Anomaly Detection in 3D Reconstruction Using Generative Adversarial Networks

Thomas Nguyen

University of Toronto, Toronto, Canada
tomasdfgh.nguyen@mail.utoronto.ca

Research conducted at Osaka University during a 2024 ESROP Global summer research position

Abstract

While traditional anomaly detection techniques have shown effectiveness in low dimensional data, their performance often diminishes in high-dimensional contexts, such as 3D imaging. To address this, the methods of Deecke et al is replicated and extended. by applying their generative adversarial network (GAN) framework to the domain of 3D image anomaly detection. The new approach leverages a GAN to map 3D data samples into a latent space, determining anomaly scores based on the quality of this mapping. If the GAN fails to find an adequate latent representation, the sample is classified as anomalous. The applicability of this technique is demonstrated to 3D datasets, achieving competitive performance and validating the effectiveness of GAN-based methods in identifying anomalies in complex, high-dimensional data.

Introduction

Anomaly detection is a critical task in machine learning, where the goal is to identify instances in data that deviate significantly from the norm. This problem is particularly challenging in high-dimensional spaces, such as those encountered in 3D reconstruction, where traditional methods often struggle due to the complexity and volume of data. Generative Adversarial Networks (GANs) have emerged as a powerful tool for anomaly detection, particularly in scenarios where abnormal data is scarce or hard to define. GANs are capable of learning the distribution of normal data and can be used to detect anomalies by assessing how well new data fits within this learned distribution.

A GAN consists of two neural networks: a generator, which creates data samples, and a discriminator, which distinguishes between real and generated samples. The generator aims to produce samples indistinguishable from the real data, while the discriminator learns to identify which samples are real and which are generated. This adversarial process encourages the generator to create increasingly realistic samples, effectively learning the underlying data distribution.

This paper builds on the method introduced by Deecke et al. [1], who demonstrated the effectiveness of GANs for anomaly detection in 2D images. Their approach is extended in this paper to the domain of 3D reconstruction, focusing on the identification of anomalies in 3D datasets. By leveraging the power of GANs, the paper's aim is to develop a robust system for detecting anomalies in complex, high-dimensional 3D data, which could have significant implications for various applications such as quality control in manufacturing, medical imaging, and autonomous systems. Our approach adapts the GAN framework to handle the unique challenges posed by 3D data, providing a novel contribution to the field of anomaly detection.

Related Works

In this section, literature on anomaly detection is reviewed, focusing on 2D and 3D image data. First discussing foundational works

in GAN architectures, particularly the DCGAN and WGAN models, which form the basis of this paper's approach. After, specific studies in anomaly detection for 2D and 3D datasets will be delved into.

Generative Adversarial Networks: Generative Adversarial Networks (GANs), introduced by Goodfellow et al. [2], have been revolutionary in generating realistic data samples. DCGANs (Deep Convolutional GANs), as proposed by Radford et al. [3], enhance GANs by incorporating convolutional neural networks, which are more effective for image data. DCGANs are known for their ability to produce high-quality images and are particularly noted for their stability and efficiency in training. Wasserstein GANs (WGANs), introduced by Arjovsky et al. [4], improve upon traditional GAN training by addressing issues related to training stability and mode collapse using the Earth Mover's (Wasserstein) distance as a loss metric.

In this paper, the DCGAN architecture will serve as the primary model for experiments, leveraging its convolutional structure to capture spatial hierarchies in 3D image data. Additionally, the WGAN training technique will be employed to potentially enhance the stability and robustness of the model training process.

Anomaly Detection in 2D: The anomaly detection task aims to identify outliers or unusual instances within a dataset, which is critical in various fields such as medical diagnostics, security, and manufacturing. The work of Deecke et al. has been influential in this domain, particularly for 2D image datasets. Their method uses GANs to map samples to a latent space, classifying those that fail to find a suitable representation as anomalies. This paper extends the methodology of Deecke et al., applying similar techniques to a 3D dataset, thus testing its applicability in a new domain.

In the realm of 2D anomaly detection, several methods have been proposed:

- Deep Neural Networks (DNNs): These models, including various forms of CNNs, have been used extensively due to their ability to learn complex patterns and features from data. Techniques often involve training on normal data to detect deviations during inference. Augusteijn and Folkert [5], looked at using a Deep Neural Network classification model to detect anomalous data that is novel from the rest of the dataset and concluded that it is not a suitable method for the task.

- Variational Autoencoders (VAEs): VAEs have been utilized for their ability to learn probabilistic representations of data. By modeling the distribution of normal data, VAEs can identify samples that deviate significantly from this distribution as anomalies. An and Cho [6] looked at a method for anomaly detection using the reconstruction probability from a variational auto-encoder. Unlike traditional methods that use reconstruction error, reconstruction probability offers a more principled and objective measure by accounting for the variability in data distribution. The results show that this method performs better than those based on auto-encoders and principal components. The VAE's generative capabilities also help in understanding the underlying causes of anomalies.

- GAN-based Approaches: Apart from the work of Deecke et al., other studies have explored using GANs for anomaly detection, capitalizing on the ability of GANs to generate data similar to the training set. These methods often involve measuring the reconstruction error or the likelihood of the data given the GAN model. Xia et al. [7] reviewed GAN-based approaches for anomaly detection and highlighted the unique advantages GANs offer in this domain. Traditional deep neural network models often struggle with anomaly detection due to the difficulty in obtaining sufficient anomalous

data, which can lead to sub-optimal performance. This issue leads to a data imbalance problem, where anomalous instances are rare compared to normal data, making it challenging for conventional models to learn effectively.

GANs address this problem by learning the distribution of normal data, which enables them to detect anomalies by assessing whether new samples fall within this learned distribution. This approach allows GANs to perform anomaly detection using only normal data, without requiring any anomalous samples. As a result, GAN-based methods have gained considerable traction and led to a significant increase in related publications, as noted in the survey.

Anomaly Detection in 3D: Anomaly detection in 3D datasets remains less explored, largely due to the complexity and high dimensionality of 3D data, as well as the scarcity of annotated datasets. The existing studies have mostly adapted 2D techniques to 3D data, but the transition is not straightforward due to differences in data representation and the additional computational burden.

Recent studies have begun to address this gap by creating datasets that include anomalous 3D reconstructions, enabling more systematic study and evaluation of anomaly detection methods in this domain. However, the availability of such datasets is limited, which poses a significant challenge for researchers. Bergmann et al. [8] introduced one of the few available 3D datasets specifically designed for anomaly detection and localization tasks. The authors themselves acknowledge the limited availability of such datasets.

This paper aims to fill this gap by applying the successful GAN-based anomaly detection techniques from 2D data to a 3D context. The challenges posed by the scarcity of suitable datasets is acknowledged and addressed by curating our own 3D dataset with labeled anomalies. This effort is crucial for the advancement of anomaly detection methodologies in 3D data and for the broader application of these techniques in fields such as medical imaging, autonomous driving, and industrial inspection. The next section contains a breakdown of the dataset of 3D models that was created specifically for this paper.

Description of the Dataset

Due to a lack of available 3D Modelling Data, a dataset is created as a part of this research. To test for anomalous data, there has to be multiple labels where one will be deemed as normal and the others anomalous. As a result, 3 types of data is created corresponding to 3 different labels: Rectangular, Cylindrical, and Spherical objects. Each label will contain 2000 different models of their respective object type at different angles and object sizes. 1500 scans will be allocated to the training set, while 500 will be allocated to the test set.

Category	# Train	# Test	Image Size
Rectangular	1500	500	1080 x 1080
Cylindrical	1500	500	1080 x 1080
Spherical	1500	500	1080 x 1080

Table 1: Statistical breakdown of the dataset used for the anomaly detection task

Defining Anomalous Data

While the definition of anomalous data can vary widely depending on its application, this paper defines anomalous data as those samples that lie outside the distribution p of the primary dataset. For the dataset used in this study, each class: rectangle, cylinder, and spheres has its own distinct data distribution: $p_{rectangle}$, $p_{cylinder}$, and $p_{spheres}$ respectively. Given any classes, any data points from a different class would fall outside of its data distribution. This classification ensures that the identification of anomalies is precise and contextual to the specific distributions of each class within the dataset.

