Anomaly Detection in 3D Point Clouds using Deep Geometric Descriptors

Paul Bergmann
MVTec Software GmbH
Technical University of Munich
paul.bergmann@mvtec.com

David Sattlegger
MVTec Software GmbH
sattlegger@mvtec.com

Abstract

We present a new method for the unsupervised detection of geometric anomalies in high-resolution 3D point clouds. In particular, we propose an adaptation of the established student-teacher anomaly detection framework to three dimensions. A student network is trained to match the output of a pretrained teacher network on anomaly-free point clouds. When applied to test data, regression errors between the teacher and the student allow reliable localization of anomalous structures. To construct an expressive teacher network that extracts dense local geometric descriptors, we introduce a novel self-supervised pretraining strategy. The teacher is trained by reconstructing local receptive fields and does not require annotations. Extensive experiments on the comprehensive MVTec 3D Anomaly Detection dataset highlight the effectiveness of our approach, which outperforms the existing methods by a large margin. Ablation studies show that our approach meets the requirements of practical applications regarding performance, runtime, and memory consumption.

1. Introduction

We address the challenging task of unsupervised anomaly detection and localization in 3D point clouds. The goal is to detect data points that deviate significantly from a training set of exclusively anomaly-free samples. This problem has important applications in various fields, such as industrial inspection [7, 9, 14, 45], autonomous driving [11, 26], and medical imaging [2, 4, 33]. It has received considerable attention in 2D, where models are typically trained on color or grayscale images with established and well-studied architectures based on convolutional neural networks. In 3D, this problem is still comparatively unexplored and only a small number of methods exist. In this work, we follow the practice in other areas of computer vision and take inspiration from recent advances in 2D anomaly detection to devise a powerful 3D method.

More specifically, we build on the success of using descriptors from pretrained neural networks for unsupervised anomaly detection. An established protocol is to extract these descriptors as intermediate features from networks trained on the ImageNet [31] dataset. Models based on pretrained features were shown to perform better than ones trained with random weight initializations [8, 12, 17]. In particular, they outperform methods based on convolutional autoencoders or generative adversarial networks.

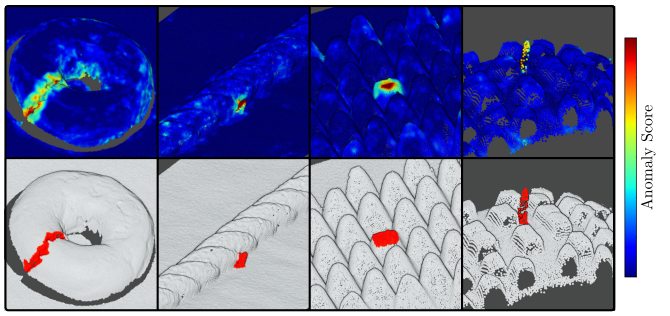

Figure 1: Qualitative results of our proposed 3D–ST method on the MVTec 3D Anomaly Detection dataset. It reliably localizes geometric anomalies in test point clouds, although it is trained only on anomaly-free samples. Top row: Anomaly scores for each 3D point predicted by our algorithm. Bottom row: Ground truth annotations of anomalous points in red.

To date, there is no established pretraining protocol for unsupervised anomaly detection in 3D point clouds. Existing work addresses the extraction of local 3D features that are highly task-specific. For point cloud registration, feature extractors often heavily downsample the input data or operate only on a small number of input points. This makes them ill-suited for anomaly localization in 3D. In this work, we develop a novel approach for pretraining local geometric descriptors that transfer well to the anomaly detection problem. We then use this pretraining strategy to introduce a new method that outperforms existing approaches in the localization of geometric anomalies in high-resolution 3D point clouds. In particular, our key contributions are:

- We present 3D Student-Teacher (3D-ST), the first method for unsupervised anomaly detection that operates directly on 3D point clouds. Our method is trained only on anomaly-free data and it localizes geometric anomalies in high-resolution test samples with a single forward pass. We propose an adaptation of the well-established student-teacher framework for anomaly detection to three dimensions. A student network is trained to match deep local geometric descriptors of a pretrained teacher network. During inference, anomaly scores are derived from the regression errors between the student's predictions and the teacher's targets. Our method sets a new state of the art on the recently introduced MVTec 3D-AD dataset. It performs significantly better than existing methods which use voxel grids and depth images.

