From Variational Autoencoder to VAE-GANs: A Comprehensive Review of Generative Models and Anomaly Detection Techniques

Kalpana Sathya Ponnada [1]

Abstract

Generative models belong to a category of machine learning models that generate new data from a given dataset. This approach has caught significant attention in the past couple of years due to its potential applications in many areas. Among generative models, GANs and VAEs are two popular approaches that have shown promising results in generating new data samples. This paper provides an overview of GANs and VAEs, showing their architectures, training, experimentation, and novel findings. This paper also shows a comparative study of GANs and GAN-VAEs, underscoring their advantages and limitations, and discussing an application in the field of anomaly detection.

1. Introduction

Over the past couple of years, generative models have gained immense popularity, with Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) being two of the most extensively used models. GANs use artificial neural networks and essentially consist of two networks - the generator(G) and the discriminator(D). G generates new images by taking a random noise vector as input, while D distinguishes between real and fake images by taking an image as input and outputting a label of 0 or 1. The primary objective of training a GAN is to train the generator such that the images generated by it can deceive the discriminator into labeling them as real. G and D are trained separately in a conflicting manner, with G trying to produce samples that are convincing enough to deceive D, while D improves by accurately identifying real and fake images.

VAEs are models that employ an encoder network for mapping data to a lower-dimensional representation. We have a variational base in this case. Autoencoders encode themselves, and the approach was inspired by the concept of image compression. VAEs are made up of two networks – an encoder(E) and a decoder(D). E produces new features from the old features by employing encoding/compressing techniques and D does the reverse process by reconstructing the output from the compressed input, making sure that the loss is minimum.

Both the VAEs and GANs have demonstrated immense potential in generating diverse and real-like samples and are being used in various areas such as image generation, anomaly detection, and data augmentation.

This paper shows a comparative study of VAE and VAE-GAN(Hybrid) models in terms of their architecture and presents experimental results that compare the performance of both models underscoring the observations. This paper also discusses how VAEs can be used in anomaly detection and highlights the advantages of using VAEs in this context.

2. Related Work

VAEs and GANs are two popular deep-learning models that have been widely used for generating high-quality synthetic data. In the past couple of years, there have been many studies that have explored the capabilities and limitations of these models for various applications. This section provides a brief overview of some of the most relevant works in this area.

Firstly, VAEs have been extensively used for image generation, and many studies have focused on improving the diversity and the quality of the generated images. One notable work is the Beta-VAE (Burgess et al., 2018) framework, is a type of Variational Autoencoder (VAE) that introduced a regularization term beta that encourages the disentanglement of the latent variables, leading to more interpretable and diverse generated images.

Secondly, GANs have been widely used for image and text generation, and have been shown to be capable of generating high-quality and realistic samples. One notable work is the StyleGAN (Karras et al., 2019) framework, which introduced a novel mapping network allowing better control over the generated images' attributes.

Finally, there have been several studies that have explored the combination of VAEs and GANs for improved image generation. One notable work is the VAE-GAN framework, which combines the reconstruction loss of VAEs with the adversarial loss of GANs to achieve better image quality and diversity.

3. Methodologies

3.1. VAEs

VAE , (*Fig 1*) is a latent variable-based model that can learn compressed input data, which can then be used to generate new data. The model consists of an encoder that associates the input data to a low-dimensional latent space, and a decoder that associates the latent space back to the original distribution. Many challenges have been faced with optimizing a latent variable model, where the data x is assumed to be generated from a latent variable z. The model is specified by a conditional probability distribution $p_\theta(x|z)$, which describes how the observed data is generated from the latent variable, and a prior distribution $p_\theta(z)$, which describes the distribution of the latent variables.

The goal is to estimate the parameters θ of the model based on a set of observed data points. One common approach is to use maximum likelihood estimation, which involves finding the parameter values that maximize the likelihood of the observed data given the model. The approximation is obtained by minimizing the KL divergence between the approximation and the true posterior.