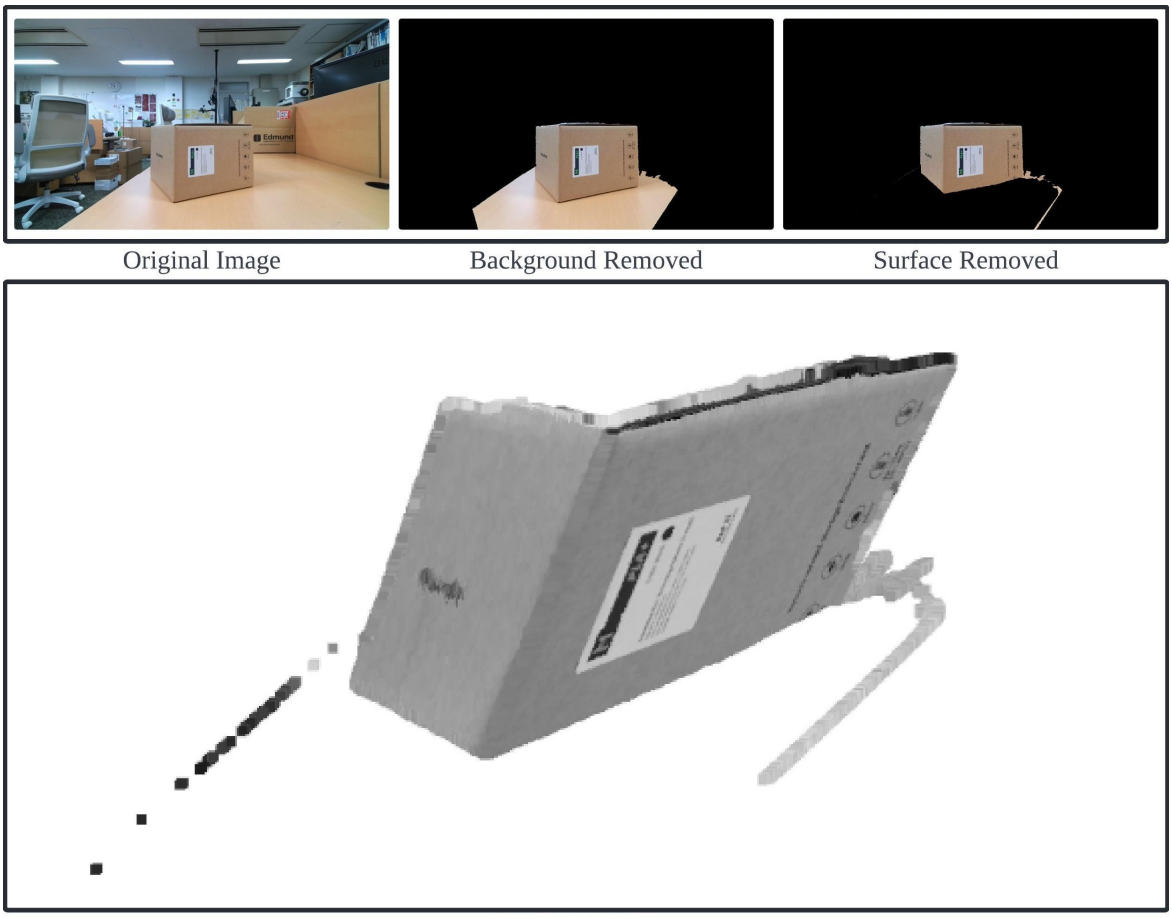


Figure 1: Example of data acquisition setup for 3D modeling. Starting with the original image with the surface and background, followed by background removed then surface removed. Each of the RGB images shown on the first row has a corresponding Depth image. This process is repeated for the entire dataset to achieve the final 3D model for each data point.

Data Acquisition

To acquire the data, an RGBD camera was employed to capture images of the object from a single angle. This approach simplifies the model's application by requiring it to identify anomalies based on a single snapshot rather than a complete view of the object. The RGBD camera captures both an RGB image, with three color channels, and a depth image of the object. These images are then used to reconstruct a 3D model of the object by converting the data into a point cloud.

For each data point, background information is removed by discarding any pixels with a depth value above 700, ensuring the focus remains on the object itself. Additionally, the surface on which the object rests is filtered out by eliminating pixels where the depth value difference between the empty surface and the surface with the object exceeds 90%. This threshold indicates the presence of the object, effectively isolating it from the background and underlying surface. Figure 1 shows what the data sample looks like after each data processing step.

It is important to note that while the input data inherently represents a 3D model, the approach leverages RGB and Depth images to simplify the analysis. By focusing on these two-dimensional representations, the complex

task of 3D anomaly detection is effectively reduced to an image anomaly detection problem. This method allows for the application of well-established techniques in image processing and machine learning, streamlining the detection of anomalies and making the overall system more efficient and accessible. Moreover, despite this simplification, the approach retains the capacity for 3D applications, as the depth information still encapsulates critical spatial details that can be used to reconstruct or analyze the 3D structure. This dual capability ensures that the system benefits from the robustness and clarity of 2D techniques while preserving the essential 3D characteristics of the data, making it versatile and powerful for various anomaly detection tasks.

Generative Adversarial Networks (GANs)

GANs represent a significant advancement in the field of generative modeling. GANs consist of two neural networks: the generator (G_θ) and the discriminator (D_ω). These two networks are trained simultaneously through a competitive game process, where the generator creates data and the discriminator evaluates its authenticity. This section delves into the underlying theory of GANs, explores the roles and interactions of both the generator and the discriminator in detail, and discusses the stability improvements introduced by Wasserstein GANs (WGANs). Additionally, it highlights the significant contributions of WGANs to more robust and effective generative modeling, emphasizing their impact on reducing training instability and improving the quality of generated data.

Theory Behind GANs

The core concept of GANs revolves around a min-max optimization problem where the generator G_θ and the Discriminator D_ω engage in a two-person game. The objective of the Generator is to generate data samples that are indistinguishable from real data samples, while the discriminator is tasked with distinguishing between real and generated samples. This interaction can be formalized using the equation 1.

$$\min_G \max_D \mathbb{E}_{x\sim p}[\log D_\omega(\mathrm{x})]+ \mathbb{E}_{z\sim p_z(z)}[\log(1 - D_\omega(G_\theta(z)))] \tag{1}$$

Where:

- p represents the distribution of real data samples.
- p_z is the prior distribution of the latent space. This paper uses normal distribution
- $G_\theta(z)$ denotes the generator's output given a latent vector z
- $D_\omega(x)$ is the probability that a given sample x is real

With this GAN objective function setup:

- The generator G_θ aims to minimize $\log(1-D_\omega(G_\theta(z)))$, which is the log probability of the discriminator classifying a generated sample as fake.
- The discriminator D_ω aims to maximize $\log D_\omega(x) + \log(1-D_\omega(G_\theta(z)))$, which is the log probability of correctly classifying real and fake samples.

Mapping Latent Space to Data Space

Generative Adversarial Networks (GANs) map latent vectors z from the latent space, following a normal distribution in this experiment, to the data space, where the generator G_θ approximates the data distribution p. The generator transforms these noise vectors z into samples mimicking the data distribution p. Ideally, if well-trained, the generated data distribution p_θ should closely approximate the real data distribution, such that $p_\theta \sim p$. This can be effectively used for anomaly detection. Since $p_\theta \sim p$, if a data sample falls outside of p, it likely falls outside of p_θ. Consequently, the generator's ability to produce data similar to the real distribution helps it flag non-conforming data, thus detecting anomalies.

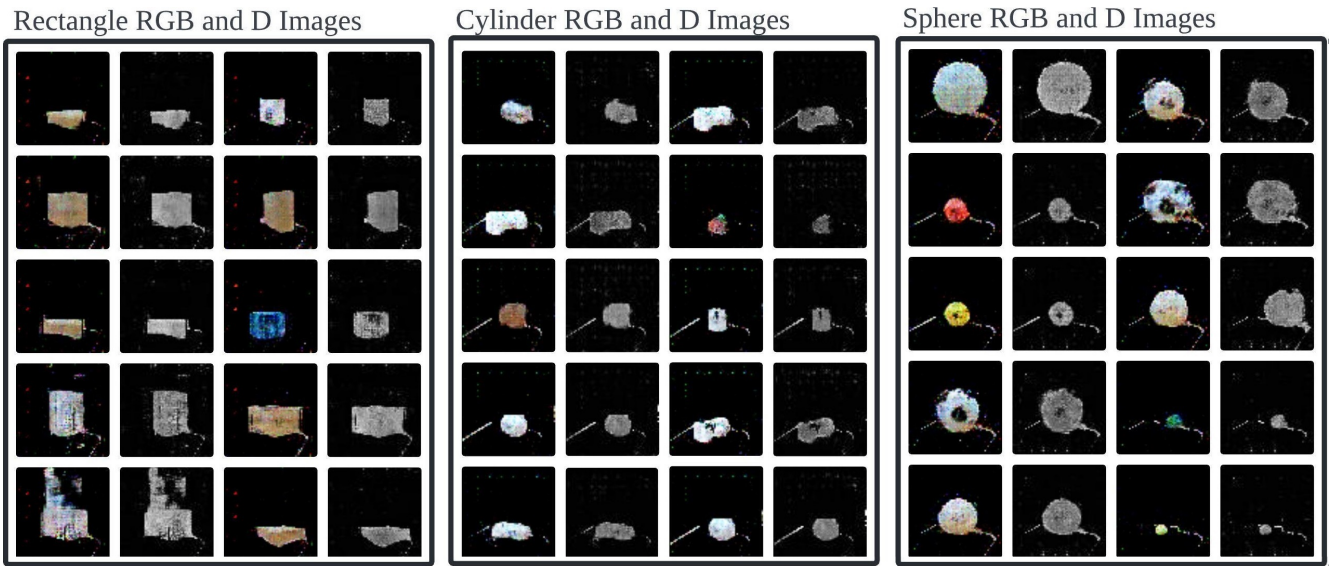


Figure 2: Generated examples for generators trained on rectangular, cylindrical, and spherical dataset using the normal GAN training process. Each example shown in this figure contain an RGB and Depth Image