- We develop a self-supervised training protocol that allows the teacher network to learn generic local geometric descriptors that transfer well to the 3D anomaly detection task. The teacher extracts a geometric descriptor for each input point by aggregating local features within a limited receptive field. A decoder network is trained to reconstruct the local geometry encoded by the descriptors. Our pretraining strategy provides explicit control over the receptive field and dense feature extraction for a large number of input points. This allows us to compute anomaly scores for high-resolution point clouds without the need for intermediate subsampling.

2. Related Work

Our work touches on several aspects of computer vision, namely unsupervised detection of anomalies in two and three dimensions and extraction of deep local geometric descriptors for 3D data.

2.1. Anomaly Detection in 2D

There is a large body of work on the unsupervised detection of anomalies in two dimensions, i.e., in RGB or grayscale images. Ehret et al. [22] and Pang et al. [36] give comprehensive overviews. Some of the existing methods are trained from scratch with random weight initialization, in particular, those based on convolutional autoencoders (AEs) [10, 27, 32, 47, 48] or generative adversarial networks (GANs) [13, 38, 43].

A different class of methods leverage descriptors from pretrained networks for anomaly detection [8, 17, 18, 25, 34, 39, 40]. The key idea behind these approaches is that anomalous regions produce descriptors that differ from the ones without anomalies. These methods tend to perform better than methods trained from scratch, which motivates us to transfer this idea to the 3D domain.

Bergmann et al. [8] propose a student-teacher framework for 2D anomaly detection. A teacher network is pretrained on the ImageNet dataset to output descriptors represented by feature maps. Each descriptor captures the content of a local region within the input image. For anomaly detection, an ensemble of student networks is trained on anomaly-free images to reproduce the descriptors of the pretrained teacher. During inference, anomalies are detected when the students produce increased regression errors and predictive variances. Closely following this idea, Salehi et al. [41] train a single student network to match multiple feature maps of a single teacher.

2.2. Anomaly Detection in 3D

To date, there are very few methods that address the task of unsupervised anomaly detection in 3D data. None of them leverages the descriptiveness of feature vectors from pretrained networks.

Viana et al. [44] propose Voxel f-AnoGAN, which is an extension of the 2D f-AnoGAN model [43] to 3D voxel grids. A GAN is trained on anomaly-free data samples. Afterwards, an encoder is trained to predict the latent vectors of anomaly-free voxel grids that, when passed through the generator network, reconstruct the input data. During inference, anomaly scores are derived by a per-voxel comparison of the input to the reconstruction. Bengs et al. [6] introduce a method based on convolutional autoencoders that also operates on 3D voxel grids. A variational autoencoder is trained to reconstruct input samples through a low-dimensional bottleneck. Again, anomaly scores are derived by comparing each voxel element of the input to its reconstruction.

Recently, Bergmann et al. introduced MVTec 3D-AD [9], a comprehensive dataset for the evaluation of 3D anomaly detection algorithms. So far, this is the only public dataset specifically designed for this task. They show that the existing methods do not perform well on challenging high-resolution point clouds and that there is a need for the development of new methods for this task.

2.3. Learning Deep 3D Descriptors

Geometric feature extraction is commonly used in 3D applications such as 3D registration or 3D pose estimation. The community has recently shifted from designing handcrafted descriptors [42, 46] to learning-based approaches.

