Generative Models for Data Synthesis

Kunwar Saaim, Supreeth Srinath, Shasha Fu

Generative Models for Data Synthesis

Kunwar Saaim[1] Supreeth Srinath[1] Shasha Fu[1]

[1]Department of Computing Science, University of Alberta
{ksaaim, srinath1, shasha5}@ualberta.ca

Abstract

Finding large quantities of high-quality data to train neural networks is one of the most challenging aspects for researchers since privacy restrictions and associated financial obligations make it difficult to gather the data. In recent years, Generative Adversarial Networks (GANs) have been extensively used in the generation of different types of datasets. Despite this, we cannot control the attributes that will be associated with a data sample generated by GANs. Combining GANs with Variational Autoencoders (VAEs) is an effective way of obtaining outputs which have the desired attributes. Our study examined the sampling capacity and quality of VAE and VAE-GAN for MNIST and Fashion MNIST datasets. Additionally, we introduce a new metric, Transformer Score (TS), to determine the quality of the generated data. It is based on a vision transformers network and demonstrates superiority over Inception Score.

Keywords: Data Synthesis, Generative Models, Variational Autoencoders, Neural Networks

1 Introduction

In order to train deep learning models, it is essential to have high-quality data sets. Therefore, the data must be representative of real-world scenarios and large enough to cover as many cases as possible. With a good data set, it's usually easier to generalize the characteristics of the data and to build a better model. Synthetic data can be useful for a variety of reasons, such as oversampling minority classes and generating new data sets in order to maintain the privacy of the originals [1]. The augmentation of data results in better performing models and can reduce generalization errors. However, in many cases, there is a requirement for more samples to be generated from some distribution and traditional augmentation techniques may not be good enough for certain tasks [2]. Generative Adversarial Networks [3] are generative models that are capable of producing synthetic data based on examples they have encountered in training. As mentioned in the paper [3], noise samples from a prior distribution function are used as inputs. In Variational Autoencoders (VAE) [4], instead of mapping the input to a fixed vector, it is mapped to a distribution and the bottleneck vector is replaced by a mean vector and a standard deviation vector. To pick samples for the generator, we propose using the latent space distribution of VAE. In other words, we plan on

studying the effects of combining VAE and GAN and collapsing the decoder and generator into one entity. Furthermore, we propose to evaluate the resulting models quantitatively and qualitatively, and improve the network accordingly.

2 Background

Generative modeling generates new samples from the same distribution with given training data. It aims to learn a generative model $P_{model}(x)$ that approximates $P_{data}(x)$. Generative modeling has broad applications. It can create realistic samples for artwork, super-resolution and colorization. Besides, generative models are useful to get insights from high-dimensional data in physics and medical imaging fields. It is also implemented to model the physical world for simulation and planning with robotics and reinforcement learning applications.

Deep generative models have a large family and variable categories such as autoregressive models (e.g., PixelCNN [5]), flow-based models (e.g., RealNVP [6]), latent variable models and energy-based models. Here we will introduce and discuss two most popular types of generative models: VAEs [4] and GANs [3], which belong to the category of latent variable models.

2.1 Variational Autoencoder

Variational Autoencoders (VAEs) is a latent variable model that combines the ideas of Autoencoder, reparameterization trick, amortization inference and variational approximation. VAE is composed of encoder and decoder, which are probabilistic. The structure of VAE is displayed as follows.

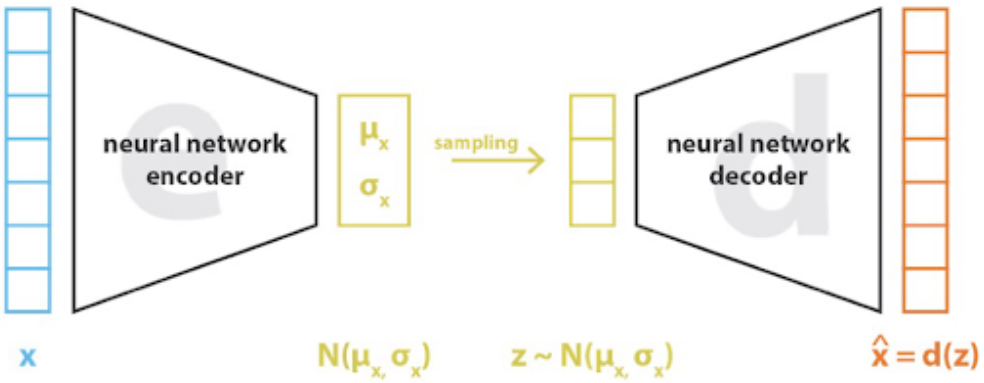

Figure 1: VAE structure Source: Adapted from [7]

Encoder network outputs mean and variance of Normal distribution as following equation.

$$q_\phi(z|x) = N(\mu_\phi(x) - \sigma_\phi(x)) \tag{1}$$

Decoder network outputs mean and optionally variance of Normal distribution.

$$p_\theta(x|z) = N(\mu_\phi(z), I) \tag{2}$$

The VAE assumes that the latent variable z follows $N(0, I)$ and the training data is generated from the distribution of unobserved (latent) representation z. The goal is to maximize the following objective function and train a generative model of the conditional probability.

$$L(x^{(i)}, \theta, \phi) = E_z(\log p_\theta(x^{(i)}|z)) = D_{KL}(q_\theta(z|(x^{(i)})||p_\theta(z)) \tag{3}$$

The loss function L provides a low bound of likelihood of $p(x)$, i.e. $p(x) >= L(x)$, so VAE is categorized in explicit density estimation methods. The reconstruction loss E_z maximizes the likelihood of original input being reconstructed. KL divergence loss makes approximate posterior distribution close to prior.