Wasserstein Generative Adversarial Networks (WGANs)

Challenges in Training GANs

Generative Adversarial Networks (GANs) have revolutionized data generation tasks but are notorious for being challenging to train. The primary difficulties include mode collapse, where the generator produces limited varieties of samples, and training instability, often due to the discriminator overwhelming the generator. These issues are typically exacerbated by the use of Jensen-Shannon divergence as the metric for training, which can cause gradients to vanish, leading to poor updates for the generator.

Wasserstein GAN: A Solution

The Wasserstein GAN (WGAN) addresses these challenges by replacing the Jensen-Shannon divergence with the Wasserstein distance (or Earth Mover's distance) to measure similarity between real and generated data distributions. The Wasserstein distance provides a smoother gradient landscape, which helps prevent the vanishing gradient problem and allows for more stable generator updates.

WGANs use a critic network instead of a discriminator to estimate the Wasserstein distance between data distributions. The critic is constrained to be Lipschitz continuous, typically enforced by weight clipping or gradient penalty methods.

WGAN Objective Function

The objective function for WGAN is:

$$\min_{G} \max_{C \in \mathcal{C}} \mathbb{E}_{x \sim p}[C(x)] - \mathbb{E}_{z \sim p_z}[C(G(z))] \qquad (2)$$

Where:

- G_θ is the generator
- C is the critic network
- $\mathcal{C}$ is the set of 1-Lipschitz functions
- p is the real data distribution
- p_z is the latent space distribution, normal distribution for this paper

The critic maximizes the difference between its output expectations for real and generated samples, while the generator minimizes this difference, aiming to produce indistinguishable samples from real data.

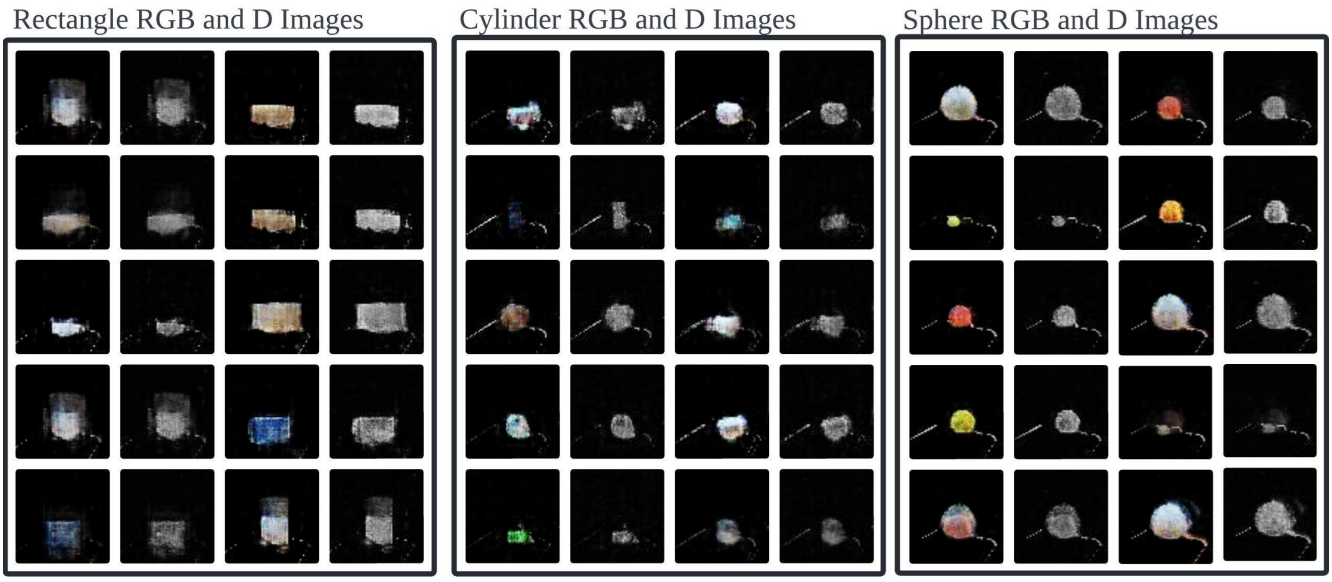


Figure 3: Generated examples for generators trained on rectangular, cylindrical, and spherical datasets using the WGAN training process. Each example shown in this figure contains an RGB and Depth Image.

Impact and Results

WGANs have significantly improved the generation of diverse and realistic samples compared to traditional GANs. The stable training process leads to better convergence and reduces issues like mode collapse. The success of WGANs has led to widespread adoption and innovations, including WGAN-GP (WGAN with Gradient Penalty), which addresses the limitations of weight clipping. In this paper, generators are trained using both the traditional GAN training technique as well as WGAN training technique, and both employed the same DCGAN architecture described in [3]. Visually, both training methods successfully captured the data distribution of their respective training datasets. However, the models trained with WGANs produced results that were clearer and more accurately representative of the dataset.

Algorithm

The algorithm employed in this experiment is based on the approach used by Deecke et al. [1], but it has been adapted to accommodate a 4-channel data frame (incorporating both RGB and Depth information, instead of just RGB). The core principle behind the algorithm is that if the generator has accurately captured the distribution of real data through training, then for any given normal data sample $x \sim p$, there exists a latent vector z that enables the generator to produce a 3D model closely resembling the sample x. Conversely, if a data sample x does not belong to the data distribution p, there will be no corresponding latent vector z that can accurately reproduce the sample. In such cases, the model will struggle to find a suitable z, and the data sample will be classified as anomalous.

Latent Space and Reconstruction Loss Propagation

The algorithm is performed by propagating through the latent space to find a latent vector that would allow $G_\theta(z)$ to most closely resemble x. z_0 is initialized from p_z, where p_z is the distribution of the latent space. A reconstruction loss l is obtained from $G_\theta(z_0)$ and x, defined as $l(G_\theta(z_0), x)$. This loss is propagated back to the latent space to find latent vector z_1. This process is repeated for k steps, following through different generations of generated samples of

Algorithm 1 Anomaly Detection using GANs for 4 channels (RGBD) data frame

Input: learning rate for latent vector and generator (γ_z, γ_θ), n_{seed}, data sample x, latent space distribution p_z, generator G_θ, reconstruction loss l
Output: Anomaly Score

Initialize $\{z_{j,0} \mid z_{j,0} \sim p_z, j = 1, \ldots, n_{\text{seed}}\}$ and $\{G_{\theta_{j,0}} \mid G_{\theta_{j,0}} \triangleq G_\theta, j = 1, \ldots, n_{seeds}\}$
for $j = 1$ **to** n_{seed} **do**
  for $t = 1$ **to** k **do**
    $z_{j,t} \leftarrow z_{j,t-1} - \gamma_z \cdot \nabla_{z_{j,t-1}} l(G_{\theta_{j,t-1}}(z_{j,t-1}), x)$
    $\theta j, t \leftarrow \theta_{j,t-1} - \gamma_\theta \cdot \nabla_{\theta_{j,t-1}} l(G_{\theta_{j,t-1}}(z_{j,t-1}), x)$
  end for
end for
Return: $(1/n_{seed}) \sum_{j=1}^{n_{seed}} l(G_{\theta_{j,k}}(z_{j,k}), x)$

$G_\theta(z_0)$, $G_\theta(z_1)$, ...,$G_\theta(z_k)$, the anomaly score can be calculated from $l(G_\theta(z_k), x)$.

