Uncracking the Bagel One Crumb at a Time

Stela Topalova Niels Sombekke Ilayda Bilgin Egoitz Gonzalez

Diego Garcia Cerdas Marten Türk

Abstract

The detection of anomalies is a critical problem with implications in various fields. While significant attention has been given to detecting defects in 2D data using convolutional neural networks, the problem in 3D remains relatively unexplored, mostly due to the lack of specific 3D datasets. In this paper, we address the task of anomaly detection in 3D point clouds by leveraging a point cloud-voxel diffusion (PVD) model. Our approach focuses on training the model to learn the data distribution of the healthy samples, introducing noise to the anomalous samples during inference, and subsequently denoising them with our trained model. By comparing the input anomalous point cloud with the reconstructed point cloud, we quantify the difference in a predicted anomaly mask, enabling effective anomaly localization in 3D space. Despite limited data availability, our method demonstrates the ability to achieve decent results.

1. Introduction

The task of detecting defects in anomalous data using computer vision plays a critical role in multiple fields such as manufacturing [14, 15], autonomous vehicles [4, 34], and medical imaging [9, 27]. This task has received a considerable amount of attention in 2D data, where models are typically based on convolutional neural networks (CNNs) and trained on a large number of images. However, for 3D data, this problem is still rather unexplored due to the limited number of datasets to train and test on.

A popular approach to anomaly detection relies on modeling healthy data distribution and tackles the anomalous samples via reconstruction [6, 7, 20]. We follow such a reconstruction-based approach for the task of unsupervised anomaly detection and localization in 3D point clouds. In this work, we employ a Point-Voxel Diffusion (PVD) model [35], which has proven to generate high-fidelity shapes, outperforming multiple state-of-the-art methods. We train a PVD model on healthy samples and, during inference, we apply noise to an anomalous sample to later denoise it with the trained model. Anomaly detection is performed by comparing the anomalous input point cloud to the reconstructed point cloud and predicting an anomaly mask. The use of masks allows us to localize anomalies in 3D space, achieving decent results on a subset of the MVTec 3D-AD dataset [3].

Our main contributions are the following:

- We present a method for anomaly detection in 3D point clouds with a PVD model, only based on the geometry of the samples.
- We show that, despite training on a small amount of data, we are able to obtain decent results for a subset of the MVTec 3D-AD dataset.

2. Related Work

In this section, we delve into the related work that has been conducted in the domains of 3D anomaly detection, point-voxel representation and diffusion-based model. By examining previous studies, experiments, and developments, we aim to establish a comprehensive understanding of the existing body of knowledge. This exploration will serve as a platform for our own research, highlighting its novelty, significance, and potential contributions to the field.

2.1. 3D Anomaly Detection

Anomaly detection methods start with classical approaches for outlier detection such as k-Nearest Neighbors [8]. Traditional handcrafted descriptors like Point Feature Histograms (FPFHs) [24] extract representations from local geometric features to detect deviations from the local surface. These handcrafted descriptors are tailored for specific tasks and are sensitive to noise. Methods based on deep learning try to learn deep representations of the data.

The code can be found at https://github.com/martentyrk/uncracking-the-bagel.