One line of work learns low-dimensional descriptors on local 3D patches cropped from larger input point clouds. In 3DMatch [51] and PPFNet [20], supervised metric learning is used to learn embeddings from annotated 3D correspondences. PPF-FoldNet [19] pursues an unsupervised strategy where an autoencoder is trained on point pair features extracted from the local patches. Similarly, Kehl et al. [30] introduce an autoencoder that is trained on patches of RGB-

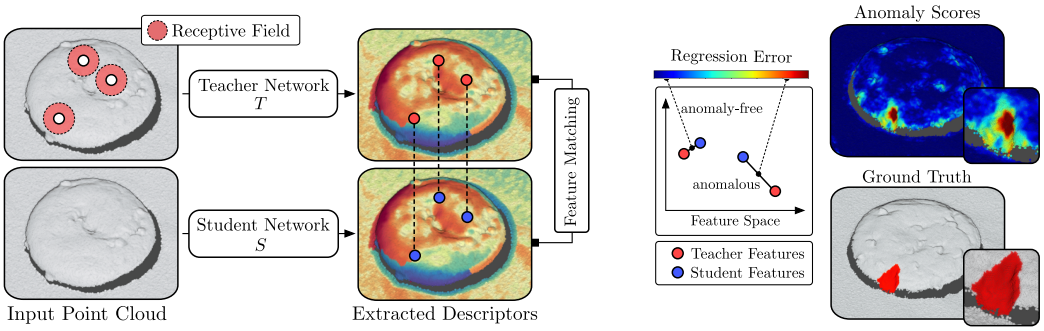

Figure 2: (a) Training of our proposed 3D-ST method on anomaly-free point clouds. A student network S is trained to match the local descriptors of a pretrained teacher network T. (b) Computation of anomaly scores during inference. Anomaly scores are derived from the regression error between the student and the teacher network. Increased regression errors correspond to anomalous 3D points.

D images to obtain local features. These methods have the disadvantage that a separate patch needs to be cropped and processed for each feature. This quickly becomes computationally intractable for a large number of points.

To mitigate this problem, recent 3D feature extractors attempt to densely compute features for high-resolution inputs. Choy et al. [16] propose FCGF, a fully convolutional approach to local geometric feature extraction for 3D registration. They design a network with sparse convolutions to efficiently processes high-resolution voxel data. Given a large number of precisely annotated local correspondences, their approach is trained using contrastive losses that encourage matching local geometries to be close in feature space. PointContrast [50] learns descriptors for 3D registration in a self-supervised fashion and does not rely on human annotations. Correspondences are automatically derived by augmenting a pair of overlapping views from a single 3D scan. While being computationally efficient, these methods require a prior voxelization that can lead to discretization inaccuracies. Furthermore, all of the discussed methods are designed to produce feature spaces that are ideally invariant to 3D rotations of the input data. In unsupervised anomaly detection, however, anomalies can manifest themselves precisely through locally rotated geometric structures. Such differences should therefore be reflected in the extracted feature vectors. This calls for the development of a different pretraining strategy that is sensitive to local rotations.

3. Student-Teacher Anomaly Detection in Point Clouds

In this section, we introduce 3D Student-Teacher (3D-ST), a versatile framework for the unsupervised detection and localization of geometric anomalies in high-resolution 3D point clouds. We build on the recent success of leveraging local descriptors from pretrained networks for anomaly detection and propose an adaptation of the 2D student-teacher method [8] to 3D data.

Given a training dataset of anomaly-free input point clouds, our goal is to create a model that localizes anomalous regions in test point clouds, i.e., assigns a real-valued anomaly score to each point. To achieve this, we design a dense feature extraction network T, called *teacher network*, that computes local geometric features for arbitrary point clouds. For anomaly detection, a *student network* S is trained on the anomaly-free point clouds against the descriptors obtained from T. During inference, increased regression errors between S and T indicate anomalous points. An overview of our approach is illustrated in Figure 2.

To pretrain the teacher, we present a self-supervised protocol. It works on any generic auxiliary 3D point cloud dataset and requires no human annotations.

3.1. Self-Supervised Learning of Dense Local Geometric Descriptors

We begin by describing how to construct a descriptive teacher network T. An overview of our pretraining protocol is displayed in Figure 3. Given an input point cloud $\boldsymbol{P} \subset \mathbb{R}^3$ containing n 3D points, its purpose is to produce a d-dimensional feature vector $\boldsymbol{f_p} \in \mathbb{R}^d$ for every $\boldsymbol{p} \in \boldsymbol{P}$. The vector $\boldsymbol{f_p}$ describes the local geometry around the point $\boldsymbol{p}$, i.e., the geometry within its receptive field.