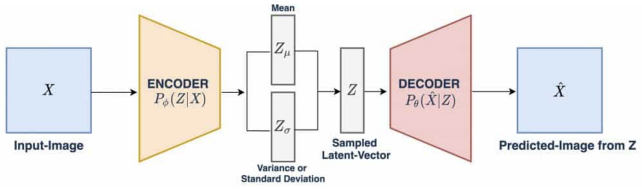

Figure 1. VAE architecture

3.1.1. VAEs for Out of Distribution Detection

Out-of-distribution (OOD) (Ran et al., 2022) detection is a task in machine learning that involves identifying samples or data points that are essentially different from the training data distribution. In other words, OOD detection is the process of detecting inputs that are outside the range of the data that a model was trained on.

OOD detection is important because many machine learning models are designed to work only within the range of data that they were trained on, and may give unreliable or incorrect predictions when presented with inputs that are significantly different from the training data. This is particularly important in safety-critical applications where the consequences of incorrect predictions can be severe.

Out of Distribution detection using machine learning has become significantly popular in recent years, with various approaches available. This is currently being solved using SVMs by identifying the normal region.

Reconstruction methods have gained traction for anomaly detection since the availability of deep learning. The idea behind these methods is that a model that learns to compress and reconstruct normal data will be unable to do so with anomalous data because it has only been trained on normal data. Therefore, a high reconstruction error can indicate the presence of anomalous data.

The main concept is that if an autoencoder is trained effectively with ample data and can produce a precise replica of the input data, it should have a stable and minimal reproduction error when given data that is comparable to the training data. However, if the autoencoder encounters input that is substantially different from what it was trained on, it will generate an unusual or extreme reproduction error, indicating that it was unable to reproduce it accurately. Therefore, if an input has a high reproduction error, it is likely to be an anomaly if it should resemble the training data. This approach is especially beneficial when the data has a high dimensionality, making it challenging to establish what is normal behavior or identify what is too extreme to be considered normal. For instance, a VAE that has been trained on MNIST images will struggle to reproduce a picture of a human face, and if presented with such an image, it will generate a relatively high reproduction error, indicating an anomaly.

VAE has been implemented so as to detect anomalies and used a loss function to train the VAE that balances reconstruction error and KL divergence. VAE labels samples that are out of distribution as 1 and those that belong to the distribution as 0.

3.2. GANs

GANs,*Fig 2* on the other hand, are composed of a generator network that learns to generate new data, and a discriminator that learns to distinguish between the generated samples and the real data. The architecture can be seen in Discriminator $D : R^n \rightarrow [0, 1]$ estimates the probability that a sample comes from the data distribution and generator $G : R^m \rightarrow R^n$ when given a latent variable z, captures probability to fool the discriminator by generating real-like samples. The two networks are trained in a conflicting manner, where the generator tries to fool the discriminator by generating realistic samples, and the discriminator tries to correctly distinguish between the generated and real samples. GANs can generate highly realistic and diverse samples but can be difficult to train and may suffer from mode collapse.

3.3. VAE-GANs

Combining GAN and VAE into a single network (Larsen et al., 2016) has been explored in some recent state-of-the-art research. The key change is the discriminator in GAN can be used in place of a VAE's decoder to learn the loss

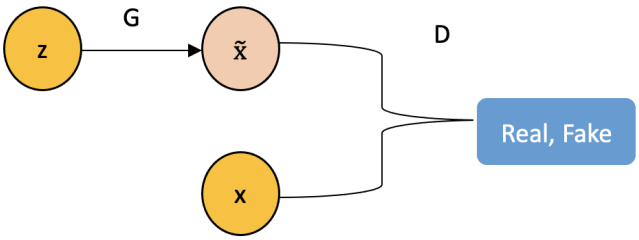

Figure 2. GAN architecture

function. This model simultaneously encodes, generates and discriminates the images. In the VAE-GAN hybrid model, the VAE and GAN are combined using the VAE decoder network as the generator network of the GAN. The VAE encoder network is used to encode the input data and generate the latent representation, which is then fed into the generator network. The discriminator network is trained to distinguish between real data samples and fake data samples generated by the VAE decoder. The generator network is trained to generate data samples that can fool the discriminator network.

By combining the VAE and GAN, the VAE-GAN hybrid model architecture (*Fig 3*) is able to generate high-quality data that are similar to the input data. The VAE part of the architecture allows for learning a low-dimensional representation of the data, which helps to reduce the data dimensionality by compressing the data and improving the quality of the generated samples. The GAN part of the architecture helps to produce unique and real like samples.

