# Improved Density-Estimation and Restoration based Vector-Quantized Anomaly Detection

Student Name: Mr. Rei Ishii
Supervisor Name: Dr Amir Atapour-Abarghouei
Submitted as part of the degree of BSc Computer Science to the
Board of Examiners in the Department of Computer Sciences, Durham University

**Abstract**—Vector-Quantized anomaly detection remains a new and relatively unexplored area of research, thus it is necessary to determine the optimal architectures for existing methods, and hence compare their performances. This paper explores how different designs of the Vector-Quantized model and Auto-Regressive model impact the performance of Density-Estimation and Restoration based anomaly detection. This was done by introducing perceptual quality to the learned vector codebook, as well as comparing PixelSnail and the GPT-2 Transformer as Auto-Regressive models to determine the need for learning local spatial dependencies. Extensive experimentation found that Vector-Quantized models utilizing a perceptually rich codebook allowed it to perform better than existing state of the art methods. Furthermore, learning local spatial dependencies was found to be of importance for Restoration methods, which encourages the use of Auto-Regressive models that include convolutional layers. It was also found that unlikely latent variables do not necessarily correspond to anomalous regions, thus revealing an inherent weakness of Restoration-based methods. Lastly, this paper also outlined the high computational cost of the current Vector-Quantized Restoration method, and proposes Light-Pixel-wise detection, a much faster, light-weight variant with improved performance.

**Index Terms**—Classifier design and evaluation, Computer vision, Image models, Neural nets

## 1 Introduction

Anomaly detection is the process of identifying data samples which greatly deviate from the majority of data [1] [2] [3]. Anomaly detection is used in many industries, ranging from risk management [4], health [5] [6] [7] [8], video surveillance [9] [10] [11], etc. It is especially useful in security and intrusion detection systems [12], as it can alert suspicious behavior which is unknown, instead of only searching for known threats. Image-based anomaly detection models aim to learn the qualities of normal images, such that an image containing an anomalous entity could be identified [1] [2].

Deep learning methods are the most popular amongst current image-based anomaly detection methods [2] [5]. In particular, unsupervised learning methods [5] [11] [13] [7] [8] are often favored over supervised learning methods [14] [15]. This is because the latter requires a large and diverse annotated dataset, and are constrained to the specific anomalies in the data and are hence unable to generalize to unseen abnormalities [1] [5].

Regenerative models are commonly used in image based unsupervised anomaly detection [1] [7] [8]. These models aim to train a generative model to reconstruct only normal images, such that when it is fed an abnormal image it fails to do so and thus identifies it as an anomaly. An example of this is the use of Variational Autoencoders (VAEs) for anomaly detection [7]. During training, the VAE model minimizes a loss function comprising of the reconstruction loss (typically L1 loss between original image and reconstruction) as well as a Kullback-Leibler (KL) divergence term [16] (which measures the distance between the prior and latent distribution). During inference, the reconstruction loss is commonly used as the anomaly score, based on the assumption that the model can only reconstruct normal images accurately.

However, recent papers [8] have found that the high representational power of VAEs allow them to reconstruct unseen anomalies, suggesting that the reconstruction loss is an ineffective anomaly score. Furthermore, Restoration based anomaly detection [5] [17] has been proposed as a newer alternative method, where the aim is to remove anomalies from the image (effectively restoring it) and compare it to the original anomalous sample. An example of this is presented by Chen *et al.* [17], whom models anomalies as spatially localized deviations from a prior distribution of normal samples, running gradient descent in the pixel space to allow them to remove the anomaly and restore the image. These kind of approaches have been shown to outperform their Regenerative counterpart [7].

Likewise, new methods have been put forward which hypothesize anomalies to be encoded in a lower density region in the latent space [5], and hence use the latent Density-Estimation as the anomaly score. These are categorised as Density-Estimation based anomaly detection methods [5] [18], which are widely used in unsupervised anomaly detection. Density-estimation methods aim to learn the probability density function of normal samples, such that it can classify data points with low densities as anomalies [19]. An example is demonstrated by Marimont *et al.* [5], where the they train a Vector-Quantized Variational AutoEncoder (VQ-VAE) [20] to learn a discrete latent distribution of normal samples, and models this with an Auto-Regressive (AR) model which have been shown to perform well in Density-Estimation of images [5] [20] [21] [22]. This method also leverages the generative capabilities of the AR model

to achieve latent space Restoration, which imputes potential anomalous latent variables with samples from the learnt prior AR model, effectively 'restoring' the anomalous latent space. This is referred to as Pixel-wise anomaly detection [5], where the Pixel-wise residuals between the original image and the image generated from the restored latent space is used as the anomaly score. The authors also introduce Sample-wise anomaly detection [5], where the sum of all negative log likelihoods of the latent variables (which are also estimated by the AR model) is used as the anomaly score.

One can see that both of these methods rely heavily on the quality of the learnt latent space, as well as the AR model's ability to estimate the prior probability. This paper hence aims to determine if the performance of the aforementioned Pixel-wise and Sample-wise anomaly detection could be improved, by learning a more accurate codebook with the introduction of a discriminator and perceptual loss for the VQ model, and utilizing a Transformer [23] instead of PixelSnail [24] as the AR model.

The introduction of an adversarial training procedure as well as perceptual loss has been shown to improve the quality of the latent space, and likewise improved performance in image modeling and image generation [21] [26]. Moreover, the key difference between a Transformer and the PixelSnail model is that the latter is purposely designed to capture local spatial dependencies [24], as it is meant to work in the pixel space. This is explained in further depth alongside the PixelSnail architecture in 2.3.

Spatial dependence is often cited as the first law of geography [27], and it refers to the degree of spatial correlation between independent values in a geographical space [28]. The existence of local spatial dependencies formulates the first property of spatial data, which is that nearby elements are more related than those far away [28]. An example of local spatial dependency in computer vision is pixel values; the value of a pixel is more influenced by its neighbors than pixels which are distant.

While local spatial dependencies are important when modeling pixel values, this may not necessarily be true for modeling codebook vectors. This is because these vector encodings are an abstraction of the pixel level image, and could contain more complex, long range dependencies. Previous works has shown Transformers to outperform PixelSnail in modeling latent priors for generative tasks [21] [22], and so they may be more appropriate in modeling the latent prior in the context of anomaly detection as well.

Hence, this paper will aim to answer the research question:

> "How does learning local spatial dependencies between codebook vectors, as well an adversarially trained perceptually rich codebook in Vector-Quantized models impact the performance of Density-Estimation and Restoration based anomaly detection in images?"

Vector-Quantized models are widely used in the field of image generation [20] [21] [26] and image modeling [22], but their applications to anomaly detection is not explored with as much depth [2]. Previous studies [5] proposing anomaly detection methods which leverage the VQ-VAE do not experiment with different AR models, nor do they incorporate more modern designs such as elements from the Vector-Quantised Generative Adversarial Network (VQ-GAN) [21] which could potentially improve performance at a low cost. Hence the topic question is important to explore, as the impact of learning local spatial dependencies will determine the optimal AR model for anomaly detection tasks. Likewise the impact of a perceptually rich codebook will determine if the VQ-GAN architecture, which has been shown to greatly improve results in image generation tasks [21] [26] can be applied to the context of anomaly detection as well.

Furthermore, the objectives of this paper can be split into three as follows:

- **Basic Objective:** To determine the impact of a perceptually rich codebook on anomaly detection performance.
- **Intermediate Objective:** To determine the impact of learning spatial dependencies on anomaly detection performance.
- **Advanced Objective:** To introduce a new restoration-based method which leverages the improved codebook.

Note that the research question does not ask whether or not spatial dependencies *exist* within codebook vectors, but rather if it is necessary for the AR model to learn it for the two anomaly detection methods. Hence the paper will compare the use of two AR models; PixelSnail and an attention based Transformer, where the former is designed to learn local spatial dependencies in the input via convolutional layers. While it is certainly possible for a Transformer to do the same as it looks at interactions between all elements in a sequence (and thus the interactions between neighbouring elements as well) the idea is that it is not forced to capture them (since it has no convolutional layers), and is able to focus on learning more complex, long-range dependencies. Therefore if there is a case where the Transformer outpreforms PixelSnail as an AR model, it could be an indication that learning local spatial dependencies for that instance is unnecessary.

The basic and intermediate objectives will be achieved by comparing 4 different model combinations:

- **VQ-VAE+PixelSnail:** This models both spatial and long-range dependencies in a non-perceptually rich codebook.
- **VQ-VAE+Transformer:** This focuses on long-range dependencies of a non-perceptually rich codebook.
- **VQ-GAN+PixelSnail:** This models both spatial and long-range dependencies of a perceptually rich codebook.
- **VQ-GAN+Transformer:** This focuses on long-range dependencies of a perceptually rich codebook.

Each of these combinations will be tested with the aforementioned Sample-wise anomaly detection which is a Density-Estimation method, as well as Pixel-wise anomaly detection which is a Restoration method. Additionally, the VQ-VAE and VQ-GAN's Regenerative anomaly detection capabilities will be tested as a baseline score.

The advanced objective will be achieved by proposing Light-Pixel-wise detection, a much faster, less computationally expensive variant of Pixel-wise detection. Unlike its

precursor, Light-Pixel-wise detection does not rely on the variety of restorations, and instead makes use of its improved restoration capabilities via the improved codebook.

It is important to note that, although there are a large variety of Density-Estimation and Restoration based anomaly detection techniques [2] [5] [7] [18] [29], this paper will only look at those using Vector-Quantized models. Hence in this paper the phrase "Density-Estimation method" would refer to Sample-wise detection, and "Restoration method" would refer to Pixel-wise and Light-Pixel-wise detection.

The experimental results in this paper show that the impact of learning local spatial dependencies and hence the best choice of AR models between PixelSnail and a Transformer is dependent on whether the anomaly score is computed in the pixel space or the latent space. The paper will also demonstrate that incorporating a perceptually rich codebook greatly increases anomaly detection performance, especially for Density-Estimation methods which is shown to outperform other state of the art models. Lastly, the paper will also show that the proposed Light-Pixel-wise detection method achieves similar performance to Pixel-wise detection, at a fraction of the running time.

## 2 Related Work

### 2.1 Neural Discrete Representation Learning

The authors of [20] propose the Vector-Quantized Variational Autoencoder (VQ-VAE) which aims to learn discrete latent representations of images for image generation tasks. The defining characteristics of the VQ-VAE is that the prior is learnt rather than static, and the encoder produces discrete rather than continuous outputs [20].

Specifically, given a latent space $e \in \mathbb{R}^{h \times w \times C}$ where $h, w$ are the height and width of the latent space, and $C$ is the channel dimension, the encoder $E$ takes an input $x$ and encodes this as $z_e(x)$ in to this space such that $z_e(x) \in \mathbb{R}^{h \times w \times C}$. The discrete latent variables are then calculated by simply taking the nearest neighbours as shown in (1).

