. BCE loss between (x: true image on Discriminator score, y: true labels).

Generator loss is the BCEloss between (x: another random z generation score out of Discriminator, y: true labels, since this loss is required to be small which means that discriminator judges generated images as true.

2.5 Loss Functions ED-VAEGAN

Encoder loss: KL-divergence + reconstruction loss + reconstruction score. The KL divergence was computed between a normal Gaussian distribution and the real image's encoded z. The reconstruction loss is the same as feature-wise (f-w) reconstruction loss, this study used the last but one layer of discriminator as the feature extraction layer, which should have encoded the performance of the input images to a 512 latent space. Mean squared error between True images encoded space and Reconstructed images encoded space should be the reconstruction loss. Reconstruction score is the BCEloss between real labels and reconstructed images. This additional loss would accelerate the whole training process by increasing the gradient for each optimizer step, and when the encoder performs better, the generator would perform better too.

Discriminator loss is the sum of the following 3 parts. 1: BCE loss of real image score and true label. 2: BCE loss of random z generated fake image score and fake

labels. 3: BCE loss of real image encoded z then decoded half true image score (the aim is to eventually make it close to "true" but to update discriminator, the discriminator is punished to say it true) about fake labels.

Decoder loss is the sum of three parts. 1: BCE loss (random z generated image's discriminator score, real labels). 2: BCE loss (reconstructed image's discriminator score, real labels). 3: Reconstruction loss.

This paper proposes such an approach different from f-w VAEGAN because the minus discriminator loss in f-w VAEGAN's decoder loss was confusing. The explanation of f-w VAEGAN is that the discriminator loss will be higher when the generator generates images that fools the discriminator, so that minus the discriminator is reasonable. In ED-VAEGAN one part of the discriminator, BCEloss(Dis(X),real labels) is discarded since it has nothing to do with decoder. And the two minus term becomes adding up, which is more common sense, and just change the fake labels to real labels.

2.6 Loss Functions f-w VAEGAN

As this paper instructed [4], it is reasonable to use weighted Reconstruction loss minus Discriminator loss as the Decoder loss. But in this paper this gamma was not applied.

For Discriminator loss, same as ED-VAEGAN. For decoder loss, use reconstruction loss minus discriminator loss. For encoder loss, use KL-divergence + Reconstruction error. No reconstruction score in consideration comparing to ED-VAEGAN and the reconstruction loss is the same with ED-VAEGAN.

2.7 Loss Functions p-w VAEGAN

The only difference between f-w and p-w VAEGAN is that the reconstruction loss should be the mean squared error between the reconstructed image from the real image's encoded z and the real image itself.

2.8 Latent Space Continuity Quality

In order to do feature transfer well, it is required that the latent space is smooth. In another word, the latent space should be meaningful almost everywhere. Latent walk is done by encoding two images into z1 and z2, and then set the total steps as number_int, for this paper 10 steps is used, meaning that there will be in total 10 images for 2 imgs in total, from z1, and eventually it gets to z2. For each step, alpha is updated from the beginning 1/11 to 2/11 all the way to 10/11, and the z_intp is defined as z1*alpha + z2*(1.0-alpha). This makes the transition from z1 to z2.

In this paper [9], the application on 3D face reconstruction shows how important it is to train a powerful latent space to store the information. And in the experiment part, this paper will compare different model's performance in latent walk feature transfer. Also, in DCGAN [3], it showed a latent walk over two images of bedroom, and the transfer was smooth, without sudden change in the scene. And the discussion of sharp transitions was explained as the latent space was hierarchically collapsed.

What's more, CycleGAN is used as a good standard of transferring horses to zebras and vice versa [10]. CycleGAN is a type of GAN used for image-to-image translation

Table 1. Architecture of ED-VAEGAN

Encoder k = 5	Decoder k = 5	Discriminator k = 5
↓64 × 64 × 64 BatchNorm2d,Relu	Linear,256*8*8 BatchNorm1d,Relu	↓32 × 64 × 64 Relu
↓128 × 32 × 32 BatchNorm2d,Relu	↑256 × 16 × 16 BatchNorm2d,Relu	↓128 × 32 × 32 BatchNorm2d,Relu
↓256 × 16 × 16 BatchNorm2d,Relu	↑128 × 32 × 32 BatchNorm2d,Relu	↓256 × 16 × 16 BatchNorm2d,Relu
↓256 × 8 × 8 BatchNorm2d,Relu	↑64 × 64 × 64 BatchNorm2d,Relu	↓256 × 8 × 8 BatchNorm2d,Relu
Linear,2048 BatchNorm1d,Relu	↑32 × 128 × 128 BatchNorm2d,Relu	Linear,512 BatchNorm1d,Relu
2 × Linear 128 for mean and logvar	→ 3 × 128 × 128 Tanh	Linear 1, Sigmoid