2.2 Generative Adversarial Network

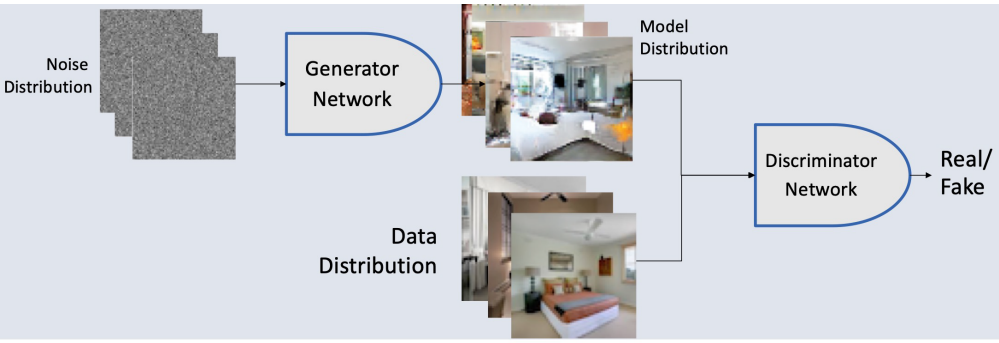

Figure 2: GAN structure Source: Adapted from [7]

Different from VAE, generative adversarial networks (GAN) focus on sample generation. GAN samples from a simple distribution (e.g. random noise) and learn transformation to training distribution. A GAN consists of a discriminator network and a generator network as the displayed structure above. The discriminator tries to distinguish between real and fake images. The generator tries to fool the discriminator by generating real-looking images. During the training process, the model tries to minimax the following objective function to make discriminator push up and generator push down.

$$\min_G \max_D V(D, G) = E_{x\ p_data(x)}[\log D(x)] + E_{z\ pz(z)}[\log(1 - D(G(z)))] \tag{4}$$

A generative model describes how a dataset is generated, in terms of a probabilistic model with probability density functions and prior probabilities. By sampling from this model, synthetic can be generated. The discriminator model evaluates the quality of the data created by the generator model. It receives input data samples from either the original data set, or created by the generator, and tries to predict the source of the sample. The generator learns to map a latent space to the distribution of the data it aims to replicate and imitate, so that when fed with a random vector from the latent space, it predicts a sample from the estimated distribution. The generator is evaluated by the discriminator, meaning that its aim is to create data samples that are similar to those in the original data set. The discriminator and the generator are trained simultaneously and they get better

by "competing" against each other and hence derive their name, "Generative Adversarial Networks".

3 Literature Review

To achieve generalizable deep learning models, large amounts of data is needed. But, labeled data is not always readily available. Recently, GAN based data augmentation has been used in CT scan segmentation [2] and retinal fundus images generation and segmentation [8]. Synthetic data can indeed boost model performance.

3.1 Generative Adversarial Networks (GANs)

Goodfellow et al. [3] first introduced generative adversarial networks that trained two fully connected networks (generator and discriminator) to generate images. Several approaches have been developed since then to generate photo-realistic images. DCGAN [9] adopted a convolution neural network for both the generated and discriminator. As an alternative to traditional GAN training, Arjovsky et al. [10] introduced WGAN that improves training stability and reduces model collapse risk. By progressively increasing the size of the image and adding layers to the network, the ProgressiveGAN [11] was successful in generating high resolution images. The Pix2Pix GAN [12], that uses a conditional GAN, mapped images from one domain to another, for instance, mapping edges to photos. Labels were required for the training of the network. Through the use of two GANs with cycle consistency loss, CycleGAN [13] further removed the requirement for labels. Recently, Vision Transformers (ViT) [14, 15] have also been used in GANs. As a result of using vision transformers architecture in GANs, ViTGAN [16] demonstrated that the regularization technique of convolutional GANs does not work on ViT GANs.

3.2 Autoencoders

The autoencoder, designed in order to denoise data by compressing it by encoding and reconstructing it through decoding [17], proved to be a good feature extraction tool. As a result of the latent vector (Rn) (compressed input representation), input features are captured in a low dimension, which follow simple arithmetic. As a result of their ability to extract features from an image, masked autoencoders [18] have been used in self-supervised learning for image classification. It has also been shown that variational autoencoders can be used for anomaly detection [19].