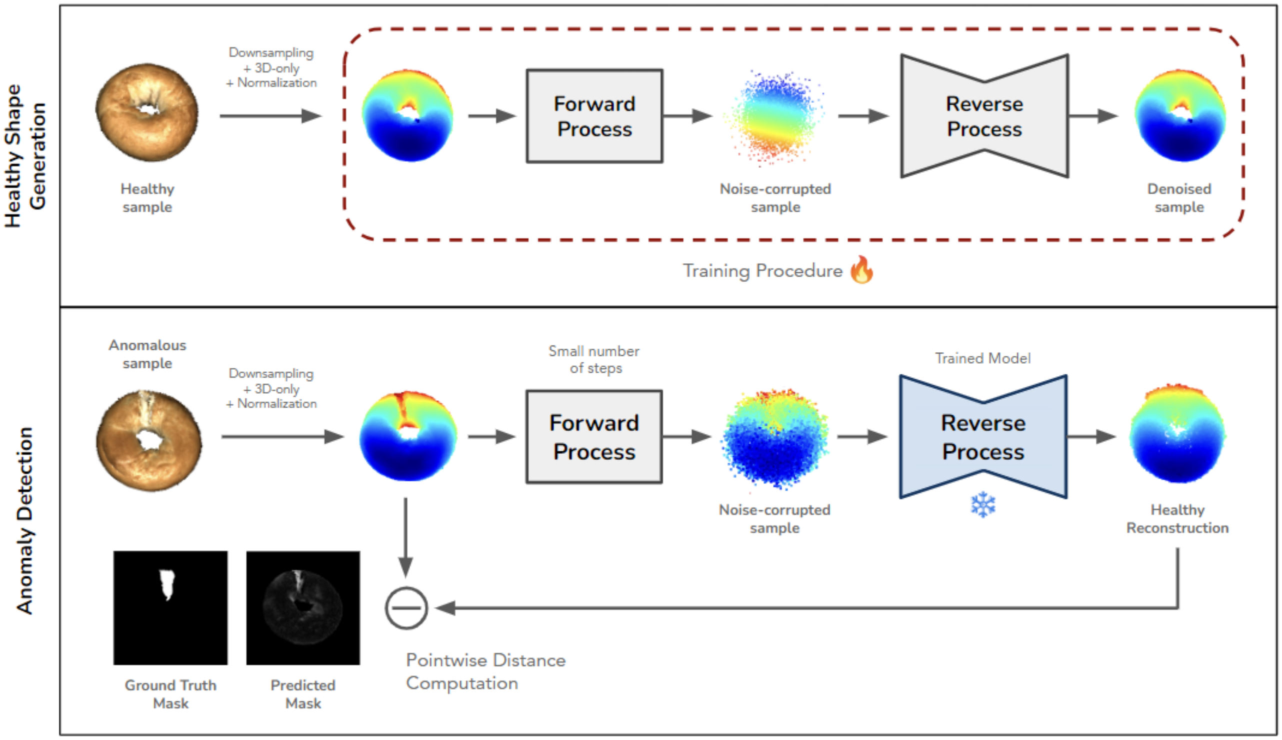

Figure 1. Pipeline overview. The top row corresponds to the training of the Point-Voxel Diffusion (PVD) model on healthy point cloud samples. The bottom row corresponds to the inference and comparison process of the pipeline. Noise is introduced to an anomalous sample and subsequently removed by the trained PVD model transforming it into a healthy reconstruction. A segmentation mask is computed by using the minimum pointwise distances between the anomalous input sample and the corresponding healthy reconstruction.

SpinNet [1] is a neural descriptor for learning deep representations of 3D data. Apart from 3D, anomaly detection has also been applied on 2D image data, using only pixel color information. However, just by looking at color information, it is difficult to determine the underlying geometry of an object. Newer approaches like Voxel f-AnoGAN [28] include generative neural networks to exploit the 3D structure. These methods might also include RGB information to improve their performance on the anomaly detection task. While other descriptor-based methods might get better results for specific tasks [11], we propose a novel approach using diffusion models that we will compare against previous state-of-the-art work in anomaly detection with generative methods [3].

2.2. Point-Voxel Representation

In the past, 3D shapes have been represented as voxel grids and processed using 3D convolution [18, 29, 31]. Furthermore, due to the correlation between 2D pixels and voxels, many works have experimented on voxel-based generation, which has proven to work very well [5, 12, 18, 30]. Even so, volumetric representations are memory-heavy and grow rapidly as dimensions increase, meaning that it is not easy to scale them to higher resolutions.

Point clouds, however, require less memory to process in higher resolutions, while having detailed samples from smooth surfaces. Additionally, most assume point cloud processing networks to be permutation invariant [19,21,33], which is a difficult requirement to pose on a model architecture since the 3D point cloud representation is unordered. Furthermore, [35] notes that applying 2D methods without modifications to permutation-invariant point clouds or voxels does not lead to desirable outcomes.

Thus, the model of our choice explores a separate point-voxel 3D representation [13, 25]. In this work we rely on an architecture proposed by [35], which uses the point-voxel CNN model [16] as its backbone. Point-voxel CNN introduces a methodology to voxelize point cloud data for 3D convolution and has shown to take advantage of the spatial correlation inherent in point cloud data.