During training, the codebook embedding layer *VQ* replaces each value of the encoder output by the closest distribution's categorical value. This process is referred to as quantization, and allows the latent space to be discrete. Likewise the posterior distribution for the VQ-VAE $q(z|x)$ can be defined as:

$$q(z = k|x|) \begin{cases} 1 \text{ for } k = argmin_j \left\| z_e(x) - e_j \right\|_2 \\ 0 \text{ otherwise} \end{cases} \tag{1}$$

The corresponding codebook vector $e_k$ shown in (2), is then fed to the decoder, where the original image is reconstructed (using only the codebook vectors).

$$z_q(x) = e_k, where\ k = argmin \left\| z_e(x) - e_j \right\|_2 \tag{2}$$

It is also important to note that, since the *argmin* function is not differentiable (and hence gradients will not flow through during backpropagation), the gradients from $z_q(x)$ are copied over to $z_e(x)$. This also means that the L2 loss (nearest neighbours loss) is not being minimized during training.

Hence, the reconstruction of an image by the VQ-VAE can be represented as:

$$\hat{x} = G(VQ(E(x))) \tag{3}$$

Where *E* is the encoder, *G* is the decoder, and *VQ* is the codebook embedding layer.

Training the VQ-VAE is a bidirectional task. The codebook vectors must learn to align with the encoder outputs, and vice versa. Firstly, the visual features are learnt by the model by minimizing the MSE of the encoder outputs:

$$L_{vq} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{|z|} (sg(z_j^i) - \hat{z}_j^i)^2 \tag{4}$$

Here, $z$ is the encoder output, $\hat{z}$ is it's quantized latent code, $sg$ is the stop gradient operator, and N is the number of observations. The stop gradient prevents its inputs from contributing to the gradient calculations. The encoder also learns to produce latent codes similar to the learnt features through the commitment loss.

$$L_c = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{|z|} (z_j^i - sg(\hat{z}_j^i))^2 \tag{5}$$

Here, the stop gradient is applied on the latent codes (since this MSE is intended to help the encoder learn). Lastly, the standard reconstruction loss for the auto encoder is represented as:

$$L_r = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{h*w} (x_j^i - \hat{x}_j^i)^2 \tag{6}$$

Where $x$ is the input image and $\hat{x}$ is the reconstruction. It is important to note that unlike a traditional VAE, the KL divergence [16] loss is not used as the divergence will always be constant (with respect to the encoder parameters) [20] for uniform categorical distributions. Hence the overall loss can be expressed as:

$$L = L_{vq} + L_r + \beta L_c \tag{7}$$

Where $\beta$is simply a hyper-parameter that controls the influence of the commitment loss. The authors found that the model is quite robust to $\beta$, and results did not vary between values of $1.0 \sim 2.0$.

The VQ-VAE is important to this paper as it is able to learn a discrete representation of normal images in an unsupervised manner. And hence allows it to be used together with the Density-Estimation and Restoration methods explained in 2.4.

### 2.2 Transformer architectures

The unique characteristic of Transformers is that they can model dependencies between elements in a sequence solely with attention mechanisms [23], allowing them to capture interactions between inputs regardless of their individual positions. Although they were originally built for natural language processing tasks [23] [30], many variations which handle audio [31] or image [22] data have been developed. The general Transformer architecture consists of a self-attention mechanism in each layer which captures the interactions between the sequence elements, as well as a

position-wise fully-connected network, which is independently applied to all positions [30].

The self-attention mechanism uses three position-wise linear layers to map an intermediate representation into three vectors; the query $Q \in \mathbb{R}^{N \times d_k}$, key $K \in \mathbb{R}^{N \times d_k}$, and value $V \in \mathbb{R}^{N \times d_v}$, where $d_k$ and $d_v$ are the dimensions of the key and value vectors respectively. These are used to compute the final output which can be represented as:

$$Attn(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \in \mathbb{R}^{N \times d_k} \quad (8)$$

In the context of modeling elements via maximum likelihood estimation in an Auto-Regressive manner, the output of the Transformer is produced after a linear, pointwise transformation to predict the logits of the next element in the sequence [21]. Likewise the attention mechanism involves computing the dot products between all pairs of elements in the sequence, hence the computational complexity increases quadratically with the sequence length [21]. This makes Transformers unfit for modeling raw images as the sequential length directly correlates to the resolution. However, in smaller sequences (such as sequences of codebook vectors) this Transformer's ability to consider all interactions between elements is especially effective, as it can capture long-range dependencies [21] [26].

The Transformer architecture in this paper is that of GPT-2 [30], which is explained in further depth in 3.3. The use of this Transformer architecture is necessary as it uses no convolutional layers. This makes it very important in determining the impact of learning spatial dependencies between codebook vectors for anomaly detection; Unlike PixelSnail, the Transformer is not forced to learn local spatial dependencies as it works only through attention [23]. Hence if using the Transformer as the AR model consistently produces better results than PixelSnail, it could indicate that learning local spatial dependencies is potentially a limiting factor.

### 2.3 The PixelSnail model

PixelSnail [24] is an AR model that combines the self attention mechanism explained in 2.2 together with causal (masked) convolutions [25]. Causal convolution is a technique which applies convolutions over a masked sequence, such that only the previous elements influence the current prediction. Hence causal convolution is able to provide a high rate of access for earlier parts of the sequence [24]. However, since this is a convolutional technique it has a finite receptive field and thus struggles to capture dependencies between elements far away from each other [24]. Hence by combining this with self attention mechanisms, PixelSnail is able to overcome the finite receptive field and access a larger context.

As shown in Fig. 1, the causal convolution layers (which are also part of the residual blocks) aggregate information from the input to build a context to which the self-attention mechanism is applied [24]. Thus the attention mechanism is applied on top of the spatial information captured by the earlier convolutional layers, making it inherently different to Transformers where the attention mechanism is applied directly on the sequence. This makes sense working in a pixel space, as it is computationally expensive to pass a raw image directly to a Transformer. However, PixelSnail's effectiveness of modeling data which may not require such a strong emphasis on spatial dependencies (e.g. codebook vectors) is yet to be determined, making it relevant to compare against the Transformer as an AR model.

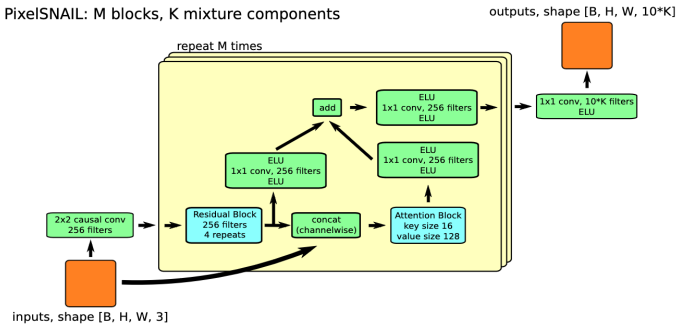



Fig. 1: High-level PixelSnail Architecture [24]

### 2.4 Anomaly Detection through Latent Space Restoration using Vector-Quantized Variational Autoencoders

The authors of this paper [5] propose an anomaly detection method that combines Density-Estimation [19] and Restoration [7] based approaches, based on the VQ-VAE model introduced in 2.1. They train the VQ-VAE to learn discrete latent representations of images, and model the prior distribution of latent codes using PixelSnail as the auto-regressive (AR) model.

The authors proposes Sample-wise detection, which utilizes the prior probabilities estimated by the AR model. The Sample-wise anomaly score indicates the likelihood of a given sample containing an anomaly. To find the Sample-wise anomaly score $AS_{sample}$ for a discrete latent encoding, the negative log likelihood (NLL) corresponding to the latent variable $z$ which are above a threshold $\lambda_S$ is summed:

$$AS_{sample} = \sum_{i}^{N} \xi(p(x_i)) \quad (9)$$

Where

$$\xi(z) \begin{cases} -log(z) \text{ if } -log(z) > \lambda_S \\ 0 \text{ otherwise} \end{cases} \quad (10)$$

The authors also propose Pixel-wise detection, which is a Restoration approach that quantifies the likelihoods of each pixel in the image containing an anomaly. This method replaces the aforementioned unlikely latent variables with new samples from the learnt prior AR model, effectively removing the potentially anomalous latent variables and restoring the final output generated by the decoder. Similar to Sample-wise detection, the negative log likelihood of each latent variable is thresholded via $\lambda_P$. A residual image is then computed by taking the absolute error between the original and restored image ($|Y - X|$).

In order to reduce the variance in the estimated anomaly score, multiple Restorations $j \in 1, 2, ...S$ are produced for each image, and hence multiple residual images are also

produced. These residual images are then each separately weighted by a factor $w_j$:

$$w_j = softmax(\frac{k}{\sum_i^P |Y^i - X_j^i|}) \tag{11}$$

Where $k$ is a softmax [32] temperature parameter and $P$ is the pixel space of the images. It can be seen that $w_j$ essentially reduces the weightings of image-restoration pairs that have low consistency in their residuals. The final Pixel-wise anomaly score $AS_{pixel}$ is calculated as the mean of all weighted residuals:

$$AS_{pixel} = \sum_j^S w_j |Y - X_j| \tag{12}$$

Note that the weighted residuals $w_j|Y - X_j|$ are smoothed via a 3x3 MinPooling and a 7x7 Average Pooling layer beforehand. While Pixel-wise detection is a strong method which is capable constructing high quality Restorations, it comes with severe computational costs, which is explained in further depth under 3.5 together with Light-Pixel-wise detection.

Sample-wise detection and Pixel-wise detection are extremely relevant for this paper, as they are the current state of the art Density-Estimation and Restoration based Vector-Quantized anomaly detection methods respectively [5].

### 2.5 Regeneration, Restoration, and Density-Estimation

For the purpose of comparing methods later on, it is important to fully define, and thus highlight the differences between each anomaly detection method.

Density-Estimation in Deep Learning refers to the process of training a neural network to predict the underlying probability density function for an observable dataset [19]. Hence Density-Estimation based anomaly detection uses the learnt probability density function to predict whether a given observation is unlikely (i.e. an anomaly) or not [18]. The aforementioned Sample-wise detection is a Density-Estimation method as anomaly scores are calculated based on the negative log likelihood of the codebook vectors for an image encoding.

Regeneration and Restoration methods can produce similar results, but have clear inherent differences. The former aims to train a generative model to learn the representation of normal samples, such that it fails to reconstruct anomalous samples [1] [7]. On the other hand, the latter aims train a network to remove anomalous elements from an input, thereby restoring the data into a form similar to normal samples [5]. In the context of image data, both methods typically use the residuals between the original anomalous input and the regeneration/restoration as the anomaly score. While both have different goals, they can produce similar outputs [7], as both methods exploit the absence of the anomaly in the output image.