that works by training two separate GANs, one to convert an image from one domain to another, and the other to convert it back to the original domain. CycleGAN works by forcing the two networks to "cycle" through the two domains, thus ensuring that the model is able to generate realistic images in both directions. This process is referred to as "cycle consistency", which is why CycleGAN works so well and this paper tries to use latent walk on horse2zebra dataset to see what results 10 steps of latent space expedition would present. Would the zebra lines appear on a horse during two horse transitions? Would the zebra lines turn to brown and red just like the color of horses' skin? The Table 1 presents the architecture of ED-VAEGAN.

3 Experiments

This paper mainly used two datasets: CelebA and horse2zebra [11, 12]. The CelebA is a widely used human face dataset while the horse2zebra is used in CycleGAN. Latent dimension is set to 128, and the images are preprocessed to size (bs,3,128,128) using torchvision.transforms, center cropped to 128 × 128 and normalized using mean = (0.5,0.5,0.5) and std = (0.5,0.5,0.5), which means this study is applying mean = 0.5 and std = 0.5 to every 3 channels. Then split the dataset to training set and test set. And only use the training set to do training work while the test set are for all experiments.

For optimizer choice, this paper used Adam for VAE, DCGAN and RMSprop for p-w VAEGAN, f-w VAEGAN and ED-VAEGAN. The learning rate for RMSprop is originally 3e-4, and continuously decaying after each epoch using torch.optim.lr_scheduler which has its gamma set to 0.75. Other RMSprop hyperparameters are alpha = 0.9, eps = 1e-8, weight_decay = 0, momentum = 0, centered = False. For DCGAN training, it was suggested to choose a learning rate of 0.0002 and betas (0.5,0.999) [3]. The training batch size and iterations are provided in Table 2.

For one iteration, there are bs numbers of images being trained. This means that f-w VAEGAN trained more than ED-VAEGAN both for iterations and total number of

Table 2. Training batch size and iterations

	VAE	DCGAN	p-w VAEGAN	f-w VAEGAN	ED-VAEGAN
Iterations	279000	219300	339900	161100	129300
Batch size	64	64	64	32	16

images and per iteration images, but actually the parameters only update once for each iteration, so batch size doesn't matter much.

4 Results and Discussions

From the generations shown in Fig. 1 and Fig. 2, it is obvious that VAE and p-w VAEGAN doesn't differs much. They both have the problem of lack of details. This indicates that the guess of discriminator contributes feature wise error does not balance the error of pixel-wise MSE. Which implies that as long as there is pixel-wise error, the training result will be lack of details. For f-w VAEGAN and ED-VAEGAN, the results are similarly good in both random z generation and reconstruction. But sometimes f-w VAEGAN generates strange results. All the results that did not use pixel-wise error (DCGAN, f-w VAEGAN and ED-VAEGAN) suffered gray color problem. Where the color of the images becomes grayer when training process goes. To solve this problem, the dataset should have more bright color images. While from testing set generations it is not hard to find that many human faces are in dark light or there are some black people images that are totally black for both faces and background (Fig. 3). To do data augmentation for celebA dataset, this comprehensive paper introduced many methods used by other papers [13], in conclusion, the most convenient way to improve feature-wise models generation results is by adding flipped images and do color space augmentations such as changing the brightness, contrast, saturation, and hue of the image. For DCGAN, generation results are as good as f-w VAEGAN and ED-VAEGAN when the training iterations is supervised, but most of the second half training results, the generated images are about the same, and sometimes even restarted from random noise. This is because the discriminator can't perform its job and thus the generation results are wrongly guided.

For latent walk of VAE and p-w VAEGAN shown in Fig. 4 and Fig. 5, it is obvious that the sharp change occurs at the fourth image, while the f-w VAEGAN and ED-VAEGAN avoided sharp change. And the images by ED-VAEGAN has higher fidelity comparing to f-w VAEGAN, VAE and p-w VAEGAN in Fig. 5 first few steps. The sharp change eventually appeared for horse2zebra dataset. ED-VAEGAN was merely trained in 16 epochs for 311400 iterations, and apparently the latent space was not well defined by generator. Maybe longer training time and better tuned hyperparameters such as learning rate would help define the latent space more continuously to avoid sharp changes. For the image overall color, since all the images in horse2zebra dataset are photographed during the day, so there is no color graying problem comparing to celebA dataset.