2.3. Diffusion-Based Models

Diffusion-based models are a special type of latent variable model that treats generation as an iterative refinement procedure [26]. More specifically, they can be dissected into a *forward diffusion process* that gradually perturbs the data until all signal is lost, and a *reverse diffusion process*, which is parameterized by a neural network that learns to invert the perturbation process. Different formulations have been proposed [32]. Among them, Denoising Diffusion Probabilistic Models (DDPMs) [10] have recently gained significant attention for their ability to generate diverse, high-resolution samples with exceptional fidelity [22, 23].

There has been previous work on using DDPMs in 3D space for generating point clouds and modeling the point distribution. An example to that can be viewing point clouds as a conditional generation problem, inspired by the diffusion process in thermodynamics [17]. In the approaches where diffusion is applied directly on 3D voxel representations or point clouds, a high generation quality has not been achieved. To address the limitations of previous models in the 3D space, PVD was proposed as a combination of denoising diffusion models and point-voxel representation of 3D shapes [35]. This will be further explained in section 3.1.

3. Methodology

We employ PVD [35] for the task of anomaly detection. Figure 1 displays an overview of our pipeline.

We first train the model for shape generation on anomaly-free samples (Section 3.1). Then, we apply noise to an anomalous sample and use the trained model to denoise it, creating a healthy reconstruction of the original point cloud. Finally, we predict an anomaly mask by computing the distance between the original point cloud and its healthy reconstruction (Section 3.2).

3.1. PVD: Healthy Shape Generation

PVD uses 3D point clouds, represented as a data point $\mathbf{x} \in \mathbb{R}^{N\times 3}$, where N is the number of points and each point has xyz-coordinates. In order to model the distribution of healthy data, we restrict the training dataset to anomaly-free point clouds.

The *forward diffusion process* gradually injects Gaussian noise into the data until all signal is removed. For a finite number of timesteps T, the signal removal is defined by a probabilistic Markov chain:

$$q(\mathbf{x}_{0:T}) = q(\mathbf{x}_0)\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) = q(\mathbf{x}_0)q(\mathbf{x}_{1:T}|\mathbf{x}_0),$$

where $q(\mathbf{x}_0)$ is the data distribution and $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ defines a noise corruption step.

We invert the noise corruption process via a *reverse diffusion process*. Since the true denoising distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is unknown, we employ a probabilistic Markov chain:

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t),$$

where $p(\mathbf{x}_T)$ is a standard Gaussian prior and $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ defines a denoising step we learn from the data. The joint distribution $p_\theta(\mathbf{x}_{0:T})$ defines our *generative model*.

Parameterizing the transition probabilities $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ and $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ as Gaussian distributions enables closed form evaluation:

$$\begin{aligned} q(\mathbf{x}_t|\mathbf{x}_{t-1}) &:= \mathcal{N}(\sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \\ p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) &:= \mathcal{N}(\boldsymbol{\mu}_\theta(\mathbf{x_t}), \sigma_t^2\mathbf{I}), \end{aligned}$$

where $\beta_1, ..., \beta_T$ define the noise variance schedule and $\boldsymbol{\mu}_\theta(\mathbf{x_t})$ is the predicted shape of our generative model at timestep $t-1$. We use $\sigma_t^2 = \beta_t$ as a general heuristic.

The marginal distribution:

$$p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}_{0:T})d\mathbf{x}_{1:T}$$

allows us to generate new data points by sequentially denoising a sample obtained from the noise prior $p(\mathbf{x}_T)$. We can learn $p_\theta(\mathbf{x})$ by maximizing a variational lower bound of the log data distribution:

$$\mathbb{E}_{q(\mathbf{x}_0)}\left[\log p_\theta(\mathbf{x}_0)\right] \geq \mathbb{E}_{q(\mathbf{x}_{0:T})} \log\left[\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right].$$

Given our assumptions, the training objective reduces to maximum likelihood estimation:

$$\max_\theta \mathbb{E}_{\mathbf{x}_0\sim q(\mathbf{x}_0), \mathbf{x}_{1:T}\sim q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\sum_{t=1}^{T}\log p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)\right].$$