## 3 METHODOLOGY

The methodology will fulfill the objectives listed in 1 by achieving the following specifications:

- The basic objective will be fulfilled by extending the VQ-VAE to the VQ-GAN; An adversarial training method as well as perceptual loss will be introduced in order to learn a higher quality, perceptually rich codebook. The performance of models utilizing the VQ-VAE and VQ-GAN will then be compared across all methods.
- The intermediate objective will be fulfilled by comparing the performance of PixelSnail and an attention based Transformer across all methods, thus determining the necessity for the AR model to learn local spatial dependencies.
- The advanced objective will be achieved by implementing Light-Pixel-wise detection, and demonstrating that it outpreforms Pixel-wise detection when used in conjunction with a perceptually rich codebook.

### 3.1 VQ-GAN

The VQ-GAN [21] is a variant of the aforementioned VQ-VAE. As the name suggests, the key difference is that this model introduces an adversarial training procedure with a Patch Discriminator [33] $D$, which differentiates between the original and reconstructions:

$$L_{GAN}(\{E, G, Z\}, D) = [logD(x) + log(1 - D(\hat{x}))] \tag{13}$$

Where $E$ is the encoder, $G$ is the decoder, and $Z = \{z_k\}_{k=1}^{h \times w} \in \mathbb{R}^C$ is the learned discrete codebook. The Patch discriminator works by penalizing local image patches; it will run convolutionally across image patches, and will classify if each patch of size $n \times n$ is real or fake. It will then average the classifications to produce the final output of $D$ [33].

Additionally, perceptual loss will be employed to achieve a perceptually rich codebook. Specifically, the learned perceptual image patch similarity [34] (Lpips) metric which is based on a pretrained VGG16 [35] network will be used. Lpips is a perceptual loss metric derived by evaluating feature distances in convolutional networks [34].

In order to compute the (perceptual) distance between two image patches $x$ and $x_0$ given some network $F$, feature stacks from $L$ different layers are extracted and unit-normalized in the channel dimension $C_l$. These feature stacks are denoted as $\hat{y}^l, \hat{y}_0^l \in \mathbb{R}^{H_l \times W_l \times C_l}$ respectively. These are then scaled in the channel dimension by a learned vector $w^l \in \mathbb{R}^{C_l}$, and the $L2$ loss is calculated. The results are then averaged spatially and across all layers. Hence to calculate $d(x, x_0)$:

$$d(x, x_0) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \left\| w_l \odot (\hat{y}^l - \hat{y}_0^l) \right\|_2^2 \tag{14}$$

Hence the final loss function for the VQ model is as follows:

$$L = L_r + L_p + L_{vq} + \beta L_c + \lambda L_{GAN}(\{E, G, Z\}, D) \tag{15}$$

where $L_p$ is the perceptual loss, and $\lambda$ is an adaptive weight computed as:

$$\lambda = \frac{\nabla_{G_L}[L_{rec}]}{\nabla_{G_L}[L_{GAN}] + \delta} \tag{16}$$

where $\bigtriangledown_{G_L}[L]$ denotes the gradient of input $L$ w.r.t the last layer of the decoder $G$, and $\delta = 10^{-6}$ is for numerical stability.

It can be seen from equation 15 above that the key difference between VQ-VAE and VQ-GAN is the training procedure (adversarial training and perceptual loss). Hence the architecture of the two models will be kept the same to ensure it does not have an impact on the experiments. Specifically, both will follow the original architecture from [21]. A high level overview of the Encoder and Decoder architecture for both models are shown in Table. 1 and Table. 2 below. The shape of the output for each component is in the form of $\mathbb{R}^{h \times w \times C}$, where $h$, $w$, and $C$ are the height, width, and channel dimensions respectively.

| Encoder |
|---|
| $x \rightarrow \mathbb{R}^{128 \times 128 \times 3}$ |
| $Conv2D \rightarrow \mathbb{R}^{128 \times 128 \times 64}$ |
| $\{Residual \times 2 + Downsample\} \rightarrow \mathbb{R}^{64 \times 64 \times 64}$ |
| $\{Residual \times 2 + Downsample\} \rightarrow \mathbb{R}^{32 \times 32 \times 64}$ |
| $\{Residual \times 2 + Downsample\} \rightarrow \mathbb{R}^{16 \times 16 \times 128}$ |
| $\{(Residual + NonLocal) \times 2 + Downsample\} \rightarrow \mathbb{R}^{8 \times 8 \times 128}$ |
| $\{Residual \times 2\} \rightarrow \mathbb{R}^{8 \times 8 \times 256}$ |
| $\{Residual + NonLocal + Residual\} \rightarrow \mathbb{R}^{8 \times 8 \times 256}$ |
| $\{GroupNorm, Swish, Conv2D\} \rightarrow \mathbb{R}^{8 \times 8 \times 256}$ |

TABLE 1: VQ model Encoder Architecture

| Decoder |
|---|
| $\hat{x} \rightarrow \mathbb{R}^{8 \times 8 \times 256}$ |
| $Conv2D \rightarrow \mathbb{R}^{8 \times 8 \times 256}$ |
| $\{Residual + NonLocal + Residual\} \rightarrow \mathbb{R}^{8 \times 8 \times 256}$ |
| $\{(Residual + NonLocal) \times 3\} \rightarrow \mathbb{R}^{8 \times 8 \times 256}$ |
| $\{(Residual + NonLocal) \times 3 + Upsample\} \rightarrow \mathbb{R}^{16 \times 16 \times 128}$ |
| $\{Residual \times 3 + Upsample\} \rightarrow \mathbb{R}^{32 \times 32 \times 128}$ |
| $\{Residual \times 3 + Upsample\} \rightarrow \mathbb{R}^{64 \times 64 \times 64}$ |
| $\{Residual \times 3 + Upsample\} \rightarrow \mathbb{R}^{128 \times 128 \times 64}$ |
| $\{GroupNorm, Swish, Conv2D\} \rightarrow \mathbb{R}^{128 \times 128 \times 3}$ |

TABLE 2: VQ model Decoder Architecture

### 3.2 Auto-Regressive models in a Latent Space

This paper will compare the performance of two different auto-regressive models to estimate the prior probability; PixelSnail and the GPT-2 Transformer. Both will learn the composition of codebook vectors of an image, by maximising the log-likelihood of the full latent representation.

Given the encoder $E$, decoder $G$ and codebook layer $VQ$ of the VQ model, a Vector-Quantized encoding of an image $x$ can be represented as:

$$z_q = VQ(E(x)) \in \mathbb{R}^{h \times w \times C} \tag{17}$$

This is equivalent to the sequence of codebook indices $s \in \{0, ...|Z| - 1\}^{h \times w}$ where each index maps to a vector in the codebook $Z$:

$$s_{ij} = k \text{ such that } (z_q)_{ij} = z_k \tag{18}$$

The image encoded to $s$ can be recovered by mapping the indices back to their codebook vectors $z_q = (z_{s_{ij}})$ and decoded via $\hat{x} = G(z_q)$. Hence, the AR model can be trained to learn the latent prior of normal images via Auto-Regressive next-index prediction. Given indices $s_{<i} = \{0, 1, ...i - 1\}$, the AR model will predict $p(s_i|s_{<i})$ (i.e. distribution of next possible indices given $s_{<i}$). This will in turn allow it to compute the likelihood of the full latent representation $p(s) = \prod_i p(s_i|s_{<i})$. By maximizing the log-likelihood of these representations during training, the AR model will learn to model the latent prior of normal images:

$$L_{AR} = \mathbb{E}_{x \sim p(x)}[-logp(s)] \tag{19}$$

During inference, negative log likelihood of each individual codebook index will be used to detect the unlikely latent variables.

### 3.3 Transformer Architecture

For this paper, gpt2-medium [30] was used as the Transformer architecture, which is the 335M parameter version of GPT-2 [30]. Gpt2-medium contains 24 layers of the Transformer block shown below. Each block has 16 independent attention mechanisms (i.e. heads)

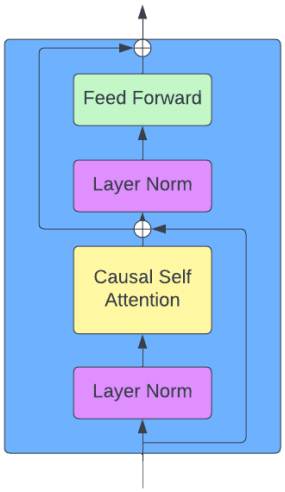



Fig. 2: High level overview of Transformer Block

Layer normalization [36] normalizes each sample independently across all features and is not influenced by other samples in the batch. This makes it useful when working with small batch sizes, including the batch size of 10 which was used in this paper.

In causal attention mechanisms [37], the queries are limited to their current position as well as the positions of preceding key-value pairs; given index $i$, all attention values after $i$ are masked such that $A_{>i} = -\infty$.

Lastly, the feed-forward block simply consists of two linear layers, a GELU [38] activation function, and a dropout layer [39] with dropout probability of 0.1.

### 3.4 PixelSnail Architecture

Following the original design in [24], the PixelSnail implementation in this paper contains 4 layers of the pixel block shown below in Fig. 3:

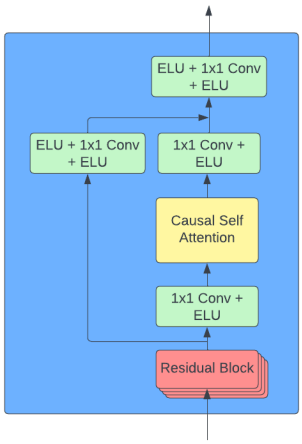



Fig. 3: High-level overview of Pixel Block

The pixel block contains 4 residual blocks, the causal attention [37] mechanism, as well as causal convolution layers and ELU [40] activation functions. Note that the architecture for residual blocks in PixelSnail is different to the residual blocks used in the VQ-VAE and VQ-GAN models, in that it uses ELU [40] activation functions as opposed to SWISH [41], and also contains a dropout layer [39] with dropout probability of 0.1.

### 3.5 Light Pixel-wise detection with improved codebooks

While the Pixel-wise detection method is a powerful approach as it enables the Restoration of the latent space and subsequently the decoded image [5], it suffers from high computational costs. This is because it recomputes all of the log likelihoods (i.e. runs through the forward pass of the AR model) after each Restoration of a latent variable. Hence the worst-case running time of Pixel-wise detection which creates $N$ Restorations is $O(H^2 \times W^2 \times N)$, where $H$ and $W$ are the height and width of the latent space respectively. This is because to compute the log likelihoods once, the AR model will need to process the sequence of codebook vectors which is of length $H \times W$. And since it recomputes the new log likelihoods after each time it restores a latent variable, in the worst case where it restores all of them, it would need to run through the sequence $H \times W$ times to create one Restoration (hence to create $N$ Restorations would be $O(H^2 \times W^2 \times N)$).

To solve this issue, this paper proposes Light Pixel-wise detection, which only computes the log likelihoods once per image restoration in the beginning. The latent variables where the negative log-likelihoods exceed $\lambda_{LP}$ are then replaced by samples from the learnt latent prior. This makes the process of detecting unlikely latent variables for Light Pixel-wise detection closer to Sample-wise detection, since the prior probability estimates do not change with each restored variable.