Since optimizing in the latent space is a non-convex problem, it is helpful to search through multiple latent vectors initialized at different locations to avoid seeding at an unsuitable region. As a result, the final anomaly score can be found by averaging the reconstruction loss of all the generated samples at different seeds after the k^{th} search. Additionally, after every iteration of k, the recreation loss l is also propagated to the generator's weights θ. Even though these changes do not affect the weight of the generator permanently and is reset back to its original weight for each new testing point. Changing the weights of the generator would allow the generator itself to improve on its representative capacities. Mathematically, the overall anomaly score of a given data sample can be defined as equation 3.

$$loss = \frac{1}{n_{seed}} \left\{ \sum_n l(G_{\theta_{n,k}}(z_{n,k}), x) \right\} \quad (3)$$

Where z_k and G_{θ_k} is the latent vector and generator after k^{th} iteration of reconstruction loss propagation. If the final loss value is low for that data sample, then the data is deemed as less anomalous, and vice versa.

Learning Rate for Latent Space (γ_z) and Generator (γ_θ)

While searching through the latent space to detect anomalous data is a common technique using GANs, it is uncommon to also change the generator's weights to enhance its representational capabilities. While both improve accuracy in assigning an anomaly score to the data sample, there is a difference between propagating through the latent space and adjusting the generator's weights. The learning rates for each have different implications that can affect various applications of anomaly detection.

Changing the latent space learning rate affects the rate of difference in the generated data with each iteration of k; however, the generated data will still closely resemble the data distribution p on which the generator was trained. In this experiment, if the Rectangle GAN is used to predict anomalous data with a high latent space learning rate, the generated data will change significantly after each step but remain a rectangle since the generated data distribution p_θ has remained unchanged and still resemblance of $p_{rectangle}$.

Conversely, the generator's learning rate affects the generated data distribution p_θ itself. This parameter is important because p_θ will never exactly match p. This discrepancy could result from various factors, including a lack of diversity in the dataset. Consequently,

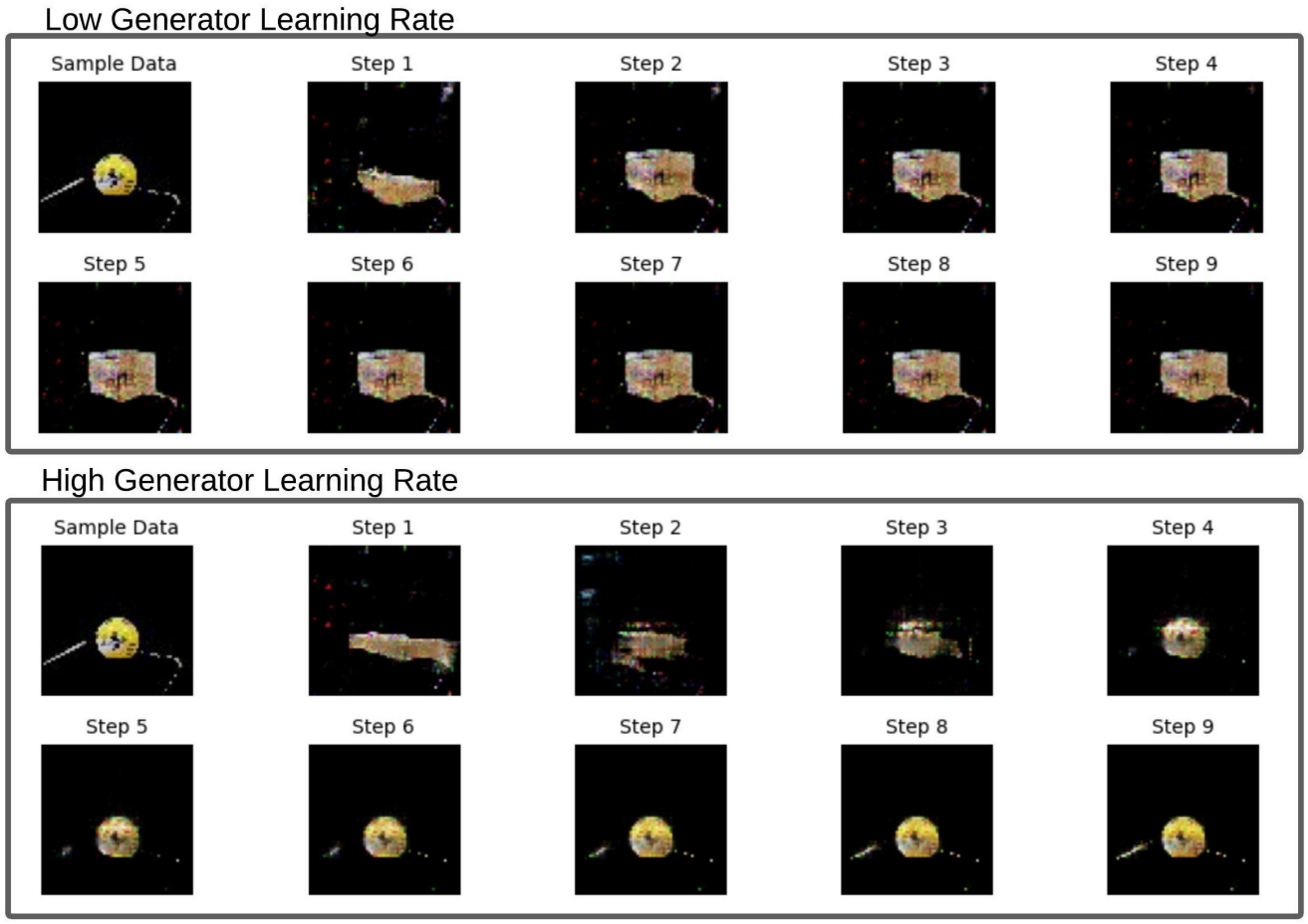


Figure 4: Generated samples through multiple iterations of latent search with low and high generator learning rate for the rectangular GAN. Samples with low generator learning rates remain in the same realm of rectangular boxes, while samples with high generator learning rates changes the generator's p_θ to no longer belonging within the realm of rectangular boxes.

when given a data sample that belong to the actual data distribution p but not p_θ, adjusting the model's weights allow p_θ to approach closer to p. However, a high generator's learning rate could result in p_θ changing completely outside of p if the data is anomalous. Therefore, the generator's learning rate is typically set low. See Figure 4 for a visual comparison of the effects of setting the generator learning rate to high and low values. Both examples use the same generator, trained on rectangular 3D model data, and evaluated with the same spherical ball data sample. Each example shows the generated data over iterations of the search process. In both cases, the generated data adjusts in size and shape to more closely resemble the data sample. However, with a low learning rate, the generated data retains its rectangular form, while with a high learning rate, the shapes become non-rectangular. This indicates that p_θ has deviated significantly from $p_{rectangle}$.

Hyperparameters

The same set hyper-parameter values were employed through out the different experiments conducted. Latent space (γ_z) and Generator (γ_θ) Learning Rate were set to be $5 \cdot 10^{-2}$ and $6 \cdot 10^{-9}$ respectively, n_{seed} as 3, and k as 19. Adam optimizer used for the generator weights loss propagation and latent space loss propagation. MSE loss is used as the reconstruction loss.

Label Category	Model	Maximum Accuracy (%)	Optimal Accuracy (%)	AUC	Average Loss		
					Rec	Cyl	Sph
Rectangle	GAN	80	64	0.71	**0.09**	0.11	0.13
	WGAN	78.7	62.0	0.68	**0.05**	0.05	0.09
Cylinder	GAN	78	68	0.76	0.14	**0.08**	0.20
	WGAN	86.7	78	0.87	0.08	**0.03**	0.12
Spheres	GAN	82.0	76.0	0.79	0.17	0.13	**0.10**
	WGAN	87.3	72.0	0.81	0.11	0.07	**0.04**

Table 2: Accuracies, average loss, and AUC for each category using both GANs and WGANs. The average loss shows the rectangular, cylindrical, and spherical loss for each generator's class training data. The lowest average loss for each class is written in bold.

Experiments

Two sets of experiments were conducted using GANs after they had converged. The first set involved using GANs for a classification task. Here, each data sample was labeled, and the GANs were tasked with detecting anomalies. The objective was for each generator to label data samples from the class it was trained on as normal and the rest as anomalous. A threshold was established: data samples with an anomaly score below the threshold were considered normal, while those with a score above were considered anomalous. The expectation was that anomalous class data samples would have higher anomaly scores than those from the normal class. This experiment was repeated six times, with a generator trained using GAN and WGAN for each class.

The second set of experiments involved unsupervised anomaly scoring. Data samples were fed into the generator without providing any ground truth. The generator assigned an anomaly score to each sample, which was then ranked from lowest to highest. Similar to the first set of experiments, it was expected that data samples belonging to the same class as the data the generator was trained on would receive lower anomaly scores.

Experiment 1: Supervised Classification

To classify data as anomalous or normal, a threshold is set to determine the generator's accuracy. A data sample with an anomaly score below the threshold is considered normal, while one above is deemed abnormal. Thus, the threshold value is crucial for the model's accuracy.