Local Feature Aggregation. The network architecture of T has two key requirements. First, it should be able to efficiently process high-resolution point clouds by computing a feature vector for each input point without downsampling the input data. Second, it requires explicit control over the receptive field of the feature vectors. In particular, it has to be possible to efficiently compute all points within the receptive field of an output descriptor.

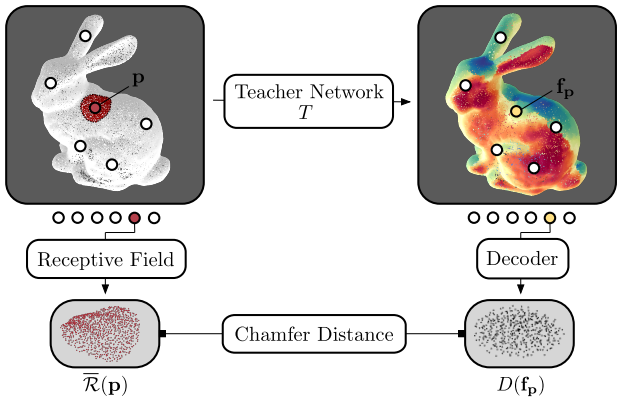

Figure 3: Overview of our proposed self-supervised pretraining strategy. A teacher network is trained to output local geometric descriptors for each 3D point in the input sample with a single forward pass. Simultaneously, a decoder network transforms randomly sampled descriptors of the teacher and attempts to fit the local receptive field around its respective input point.

To meet these requirements, we construct the k-nearest neighbor graph of the input point cloud and initialize $\boldsymbol{f_p} = \mathbf{0}$. We then pass the input sample through a series of residual blocks, where each block updates the feature vector of each 3D point $\boldsymbol{p}$ from $\boldsymbol{f_p} \in \mathbb{R}^d$ to $\tilde{\boldsymbol{f}}_{\boldsymbol{p}} \in \mathbb{R}^d$. These blocks are inspired by RandLA-Net [28, 29], an efficient and lightweight neural architecture for semantic segmentation of large-scale point clouds. In semantic segmentation tasks, the absolute position of a point is often related to its class, e.g., in autonomous driving datasets. Here, we want our model to produce features that describe the local geometry of an object independent of its absolute location. We therefore make the residual blocks translation-invariant by removing any dependency on absolute coordinates. This significantly increases the performance when used for anomaly detection as underlined by the results of our experiments in Section 4.

The architecture of our residual blocks is visualized in Figure 4(a). The input features are first passed through a shared MLP, followed by two local feature aggregation (LFA) blocks. The output features are added to the input after processing both by an additional shared MLP. The features are transformed by a series of residual blocks and a final shared MLP with a single hidden layer that maintains the dimension of the descriptors, i.e., $\tilde{\boldsymbol{f}}_{\boldsymbol{p}} \in \mathbb{R}^d$.

The purpose of the LFA block is to aggregate the geometric information from the local vicinity of each input point. To this end, it computes the nearest neighbors $\mathrm{knn}(\boldsymbol{p}) = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_k\}$ of all $\boldsymbol{p} \in \boldsymbol{P}$ and a set of local geometric features G for each point pair defined by

$$G(\boldsymbol{p}, \boldsymbol{p}_j) = (\boldsymbol{p} - \boldsymbol{p}_j) \odot \|(\boldsymbol{p} - \boldsymbol{p}_j)\|_2, \tag{1}$$

where $j \in \{1, \ldots, k\}$. The operator $\odot$ denotes the concatenation operation and $\|\cdot\|_2$ denotes the L^2-norm. Since G only depends on the difference vectors between neighboring points, our network is by design invariant to translations of the input data. Our experiments show that this invariance of our local feature extractor is crucial for anomaly detection performance. Therefore, we make this small but important change to the LFA block. A schematic description of such a block is given in Figure 4(b).

For each LFA block, the set of geometric features $G(\boldsymbol{p}, \boldsymbol{p}_j)$ is passed through a shared MLP producing feature vectors of dimension d_{LFA}. These are concatenated with the set of input features $\{\boldsymbol{f}_{\boldsymbol{p}_1}, \ldots, \boldsymbol{f}_{\boldsymbol{p}_k}\}$. The output feature vector of the LFA block $\tilde{\boldsymbol{f}}_{\boldsymbol{p}}$ is obtained by an average-pooling operation of the concatenated features, yielding a feature vector of dimension $2d_{\mathrm{LFA}}$.