This also however, means that there will be less variance in the restorations compared to the original Pixel-wise method. This can significantly impact anomaly detection performance as Pixel-wise detection relies on the variance of restorations. An ideal Restoration method would only need to output a single, perfect restoration where only the anomalous element of an image is removed. However this is not always the case, most restorations have some form of error, such as a the normal parts of the image accidentally being restored, or the anomaly not being removed completely [5]. Pixel-wise detection aims to overcome this problem by creating a large number of varied restorations and taking the average of the residuals, such that the significance of each error is reduced. This approach would not work with Light-Pixel-wise detection, due to the aforementioned lack of variation between restorations. Hence Light-Pixel-wise detection is designed to work in conjunction with the improved, perceptually rich codebook. Previous studies found that the improved codebook of the VQ-GAN produced significantly better results than the VQ-VAE for image generation tasks [21] [26]. This indicates that that a perceptually rich codebook allows for more high quality, detailed image generation and thus should also lead to high quality image restorations as well. Higher quality restorations will have fewer errors, and hence the need for numerous variations lessens. This is the fundamental approach of the proposed Light-Pixel-wise detection; to leverage the improved image sampling quality of the perceptually rich codebook to achieve anomaly detection with fewer restorations.

### 3.6 Testing and Experimentation

This paper will compare the anomaly detection performances of the following model combinations:

- VQ-VAE + PixelSnail (VP)
- VQ-VAE + Transformer (VT)
- VQ-GAN + PixelSnail (GP)
- VQ-GAN + Transformer (GT)

Each of these models will be tested on the following anomaly detection methods:

- Sample-wise detection
- Pixel-wise detection
- Light Pixel-wise detection
- Regenerative detection

This is in order to measure how much of an impact the updated architectures have on performance, as well as to compare the effectiveness of the individual methods. Sample-wise detection and Pixel-wise detection will follow the same procedure outlined in 2.4. A common implementation issue with Pixel-wise detection was that, despite limiting the number of Restorations to 5, it was still prone to out-of-memory errors depending on the parameters of the model. Hence when running each model on the validation and testing dataset, results were written into a text file in real time, such that if an out-of-memory error did occur, it would be able to save its progress and continue where it left off.

The Regenerative approach will utilize the VQ model and take the $L1$ loss between an image $x$ and its reconstruction $\hat{x}$, similar to [1]. Note that, since the Regenerative method does not use an auto regressive model, experiments will be conducted for the VQ-VAE and VQ-GAN models only. The best scores obtained by each model will be compared against other state-of-the-art models; Ganomaly [1] and Deep Feature Kernel Density Estimation [42] (DFKDE).

Ganomaly is a semi-supervised anomaly detection model which uses a technique similar to Regenerative detection. It uses a generator which maps an input image in to a lower dimensional vector and reconstructs this using with its decoder component. It also has an additional encoder which creates a latent vector representation of the generated image. The model then minimizes the distance between the original and reconstructed image, as well as the two latent vectors and learns the representation of normal images. On the other hand, DFKDE an anomaly detection method which uses deep feature extraction. It initially feeds an image through a ResNet50 [43] backbone pretrained on the ImageNet [44] dataset. Features from the penultimate layer are then reduced to 16 components via Principal Component Analysis [45], and Gaussian Kernel Density [46] is then used to estimate the probability density based on the features obtained from the training set. These two models were chosen to be compared as they both use similar anomaly detection approaches to the methods introduced in this paper; Pixel level residuals are used to calculate the anomaly score in Ganomaly (similar to Pixel-wise detection), and DFKDE is also a Density-Estimation method (similar to Sample-wise detection).

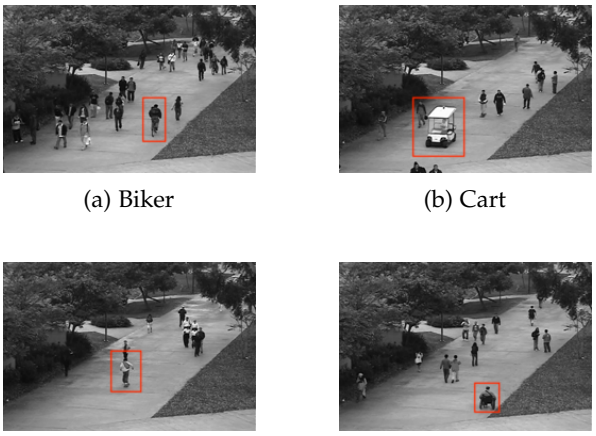
### 3.7 Tools used

The implementations for Ganomaly and DFKDE were obtained from Anomalib [47], which is an open-source benchmarking library for anomaly detection. Additionally, the implementations for all experiments and models in this paper were written in PyTorch [48], which is an open-source machine learning framework for Python based on Torch [49]. Lastly, all experiments were conducted using Durham University's NCC cluster, which contains 48 GPUs across 9 servers.

### 3.8 Datasets

The experiments in this paper were conducted on the UCSD Ped2 Dataset [11] (shortened to UCSD Dataset for the sake of redundancy), released by the University of California San Diego. The UCSD Dataset is a commonly used anomaly detection dataset [9] [50] [51] with images taken from a stationary camera overlooking a pedestrian walkway. Normal images consist of only people walking along this walkway, and anomalous images include non pedestrian objects. Such examples of anomalous images are shown in Fig. 4.

Moreover, since this paper explores unsupervised anomaly detection methods, the training dataset is made up of only 80% of normal images. 10% of normal images as well as 90% of anomalous images will comprise of the test dataset, and the remaining 10% of normal images and 10% of anomalous images will represent the validation dataset. All images were resized to dimensions of 128x128 and a batch size of 10 was used.

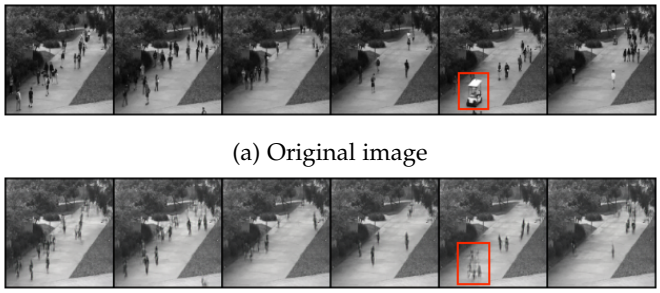


(a) Biker (b) Cart

(c) Skater (d) Wheelchair

Fig. 4: Anomaly examples in the UCSD dataset

### 3.9 Parameter tuning and Validation



(a) Original image

(b) VQ-GAN reconstruction with latent channel dimension of 2048. Red region indicates reconstructed anomaly.

Fig. 5: Example of unseen Anomaly being reconstructed

This section demonstrates the necessity of tuning the channel dimension of the latent space as well as the sizes of the vector codebook. [8] found that VAEs with a large number of channels were capable of reconstructing unseen anomalies, making them ineffective to use for Regenerative detection. A similar phenomenon can be observed here as well; Fig. 5 shows 6 pairs of anomalous images and their reconstructions from a VQ-GAN trained in a channel dimension of 2048 and with a codebook vector size of 1048. It can be seen that the VQ-GAN is able to reconstruct blurry yet similar shapes of the unseen anomalies. This is especially noticeable for larger objects such as the cart, as shown in the red bounding box. The likely reason for this phenomenon is that with more channels, the model is able to capture more complex features learned by its deeper layers thus learning more varied codebook vectors, which it can remember with a large enough codebook size. While in an image modeling scenario this would be beneficial, due to the majority of images in anomaly detection datasets (such as UCSD) being visually similar, they don't normally require a large number of channels or a large codebook for the model to learn an accurate representation. Thus increasing the two parameters results in the model having such a large, varied codebook that it can reconstruct unseen anomalies.

Furthermore, this can impact the model's anomaly detection capabilities as well. Table. 3 shows the Sample-wise detection scores for latent channel dimensions 256, 512, 1024, and 2048 for the GAN+Transformer model on the validation dataset. It can be seen that performance worsens as the channel dimension increases. The same can also be said for Table. 4, which shows the Sample-wise detection scores for codebook sizes 128, 512, 1048 and 2048.

| Dimension | F1 | AUC |
|---|---|---|
| 256 | 0.995 | 0.999 |
| 512 | 0.995 | 0.998 |
| 1024 | 0.992 | 0.993 |
| 2048 | 0.985 | 0.989 |

TABLE 3: Sample-wise detection performance for different latent channel dimensions of GAN+Transformer

| Codebook size | F1 | AUC |
|---|---|---|
| 128 | 0.995 | 0.999 |
| 512 | 0.993 | 0.996 |
| 1024 | 0.992 | 0.994 |
| 2048 | 0.986 | 0.997 |

TABLE 4: Sample-wise detection performance for different codebook sizes of GAN+Transformer

Intentionally limiting the extent to which the model learns the representation of images can be seen to improve anomaly detection performance, hence the 4 models VP, VT, GP and GT will all have channel dimensions of 256 as well as a codebook size of 128.

## 4 Results

### 4.1 Implementation details

The lpips metric used in this paper is based on a pre-trained VGG16 [35] network. This is because many literatures consider the VGG16 network to be the de facto standard for transfer learning in computer vision tasks [34], and it was also shown that the VGG16 based lpips metric outpreformed its SqueezeNet [52] and Alexnet [53] counterparts [34]. The PixelSnail architecture will follow the design described in its original paper. Similar to [5], a small latent space of size 8x8 was found to have the best trade-off between performance and computational complexity.

The learning rates of the VQ-VAE, Transformer, and PixelSnail model were set to 2.25e-5, 2.25e-5, and 1e-4 respectively. The VQ-GAN was found to work better with a higher learning rate of 5e-5, likely due to the more informative loss function. The channel dimensions of the latent space and codebook vector size were set to 256 and 128 respectively, as they were found to produce the best results, as demonstrated in 4.2. $\lambda_S$, $\lambda_P$, and $\lambda_{LP}$ were tuned separately for each model. All other parameters for each model were set to the values used in their respective papers [5] [21].

The models were not trained for a set number of epochs, and instead the training process utilizes early stopping [54]. This is to prevent the neural networks from overfitting to the training data. After every other epoch during training, the model calculates the average training loss. The model is then tested against the validation dataset and calculates the average validation loss. If the average validation loss is seen to be larger than the average training loss for more than 5 times, training is terminated.

### 4.2 Evaluation methods

The performance of the anomaly detection methods will be measured via the F1 score [55] and AUC score [56]. The F1 score is the harmonic mean of precision and recall; i.e. it measures the relative impact of both false positives and false negative errors. It was chosen over a simple accuracy metric as there is a significant class imbalance in the test dataset. The AUC score was also chosen as an additional metric to measure how well the model is capable of distinguishing between normal and anomalous samples; The ROC (receiver operating characteristics curve) is a probability curve created by plotting the true positive rate against the false positive rate, and the AUC represents the degree of separability between the two.