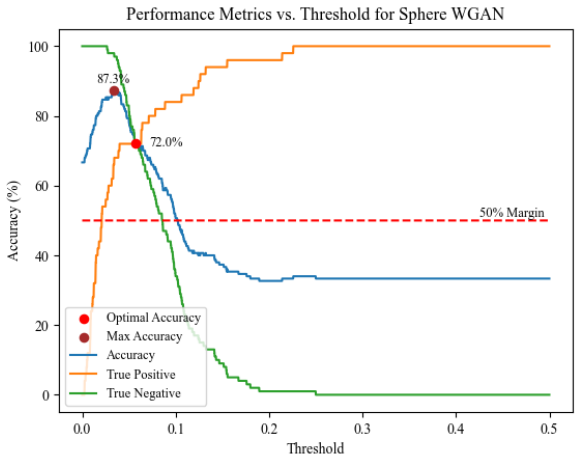


Figure 5: Overall accuracy, true positive, and true negative rate for data samples of a DCGAN trained with WGAN on the spherical dataset as a function of threshold

Figure 5 illustrates the different accuracies of a generator trained using WGAN on a dataset of spherical 3D models at various threshold values. When the threshold is set to 0, all normal and anomalous data points are classified as abnormal, resulting in a true positive value of 0 and a true negative value of 100. Conversely, if the threshold is set very high, all data points are classified as normal, leading to true positive and true negative values of 100 and 0, respectively.

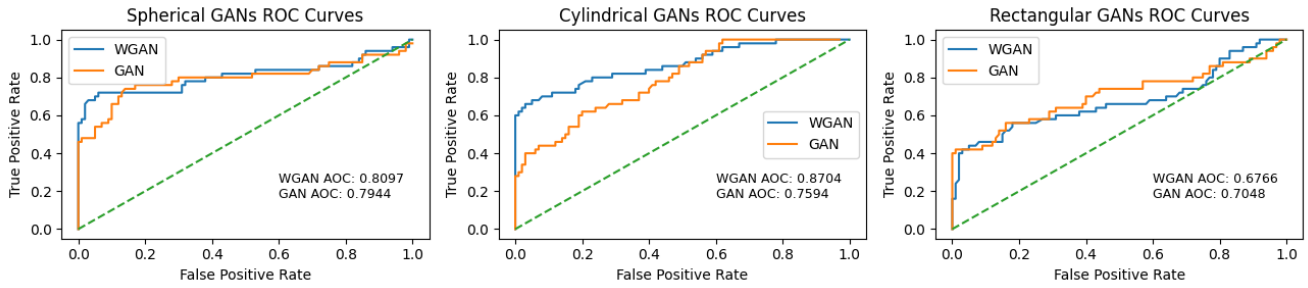


Figure 6: ROC and AUC values for WGAN and GAN generators for every class.

Given that the test set contains an equal number of data points for each of the three classes, the overall accuracy is 66% when the threshold is 0, as all abnormal data points are correctly classified, and 33% when the threshold is high, as all normal data points are correctly classified. Due to this trade off, two types of accuracies are determined for each label's WGAN and GAN generator: Maximum and Optimal accuracy. Maximum accuracy is the accuracy of the model at its highest, while optimal accuracy is the accuracy of the model when the trade off between true positive and negative is at a minimum. Effectively, optimal accuracy is found to be the accuracy of when the true positive and true negative converge to the same value.

It is important to note that the anomaly score (loss) of each data sample determines the model's overall accuracy. If the average loss of the generator's respective label is lower than that of other labels, the generator's overall accuracy should be higher. As shown in Table 2, the average loss for each type of generator is indeed lowest for its respective label, confirming this relationship.

A better method of exploring the performance of the generators could be to look at their respective ROC curves and AUC scores. ROC curves show a trade off between the true positive rate as a function of the false positive rate. Figure 6 shows the ROC curve for each model type with their respective class of training data. On average, generators trained using WGAN recieved a slightly higher or significantly higher AUC score than GAN, with the exception of Rectangle generators where WGAN scored slightly below GAN. Rectangular Generators scored the highest AUC score of **0.71**, Cylindrical **0.87**, and Spheres **0.81**.

Experiment 2: Unsupervised Anomaly Scoring

Experiment 2 is to rank the data samples based on their anomaly score. Samples are ranked based on what the generator deemed from least anomalous to most anomalous. Figure 7 shows the ranking for generator trained on cylindrical data using wgan. The generator ranked their own respective label to be the least anomalous. The purpose of this experiment is to show another practical usage of the anomalous scoring algorithm.

Conclusions

GANs were desmonstrated to be effective for anomaly detection in 3D reconstruction tasks by leveraging the latent space of the generator. The approach was validated through experiments using both GANs and WGANs, showing notable improvements in anomaly detection performance. Key findings include:

1. **Classification Accuracy:** GANs and WGANs effectively distinguished between normal and anomalous data samples, achieving high classification accuracy across different datasets.

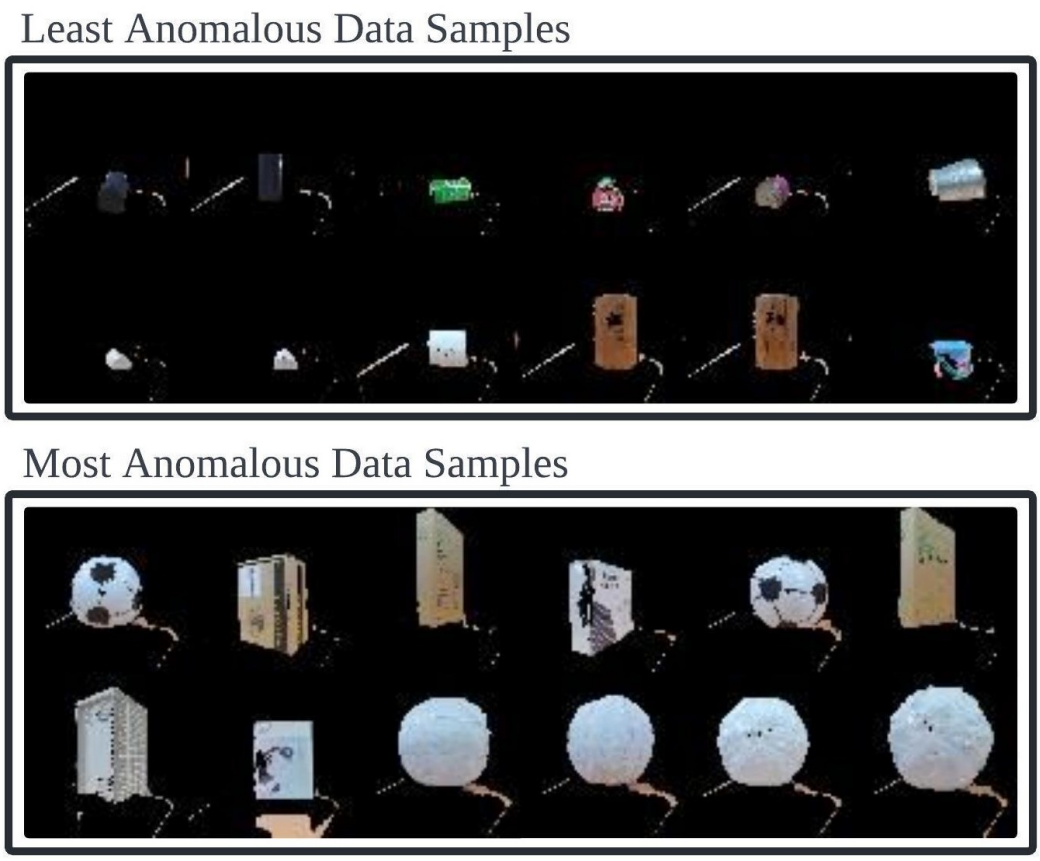


Figure 7: Data samples from dataset ranked by the wgan generator trained on cylindrical models. The data samples are ranked from least anomalous to most anomalous with the top two rows being least anomalous and the bottom two rows being most anomalous.

2. **Robustness of WGANs:** WGANs exhibit superior stability and accuracy in generating data samples, as evidenced by lower average loss values and clearer generated outputs compared to traditional GANs.

These experiments underscore the importance of fine-tuning GAN architectures and training procedures for optimal detection performance. Future work may explore enhancements such as incorporating advanced reconstruction loss functions or leveraging more sophisticated initialization techniques for latent vectors. By continuing to refine these models, the boundaries of anomaly detection capabilities in various applications can advance.

References

[1] L. Deecke, R. Vandermeulen, L. Ruff, S. Mandt, and M. Kloft, "Image anomaly detection with generative adversarial networks," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18*, pp. 3–17, Springer, 2019.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems* (Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, eds.), vol. 27, Curran Associates, Inc., 2014.