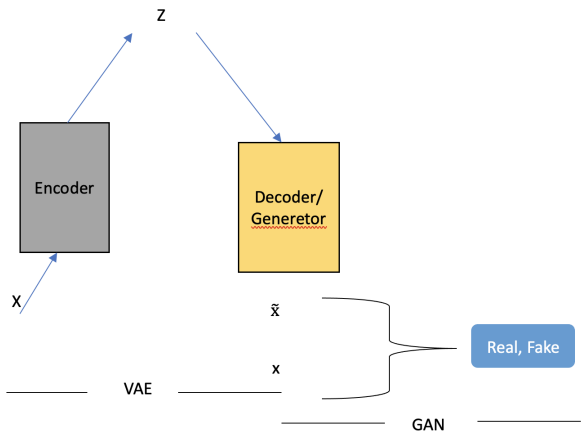

Figure 3. VAE-GAN architecture

4. Experimentation and Results

Experimentation is a crucial part for the comparison of GANs, VAEs, and VAE-GANs. Here are few experimentational steps, that have been followed to compare the above three models and for detecting out-of-distribution data.

4.1. Dataset selection and preprocessing

To compare all the above models, MNIST dataset (Deng, 2012) has been used.MNIST is a widely used dataset in machine learning for image recognition and classification tasks. It consists of 70,000 grayscale images of handwritten digits, with 60,000 images used for training and 10,000 for testing. Each image is 28x28 pixels in size, and each pixel has a value between 0 and 255 representing the intensity of the pixel. MNIST has become a standard dataset for evaluating the performance of different machine learning algorithms, and many state-of-the-art methods have been evaluated on this dataset. To implement all the above models, data has been preprocessed to ensure that data has been normalized, centred and scaled.The goal is to train all the three models so that they can accurately generate images that resemble MNIST images. To identify out of distribution data, the VAE model was trained on MNIST data and the Fashion-MNIST dataset was utilized for classification purposes. The Fashion-MNIST dataset comprises 70,000 grayscale images of clothing items from ten distinct categories, with a training set of 60,000 images and a test set of 10,000 images.

4.2. Model Configuration

4.2.1. VAE

The encoder is a neural network consisting of four convolutional layers that extract important features from the input image through batch normalization and LeakyReLU activation functions. The layers have different numbers of filters, with the first layer having one filter and subsequent layers having 32, 64, and 64 filters. All convolutional layers have a kernel size of (3, 3) with padding to preserve the spatial dimensions of the input image. The third and fourth convolutional layers reduce the spatial dimensions of the feature maps using a stride of 2, while the fifth convolutional layer maintains the spatial dimensions of the feature maps with a stride of 1.

The decoder is a neural network that takes the latent space representation as input and uses a dense layer to produce a vector with the size of the output shape before flattening. This vector is then reshaped back to the original shape. The decoder then uses a series of transposed convolutional layers, with the first layer having 64 filters and a 3x3 kernel size, to produce the final reconstructed image. Each layer is followed by batch normalization and LeakyReLU activation. The final transposed convolutional layer has a single filter

and produces the final reconstructed image, which is passed through a LeakyReLU activation function.

The model is trained using two losses- reconstruction loss and KL divergence loss, using the mean and log variance of the encoder output. The reconstruction loss is the difference between the original input image and the reconstructed image generated by the decoder part of the network whereas the KL divergence loss is the difference between the distribution of the latent space and a normal distribution. The loss function encourages the VAE to generate images that are as close as possible to the input data while also encouraging the learned latent space to be compact and smooth..

4.2.2. **VAE-GANs**

The VAE-GAN hybrid architecture combines the encoder-decoder architecture of a VAE with the generator-discriminator architecture of a GAN. The encoder-decoder part takes the input image and passes it through four convolutional layers with LeakyReLU activation functions and batch normalization. These layers extract features from the input image and reduce its spatial dimensions while increasing the number of channels. The output of the last convolutional layer is then flattened and fed into two dense layers that produce the mean and variance of the latent space. The mean and variance are used to generate a latent vector using a sampling layer. This latent vector is then fed into the generator part of the network.

The generator network takes the latent vector as input and generates a fake image that is similar to the real images in the training dataset. The generator part of the network consists of several transposed convolutional layers with batch normalization and LeakyReLU activation functions. These layers gradually increase the spatial dimensions of the input while decreasing the number of channels until the final output has the same dimensions and number of channels as the input image.