Reconstructing Local Receptive Fields. To pretrain T in a self-supervised fashion, we employ a network D that decodes the local receptive field of a feature vector. The design of our network architecture allows an efficient computation of all points within the receptive field $\mathcal{R}(\boldsymbol{p})$ of a point $\boldsymbol{p}$, i.e., all points that affect the feature vector $\boldsymbol{f_p}$. Each LFA block depends on the features of the surrounding nearest neighbors $\mathrm{knn}(\boldsymbol{p})$. Whenever an LFA block is executed, $\mathcal{R}(\boldsymbol{p})$ grows by one hop in the nearest-neighbor graph. The receptive field can therefore be obtained by iteratively traversing the nearest neighbor graph:

$$\mathcal{R}(\boldsymbol{p}) = \bigcup_{l=0}^{L} \mathrm{knn}^l(\boldsymbol{p}),\ \mathrm{knn}^l(\boldsymbol{p}) = \bigcup_{\boldsymbol{q} \in \mathrm{knn}^{l-1}(\boldsymbol{p})} \mathrm{knn}(\boldsymbol{q}) \tag{2}$$

and $\mathrm{knn}^0 = \{\boldsymbol{p}\}$. L denotes the total number of LFA blocks in the network. Figure 4(c) visualizes this definition of the receptive field.

The decoder $D : \mathbb{R}^d \to \mathbb{R}^{3\times m}$ upsamples a feature vector to produce m 3D points by applying an MLP. For pretraining, we extract descriptors from an input point cloud by passing it through the local feature extractor. We then randomly sample a set of points $\boldsymbol{Q}$ from the input. For each $\boldsymbol{p} \in \boldsymbol{Q}$, we compute the receptive fields $\mathcal{R}(\boldsymbol{p})$ and pass their respective feature vectors through the decoder. To train D, we minimize the Chamfer distance [3] between the decoded points and the receptive fields. Since our network architecture is not aware of the absolute coordinates of $\boldsymbol{p}$, we additionally compute the mean $\bar{\boldsymbol{p}}$ of all $\boldsymbol{p} \in \mathcal{R}(\boldsymbol{p})$ and subtract it from each point, yielding the set $\overline{\mathcal{R}}(\boldsymbol{p}) = \mathcal{R}(\boldsymbol{p}) - \bar{\boldsymbol{p}}$. The loss function for our self-supervised training procedure can then be written as:

$$\mathcal{L}_C = \frac{1}{|\boldsymbol{Q}|} \sum_{\boldsymbol{p} \in \boldsymbol{Q}} \mathrm{Chamfer}(D(\boldsymbol{f_p}), \overline{\mathcal{R}}(\boldsymbol{p})). \tag{3}$$

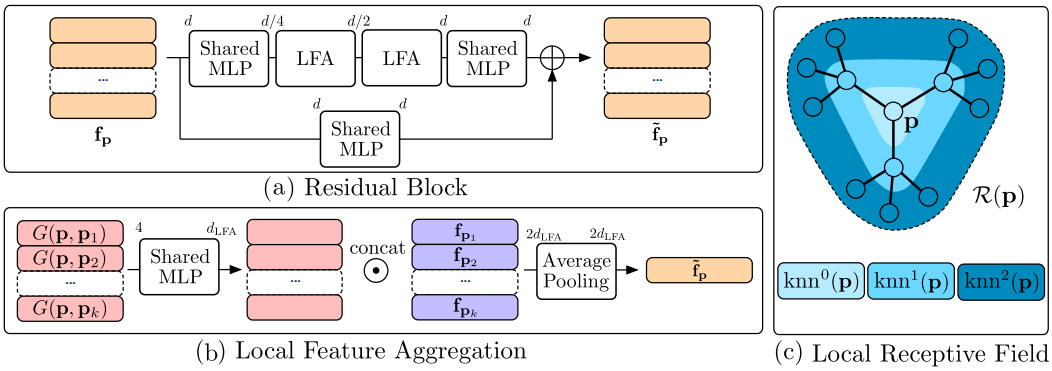

Figure 4: Overview of our network architecture. (a) Residual block that performs a series of local feature aggregation steps to update the feature vectors. (b) The local feature aggregation block aggregates geometric information of surrounding points. (c) Visualization of the receptive field of a point $\boldsymbol{p}$.