Inception v3 is a convolutional neural network architecture developed by Google for image recognition tasks. It is based on the earlier Inception model and is capable of recognizing complex patterns in images with high accuracy. It has been trained on

Fig. 1. Result of random sampled z generations. From top to bot: VAE, GAN, p-w VAEGAN, f-w VAEGAN, ED-VAEGAN

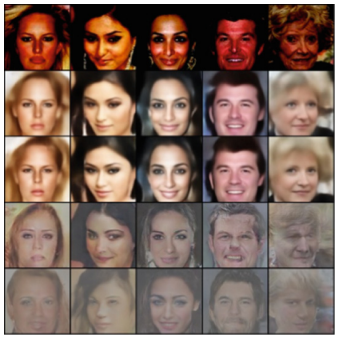

Fig. 2. Reconstruction results. From top to bot: Original image, VAE, p-w VAEGAN, f-w VAEGAN, ED-VAEGAN

Fig. 3. Examples of dark original images

the ImageNet dataset and is widely used in many computer vision applications. The mean Inception Score shown in Table 3 indicates the overall quality of the generated images. A higher mean score indicates that the images are of higher quality. The standard deviation of the score indicates the amount of variation in the scores, which can be used to determine the reliability and consistency of the generated images. A lower standard deviation indicates that the images are more consistent, while a higher standard deviation indicates that the images are more varied. From the table, the statistics are all average values. ED-VAEGAN and p-w VAEGAN have the higher mean and standard deviation than other 3 models, meaning they better recognize complex patterns in the images.

From the loss graph shown in Fig. 6, ED-VAEGAN has a lower encoder loss and decoder loss comparing to f-w VAEGAN while ED-VAEGAN has one half the batch size comparing to f-w VAEGAN and the encoder loss is defined with an extra term

Fig. 4. Latent walk of 2 images, from the top to the bottom they are VAE, p-w VAEGAN, f-w VAEGAN and ED-VAEGAN

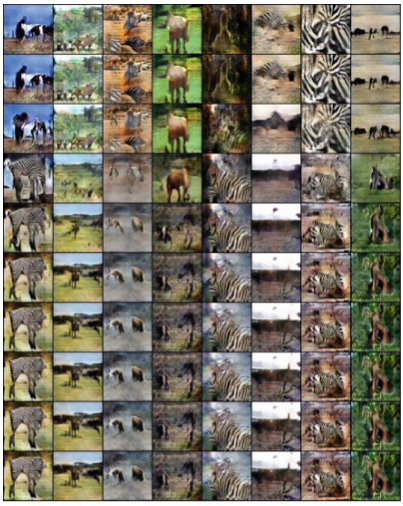

Fig. 5. 2 Images of h2z latent walk by ED-VAEGAN

comparing to f-w VAEGAN. For discriminator loss, the ED-VAEGAN rise and fall rapidly, the reason might be the fast-training speed makes generation process performs well so that the discriminator is fooled by the generation results, so the loss sometimes goes up high.

In Fig. 7, ED-VAEGAN takes the lead over training time, and for the last column of image, ED-VAEGAN's result started to become grayer earlier than traditional method.

Figure 8 results are trained in epochs 30, batchsize 16, with input images randomly flipped and randomly brightened to 100%–150% using transforms. Colorjitter, the training results avoided dark color problem and the details were kept as good.

Table 3. Inception score

	Inception v3-mean	V3-std
VAE	1.7474	0.0757
DCGAN	1.6555	0.0762
p-w VAEGAN	1.7189	0.0764
f-w VAEGAN	1.9254	0.0951
ED-VAEGAN	1.8685	0.1984