In practice, $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is parameterized by a single point-voxel CNN [16] that predicts a noise value $\boldsymbol{\epsilon}_\theta(\mathbf{x}, t)$ for a given timestep t. Finally, the loss is an $\mathcal{L}_2$ loss between $\boldsymbol{\epsilon}_\theta(\mathbf{x}, t)$ and the true noise $\boldsymbol{\epsilon}$:

$$\mathcal{L}_t = \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}, t)\|^2, \text{ with } \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}).$$

A healthy sample can be generated by sequentially sampling from $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ for $t = T, ..., 1$ via

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\tilde{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}, t)\right) + \sqrt{\beta_t}\mathbf{z},$$

where $\alpha_t = 1 - \beta_t$, $\tilde{\alpha}_t = \prod_{s=1}^{t}\alpha_s$, and $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$.

As we have now defined the process of generation, the next step in the pipeline is using it to detect anomalies.

3.2. Anomaly Detection

We break the process of anomaly detection down into two separate steps: healthy reconstruction and prediction of anomaly masks. In the first step, we leverage the model trained on healthy data to create a healthy reconstruction $\hat{\mathbf{x}}_H$ of an anomalous sample $\mathbf{x}_A$. We first apply N_{noise}

noise corruption steps to $\mathbf{x}_A$ and later use the PVD model to denoise it for the same number of steps to retrieve $\hat{\mathbf{x}}_H$, which will be reconstructed as a healthy sample, without the anomalous area.

In order to predict and identify anomalies, we look for the regions where the differences between the input anomalous point cloud and the healthy reconstruction are the biggest. More specifically, for each point in $\mathbf{x}_A$, we compute the distance to the closest point in $\hat{\mathbf{x}}_H$. Then, we map this value in our predicted mask P to the pixel-wise location of the point in $\mathbf{x}_A$. This allows for a direct comparison to the ground-truth anomaly mask and avoids having to render the mask from camera parameters. The use of soft (non-binary) masks also introduces uncertainty about our prediction and allows for thresholding during performance measurement.

4. Experiments

We give an overview of the dataset and our experimental setup in sections 4.1, 4.2 and 4.3. From 4.4 and onwards we focus on the results of the experiments and show how our results compare to other related methods. We found that the most comparable models are Voxel- GAN, AE, and VM as well as Depth- GAN, AE and VM introduced in [3] as these models are also generative and make their predictions solely on 3D data, without any extra conditioning.

4.1. Dataset

All experiments were run on the MVTec 3D-AD Dataset [3], which is specifically meant for 3D anomaly detection and localization. The dataset includes 10 different categories such as bagels, carrots, rope, cookies, etc., which have been split into train, validation, and test, where all training and validation samples are healthy. The test set however, contains both healthy and anomalous data. The training and validation data include TIFF images for the 3D coordinates and RGB images to represent the color of the object. In addition to the coordinates and RGB representations, the test set also includes binary ground truth masks for the anomalous areas.

In the interest of time and complexity, we solely focus on the bagel class, which includes 244 healthy training samples and 22 healthy validation samples. The test set contains a total of 110 3D objects, out of which 22 are non-anomalous. In order to start training we first need to preprocess the whole dataset.

4.1.1 Preprocessing

We perform a 3D-based preprocessing protocol to remove artifacts in the input TIFF files [11]. First, RANSAC is applied on the point cloud to remove the background plane. Then, outliers are removed via a connected-component-based algorithm. The resulting point cloud contains a large number of zero points resulting from the original background pixels. To counteract their effect on training, we remove them from the input data. We compute the mean and standard deviation of xyz-coordinates on the training and validation data, and perform normalization on the input point clouds.

4.2. Experimental setup

Our best-performing model has been trained for 2500 epochs with a learning rate of 2e-4 and a batch size of 16 on an Nvidia Titan RTX GPU, which took ∼24h to complete. During training we randomly sample 10k points from each input point cloud and add noise for 1000 timesteps, which is later denoised for the same amount of steps. During inference, we add noise for 50 timesteps and then denoise the 3D object from that.

4.3. Performance Metric

We follow the standard evaluation procedure for MVTec 3D-AD and compute the per-region overlap (PRO) metric [2], defined as the average relative overlap of a binary prediction P_B with each ground truth connected component C_k:

$$\text{PRO} = \frac{1}{K}\sum_{k=1}^{K}\frac{|P_B \cap C_k|}{|C_k|},$$

where K is the total number of ground truth connected components.