3.3 Controlled Image Generation

3.3.1 GANs

Vanilla GANs that play two player min-max game with a generator and discriminator are also able generate images with desired attributes. Conditional generative adversarial nets (CGAN) [20] extend the original generative adversarial nets to a conditional model so that both generator and discriminator are conditioned on some extra information y. The model

can generate images conditioned on class labels and learn multi-modal models like text language to image. Antipov et al.[21] further enhanced the cGAN to Age-cGAN that conditions on age and preserve the identity of the generated face. The face aging is simply the change of condition y at the input of the generator. Reed et al. [22] conditioned on text descriptions to generate desired attributes. Tan et al. [23] proposed ArtGAN to generate artistic images conditioned on some attributes K, the discriminator unlike a binary classifier also predicted K attributes which conditioned the generator.

3.3.2 VAEs

Variational Autoencoders (VAEs) [4] are enhanced Autoencoders where the probability distribution of the input data is learned using a Bayesian approach. A VAE generates new samples by utilizing the decoder part of Autoencoder, which has learned a mapping from latent space to data. Latent space is modeled as a standard normal distribution from which we can sample data points. Adversarial Autoencoder (AAE) introduced by Makhzani et al. [24] uses adversarial training to match the aggregated posterior of the hidden code vector with an arbitrary prior distribution. Wasserstein Auto-Encoder (WAE) [25] minimizes a penalised form of Wasserstein distance [10] between the model distribution and the target distribution to build generative model of data distribution. With stable training WAE is generalization of AAE.

3.3.3 VAE-GAN

Larsen et al. [26] first combined VAE and GAN by collapsing the decoder and generator into one. On the latent vector, simple arithmetic was used to obtain desired attributes. Hybrid VAE-GAN method [27] learns an inference model from a GAN model and capture data representation specific to the VAEs. The aim is to match the latent variables distribution with the data distribution jointly. The L-VAE-GAN method [28] allows one to discover disentangled representations across domains and automatically learn shared latent variables.

Weidong Yin et al. [29] introduced Semi-Latent GANs (SL-GANs) for generating and modifying facial images using high-level semantic attributes. The SL-GAN is composed of an encoder-decoder network, a GAN, and a recognition network. An encoder projects the facial images into a semi-latent attribute space containing both user-defined and latent attributes. As a generator, the decoder generates an image based on an attribute vector. The user-defined and latent attributes are jointly learned by the recognition network.

Jianmin Bao et al. [30] proposed a variational generative adversarial networks framework CVAE-GAN for synthesizing images in fine-grained categories such as faces of a specific person or objects in a category. The method consists of four components: an encoder network E, a generative network G, a discriminative network D and a classification network C. As in conditional variational auto-encoder (CVAE) [31], networks E are designed to learn structured output representations by utilizing deep conditional generative models. The function of networks G and D is the same as that of GANs [3].

Mingqi Hu et al. [32] developed a variational generator framework to capture semantic details behind conditional GANs and achieve fine-grain images with rich diversity. A variational inference is introduced into the generator to infer the posterior of a latent variable

only from the conditional input, resulting in a variable augmented representation for image generation. They use a novel auxiliary classifier that reduces adversarial training time and avoids mode collapse while respecting class-conditional constraints.

Zhang et al. [33] used U-Net [34] as auto-encoder for generating images in adversarial fashion. There are three components to the network: a U-Net based generator, an adversarial network D_z on the latent space, a discriminator D_x and an attribute classification network. U-Nets were trained using reconstruction losses on the generated and true images, and adversarial losses on the latent space. As a result of the attribute classification loss and adversarial loss of the discriminator D_x, realistic images with the desired attributes were guaranteed.

Yu et al. [**?**] combined VAEs and GANs to develop a VAE-GAN recommendation system. The authors presented a variant of VAEs that uses adversarial training for collaborative filtering. In order to train VAEs with an arbitrarily expressive inference model, adversarial variational bayes (AVB) is introduced. GANs are then used for implicit variational inference, which provides an approximation to posteriors and maximum likelihoods.

Hongyou Chen et al. [35] provided a general framework based on GANs and two autoencoders for the task of conditional image generation. The proposed network is intended to learn a generative model for the entire distribution of data, and to overcome the notorious problem of training instability. There are no typical problems associated with general generative models, such as mode collapse and unstable training, in the model.

Rui Gao et al. [36] proposed Zero-VAE-GAN, a joint generative model by combining VAEs and GANs for feature generation to cope with the zero-shot learning problem and generate unseen features. To enhance class level discriminability, an adversarial categorization network is incorporated into the joint framework. In order to augment features that are unlabeled and unseen, two self-training strategies have been implemented.

Feihong Li et al. [37] presented their VAE-GAN model for arterial spin labeling (ASL) image synthesis in Magnetic Resonance Imaging(MRI). In the GAN-based model, VAEs are used as a generator. In addition, Liu et al. [38] investigated VAEs with GANs objectives and proposed a dual-cycle constrained bijective VAE-GAN to solve the problem of tagged-to-cine MRI synthesis. Cine MRI is synthesized from its paired tagged MRIs using the network. Given tagged MR images, a variational autoencoder backbone and cycle reconstruction constrained adversarial training are used to produce accurate and realistic cine MR images.