### 4.3 Experimental results

| Model | F1 | AUC |
|---|---|---|
| VAE+PixelSnail | 0.881 | 0.942 |
| VAE+Transformer | 0.917 | 0.952 |
| GAN+PixelSnail | 0.993 | 0.997 |
| GAN+Transformer | 0.985 | 0.994 |
| Ganomaly | 0.986 | 0.992 |
| DFKDE | 0.962 | 0.969 |

TABLE 5: Comparison of Best F1 and AUC scores against state of the art

Table. 5 shows the best scores produced by each model compared against the state of the art.

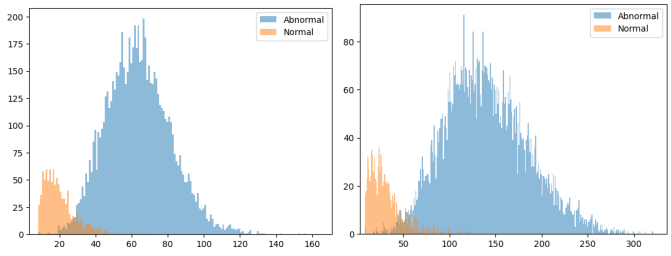
| Model | F1 | AUC | $\lambda_S$ |
|---|---|---|---|
| VAE+PixelSnail | 0.881 | 0.942 | 0.216 |
| VAE+Transformer | 0.917 | 0.952 | 0.0162 |
| GAN+PixelSnail | 0.993 | 0.997 | 1.43 |
| GAN+Transformer | 0.985 | 0.994 | 2.27 |

TABLE 6: Comparison of F1 and AUC scores for Sample-wise detection

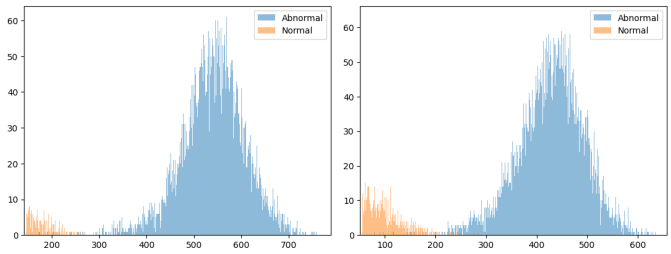
Table. 6 shows the performances of the 4 model combinations for Sample-wise anomaly detection. The listed $\lambda_S$ thresholds correspond to the 60th, 40th, 60th and 40th percentiles respectively. Fig. 6 shows the distribution of anomaly scores for Sample-wise detection.

The parameter tuning results for the $\lambda_S$ values can be found in the appendix. This is also true for the $\lambda_P$ and $\lambda_{LP}$ values shown later on as well.

Table. 7 shows the performances of the 4 models for Pixel-wise anomaly detection. The listed $\lambda_P$ thresholds correspond to the 90th, 80th, 90th and 90th percentiles respectively. Fig. 7 shows the distribution of anomaly scores for Pixel-wise detection.




(a) VAE+PixelSnail (b) VAE+Transformer

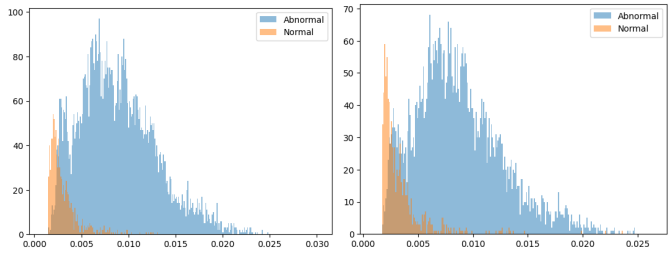



(c) GAN+PixelSnail (d) GAN+Transformer

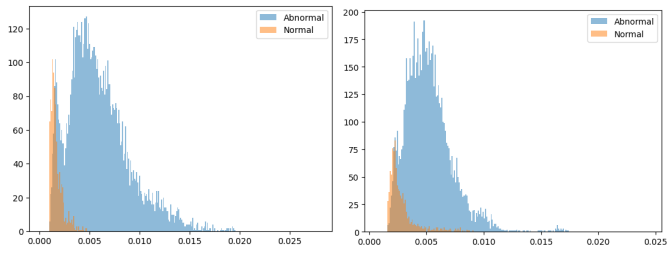
Fig. 6: Anomaly score distribution for Sample-wise Detection

| Model | F1 | AUC | $\lambda_P$ |
|---|---|---|---|
| VAE+PixelSnail | 0.752 | 0.874 | 6.01 |
| VAE+Transformer | 0.794 | 0.866 | 6.32 |
| GAN+PixelSnail | 0.790 | 0.902 | 10.7 |
| GAN+Transformer | 0.735 | 0.832 | 12.6 |

TABLE 7: Comparison of F1 and AUC scores for Pixel-wise detection




(a) VAE+PixelSnail (b) VAE+Transformer



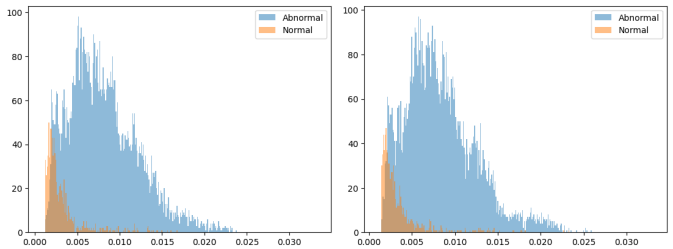

(c) GAN+PixelSnail (d) GAN+Transformer

Fig. 7: Anomaly score distribution for Pixel-wise Detection

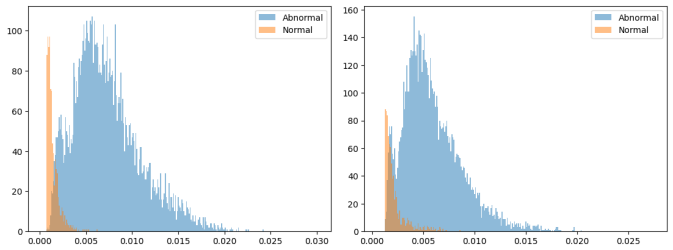
Due to computational limitations, the number of Restorations per image was set to 5 instead of the recommended 15 in the original paper. [5] found that more Restorations can improve performance as there is a larger number of varied residuals to which the anomaly score can be calculated. Hence limiting this number is also expected to limit performance, which will be discussed later.

| Model | F1 | AUC | $\lambda_{LP}$ |
|---|---|---|---|
| VAE+PixelSnail | 0.718 | 0.842 | 8.08 |
| VAE+Transformer | 0.716 | 0.837 | 3.29 |
| GAN+PixelSnail | 0.848 | 0.938 | 7.23 |
| GAN+Transformer | 0.789 | 0.875 | 11.2 |

TABLE 8: Comparison of F1 and AUC scores for Light-Pixel-wise detection




(a) VAE+PixelSnail (b) VAE+Transformer




(c) GAN+PixelSnail (d) GAN+Transformer

Fig. 8: Anomaly score distribution for Light Pixel-wise Detection

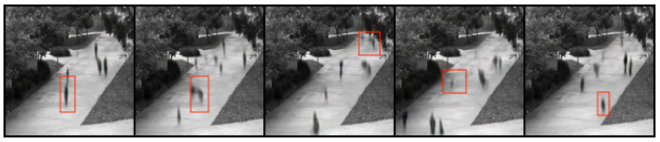
Table 8 shows the performances of the 4 models for Light Pixel-wise anomaly detection. The $\lambda_{LP}$ thresholds correspond to the 80th, 40th, 60th and 80th percentiles respectively. Likewise Fig. 8 shows the distribution of anomaly scores for Light-Pixel-wise detection. To fairly compare the performance against Pixel-wise detection, the number of restorations was also set to 5.

| Model | F1 | AUC |
|---|---|---|
| VQ-VAE | 0.796 | 0.902 |
| VQ-GAN | 0.715 | 0.866 |

TABLE 9: Comparison of F1 and AUC scores for Regenerative detection

Lastly, Table. 9 shows the performance of the Regenerative approach for the VQ models, and Fig. 9 displays the Regeneration results for anomalous images.

(a) Original image



(b) VQ-VAE Regeneration

(c) VQ-GAN Regeneration

Fig. 9: Anomalous Regeneration results for VQ-VAE and VQ-GAN

(a) Original Image (0 latent variables restored)

(b) 10 latent variables restored

(c) 20 latent variables restored

(d) 30 latent variables restored

Fig. 10: Change in Pixel-wise Restoration results for GAN+Transformer model

## 5 Evaluation

GP and GT both performed exceptionally well, achieving a higher AUC score via Sample-wise detection than all other models including state of the art, as shown in Table. 5. This indicates that a more representative, perceptually rich codebook is highly effective when using Vector-Quantized models for anomaly detection.

### 5.1 Comparison of Methods

Sample-wise detection significantly outperformed the other Restoration and Regenerative methods. Pixel-wise detection preformed the worst, having lower results than the base line Regenerative method. This suggests that Restoration methods are not fit for this task.

This could be because the VQ models are shown to be capable of restoring anomalies themselves in the initial regeneration phase. Fig. 9 shows the regeneration results of the VQ-VAE and VQ-GAN. Despite none of the unlikely latent variables being replaced yet, the anomalies appear to be restored in the reconstructions made by the VQ-GAN; the skater and biker are replaced by a person walking, and the two carts as well as the wheelchair are replaced by human-like silhouettes. This shows that the increased representational power due to the perceptually rich codebook allows the VQ-GAN to restore anomalies itself. This makes sense as VQ-Models are trained such that the discrete latent variables are computed by taking the nearest neighbour codebook vectors (refer to eq 2), thus when it encounters unknown features (i.e. features corresponding to bikes, skaters, etc) it will replace them by their nearest normal codebook vectors, hence the anomalous object will likely no longer appear in the decoded output.

Likewise, applying an additional restorative process on top of this would result in normal regions of the images being changed. This can be shown in Fig. 10, which shows the decoded outputs between 10 and 30 latent variable restorations in increments of 10. It can be seen that since the anomalous object is already non-existent, restoring the latent variables result in normal regions of the images being changed, ultimately resulting in "restorations" which are essentially completely new images. One potential way of fixing this is limiting the representational power of the VQ-Model even further. Both the channel dimension as well as the codebook size were set to be relatively small in 3.9 (256 and 128 respectively) with regards to Sample-wise detection results, but reducing them even further (particularly the size of the codebook to limit the number of available varied codebook vectors) will make it more difficult for the model to restore the anomalous regions. However, whether this is actually necessary is questionable. As explained in 2.5, Restoration methods and Regeneration based methods produce very similar outputs but in different ways, hence if one succeeds at effectively removing the anomaly, there is no reason to restrict it to force the other method.