We build the binary prediction P_B from our soft mask P via binary thresholding. The process is repeated for multiple equidistant thresholds and a curve is built by plotting the PRO values against the corresponding false positive rates [3].

The final AU-PRO metric is computed by integrating the curve up to some false positive rate l_{FPR} and normalizing the area to the interval $[0, 1]$.

4.4. Quantitative Results

To quantitatively evaluate our model we calculate the AU-PRO metric explained in 4.3. Figure 2 displays our experiments when different amounts of noise were applied to the input during inference. We found that applying noise for 50 timesteps achieved the highest AU-PRO score. This is reasonable since applying more would destroy the original point cloud, whereas applying less does not add a sufficient amount of noise for the model to reconstruct a healthy sample. Furthermore, Table 1 shows how our results compare to those of the closest methods, with our method outperforming all of them.

4.5. Qualitative results

Our qualitative results are mainly composed of observing the masks we obtain from inference and the denoised

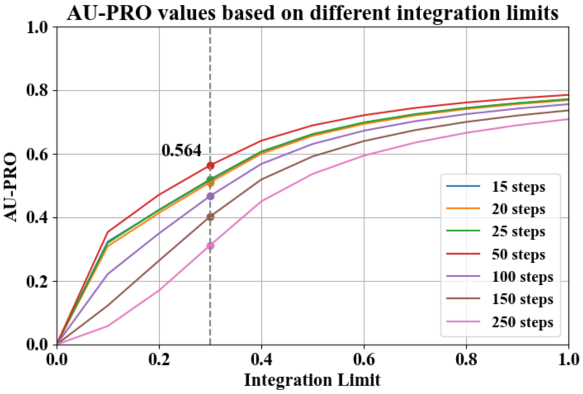

Figure 2. Our results, AU-PRO scores per integration limit for different amounts of noise applied to the input during inference.

Method		AU-PRO Value
Voxel	GAN	0.440
	AE	0.260
	VM	0.453
Depth	GAN	0.111
	AE	0.147
	VM	0.280
	Ours	**0.564**

Table 1. Quantitative results between the baselines of the original MVTec paper [3] and our results at integration limit = 0.3 for the category bagel.

samples of anomalous inputs. As for the masks, we found that other than the anomalous areas, we also see high distance values near the center of the bagel. The reason for this is most likely the fact that the centers of all bagels following a random shape distribution, making it hard for the model to perfectly depict that part.

The overall reconstructions seem to fix the anomalous areas well as can be seen from Figure 3, although comparing inputs to outputs, it can be observed that the surfaces of the outputs are not as smooth and there are still some points around the surface area. This could be the consequence of not enough training time or scarcity of data.

We also found that by increasing the number of noise time steps, the distances between input and output increase all over the sample, meaning that too much noise destroys the original shape of the input and generates a completely new instance, which is not desirable. In Figure 4 we see that choosing 50 steps gives us the less noisy results and 250, on the other hand, the anomalous region gets shadowed by false anomaly detections in other regions that appeared due to the perturbation of the overall shape.

5. Discussion

Despite the promising results we achieved for our novel method for anomaly detection, it is important to acknowledge several limitations and challenges associated with this approach:

- Deep diffusion models often require significant computational resources during training and inference. Other 3D descriptor methods such as FPFH [24] are simpler in exchange for flexibility.
- Diffusion models typically rely on a significant amount of training data, but the datasets for anomaly detection are limited. Our method requires the model to be trained to reconstruct healthy samples of an object to then detect defects in anomalous samples. Thus, the model needs to learn from healthy data, which might not always be available. However, we could observe that our method is good enough to reconstruct samples despite having trained on such a small amount of data.
- The model we chose is dependent on the hyperparameters chosen. We opted for a grid sampling of the points in the surface of the 3D shape and found that random sampling gave worse results, probably caused by the lack of control in the sampling density. The performance of the model in reconstruction and anomaly detection is also sensitive to the number of denoising steps chosen for inference time, as seen in Figure 2.
- Since the quality of the generations is not perfect, the denoising process does introduce changes in areas of the object that are not part of the anomalous area. This makes the precise localization of the anomaly in or on the object extremely difficult, since thresholding distance values might also affect our predictions near the anomalous area. Furthermore, the quantitative results we obtained are very heavily influenced by the nature of the PRO metric, which only looks at the area where the anomaly is originally located.