To tackle the generalized zero-shot learning (GZSL) problem, Yuxuan Luo et al. [39] developed a dual learning framework called Dual VAE-GAN that combines VAEs and GANs for visual feature generation. Compared to VAEs, the dual VAE-GAN model produces more clear visual features and alleviates the model collapse problem of GANs. Peirong Ma et al. [40] also proposed a latent feature generation framework GAN-MVAE for the GZSL problem. GAN-MVAE maps the real and synthetic samples to the latent space of MVAE to further align them to make the data distribution synthesized by GAN more consistent with real data distribution. In order to train the final GZSL classifier, GAN-MVAE learns a discriminative latent space through latent distribution alignment and cross-modal reconstruction.

Two-Channel VAE-GANs proposed by Shengli Wang et al. [41] can address multiple image-to-video translation tasks, such as generating multiple videos of different categories. It consists of two channel encoders based on a VAE-GAN network. As a result of combining VAEs and GANs, the authors avoid the shortcomings of both, such as blurring caused by

VAE components, and unstable gradients caused by GANs.

3.4 Metrics

3.4.1 Inception Score

Deep generative models are powerful tools that have produced impressive results in recent years. These advances have been for the most part empirically driven, making it essential that we use high quality evaluation metrics. The Inception score (IS) [42] is such a popular metric for automatically judging the image outputs of image generative models. The IS takes a list of images and returns a single floating point number, the score. The score is shown to correlate well with human scoring of the realism of generated images [43]. The IS uses an Inception v3 Network [44] pre-trained on ImageNet and calculates a statistic of the network's outputs when applied to generated images.

IS aims to measure two desirable qualities of a generative model into a metric simultaneously. First, the images generated should contain clear objects, for example, the images are clear and sharp instead of blurry. In other words, the Inception Network should be highly confident there is a single object in the image. Secondly, The output of generative algorithms should have a high diversity of images from all different classes. For example, if the model generates dogs, each output image should be a different breed of dog. If both things are true, the score will be high. If either or both are false, the score will be low. A higher score is better. It means your GAN can generate many different distinct images. The lowest score possible is zero. Mathematically the highest possible score is infinity, although in practice there will probably emerge a non-infinite ceiling.

$$IS(\mathbb{P}_g) = e^{\mathbb{E}_{x\sim\mathcal{P}_g}[KL(p_{\mathcal{M}}(y|x)||p_{\mathcal{M}}(y))]} \tag{5}$$

The above equation is used to calculate IS. where $x \sim p_g$ indicates that x is an image sampled from p_g, $D_{KL}(p||q)$ is the KL-divergence between the distributions p and q, $p(y|x)$ is the conditional class distribution, and $p(y) = \int_x p(y|x)p_g(x)$ is the marginal class distribution. The exp in the expression is there to make the values easier to compare. If both of these traits mentioned above are satisfied by a generative model, then we expect a large KL-divergence between the distributions $p(y)$ and $p(y|x)$, resulting in a large IS.

IS is a powerful metric of generative models to simulate human's sense and evaluation. However, it also has limitations that need to be considered [42]. Firstly, the score is limited by what the Inception classifier can detect, which is directly linked to the training data commonly associated with ImageNet. For example, if the model is learning to generate things not present in the classifier's training data, then it may always get low IS despite generating high quality images because that image doesn't get classified as a distinct class. Or if the classifier network can not detect features relevant to the concept of image quality, then poor quality images may still get high scores. In addition, if the generator generates only one image per classifier image class, repeating each image many times, it can score highly such as no measure of intra-class diversity. Moreover, the model can score well when the generator memorizes the training data and replicates it.

3.4.2 Kernel MMD

Kernel MMD [45] measures the dissimilarity between a real distribution and the learned parameterized distribution for some fixed kernel function k. This metric gives good results when it operates in the feature space of a pre-trained feature detector network like ResNet [46]. Given two sets of samples from both the distributions, the empirical MMD between them can be computed with finite sample approximation of the expectation. A lower MMD means that the former distribution is closer to the latter. This metric can identify generative or noise images from real images, and both its sample complexity and computational complexity are low.

3.4.3 1-NN Classifier

1-NN classifier [47] outputs a score in the interval [0, 1], similar to accuracy/error in classification problems. When the generative distribution perfectly matches the true distribution, perfect score (i.e., 50% accuracy) is attainable. Typical GAN models tend to achieve lower leave-one-out (LOO) accuracy for real samples (1-NN accuracy (real)), while higher LOO accuracy for generated samples (1-NN accuracy (fake)). GANs are able to capture modes from the training distribution, such that the majority of training samples distributed around the mode centers have their nearest neighbor from the generated images, yet most of the generated images are still surrounded by generated images as they are collapsed together. The observation indicates that the mode collapse problem is prevalent for typical GAN models. This problem cannot be detected by human evaluation or Inception Score and thus proves the superiority of 1-NN classifier metric.

After comparing various metrics, Kernel MMD and 1-NN accuracy appear to be good metrics in terms of discriminability, robustness and efficiency.