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

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

[5] M. F. Augusteijn and B. A. Folkert, "Neural network classification and novelty detection," *International Journal of Remote Sensing*, vol. 23, no. 14, pp. 2891–2902, 2002.

[6] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special lecture on IE*, vol. 2, no. 1, pp. 1–18, 2015.

[7] X. Xia, X. Pan, N. Li, X. He, L. Ma, X. Zhang, and N. Ding, "Gan-based anomaly detection: A review," *Neurocomputing*, vol. 493, pp. 497–535, 2022.

[8] P. Bergmann, X. Jin, D. Sattlegger, and C. Steger, "The mvtec 3d-ad dataset for unsupervised 3d anomaly detection and localization," *arXiv preprint arXiv:2112.09045*, 2021.

Appendix

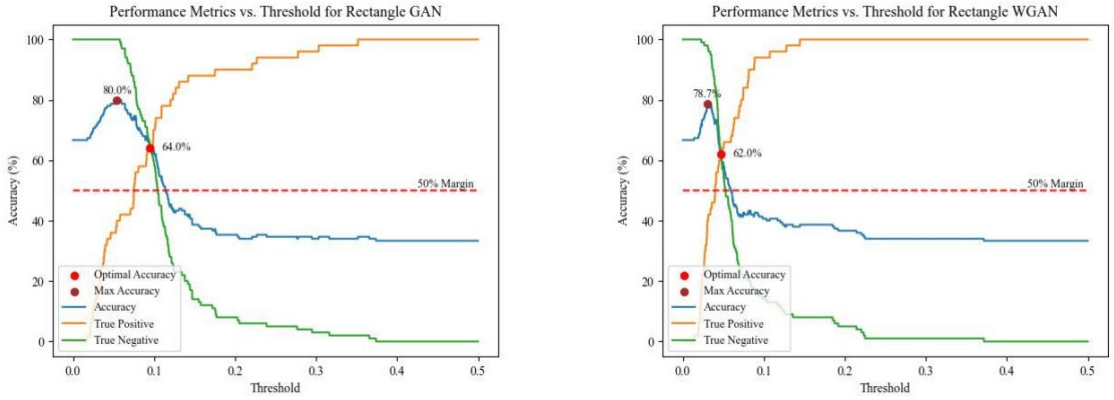


Figure 8: Rectangle generators trained using GAN and WGAN accuracies as a function of threshold

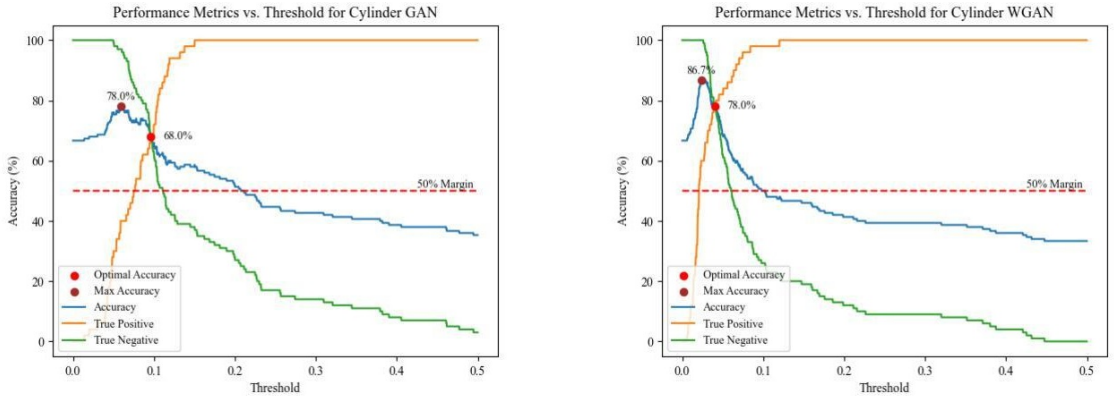


Figure 9: Cylinder generators trained using GAN and WGAN accuracies as a function of threshold

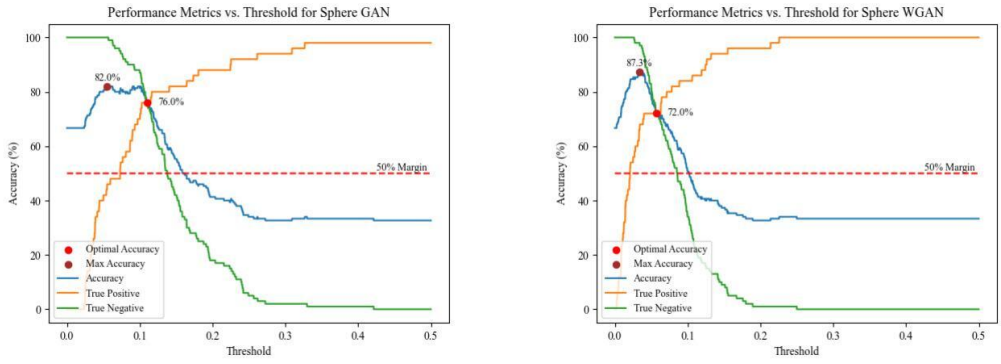
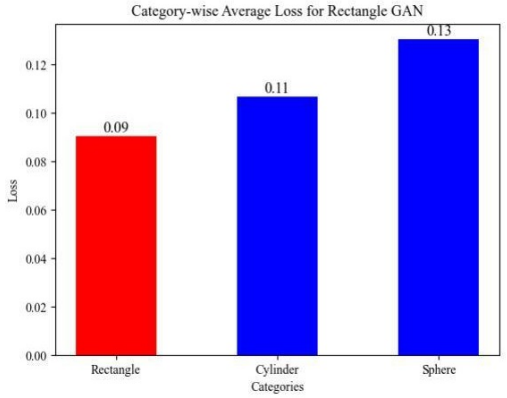


Figure 10: Sphere generators trained using GAN and WGAN accuracies as a function of threshold



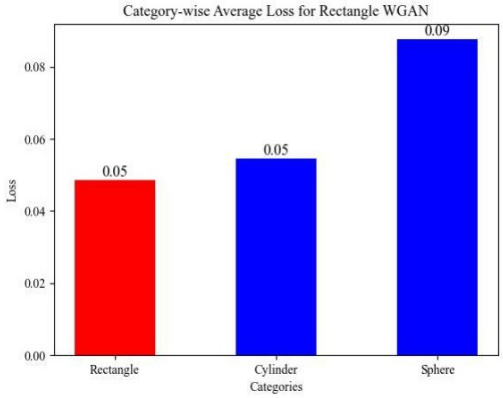
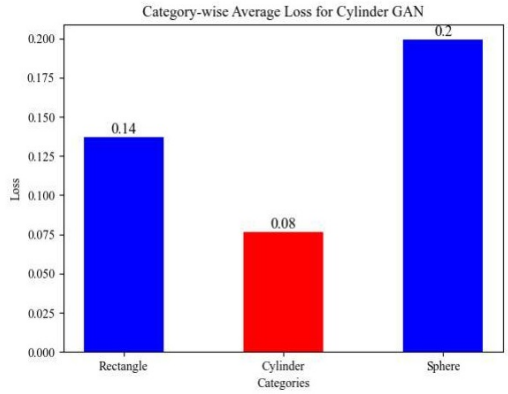


Figure 11: Average loss for data samples of each label from rectangle generators trained using GAN and WGAN



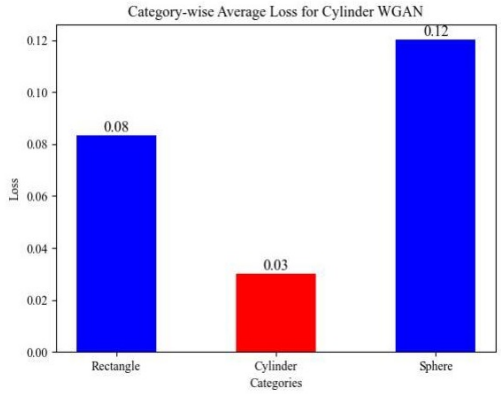
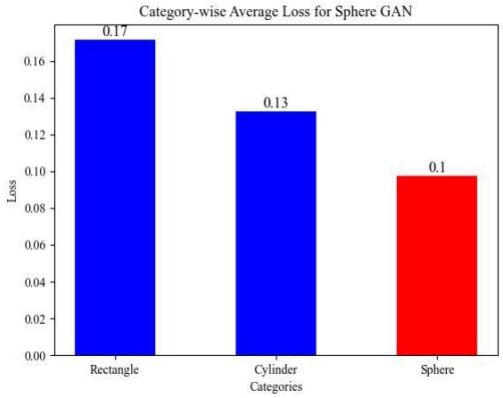


Figure 12: Average loss for data samples of each label from cylinder generators trained using GAN and WGAN