The VQ regenerations being visually restored does not seem to impact the existence of unlikely latent variables. As shown in Fig. 6, many unlikely latent variables still exist within the latent representation which all have high loss and hence are detected via density-estimation. This indicates that the anomalous images lead to the presence of unlikely codebook vectors, but they do not necessarily lead to anomalous regions in the reconstructions. This makes sense as codebook vectors do not correspond to specific objects nor regions in the image; they are a more complex, abstract representation of the image features. On the other hand, this could also raise the possibility of the AR model overfitting to the training data; if the initial encoding produced by the VQ-

Model results in a visually restored image, then intuitively the AR model shouldn't detect any unlikely latent variables. However, observing Fig. 6 shows this is not the case, as the AR model is able to clearly distinguish between the normal and anomalous images in the test dataset. Hence this shows that the initial "restoration" process by the VQ model's regeneration (i.e. vector encoding) phase results in an unusual combination of codebook vectors, however since these codebook vectors are learned from a normal representation of images, the resulting regeneration can still be visually restored.

This reveals the inherent weakness of Restoration methods that makes Density-Estimation methods superior; the latter assumes that unlikely latent variables will always correspond to anomalous regions in the image, but this is not true as demonstrated above. On the other hand, Density-Estimation methods (i.e. Sample-wise detection) does not have this issue, as the anomaly scores are calculated purely based on the likelihood of codebook indices, and hence whether or not the resulting decoded output is visually restored has no effect.

(a) Original image

(b) Restoration

Fig. 11: Anomalous Restoration via Light-Pixel-wise detection for GAN+Transformer

Furthermore, while it was expected that Light-Pixel-wise detection would produce Restorations with significantly lower variance than the Pixel-wise detection method, this does not seem to be the case. Observing the results in Fig. 10 and Fig. 11, which displays the Restorations made by the GT model using Pixel-wise and Light-Pixel-wise detection respectively, show that the majority of Restorations look very similar. Likewise, the anomaly score distributions are very similar between the two methods. This may indicate that contrary to what was expected, recomputing the log likelihoods after each restored latent variable does not significantly impact the variance in the final output.

Note that the Pixel-wise and Light-Pixel-wise restorations results from the other models (VP, VT, GP) were not included in the paper due to limited space, but are included in the appendix.

It can be seen that Light-Pixel-wise detection is significantly faster than Pixel-wise detection. Table. 10 shows the number of seconds each method took to process an image for the GT model. it can be seen that Light-Pixel-wise detection was able to process in image 67 times faster than Pixel-wise detection. The run times for VP, VT and GP are also listed in the appendix.

| Method | sec/image |
|---|---|
| Sample-wise | 0.04 |
| Pixel-wise | 8.1 |
| Light Pixel-wise | 0.12 |
| Regenerative | 0.02 |

TABLE 10: Comparison of Running times (seconds per image) for anomaly detection methods on the GAN+Transformer

| Model | F1 | AUC |
|---|---|---|
| VP (Pixel-wise) | 0.752 | 0.874 |
| VT (Pixel-wise) | 0.794 | 0.866 |
| VP (Light-Pixel-wise) | 0.718 | 0.842 |
| VT (Light-Pixel-wise) | 0.716 | 0.837 |

TABLE 11: Comparison of Light-Pixel-wise detection and Pixel-wise detection performance for VP and VT

| Model | F1 | AUC |
|---|---|---|
| GP (Pixel-wise) | 0.790 | 0.902 |
| GT (Pixel-wise) | 0.735 | 0.832 |
| GP (Light-Pixel-wise) | 0.848 | 0.938 |
| GT (Light-Pixel-wise) | 0.789 | 0.875 |

TABLE 12: Comparison of Light-Pixel-wise detection and Pixel-wise detection performance for GP and GT

Additionally, much of what was discussed in 3.5 can be seen in the results; the models which utilized the VQ-VAE (VP and VT) preformed worse with Light-Pixel-wise detection than Pixel-wise detection, while those that used the VQ-GAN (GP and GT) had the opposite results. This indicates that the perceptually rich codebook to increase representational power is indeed crucial for Light-Pixel-wise detection, as it is sacrificing variance among its restorations.

### 5.2 Comparison of Models

For Sample-wise detection, utilizing a Transformer as the AR model performed better than PixelSnail for the VQ-VAE models (VP and VT), while the opposite outcome was observed for the VQ-GAN models (GP and GT). This suggests that a perceptually rich codebook induces local spatial dependencies between codebook vectors, while a more simpler codebook learnt without any adversarial training or perceptual loss does not.

This is not necessarily the case for Restoration methods. The results for Pixel-wise detection show that both models using PixelSnail (VP and GP) outperformed their Transformer counterparts. Additionally, the worst performing model was GT, having the lowest F1 score as well as the lowest AUC score. This shows that Density-Estimation methods (Sample-wise detection) and Restoration methods (Pixel-wise detection) require different architectures in order for each to work optimally. This may also explain why the VQ-GAN models (GP and GT) do the best in Light-Pixel-wise detection. Since Light-Pixel-wise detection determines the unlikely latent variables with the initially computed log likelihoods, it is much more similar to Sample-wise detection than Pixel-wise detection despite being a Restoration

method. This is because Pixel-wise detection recomputes the log likelihoods after each restored variable, thus the Restoration is influencing the prior probability estimates made by the AR model, while this is obviously not the case for the other two methods.

However, despite this the models utilizing a Transformer (VT and GT) were outperformed by their PixelSnail counterparts in both Pixel-wise and Light Pixel-wise detection, which was not the case for Sample-wise detection. This is important as it could indicate that, regardless of the quality of the learned codebook and regardless of the way the prior probability is estimated by the AR model, local spatial dependencies indeed exist between the codebook vectors, but whether the model needs to learn it depends on whether the anomaly score is calculated in the pixel space or in the abstracted latent space. This is because despite the latent variables being detected in the same manner as Sample-wise detection, local spatial dependencies were shown to be important regardless of the quality of the vector codebook (VP and GP outperformed VT and GT) for Light-Pixel-wise detection, which was also the same for Pixel-wise detection. This shows that local spatial dependencies which are dominant in the pixel-space, are also important for the model to learn in a discrete latent space if the goal of the task would be to output an image, such as with Restoration methods. Furthermore, this indicates that the works of Chen *et al.* [17], whom modeled the anomalies as spatially local deviants in the pixel space could also translate in to a discrete latent space as well.

### 5.3 Limitations to Methodology

Although the results indicated that Restoration methods were not suited for Vector-Quantized anomaly detection tasks, it is also important to consider the limitations in the methodology which could have impacted results.

Firstly, the channel dimensions and codebook size were set in accordance to Sample-wise detection performance. Hence it is necessary to confirm the best combinations of parameters separately for Pixel-wise and Light-Pixel-wise detection as well. Furthermore, the phenomenon discussed in 5.1 where the anomalies are restored during the inital encoding of the VQ-Model could depend on the datasets; For example, it could be that for datasets where the anomalies are more larger and complex, the VQ-Model would struggle to restore them by itself, making the Restoration methods more appropriate. Hence experiments on a wider variety of datasets is imperative, especially experimentation with medical images where Restoration methods have been shown to preform well [5].

Another limitation lies in the fundamental architecture of the AR models. PixelSnail and the gpt2-medium Transformer both use causal convolution [25] and causal attention [37], which prohibits the AR model from looking forward into the sequence from a given element. While this is justified for generative tasks, where the model doesn't have access to future elements in the sequence during inference as they are not generated yet, this is not necessarily the case for anomaly detection. In both the Density-Estimation and Restoration methods, nothing is being newly generated (only detected or restored), and hence the AR model does have access to future elements of the sequence. Using causal techniques only hides this information which the AR model otherwise would have had access to, and hence could have utilized to make better estimations. Removing the causal design such that both AR models constantly have access to all elements could potentially improve its ability of estimating the prior probability, and consequently improve anomaly detection performance.

## 6 Conclusion

This paper sought to answer the research question:

> "How does learning local spatial dependencies between codebook vectors, as well an adversarially trained perceptually rich codebook in Vector-Quantized models impact the performance of Density-Estimation and Restoration based anomaly detection in images?"

This topic question was important to explore as the necessity of learning local spatial dependencies can help determine the best AR model architecture for both Density-Estimation and Restoration methods. Additionally, it also determined if the introduction of perceptual quality in the learned codebook can lead to improvements in performance. Since Vector-Quantized anomaly detection remains a relatively new area, this information will be valuable to use as reference for future research.

Through the conducted experiments it was found that local spatial dependencies are a necessity for the model to learn for Restoration based anomaly detection, but is not always the case for Density-Estimation methods. More specifically for Sample-wise detection, without a perceptually rich codebook it was shown that using a solely attention-based Transformer produced better results. Likewise, a perceptually rich codebook improved performance for both methods, regardless of the AR model. From this it can be concluded that both Density-Estimation methods as well as Restoration methods rely heavily on the quality of the learnt latent codebook, and the latter also relies on the local spatial dependencies between the individual codebook vectors and thus requires an AR model which can capture them such as PixelSnail. Sample-wise detection was shown to produce very good results, showing the best results compared to the other methods. This was not the case for Pixel-wise detection, as the experiments found that unlikely latent variables do not necessarily correspond to anomalous regions in the image.

The proposed Light-Pixel-wise detection was shown to produce adequate results relative to Pixel-wise detection but in a fraction of the time, however to fully determine its effectiveness it must also be tested on a wider variety of datasets.

### 6.1 Future works

There are many ways to extend the works of this paper. One is to change the way the AR model estimates the prior probability during inference. Currently both AR models (and subsequently both the Density-Estimation and Restoration methods) use causal blocks, which, as discussed in 5.3, may be a limiting factor for anomaly detection.

Instead of masking all elements after a given index, a better method may be to mask out all unusual elements, such that the log likelihoods are computed using only normal elements. Specifically, given a sequence of codebook indices $s \in \{0, ...|Z| - 1\}^{h \times w}$ the AR model will compute the initial log likelihoods similar to Sample-wise detection to determine the set of unusual indices $u \subset s$, and consequently the set of normal indices $\overline{u} \subset s$, such that $\overline{u} \cap u = \varnothing$. Hence, instead of computing the likelihood of the full representation in an Auto-Regressive manner via $p(s) = \prod_i p(s_i|s_{<i})$, it computes it using only the indices in $\overline{u}$:

$$p(s) = \prod_i p(s_i|\overline{u}) \tag{20}$$

This could produce better Density-Estimations as it is not only no longer restricted by the causal layers, but abnormal latent variables will no longer influence the AR model's estimated prior probability. Also, unlike Pixel-wise detection it is not computationally expensive as it would only require to run through the forward pass of the AR model twice (once for the initial estimation to mask out the unusual variables, and the second for the actual estimation).

Another extension is to apply the Restoration methods for anomaly segmentation [57]. Anomaly segmentation (or anomaly localization) is simply the process to identify where the anomalous entity is located in an image, commonly through heatmaps or bounding boxes. Since Restoration methods produce multiple Restorations of an anomalous image, observing the areas with the highest residuals potentially is a simple method to achieve anomaly localization.