3.5 Datasets

- **CelebA** [48]: CelebFaces Attributes dataset contains 202,599 face images of the size 178×218 from 10,177 celebrities, each annotated with 40 binary labels indicating facial attributes like hair color, gender and age
- **CelebA-HQ** [11]: The CelebA-HQ dataset is a high-quality version of CelebA that consists of 30,000 images at 1024×1024 resolution.
- **CIFAR-10** [49]: The CIFAR-10 dataset (Canadian Institute for Advanced Research, 10 classes) is a subset of the Tiny Images dataset and consists of 60000 32x32 color images.
- **LSUN Bedroom** [50]: The LSUN classification dataset contains 10 scene categories, such as dining room, bedroom, chicken, outdoor church, and so on.
- **MNIST** [51]: The MNIST database (Modified National Institute of Standards and Technology database) is a large collection of handwritten digits. It has a training set of 60,000 examples, and a test set of 10,000 examples.

- **Tiny ImageNet** [52]: Tiny ImageNet contains 200 classes for training. Each class has 500 images. The test set contains 10,000 images. All images are 64x64 colored images.

4 Methodology

Generating synthetic data in a controlled manner would be extremely beneficial. As an example, if the user is able to control face attributes when generating human face images, it would assist in generating some underrepresented groups that can be used in the design of unbiased face recognition systems. The vanilla GAN generates samples from the training data distribution using noise, but it is difficult to get specific attributes in a sample. It will be easier to control the output if we start with a reference image instead of noise and introduce the desired changes in the network.

We trained vanilla VAE that used reconstruction loss, and VAE-GAN that used the discriminator loss and reconstruction loss on discriminator l^{th} layer feature vector. There are times when the quantitative metric is good in terms of numbers, but the qualitative result is not satisfactory. In this regard, it is necessary to examine the various metrics available for GANs. We then compare the sampling capacity of the decoder using Inception Score (IS) and our introduced Transformer Score (TS) based on Vision Transformer [14]. The models were trained on MNIST digit [51] and Fashion MNIST [53] datasets, we also tried training on CelebA [48] but the model did not converge.

4.1 Network Structure

	MNIST
Encoder	(1, 28, 28)
Layer 1	Conv(128, 4, 2), BN, ReLU
Layer 2	Conv(256, 4, 2), BN, ReLU
Layer 3	Conv(512, 4, 2), BN, ReLU
Layer 4	Conv(1024, 4, 2), BN, ReLU
Layer 5	Linear(1024, latent_dim)*
Decoder	
Layer 1	Linear(latent_dim, 16384)
Layer 2	ConvT(512, 3, 2), BN, ReLU
Layer 3	ConvT(256, 3, 2), BN, ReLU
Layer 4	Conv(1, 3, 2), Sigmoid
Layer 5	-

*Doubled for VAE-based models

Figure 3: VAE Networks

We construct VAE and VAE-GAN with the same structures to make sure that our experiments and comparisons are fair and convincing for each model. As shown in figure 3,

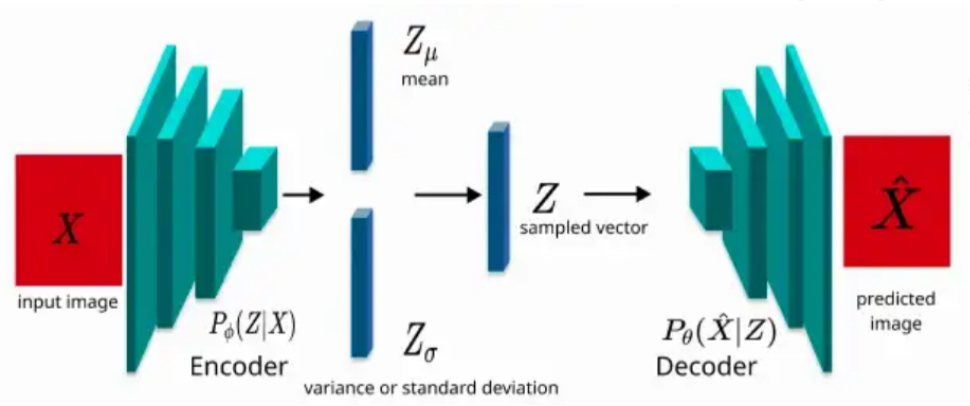

Figure 4: VAE Structure

the Encoder of VAE has the same structure as the Discriminator of GAN and the Decoder the same as the Generator of GAN. For example, training of VAE on MNIST dataset, the network is visualized in the above figure 4. The encoder takes MNIST image of size $1*28*28$ as input. The MNIST image is fed into four convolutional layers with each followed by Batch Normal layer and ReLU activation layer. In the fifth layer, the features are flattened as a latent dimension with *latent_dim* size. During the decoding process, the network takes latent dimension as input and generated an image of the same size as original input, which is achieved by a transpose convolution. In our VAE-GAN network as display in figure 5, we refer to [26] to combine VAE and GAN networks. The generator of GAN is replaced as VAE network. Therefore, we implement VAE to generate high quality fake images. Then, both real images and generated fake images are fed into discriminator to be identified.