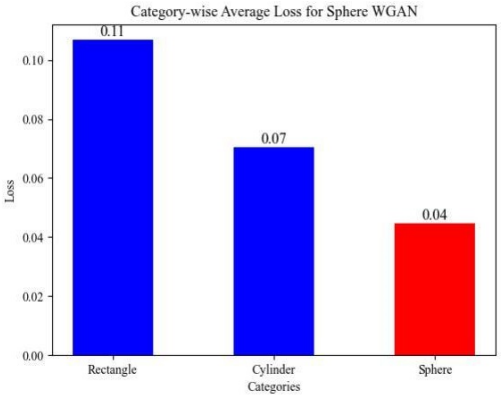


Figure 13: Average loss for data samples of each label from sphere generators trained using GAN and WGAN

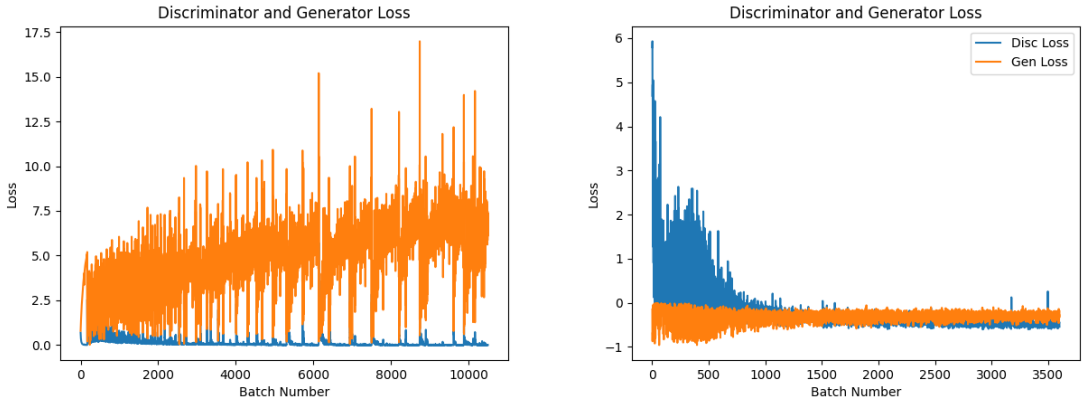


Figure 14: Rectangular GAN and WGAN training loss

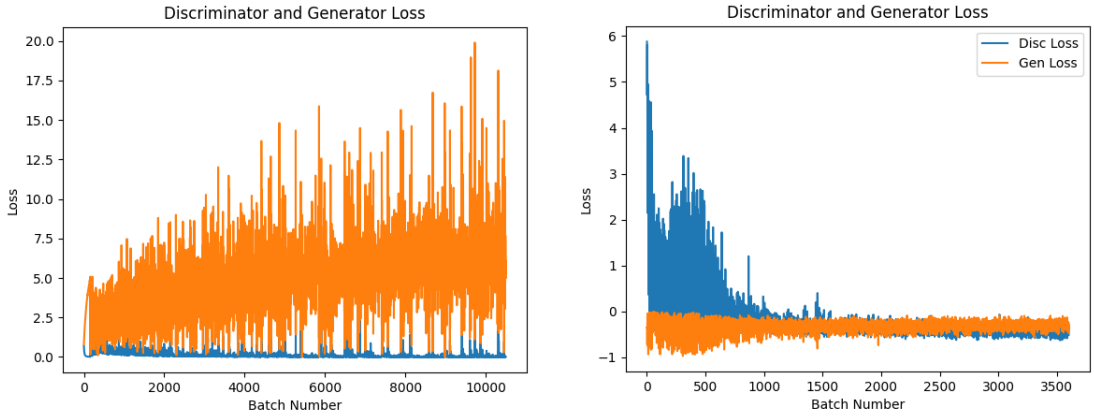


Figure 15: Cylindrical GAN and WGAN training loss

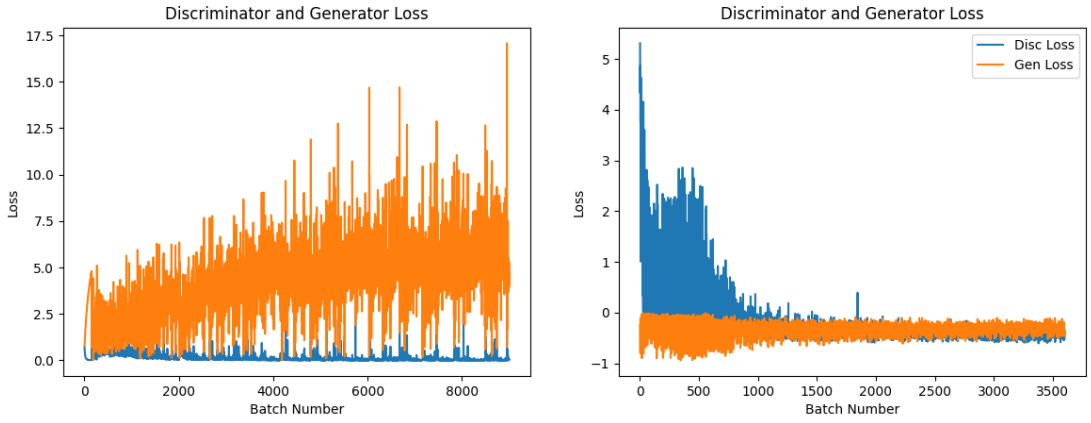


Figure 16: Spherical GAN and WGAN training loss

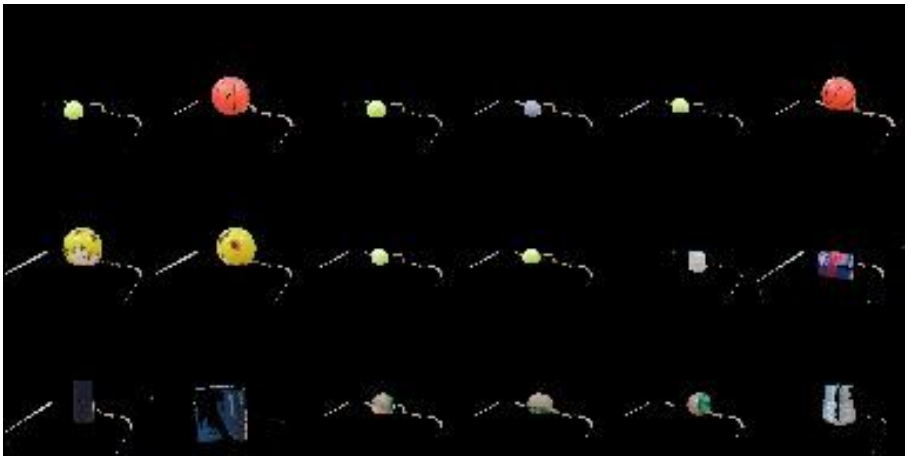


Figure 17: Experiment 2's Spherical GAN least anomalous data

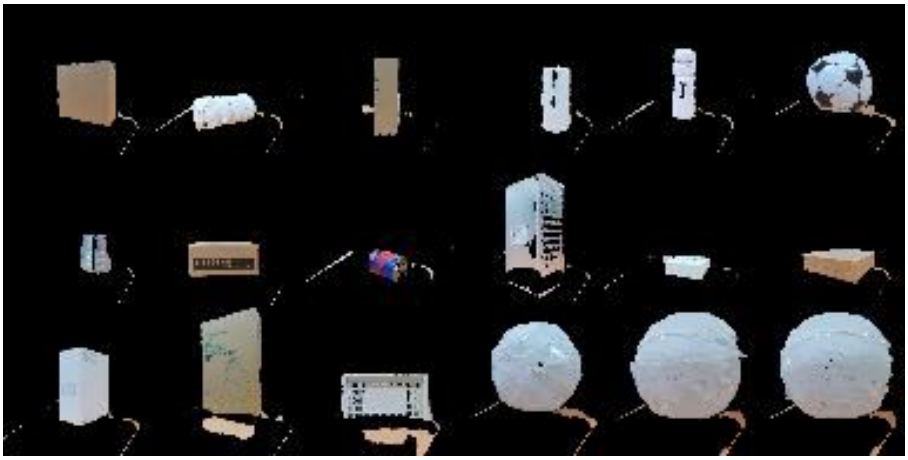


Figure 18: Experiment 2's Spherical GAN most anomalous data

Figure 19: Experiment 2's Spherical WGAN least anomalous data

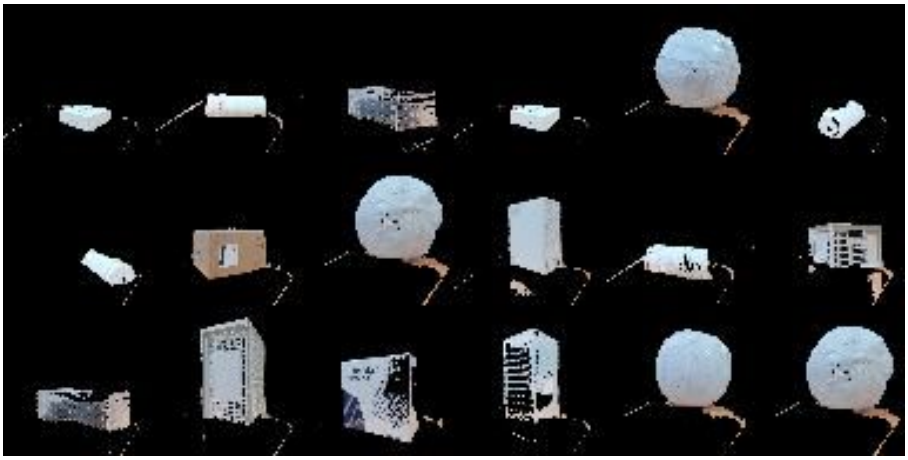