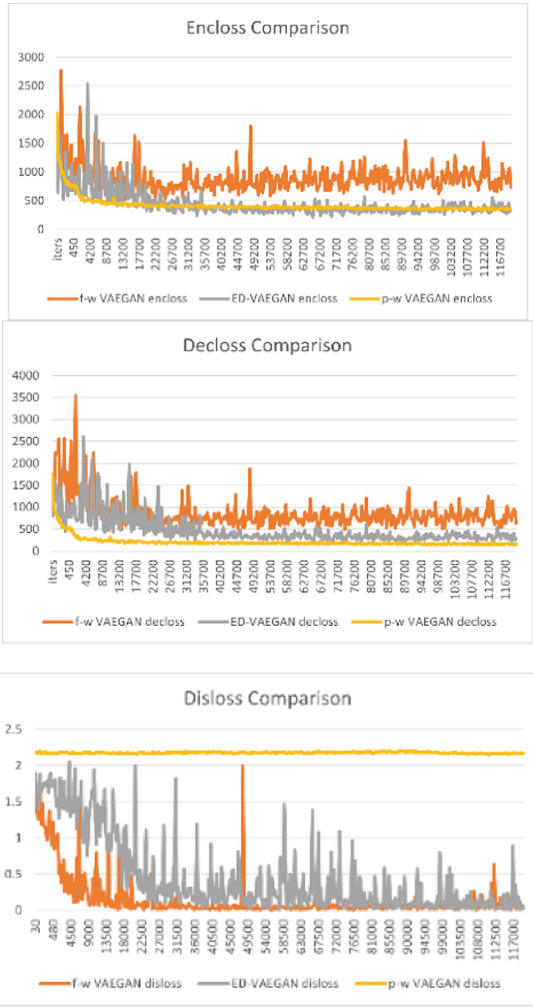

Fig. 6. p-w VAEGAN, f-w VAEGAN and ED-VAEGAN loss over iteration

Fig. 7. f-w VAEGAN (second row) and ED-VAEGAN (first row) comparison overtime

Fig. 8. Random generation after data augmentation

5 Conclusion

The present study puts forward a novel methodology, namely ED-VAEGAN, which enhances the efficacy of the encoder and decoder loss in the feature-wise VAEGAN. The study undertakes a comparative analysis of the quality of the training result images, while ensuring identical neural network and optimizers, at the same training iterations. The outcomes of the investigation reveal that the proposed ED-VAEGAN approach displays faster training times in comparison to the traditional VAEGAN. Furthermore, the exploration of the latent space highlights superior continuity in the ED-VAEGAN approach as compared to other pixel-wise models.

References

1. Kingma, Diederik P., and Max Welling. “Auto-encoding variational bayes.” arXiv preprint arXiv:1312.6114 (2013).
2. Goodfellow I Pouget-Abadie J Mirza M et al. 2020 Generative adversarial networks Communications of the ACM 63(11): 139–144Author, F.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010).
3. Radford A Metz L Chintala S Unsupervised representation learning with deep convolutional generative adversarial networks arXiv preprint arXiv:1511.06434(2015)
4. Larsen A B L Sønderby S K Larochelle H et al. Autoencoding beyond pixels using a learned similarity metric International conference on machine learning PMLR 1558–1566. (2016)
5. Zhang X Wang Z Lu K et al. Data Augmentation and Classification of Sea-Land Clutter for Over-the-Horizon Radar Using AC-VAEGAN arXiv preprint arXiv:2301.00947 (2023)
6. Ye F Bors A G. Learning latent representations across multiple data domains using lifelong VAEGAN Computer Vision–ECCV 2020: 16th European Conference Glasgow UK August 23–28 Proceedings Part XX 16 Springer International Publishing 777–795(2020)
7. Gur S Benaim S Wolf L. Hierarchical patch vae-gan: Generating diverse videos from a single sample Advances in Neural Information Processing System 33: 16761-1677(2020)
8. Doersch C. Tutorial on variational autoencoders arXiv preprint arXiv:1606.05908 (2016)

9. Gecer B Ploumpis S Kotsia I et al. Fast-ganfit: Generative adversarial network for high fidelity 3d face reconstruction IEEE transactions on pattern analysis and machine intelligence 44(9): 4879–4893(2021)
10. Zhu J Y Park T Isola P et al. Unpaired image-to-image translation using cycle-consistent adversarial networks Proceedings of the IEEE international conference on computer vision 2223–2232(2017)
11. MMLab Large-scale CelebFaces Attributes (CelebA)Dataset (2016) https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html
12. Berkeley Cycle GAN Datasets (2017) https://people.eecs.berkeley.edu/~taesung_park/CycleGAN/datasets/
13. Xu M Yoon S Fuentes A et al. A comprehensive survey of image augmentation techniques for deep learning Pattern Recognition 109347. (2023)