Lastly, another extension of this paper would be a more thorough study focusing only on Restoration based methods (i.e. Pixel-wise detection). This would include conducting experiments to see if increasing the latent space or the codebook size does indeed improve performance, or to determine how sensitive the number of Restorations are (i.e. to see if more Restorations actually result in more accurate classifications), or also to see how the performance changes with different datasets (e.g. medical images which tend to have higher variance).

## REFERENCES


[1] Akcay, S., Atapour-Abarghouei, A. & Breckon, T. GANomaly: Semi-Supervised Anomaly Detection via Adversarial Training. *ArXiv:1805.06725 [cs]*. (2018,11), https://arxiv.org/abs/1805.06725

[2] Chalapathy, R. & Chawla, S. Deep Learning for Anomaly Detection: A Survey. *ArXiv:1901.03407 [cs, Stat]*. (2019,1), https://arxiv.org/abs/1901.03407

[3] Liu, J., Xie, G., Wang, J., Li, S., Wang, C., Zheng, F. & Jin, Y. Deep Industrial Image Anomaly Detection: A Survey. *ArXiv:2301.11514 [cs]*. (2023,1), https://arxiv.org/abs/2301.11514

[4] Crépey, S., Noureddine, L., Madhar, N. & Thomas, M. Anomaly Detection on Financial Time Series by Principal Component Analysis and Neural Networks. *ArXiv:2209.11686 [math, Q-fin, Stat]*. (2022,10), https://arxiv.org/abs/2209.11686

[5] Marimont, S. & Tarroni, G. Anomaly detection through latent space restoration using vector-quantized variational autoencoders. *ArXiv:2012.06765 [cs, Eess]*. (2020,12), https://arxiv.org/abs/2012.06765

[6] Fernando, T., Gammulle, H., Denman, S., Sridharan, S. & Fookes, C. Deep Learning for Medical Anomaly Detection – A Survey. *ArXiv:2012.02364 [cs, Eess, Stat]*. (2021,4), https://arxiv.org/abs/2012.02364

[7] Baur, C., Denner, S., Wiestler, B., Albarqouni, S. & Navab, N. Autoencoders for Unsupervised Anomaly Segmentation in Brain MR Images: A Comparative Study. *ArXiv:2004.03271 [cs, Eess]*. (2020,4), https://arxiv.org/abs/2004.03271

[8] Zimmerer, D., Isensee, F., Petersen, J., Kohl, S. & Maier-Hein, K. Unsupervised Anomaly Localization using Variational Auto-Encoders. *ArXiv:1907.02796 [cs, Eess, Stat]*. (2019,7), https://arxiv.org/abs/1907.02796

[9] Abati, D., Porrello, A., Calderara, S. & Cucchiara, R. Latent Space Autoregression for Novelty Detection. *ArXiv:1807.01653 [cs]*. (2019,3), https://arxiv.org/abs/1807.01653

[10] Deshpande, K., Punn, N., Sonbhadra, S. & Agarwal, S. Anomaly detection in surveillance videos using transformer based attention model. *ArXiv:2206.01524 [cs]*. (2022,6), https://arxiv.org/abs/2206.01524

[11] Zhang, X., Yang, S., Zhang, X., Zhang, W. & Zhang, J. Anomaly Detection and Localization in Crowded Scenes by Motion-field Shape Description and Similarity-based Statistical Learning. *ArXiv:1805.10620 [cs]*. (2018,5), https://arxiv.org/abs/1805.10620

[12] Kale, R., Lu, Z., Fok, K. & Thing, V. A Hybrid Deep Learning Anomaly Detection Framework for Intrusion Detection. *2022 IEEE 8th Intl Conference On Big Data Security On Cloud (BigDataSecurity), IEEE Intl Conference On High Performance And Smart Computing, (HPSC) And IEEE Intl Conference On Intelligent Data And Security (IDS)*. (2022,5)

[13] Cui, Y., Liu, Z. & Lian, S. A Survey on Unsupervised Visual Industrial Anomaly Detection Algorithms. *ArXiv:2204.11161 [cs]*. (2022,8), https://arxiv.org/abs/2204.11161

[14] Jiang, M., Hou, C., Zheng, A., Hu, X., Han, S., Huang, H., He, X., Yu, P. & Zhao, Y. Weakly Supervised Anomaly Detection: A Survey. *ArXiv:2302.04549 [cs]*. (2023,2), https://arxiv.org/abs/2302.04549

[15] Goernitz, N., Kloft, M., Rieck, K. & Brefeld, U. Toward Supervised Anomaly Detection. *Journal Of Artificial Intelligence Research*. **46** pp. 235-262 (2013,2)

[16] Kingma, D. & Welling, M. Auto-Encoding Variational Bayes. (arXiv.org,2013,12), https://arxiv.org/abs/1312.6114v10

[17] Chen, X., You, S., Tezcan, K. & Konukoglu, E. Unsupervised Lesion Detection via Image Restoration with a Normative Prior. *ArXiv:2005.00031 [cs, Eess]*. (2020,4), https://arxiv.org/abs/2005.00031

[18] Liu, Q., Xu, J., Jiang, R. & Wong, W. Density estimation using deep generative neural networks. *Proceedings Of The National Academy Of Sciences Of The United States Of America*. **118** (2021,4)

[19] Dinh, L., Sohl-Dickstein, J. & Bengio, S. Density estimation using Real NVP. *ArXiv:1605.08803 [cs, Stat]*. (2017,2), https://arxiv.org/abs/1605.08803

[20] Oord, A., Vinyals, O. & Kavukcuoglu, K. Neural Discrete Representation Learning. *ArXiv:1711.00937 [cs]*. (2018,5), https://arxiv.org/abs/1711.00937

[21] Esser, P., Rombach, R. & Ommer, B. Taming Transformers for High-Resolution Image Synthesis. *ArXiv:2012.09841 [cs]*. (2021,2), https://arxiv.org/abs/2012.09841

[22] Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, Ł., Shazeer, N., Ku, A. & Tran, D. Image Transformer. *ArXiv:1802.05751 [cs]*. (2018,6), https://arxiv.org/abs/1802.05751

[23] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L. & Polosukhin, I. Attention Is All You Need. (arXiv.org,2017), https://arxiv.org/abs/1706.03762

[24] Chen, X., Mishra, N., Rohaninejad, M. & Abbeel, P. PixelSNAIL: An Improved Autoregressive Generative Model. *ArXiv:1712.09763 [cs, Stat]*. (2017,12), https://arxiv.org/abs/1712.09763

[25] Oord, A., Kalchbrenner, N. & Kavukcuoglu, K. Pixel Recurrent Neural Networks. *ArXiv:1601.06759 [cs]*. (2016,8), https://arxiv.org/abs/1601.06759v3

[26] Bond-Taylor, S., Hessey, P., Sasaki, H., Breckon, T. & Willcocks, C. Unleashing Transformers: Parallel Token Prediction with Discrete Absorbing Diffusion for Fast High-Resolution Image Generation from Vector-Quantized Codes. *ArXiv:2111.12701 [cs]*. (2021,11), https://arxiv.org/abs/2111.12701

[27] Tobler, W. A Computer Movie Simulating Urban Growth in the Detroit Region. *Economic Geography*. **46** pp. 234 (1970,6)

[28] Nikparvar, B. & Thill, J. Machine Learning of Spatial Data. *ISPRS International Journal Of Geo-Information*. **10** pp. 600 (2021,9)

[29] Gallego-Mejia, J., Bustos-Brinez, O. & González, F. LEAN-DMKDE: Quantum Latent Density Estimation for Anomaly Detection. *ArXiv:2211.08525 [quant-ph, Stat]*. (2022,11), https://arxiv.org/abs/2211.08525

[30] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. & Sutskever, I. Language Models are Unsupervised Multitask Learners. (2019)

[31] Verma, P. & Berger, J. Audio Transformers:Transformer Architectures For Large Scale Audio Understanding. Adieu Convolutions. *ArXiv:2105.00335 [cs, Eess]*. (2021,5), https://arxiv.org/abs/2105.00335

[32] Bridle, J. Training Stochastic Model Recognition Algorithms as Networks can Lead to Maximum Mutual Information Estimation of Parameters. (Morgan-Kaufmann,1989),

[33] Isola, P., Zhu, J., Zhou, T. & Efros, A. Image-to-Image Translation with Conditional Adversarial Networks. *ArXiv:1611.07004 [cs]*. (2018,11), https://arxiv.org/abs/1611.07004v3

[34] Zhang, R., Isola, P., Efros, A., Shechtman, E. & Wang, O. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. *ArXiv:1801.03924 [cs]*. (2018,4), https://arxiv.org/abs/1801.03924

[35] Simonyan, K. & Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. (arXiv.org,2014), https://arxiv.org/abs/1409.1556v6

[36] Ba, J., Kiros, J. & Hinton, G. Layer Normalization. *ArXiv:1607.06450 [cs, Stat]*. (2016,7), https://arxiv.org/abs/1607.06450

[37] Yang, X., Zhang, H., Qi, G. & Cai, J. Causal Attention for Vision-Language Tasks. *ArXiv:2103.03493 [cs]*. (2021,3), https://arxiv.org/abs/2103.03493

[38] Hendrycks, D. & Gimpel, K. Gaussian Error Linear Units (GELUs). *ArXiv:1606.08415 [cs]*. (2020,7), https://arxiv.org/abs/1606.08415

[39] Srivastava, N., Hinton, G., Krizhevsky, A. & Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. (unknown,2014,6), https://www.researchgate.net/publication/286794765-Dropout-A-Simple-Way-to-Prevent-Neural-Networks-from-Overfitting

[40] Clevert, D., Unterthiner, T. & Hochreiter, S. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). (arXiv.org,2015), https://arxiv.org/abs/1511.07289

[41] Ramachandran, P., Zoph, B. & Le, Q. Searching for Activation Functions. (arXiv.org,2017), https://arxiv.org/abs/1710.05941

[42] Rosenberger, J., Müller, K., Selig, A., Bühren, M. & Schramm, D. Extended kernel density estimation for anomaly detection in streaming data. *Procedia CIRP*. **112** pp. 156-161 (2022)

[43] He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition. (arXiv.org,2015), https://arxiv.org/abs/1512.03385v1

[44] Deng, J., Dong, W., Socher, R., Li, L., Li, K. & Fei-Fei, L. ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference On Computer Vision And Pattern Recognition*. (2009,6)

[45] Maćkiewicz, A. & Ratajczak, W. Principal components analysis (PCA). *Computers & Geosciences*. **19** pp. 303-342 (1993,3)

[46] Zambom, A. & Dias, R. A Review of Kernel Density Estimation with Applications to Econometrics. (2012), https://arxiv.org/pdf/1212.2812.pdf

[47] Akcay, S., Ameln, D., Vaidya, A., Lakshmanan, B., Ahuja, N. & Genc, U. Anomalib: A Deep Learning Library for Anomaly Detection. *ArXiv:2202.08341 [cs]*. (2022,2), https://arxiv.org/abs/2202.08341

[48] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J. & Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. (arXiv.org,2019), https://arxiv.org/abs/1912.01703