Figure 20: Experiment 2's Spherical WGAN most anomalous data

Figure 21: Experiment 2's Cylindrical GAN least anomalous data

Figure 22: Experiment 2's Cylindrical GAN most anomalous data

Figure 23: Experiment 2's Cylindrical WGAN least anomalous data

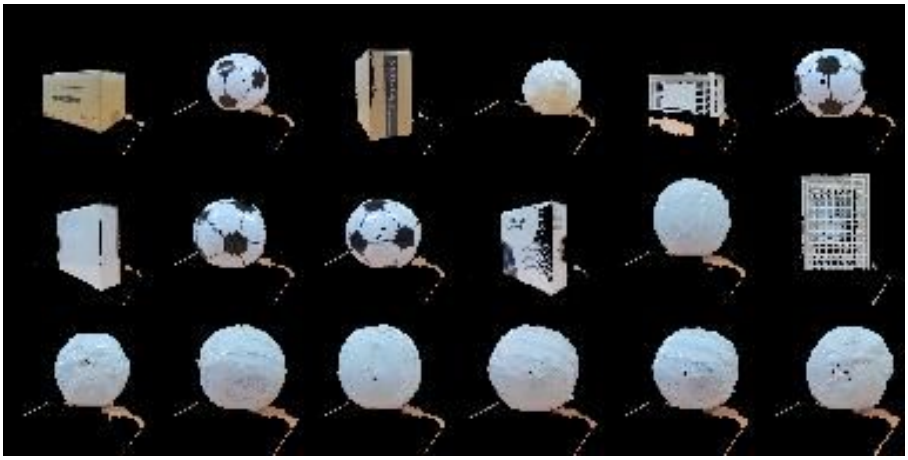


Figure 24: Experiment 2's Cylindrical WGAN most anomalous data

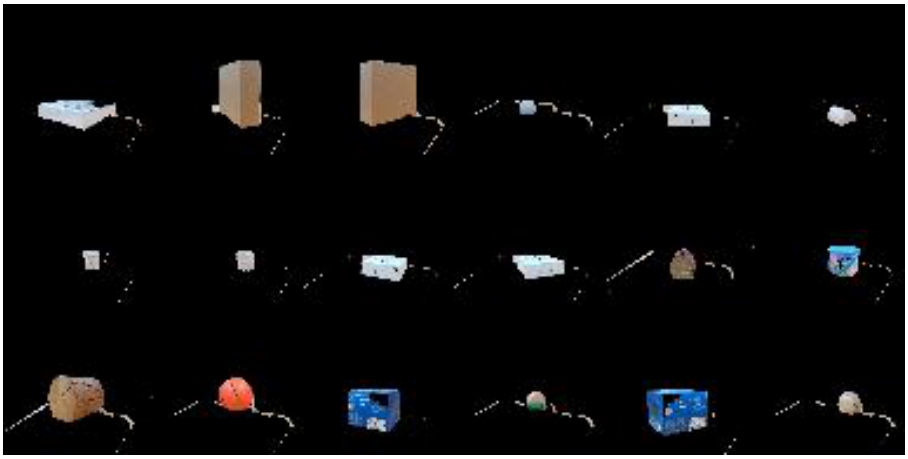


Figure 25: Experiment 2's Rectangular GAN least anomalous data

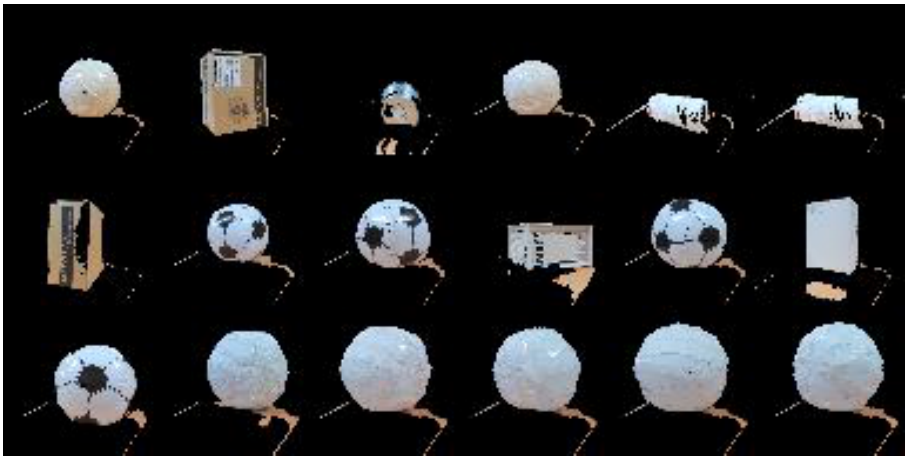


Figure 26: Experiment 2's Rectangular GAN most anomalous data

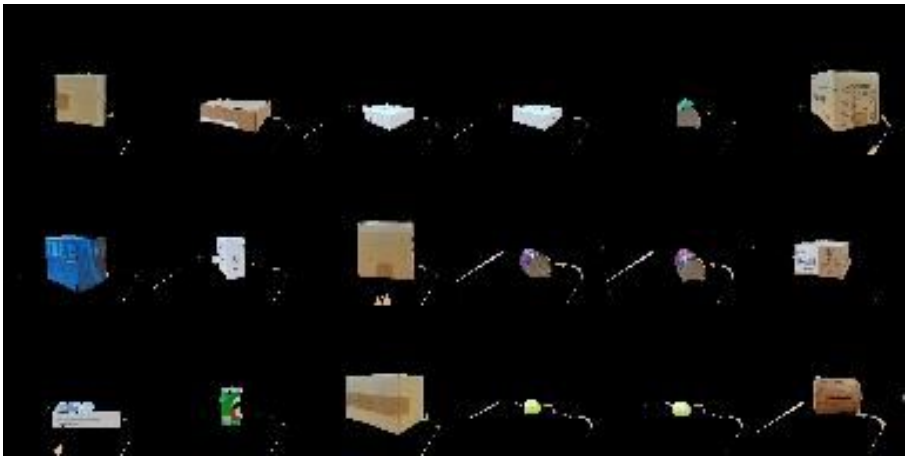


Figure 27: Experiment 2's Rectangular WGAN least anomalous data

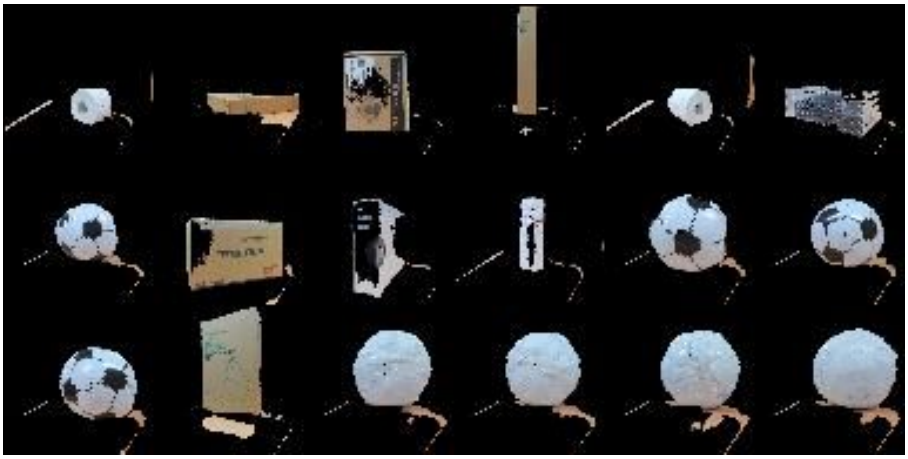


Figure 28: Experiment 2's Rectangular WGAN most anomalous data