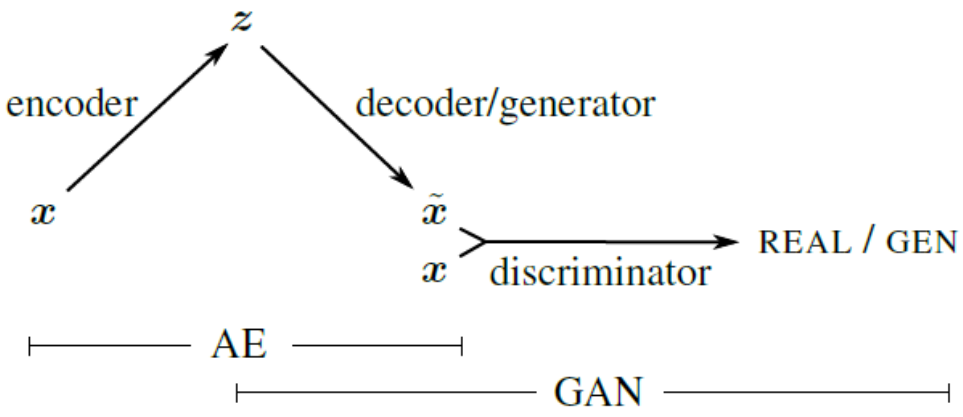

Figure 5: VAE-GAN Structure

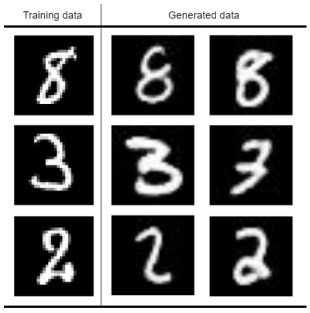

Figure 6: VAE MNIST images

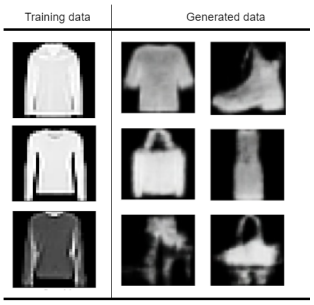

Figure 7: VAE Fashion MNIST images

4.2 Training

4.2.1 VAE

The architecture described in Figure 3 was implemented in PyTorch. We trained two models on MNIST and Fashion MNIST respectively using Mean Square Error as reconstruction loss function and KL Divergence as prior loss, with Adam optimizer and batch size of 256 for 67 epochs for MNIST and 85 epochs for Fashion MNIST. The learning rate was set to 1e-3 and the latent vector was set have 16 dimensions for MNIST model and 32 dimensions for the Fashion MNIST model.

4.2.2 VAE-GAN

The architecture of VAE-GAN is same as VAE with addition of a discriminator. The number of layers in the discriminator are same as the encoder having same number of filters corresponding to the encoder. VAE-GAN used adversarial loss, discriminator feature reconstruction loss and KL Divergence as prior loss. The learning was set to 1e-4 for MNIST and 1e-3 for Fashion MNIST. The latent dimensions were same as VAE for corresponding datasets.

4.2.3 Fine-Tuned Inception Score

Inception Scores calculate a statistic from the output of the network when applied to generated images using the Inception v3 model pre-trained on ImageNet. We fine-tuned the ImageNet-trained Inception v3 model on interpolated MNIST and Fashion MNIST images. Using 60000 images and the categorical cross entropy loss function with Adam optimizer, we trained the model over 20 epochs with a batch size of 128. In this case, the learning rate was set to 1e-3. Pre-processing involved changing the grayscale images to RGB color space and interpolating their height and width to 75 pixels. Fashion MNIST followed the same steps. Following that, we used the resulting models to evaluate the generative models by calculating IS for MNIST and Fashion MNIST data generated from VAE and VAE-GAN.

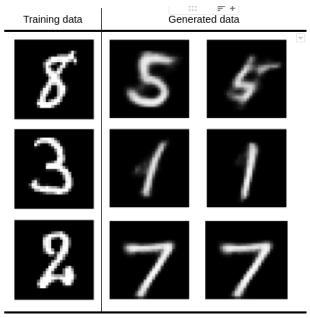

Figure 8: VAE-GAN MNIST images

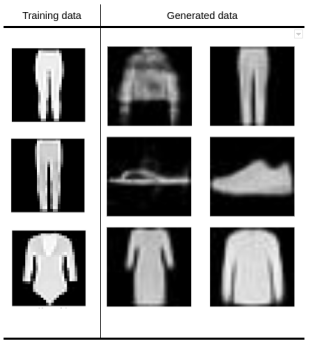

Figure 9: VAE-GAN Fashion MNIST image

4.3 Transformer Score

We propose a new metric named Transformer Score (TS) that uses ViT [14] backbone to classify images. The output probabilities of ViT are used to calculate Transformer Score similar to Inception Score. Further argument on why Transformer Score is a better metric is in discussion section.

VAE and VAE-GAN were build using Pythae [54] library. The inception networks were trained on TensorFlow and ViT was fine tuned using Transformers library, on MNIST and Fashion MNIST dataset.