The discriminator network takes as input either a real image from the training dataset or a fake image generated by the generator network. The discriminator network then outputs a probability value indicating whether the image is real or fake. The discriminator part of the network consists of several convolutional layers with batch normalization and LeakyReLU activation functions. These layers extract features from the input image and reduce its spatial dimensions while increasing the number of channels. The output of the last convolutional layer is then flattened and fed into a dense layer that produces the final probability value.

The model is trained using a combination of reconstruction loss, KL divergence loss, and adversarial loss. The reconstruction loss measures the difference between the original input image and the reconstructed image generated by the decoder part of the network. The KL divergence loss measures the difference between the distribution of the latent space and a normal distribution. The adversarial loss measures the difference between the output probability of the discriminator for real and fake images. The network is trained to minimize the total loss, which is a weighted sum of the three losses.

4.3. Training and hyperparameter-tuning

Training the models VAE and VAE-GANs on a GPU can significantly reduce training time compared to using a CPU. In terms of the training process, both models were trained for 50 epochs. An epoch is a single iteration over the entire training dataset. The batch size used was 32, which means that the model's weights were updated after processing 32 images at a time. Using a larger batch size may lead to faster training times but can also require more memory and may not generalize as well. A smaller batch size, on the other hand, can lead to slower training times but can provide more accurate weight updates.

The Adam optimizer has been shown to be effective for a wide range of deep learning tasks and was used to update the weights of both models. Adam is a popular optimization algorithm that adapts the learning rate for each parameter based on the first and second moments of the gradients.

Lastly, hyperparameter tuning is critical to achieving good performance in both VAE and VAE-GANs. Hyperparameters are parameters that are not learned during training, such as the learning rate, batch size, and number of epochs. Tuning these hyperparameters involves adjusting their values to find the optimal combination that maximizes performance on a validation dataset. The specific hyperparameters used in training these models are learning rate of the optimizer, epochs, batch size.

4.4. Results and Metric Evaluation

The performance of VAEs and VAE-GANs was evaluated based on several metrics, including the accuracy, reconstruction loss, visual quality of the generated images. The results and performance evaluation are presented in tables.

4.4.1. **VAE**

VAE has been trained for 50 epochs and the output can be seen in *Fig 4*. *Fig 5* shows the graph drawn between the loss and number of epochs for both training and testing data.
For VAEs, the average reconstruction loss was calculated, and it was found that it is less for testing data than for the training data,indicating that VAEs can better capture the underlying distribution of the data. This can be seen in the graph fig5 that was plotted between the number of epochs and the reconstruction loss.

Figure 4. VAE on MNIST dataset at 50th epoch

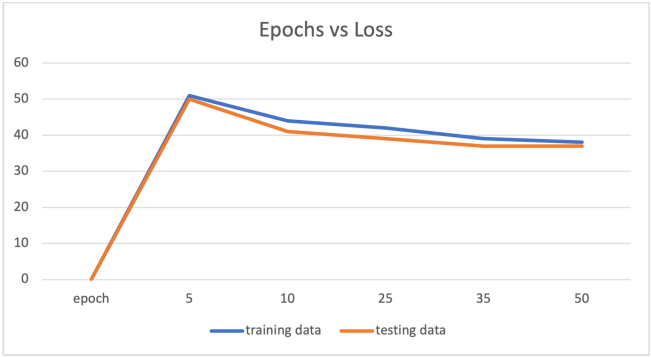

Figure 5. Loss vs number of epochs-testing and training data

4.4.2. **VAE-GAN**

VAE-GAN performed better since VAE-GAN models are trained to minimize the reconstruction loss between the input and output images. Lower reconstruction loss indicates better performance. The output of VAE-GAN at 50th epoch is in *Fig 6*.

Figure 6. VAE-GAN on MNIST dataset at 50th epoch

A graph has been plotted between the loss and number of epochs for VAE-GANs as well and can be seen in *Fig 7*.