For the future work, conditioning on RGB data with the current implementation could be explored. This could aid in the localization of the anomaly or extending the use of our diffusion model. The denoising model could be trained to generate multiple samples of different classes at once and could further be conditioned on multi-modal input data so that the reconstruction would be better, such as an image of the 3D scene containing the sample to be reconstructed.

6. Conclusion

Our method leverages the generative power of flexible deep diffusion models to tackle the downstream task of anomaly detection. We outperform previous generative

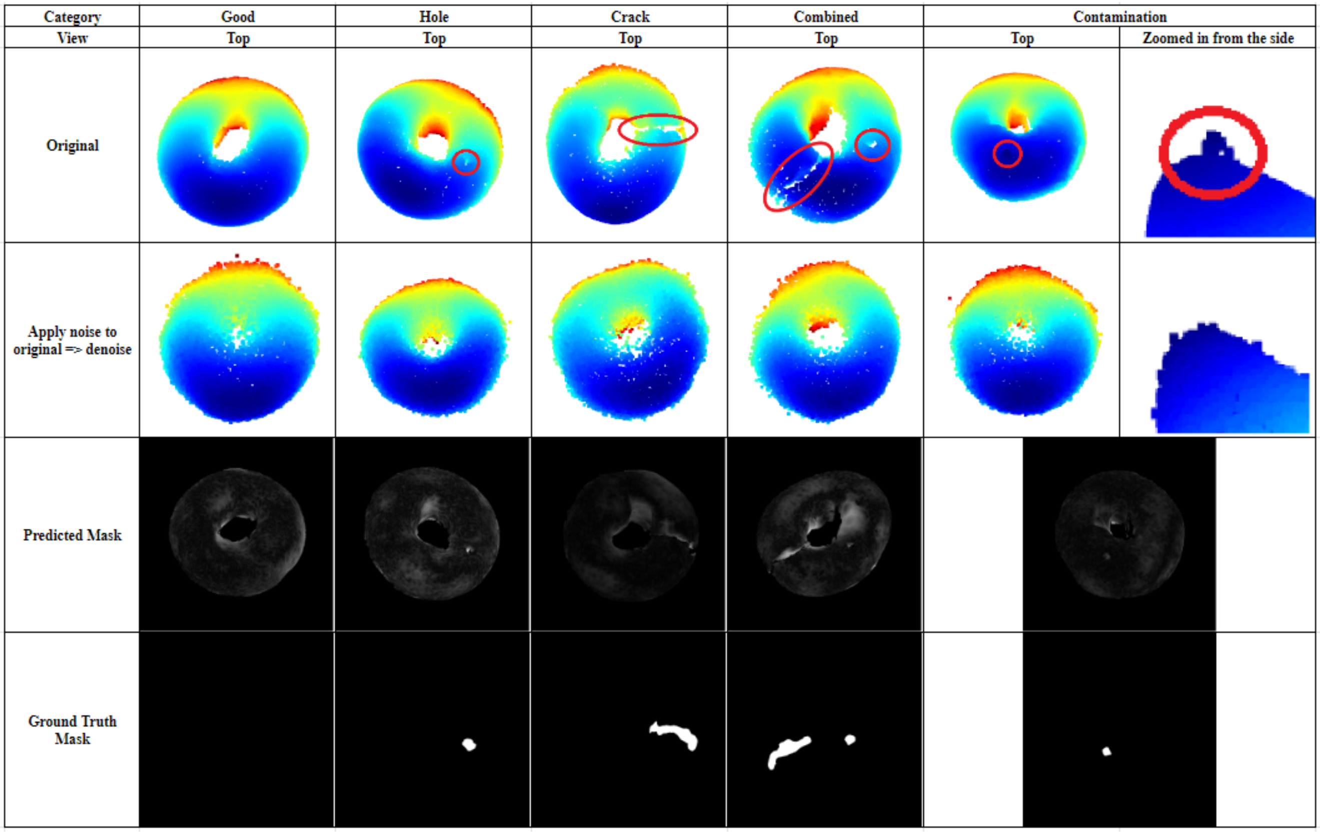

Figure 3. Qualitative Results for a healthy point cloud and each category of anomalous point cloud using 50 noise steps. The top row corresponds to the original point clouds, the middle row corresponds to the point clouds after inference and the bottom row corresponds to the computed segmentation mask.