5 Results

Model	VAE (1K)	VAE-GAN (1K)	VAE (5K)	VAE-GAN (5K)
Inception Score	8.028	8.031	8.281	8.043
Transformer Score	7.113	7.893	7.313	8.210

Table 1: Results on MNIST for 1K and 5K samples

Model	VAE (1K)	VAE-GAN (1K)	VAE (5K)	VAE-GAN (5K)
Inception Score	5.683	4.891	5.817	4.995
Transformer Score	5.572	4.593	5.827	4.791

Table 2: Results on Fashion MNIST for 1K and 5K samples

Table 1 show Inception Score and Transformer Score for MNIST dataset. We observe that for 1K samples VAE-GAN has better IS but for 5K samples VAE achieves a better IS score. Though for Transformer Score VAE-GAN has better metric for both 1K and 5K samples. The Inception Score is cluttered between a small range making it difficult differentiate between good and bad samples set. In case of Transformer Score the difference between TS of VAE and VAE-GAN is large, making it easy to differentiate between good and bad sampling models.

Table 2 shows the result for fashion MNIST on VAE and VAE-GAN. From figure 7 and figure 9 we can observe that VAE samples are better than VAE-GAN which is in line with the Inception Score and Transformer Score.

6 Discussion

In this section we discuss the drawbacks of convolutional neural networks as feature extractor and their effect on Inception Score.

6.1 Drawbacks of CNN Feature Extraction

Convolutional neural networks (CNNs) extract features from images and then classify, identify, predict or make decisions using those features. The most important step for CNN is feature extraction. There are mainly three good features of CNNs: local perception, parameter sharing, and multi-kernel. CNN, however, also has a number of drawbacks when it comes to extracting features. Firstly, there is the back propagation algorithm, which is not an efficient method in deep learning due to its high data requirements. Another one is translation invariance. Translation invariant means that neurons that recognize an object may not be activated if its orientation or position is slightly changed. If a neuron is used to recognize a cat, its parameters will change as the cat's position and rotation change. The problem has been partially solved by data augmentation, but not completely. The pooling layer also has a shortcoming. As a result of the pooling layer, a lot of very valuable information will be lost, and also the relationship between the whole and the part will be ignored. In order to recognize a face, we must combine several features (mouth, eyes, face outline, nose) together. According to CNN, if these five features appear together at the same time, it is likely that the face is human. Merging layers is a big mistake because it loses a lot of valuable information, and if we are talking about a face recognizer, it ignores the relationship between parts and wholes, which means we must combine some features to prove it is a face (mouth, eyes, oval face, nose). Consequently, CNN features are limited by their heavy reliance on local image textures for classification [55] and their invariance to translation and pooling.

6.2 Transformers as Global Feature Extractor

One key aspect of vision transformers is that they are global feature extractors, meaning that they can process the entire input image at once, rather than processing small patches or regions of the image separately. This allows them to capture long-range dependencies and context in the image, which can be important for understanding the overall scene or objects depicted in the image. Transformers rely on self-attention mechanisms, which allow the model to weight the importance of different positions in the input image when processing it. This allows the model to focus on relevant parts of the image and capture long-range dependencies and context. CNNs are well-suited for tasks that require the extraction of local features from images, while vision transformers are better at capturing global context and dependencies in the input data. In case of evaluating the quality of an image we need global

context rather than local features, therefore Transformer Score (TS) proves to be better metric than Inception Score(IS).

7 Conclusion

Based on the findings presented, it appears that combining Generative Adversarial Networks (GANs) with Variational Autoencoders (VAEs) can be an effective way of generating datasets with specific attributes. The authors also introduced a new metric called Transformer Score (TS) for evaluating the quality of the generated data, which was found to be superior to the Inception Score. Overall, the results suggest that VAE and VAE-GAN models have good sampling capacity and can produce high-quality data for training neural networks, but VAE-GAN is difficult to train due to unstable training.

References

[1] F. H. K. d. S. Tanaka and C. Aranha, "Data augmentation using gans," *arXiv preprint arXiv:1904.09135*, 2019.

[2] V. Sandfort, K. Yan, P. J. Pickhardt, and R. M. Summers, "Data augmentation using generative adversarial networks (cyclegan) to improve generalizability in ct segmentation tasks," *Scientific reports*, vol. 9, no. 1, pp. 1–9, 2019.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[4] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[5] A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, *et al.*, "Conditional image generation with pixelcnn decoders," *Advances in neural information processing systems*, vol. 29, 2016.

[6] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," *arXiv preprint arXiv:1605.08803*, 2016.

[7] "Deep Generative Models stefano ermon." `https://deepgenerativemodels.github.io/`. Accessed: 2021-09-01.

[8] P. Andreini, G. Ciano, S. Bonechi, C. Graziani, V. Lachi, A. Mecocci, A. Sodi, F. Scarselli, and M. Bianchini, "A two-stage gan for high-resolution retinal image generation and segmentation," *Electronics*, vol. 11, no. 1, p. 60, 2021.

[9] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[10] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*, pp. 214–223, PMLR, 2017.

[11] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.

[12] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

[13] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

[14] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[15] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

[16] K. Lee, H. Chang, L. Jiang, H. Zhang, Z. Tu, and C. Liu, "Vitgan: Training gans with vision transformers," *arXiv preprint arXiv:2107.04589*, 2021.

[17] L. Gondara, "Medical image denoising using convolutional denoising autoencoders," in *2016 IEEE 16th international conference on data mining workshops (ICDMW)*, pp. 241–246, IEEE, 2016.