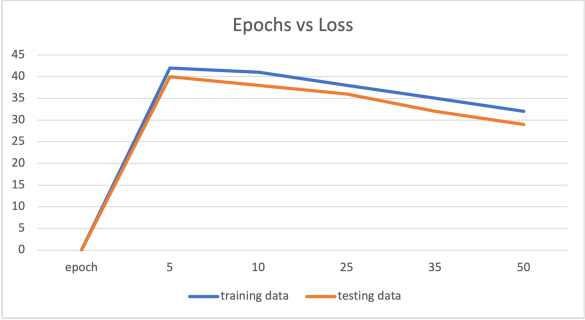

Figure 7. Loss vs number of epochs-testing and training data

4.4.3. **Comparison between VAE and VAE-GANs**

In terms of the visual quality of the generated images, VAEs were able to generate high-quality images, although some blurriness was observed in the generated images.

For VAE-GANs, the results were evaluated based on both the reconstruction loss and the quality of the images generated. The average reconstruction loss was found to be slightly higher than that of VAEs, indicating that the adversarial loss can sometimes hinder the reconstruction ability of the VAE-GAN. However, VAE-GANs were found to generate higher-quality images with better sharpness and fewer artifacts than VAEs.

Table 1 shows the loss and similarity obtained at 50th epoch for both VAE and VAE-GAN models when trained on MNIST dataset. The similarity is calculated using cosine

similarity metric that compares the input and output image vectors by calculating the angle between them.

Model	Loss	Similarity
VAE	39	73.08%
VAE-GAN	28	80.126%

Table 1. Model evaluation using reconstruction Loss

4.4.4. VAEs FOR ODD

Since we use VAE for out of distribution detection, the loss function used is the weighted average of reconstruction loss and KL loss. *Fig 8* shows the graph that has been plotted between the number of epochs and the loss.

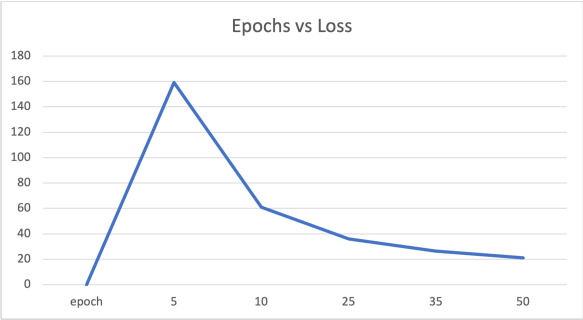

Figure 8. Loss vs number of epochs

Table 2 shows different metrics such as precision, recall and f1-score that were evaluated for the samples that were labeled as 0 and 1.

Label	precision	recall	f1-score
0	0.91	0.99	0.94
1	0.98	0.90	0.94

Table 2. Model evaluation using precision,recall and f1-score

Fig 9 shows the points that are out of distribution in orange colour that were classified by VAE where as *Fig 10* shows the true classification of all the datapoints.

4.5. Observations

4.5.1. VAE

- VAEs are able to effectively learn a compressed representation of the MNIST images using an encoder-decoder architecture. The encoder associates the input images to a lower-dimensional latent space, while the decoder maps it back to the input space.
- The latent space learned by the VAE has a continuous structure that allows for interpolation between different