model approaches while only relying on geometric data. Our model performs well even if it has only been trained on a small dataset. We set a starting point in the application of diffusion models for 3D anomaly detection tasks.

7. Workload division and difficulties

The starting point of the project was to analyze shape generation using neural implicit representations, i.e. SDF-encoded mesh representations learned via neural networks and generation using diffusion. We split the team into two groups, one focused on reproducing SDFusion results and the other group on preprocessing DiffusionSDF and trying to run it. Our intention was to have baselines for our own extension. After having some difficulties with DiffusionSDF and better ideas for the extension, we chose to work on anomaly detection, also using diffusion for generation. We decided to use the PVD model and tested it on ShapeNet and MVTec for 3D anomaly detection on the bagels class. We split the tasks for reproducing PVD generation on ShapeNet, preprocessing bagels data and setting up the environment and debugging dependency issues. After getting good ShapeNet chair generations, we ran it with bagels. Since the generations were not as expected, we had to debug an error in preprocessing. Once we observed the results, we decided to increase the number of points sampled for each model in order to improve generation quality. After generation was tested, we split the work into two groups again. Half of us were working on the inference pipeline while the rest focused on metric and segmented mask generation for evaluation. Finally, we also split the tasks of poster and report, despite all of us contributing to the final paper/report. During the project we used agile development methodology to split and parallelized teamwork as well as the Notion app for task logging and documentation, specially focused on easing the work within the team.

References

[1] Sheng Ao, Qingyong Hu, Bo Yang, Andrew Markham, and Yulan Guo. Spinnet: Learning a general surface descriptor for 3d point cloud registration, 2021. 2

[2] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. MVTec AD – a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings*

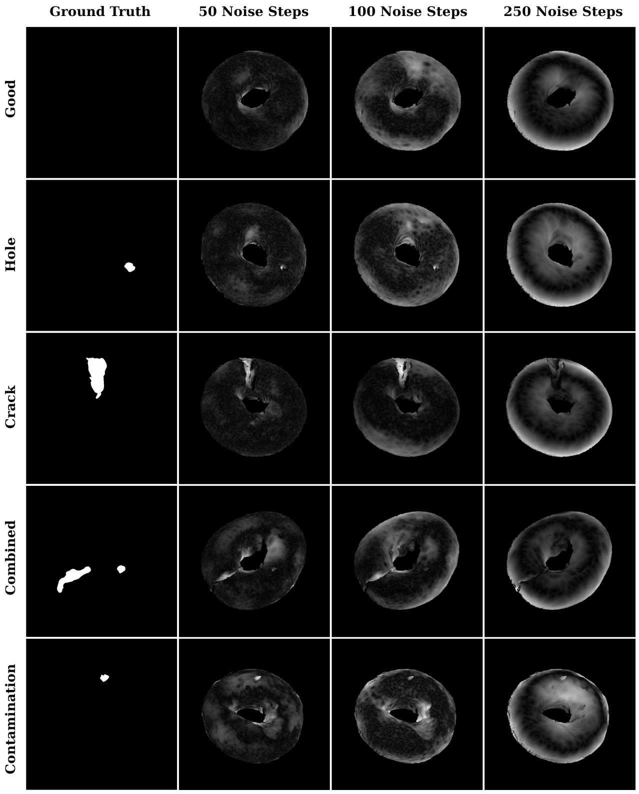

Figure 4. Predicted masks for different noise levels. From left to right, ground truth mask, and predictions for 50, 100 and 250 noise steps. In each row, we show an example of each of the anomaly classes, where good is non anomalous data.

of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9592–9600, 2019. 4

[3] Paul Bergmann, Xin Jin, David Sattlegger, and Carsten Steger. The MVTec 3d-AD dataset for unsupervised 3d anomaly detection and localization. In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SCITEPRESS - Science and Technology Publications, 2022. 1, 2, 4, 5