[49] Collobert, R., Bengio, S. & Mariéthoz, J. Torch: a modular machine learning software library. (www.semanticscholar.org,2002), https://www.semanticscholar.org/paper/Torch

[50] Park, H., Noh, J. & Ham, B. Learning Memory-guided Normality for Anomaly Detection. *ArXiv:2003.13228 [cs]*. (2020,3), https://arxiv.org/abs/2003.13228

[51] Le, V. & Kim, Y. Attention-based residual autoencoder for video anomaly detection. *Applied Intelligence*. (2022,5)

[52] Iandola, F., Han, S., Moskewicz, M., Ashraf, K., Dally, W. & Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and . *ArXiv:1602.07360 [cs]*. (2016,11), https://arxiv.org/abs/1602.07360

[53] Krizhevsky, A., Sutskever, I. & Hinton, G. ImageNet Classification with Deep Convolutional Neural Networks. (Curran Associates, Inc.,2012)

[54] Bai, Y., Yang, E., Han, B., Yang, Y., Li, J., Mao, Y., Niu, G. & Liu, T. Understanding and Improving Early Stopping for Learning with Noisy Labels. *ArXiv:2106.15853 [cs]*. (2021,12), https://arxiv.org/abs/2106.15853

[55] Lipton, Z., Elkan, C. & Naryanaswamy, B. Thresholding Classifiers to Maximize F1 Score. , https://arxiv.org/pdf/1402.1892.pdf

[56] Muschelli, J. ROC and AUC with a Binary Predictor: a Potentially Misleading Metric. *Journal Of Classification*. (2019,12)

[57] Chan, R., Lis, K., Uhlemeyer, S., Blum, H., Honari, S., Siegwart, R., Fua, P., Salzmann, M. & Rottmann, M. SegmentMeIfYouCan: A Benchmark for Anomaly Segmentation. *ArXiv:2104.14812 [cs]*. (2021,11), https://arxiv.org/abs/2104.14812

[58] Liu, B., Tan, P. & Zhou, J. Unsupervised Anomaly Detection by Robust Density Estimation. *Proceedings Of The AAAI Conference On Artificial Intelligence*. **36** pp. 4101-4108 (2022,6), https://ojs.aaai.org/index.php/AAAI/article/view/20328

## APPENDIX

| Percentile | F1 | AUC |
|---|---|---|
| 20 | 0.552 | 0.694 |
| 40 | 0.615 | 0.701 |
| 60 | 0.650 | 0.727 |
| 80 | 0.780 | 0.860 |
| 90 | 0.794 | 0.885 |

TABLE 13: Performance of Different $\lambda_P$ values for VAE+PixelSnail on validation set

| Percentile | F1 | AUC |
|---|---|---|
| 20 | 0.637 | 0.691 |
| 40 | 0.628 | 0.695 |
| 60 | 0.687 | 0.725 |
| 80 | 0.845 | 0.894 |
| 90 | 0.717 | 0.785 |

TABLE 14: Performance of Different $\lambda_P$ values for VAE+Transformer on validation set

| Percentile | F1 | AUC |
|---|---|---|
| 20 | 0.571 | 0.691 |
| 40 | 0.765 | 0.869 |
| 60 | 0.823 | 0.920 |
| 80 | 0.800 | 0.912 |
| 90 | 0.817 | 0.918 |

TABLE 15: Performance of Different $\lambda_P$ values for GAN+PixelSnail on validation set

| Percentile | F1 | AUC |
|---|---|---|
| 20 | 0.697 | 0.712 |
| 40 | 0.721 | 0.763 |
| 60 | 0.747 | 0.764 |
| 80 | 0.796 | 0.800 |
| 90 | 0.801 | 0.808 |

TABLE 16: Performance of Different $\lambda_P$ values for GAN+Transformer on validation set

| Percentile | F1 | AUC |
|---|---|---|
| 20 | 0.789 | 0.856 |
| 40 | 0.791 | 0.857 |
| 60 | 0.800 | 0.858 |
| 80 | 0.796 | 0.861 |
| 90 | 0.787 | 0.853 |

TABLE 17: Performance of Different $\lambda_{LP}$ values for VAE+PixelSnail on validation set

| Percentile | F1 | AUC |
|---|---|---|
| 20 | 0.810 | 0.862 |
| 40 | 0.809 | 0.860 |
| 60 | 0.810 | 0.859 |
| 80 | 0.810 | 0.860 |
| 90 | 0.815 | 0.860 |

TABLE 18: Performance of Different $\lambda_{LP}$ values for VAE+Transformer on validation set

| Percentile | F1 | AUC |
|---|---|---|
| 20 | 0.883 | 0.947 |
| 40 | 0.889 | 0.945 |
| 60 | 0.908 | 0.945 |
| 80 | 0.890 | 0.944 |
| 90 | 0.889 | 0.945 |

TABLE 19: Performance of Different $\lambda_{LP}$ values for GAN+PixelSnail on validation set

| Percentile | F1 | AUC |
|---|---|---|
| 20 | 0.844 | 0.884 |
| 40 | 0.849 | 0.886 |
| 60 | 0.839 | 0.886 |
| 80 | 0.860 | 0.889 |
| 90 | 0.855 | 0.889 |

TABLE 20: Performance of Different $\lambda_{LP}$ values for GAN+Transformer on validation set

| Percentile | F1 | AUC |
|---|---|---|
| 20 | 0.894 | 0.941 |
| 40 | 0.906 | 0.940 |
| 60 | 0.863 | 0.944 |
| 80 | 0.847 | 0.934 |
| 90 | 0.847 | 0.920 |

TABLE 21: Performance of Different $\lambda_S$ values for VAE+PixelSnail on validation set

| Percentile | F1 | AUC |
|---|---|---|
| 20 | 0.914 | 0.955 |
| 40 | 0.924 | 0.954 |
| 60 | 0.952 | 0.908 |
| 80 | 0.945 | 0.901 |
| 90 | 0.849 | 0.931 |

TABLE 22: Performance of Different $\lambda_S$ values for VAE+Transformer on validation set

| Percentile | F1 | AUC |
|---|---|---|
| 20 | 0.995 | 0.996 |
| 40 | 0.995 | 0.998 |
| 60 | 0.995 | 0.998 |
| 80 | 0.983 | 0.995 |
| 90 | 0.681 | 0.866 |

TABLE 23: Performance of Different $\lambda_{LP}$ values for GAN+PixelSnail on validation set

| Percentile | F1 | AUC |
|---|---|---|
| 20 | 0.992 | 0.997 |
| 40 | 0.993 | 0.998 |
| 60 | 0.982 | 0.995 |
| 80 | 0.945 | 0.985 |
| 90 | 0.864 | 0.950 |

TABLE 24: Performance of Different $\lambda_{LP}$ values for GAN+Transformer on validation set

| Method | sec/image |
|---|---|
| Sample-wise | 0.04 |
| Pixel-wise | 7.8 |
| Light Pixel-wise | 0.11 |
| Regenerative | 0.02 |

TABLE 25: Comparison of Running times (seconds per image) for Anomaly Detection methods on the VAE+PixelSnail

| Method | sec/image |
|---|---|
| Sample-wise | 0.04 |
| Pixel-wise | 8.0 |
| Light Pixel-wise | 0.12 |
| Regenerative | 0.02 |

TABLE 26: Comparison of Running times (seconds per image) for Anomaly Detection methods on the VAE+Transformer

| Method | sec/image |
|---|---|
| Sample-wise | 0.04 |
| Pixel-wise | 7.8 |
| Light Pixel-wise | 0.11 |
| Regenerative | 0.02 |

TABLE 27: Comparison of Running times (seconds per image) for Anomaly Detection methods on the GAN+PixelSnail

(a) Original image

(b) VAE+PixelSnail Restoration

(c) VAE+Transformer Restoration

(d) GAN+PixelSnail Restoration

(e) GAN+Transformer Restoration

Fig. 12: Different Normal Restoration results between each model for $N = 5$ Restorations via Pixel-wise detection

(a) Original image

(b) VAE+PixelSnail Restoration

(c) VAE+Transformer Restoration

(d) GAN+PixelSnail Restoration

(e) GAN+Transformer Restoration

Fig. 13: Different Normal Restoration results between each model for $N = 5$ Restorations via Light-Pixel-wise detection

(a) Original image

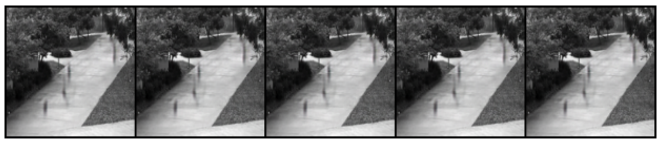
(b) VAE+PixelSnail Residuals

(c) VAE+Transformer Residuals

(d) GAN+PixelSnail Residuals

Fig. 14: Different Anomalous Restoration results between VP, VT and GP (GT results shown in paper) for $N = 5$ Restorations via Pixel-wise detection

(a) Original image



(b) VAE+PixelSnail Residuals

(c) VAE+Transformer Residuals

(d) GAN+PixelSnail Residuals

Fig. 15: Different Anomalous Restoration results between VP, VT and GP (GT results shown in paper) for $N = 5$ Restorations via Light-Pixel-wise detection

(a) Original image

(b) VQ-VAE Regeneration

(c) VQ-GAN Regeneration

Fig. 16: Normal image Regeneration results for VQ-VAE and VQ-GAN

(a) Original image

(b) VAE+PixelSnail Residuals

(c) VAE+Transformer Residuals

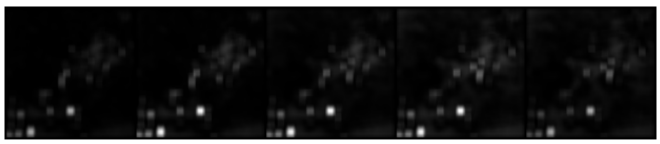
(d) GAN+PixelSnail Residuals

(e) GAN+Transformer Residuals

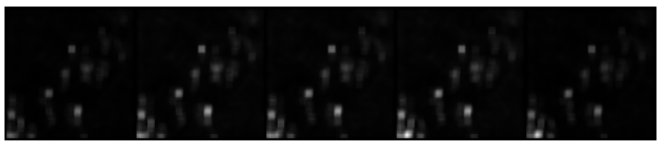
Fig. 17: Anomalous Residuals from Pixel-wise Restorations

(a) Original image

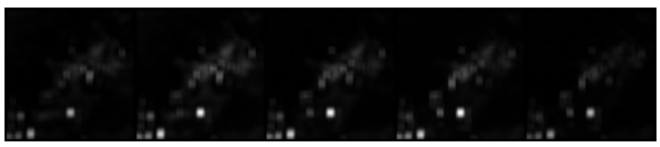
(b) VAE+PixelSnail Residuals



(c) VAE+Transformer Residuals



(d) GAN+PixelSnail Residuals



(e) GAN+Transformer Residuals

Fig. 18: Normal Residuals from Pixel-wise Restorations