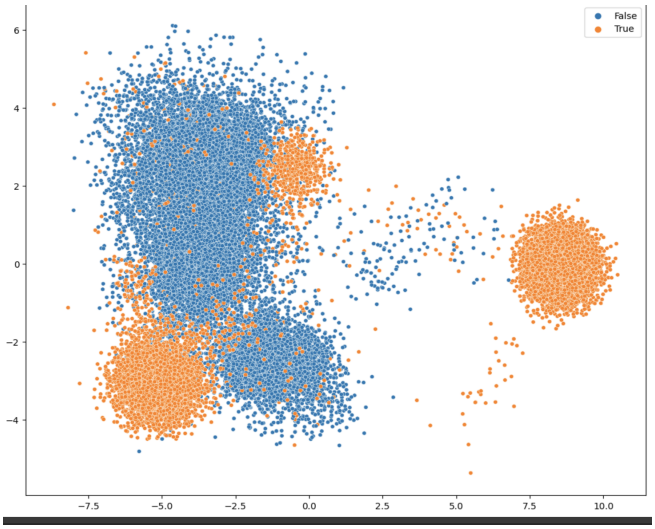

Figure 9. Predicted Out of Distribution Detection

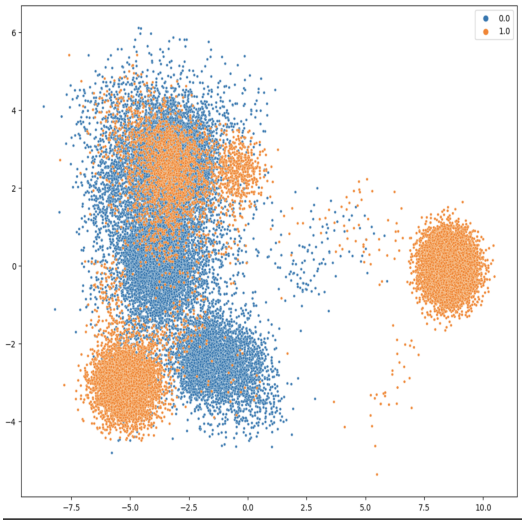

Figure 10. True Out of Distribution Detection

digit classes. This means that the VAE can generate new digits by sampling from the latent space and decoding the samples back to the input space.
- The reconstruction loss, which measures the difference between the input images and the generated images, is used to train the VAE. The VAE is able to achieve low reconstruction loss on the MNIST dataset, indicating that it can effectively learn a compressed representation of the images.
- VAEs can be further improved by incorporating techniques such as convolutional layers and regularization methods such as dropout.

4.5.2. **VAE-GAN**

- VAE-GANs combine the benefits of VAEs and GANs to improve the quality of generated images. The VAE-GAN architecture includes a VAE encoder-decoder and a GAN discriminator-generator.
- The VAE-GANs have less loss when compared to plain VAEs implying that the VAE-GANs are able to generate high-quality images with sharp edges and more realistic textures than VAEs.

4.5.3. **OOD**

- VAEs capture the data points that are out of distribution with a precision of 98% when trained correctly.
- While the vast majority of the points that are out-of-distribution were correctly identified, still there is a small group of data points that were not detected, due to the similarity to the normal points.

5. Applications

VAEs and GANs have many practical applications in various fields, and their development and use are an active area of research in the machine-learning community.
Some applications of VAEs include:

- Removing noise from images, which has applications in medical imaging and other fields where image quality is important.
- Detecting anomalous data points that do not fit the normal distribution of the training data. This has applications in fraud detection and security.

Some applications of GANs include:

- Translating images from one domain to another. This has applications in style transfer and image editing.
- Generating images from textual descriptions. This has applications in creating images for virtual assistants and chatbots.

6. Conclusion

In conclusion, this paper discusses interesting and useful models - Variational Autoencoder (VAE), VAE-Generative Adversarial Network (VAE-GAN), and an application - Out of Distribution (OOD) detection.

VAE is a generative model that uses a latent variable representation to capture the underlying distribution of the data. The model is trained using variational inference, which enables efficient computation of the maximum likelihood of the data. VAE-GAN is a hybrid model that combines the VAE with the GAN framework. This model aims to overcome some of the drawbacks of the VAE, such as blurry image generation, by using the GAN's discriminative framework to provide sharper and diverse images. The VAE-GAN has been shown to generate high-quality images in various datasets. The OOD detection is an application of VAE for detecting anomalous data points that are not represented in the training data. The method uses the reconstruction error of the VAE as a measure of the likelihood of the input being from the in-distribution or out-of-distribution. This method has been shown to be effective in detecting anomalous data points. Overall, this paper presents interesting and useful observation on generative modeling and anomaly detection, which can be applied to various fields such as computer vision, natural language processing, and healthcare.

References

Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018. URL https://arxiv.org/abs/1804.03599.

Deng, L. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. doi: 10.1109/MSP.2012.2211477. URL https://ieeexplore.ieee.org/document/6296535.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019. URL https://ieeexplore.ieee.org/document/8953766.

Larsen, A. B. L., Sønderby, S. K., Larochelle, H., and Winther, O. Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning*, pp. 1558–1566. PMLR, 2016. URL https://arxiv.org/abs/1512.09300.

Ran, X., Xu, M., Mei, L., Xu, Q., and Liu, Q. Detecting out-of-distribution samples via variational auto-encoder with reliable uncertainty estimation. *Neural Networks*, 145:

199–208, 2022. URL https://arxiv.org/abs/2007.08128.