[18] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

[19] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng, *et al.*, "Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications," in *Proceedings of the 2018 world wide web conference*, pp. 187–196, 2018.

[20] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[21] G. Antipov, M. Baccouche, and J.-L. Dugelay, "Face aging with conditional generative adversarial networks," in *2017 IEEE international conference on image processing (ICIP)*, pp. 2089–2093, IEEE, 2017.

[22] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, "Learning what and where to draw," *Advances in neural information processing systems*, vol. 29, 2016.

[23] W. R. Tan, C. S. Chan, H. E. Aguirre, and K. Tanaka, "Artgan: Artwork synthesis with conditional categorical gans," in *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3760–3764, IEEE, 2017.

[24] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," *arXiv preprint arXiv:1511.05644*, 2015.

[25] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf, "Wasserstein auto-encoders," *arXiv preprint arXiv:1711.01558*, 2017.

[26] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *International conference on machine learning*, pp. 1558–1566, PMLR, 2016.

[27] L. Chen, S. Dai, Y. Pu, E. Zhou, C. Li, Q. Su, C. Chen, and L. Carin, "Symmetric variational autoencoder and connections to adversarial learning," in *International Conference on Artificial Intelligence and Statistics*, pp. 661–669, PMLR, 2018.

[28] F. Ye and A. G. Bors, "Learning latent representations across multiple data domains using lifelong vaegan," in *European Conference on Computer Vision*, pp. 777–795, Springer, 2020.

[29] W. Yin, Y. Fu, L. Sigal, and X. Xue, "Semi-latent gan: Learning to generate and modify facial images from attributes," *arXiv preprint arXiv:1704.02166*, 2017.

[30] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, "Cvae-gan: fine-grained image generation through asymmetric training," in *Proceedings of the IEEE international conference on computer vision*, pp. 2745–2754, 2017.

[31] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Advances in neural information processing systems*, vol. 28, 2015.

[32] M. Hu, D. Zhou, and Y. He, "Variational conditional gan for fine-grained controllable image generation," in *Asian Conference on Machine Learning*, pp. 109–124, PMLR, 2019.

[33] J. Zhang, A. Li, Y. Liu, and M. Wang, "Adversarially regularized u-net-based gans for facial attribute modification and generation," *IEEE Access*, vol. 7, pp. 86453–86462, 2019.

[34] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[35] H.-Y. Chen and C.-J. Lu, "Nested variance estimating vae/gan for face generation," in *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2019.

[36] R. Gao, X. Hou, J. Qin, J. Chen, L. Liu, F. Zhu, Z. Zhang, and L. Shao, "Zero-vae-gan: Generating unseen features for generalized and transductive zero-shot learning," *IEEE Transactions on Image Processing*, vol. 29, pp. 3665–3680, 2020.

[37] F. Li, W. Huang, M. Luo, P. Zhang, and Y. Zha, "A new vae-gan model to synthesize arterial spin labeling images from structural mri," *Displays*, vol. 70, p. 102079, 2021.

[38] X. Liu, F. Xing, J. L. Prince, A. Carass, M. Stone, G. El Fakhri, and J. Woo, "Dual-cycle constrained bijective vae-gan for tagged-to-cine magnetic resonance image synthesis," in *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pp. 1448–1452, IEEE, 2021.

[39] Y. Luo, X. Wang, and F. Pourpanah, "Dual vaegan: A generative model for generalized zero-shot learning," *Applied Soft Computing*, vol. 107, p. 107352, 2021.

[40] P. Ma, H. Lu, B. Yang, and W. Ran, "Gan-mvae: A discriminative latent feature generation framework for generalized zero-shot learning," *Pattern Recognition Letters*, vol. 155, pp. 77–83, 2022.

[41] S. Wang, M. Xieshi, Z. Zhou, X. Zhang, X. Liu, Z. Tang, Y. Dai, X. Xu, and P. Lin, "Two-channel vae-gan based image-to-video translation," in *International Conference on Intelligent Computing*, pp. 430–443, Springer, 2022.

[42] S. Barratt and R. Sharma, "A note on the inception score," 2018.

[43] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in neural information processing systems*, vol. 29, 2016.

[44] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[45] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[47] Q. Xu, G. Huang, Y. Yuan, C. Guo, Y. Sun, F. Wu, and K. Weinberger, "An empirical study on evaluation metrics of generative adversarial networks," 2018.

[48] Z. Liu, P. Luo, X. Wang, and X. Tang, "Large-scale celebfaces attributes (celeba) dataset," *Retrieved August*, vol. 15, no. 2018, p. 11, 2018.

[49] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[50] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv preprint arXiv:1506.03365*, 2015.

[51] Y. LeCun, "The mnist database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.

[52] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, no. 7, p. 3, 2015.

[53] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[54] C. Chadebec, L. J. Vincent, and S. Allassonnière, "Pythae: Unifying generative autoencoders in python–a benchmarking use case," *arXiv preprint arXiv:2206.08309*, 2022.

[55] W. Brendel and M. Bethge, "Approximating cnns with bag-of-local-features models works surprisingly well on imagenet," *arXiv preprint arXiv:1904.00760*, 2019.