[4] Daniel Bogdoll, Maximilian Nitsche, and J Marius Zöllner. Anomaly detection in autonomous driving: A survey. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4488–4499, 2022. 1

[5] Andrew Brock, Theodore Lim, J. M. Ritchie, and Nick Weston. Generative and discriminative voxel modeling with convolutional neural networks, 2016. 2

[6] Hanqiu Deng and Xingyu Li. Anomaly detection via reverse distillation from one-class embedding. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, page 9737–9746, 2022. 1

[7] Vuong Le Budhaditya Saha Moussa Reda Mansour Svetha Venkatesh Dong Gong, Lingqiao Liu and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, page 1705–1714, 2019. 1

[8] Eleazar Eskin, Andrew Arnold, Michael Prerau, Leonid Portnoy, and Sal Stolfo. *A Geometric Framework for Unsupervised Anomaly Detection*, pages 77–101. Springer US, Boston, MA, 2002. 1

[9] Tharindu Fernando, Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. Deep learning for medical anomaly detection–a survey. *ACM Computing Surveys (CSUR)*, 54(7):1–37, 2021. 1

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 3

[11] Eliahu Horwitz and Yedid Hoshen. Back to the feature: Classical 3d features are (almost) all you need for 3d anomaly detection, 2022. 2, 4

[12] Truc Le and Ye Duan. Pointgrid: A deep network for 3d shape understanding. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9204–9214, 2018. 2

[13] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on $\mathcal{X}$-transformed points, 2018. 2

[14] Benjamin Lindemann, Benjamin Maschler, Nada Sahlab, and Michael Weyrich. A survey on anomaly detection for technical systems using lstm networks. *Computers in Industry*, 131:103498, 2021. 1

[15] Jiaqi Liu, Guoyang Xie, Jingbao Wang, Shangnian Li, Chengjie Wang, Feng Zheng, and Yaochu Jin. Deep visual anomaly detection in industrial manufacturing: A survey. *arXiv preprint arXiv:2301.11514*, 2023. 1

[16] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning, 2019. 2, 3

[17] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2837–2845, 2021. 3

[18] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, 2015. 2

[19] Junier B. Oliva, Avinava Dubey, Manzil Zaheer, Barnabás Póczos, Ruslan Salakhutdinov, Eric P. Xing, and Jeff Schneider. Transformation autoregressive networks, 2018. 2

[20] Ramesh Nallapati Pramuditha Perera and Bing Xiang. Ocgan: One-class novelty detection using gans with constrained latent representations. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, page 2898–2906, 2019. 1

[21] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation, 2017. 2

[22] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 3

[23] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 3

[24] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE International Conference on Robotics and Automation*, pages 3212–3217, 2009. 1, 5

[25] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection, 2021. 2

[26] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 2

[27] Maximilian E Tschuchnig and Michael Gadermayr. Anomaly detection in medical imaging-a mini review. In *Data Science–Analytics and Applications: Proceedings of the 4th International Data Science Conference–iDSC2021*, pages 33–38. Springer, 2022. 1

[28] Jaime Simarro Viana, Ezequiel de la Rosa, Thijs Vande Vyvere, David Robben, Diana M. Sima, CENTER-TBI Participants, and Investigators. Unsupervised 3d brain anomaly detection. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pages 133–142. Springer International Publishing, 2021. 2

[29] Zongji Wang and Feng Lu. Voxsegnet: Volumetric cnns for semantic part segmentation of 3d shapes, 2018. 2

[30] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T Freeman, and Joshua B Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016. 2

[31] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, 2015. 2

[32] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Yingxia Shao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *arXiv preprint arXiv:2209.00796*, 2022. 3

[33] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep sets, 2018. 2

[34] Mingming Zhang, Chao Chen, Tianyu Wo, Tao Xie, Md Zakirul Alam Bhuiyan, and Xuelian Lin. Safedrive: Online driving anomaly detection from large-scale vehicle data. *IEEE Transactions on Industrial Informatics*, 13(4):2087–2096, 2017. 1

[35] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5826–5835, 2021. 1, 2, 3