Data Normalization. In order for our teacher network to be applied to any point cloud not included in the pretraining dataset, some form of data normalization is required. Since our network operates on the distance vectors of neighboring points, we choose to normalize the input data with respect to these distances. More specifically, we compute the average distance between each point and its nearest neighbors over the entire training set, i.e.,

$$s = \frac{1}{k\,|\boldsymbol{P}|} \sum_{\boldsymbol{p}\in\boldsymbol{P}} \sum_{\boldsymbol{q}\in\text{knn}(\boldsymbol{p})} \|p - q\|_2. \tag{4}$$

We then scale the coordinates of each data sample in the pretraining dataset by $1/s$. This allows us to apply the teacher network to arbitrary point cloud datasets, as long as the same data normalization technique is used.

3.2. Matching Geometric Features for 3D Anomaly Detection

Finally, we describe how to employ the pretrained teacher network T to train a student network S for anomaly detection. Given a dataset of anomaly-free point clouds, we first calculate the scaling factor s for this dataset as defined in (4). The weights of T remain constant during the entire anomaly detection training. S exhibits the identical network architecture as T and is initialized with uniformly distributed random weights. Each training point cloud $\boldsymbol{P}_t \subset \mathbb{R}^3$ is passed through both networks, T and S, to compute dense features $\boldsymbol{f}_{\boldsymbol{p}}^T$ and $\boldsymbol{f}_{\boldsymbol{p}}^S$ for all $\boldsymbol{p} \in \boldsymbol{P}_t$, respectively. The weights of S are optimized to reproduce the geometric descriptors of T by computing the feature-wise L^2-distance:

$$\mathcal{L}_{ST} = \frac{1}{|\boldsymbol{P}_t|} \sum_{\boldsymbol{p}\in\boldsymbol{P}_t} \left\| \boldsymbol{f}_{\boldsymbol{p}}^S - (\boldsymbol{f}_{\boldsymbol{p}}^T - \boldsymbol{\mu})\,\text{diag}(\boldsymbol{\sigma})^{-1} \right\|_2^2. \tag{5}$$

We transform the teacher features to be centered around $\mathbf{0}$ with unit standard deviation. This requires the computation of the component-wise means $\boldsymbol{\mu} \in \mathbb{R}^d$ and standard deviations $\boldsymbol{\sigma} \in \mathbb{R}^d$ of all teacher features over the whole training set. We denote the inverse of the diagonal matrix filled with $\boldsymbol{\sigma}$ by $\text{diag}(\boldsymbol{\sigma})^{-1}$.

During inference, anomaly scores $A(\boldsymbol{p})$ are derived for each point $\boldsymbol{p} \in \boldsymbol{P}_i$ in a test point cloud $\boldsymbol{P}_i \subset \mathbb{R}^3$. They are given by the regression errors between the respective features of the student and the teacher network, i.e.,

$$A(\boldsymbol{p}) = \|\boldsymbol{f}_{\boldsymbol{p}}^S - (\boldsymbol{f}_{\boldsymbol{p}}^T - \boldsymbol{\mu})\,\text{diag}(\boldsymbol{\sigma})^{-1}\|_2. \tag{6}$$

The intuition behind this is that anomalous geometries produce features that the student network has not observed during training, and is hence unable to reproduce. Large regression errors indicate anomalous geometries.

4. Experiments

To demonstrate the effectiveness of our approach, we perform extensive experiments on the MVTec 3D Anomaly Detection (MVTec 3D-AD) dataset [9]. This dataset was designed to evaluate methods for the unsupervised detection of geometric anomalies in point cloud data (PCD). Currently, this is the only publicly available comprehensive dataset for this task. It contains over 4000 high-resolution 3D scans of 10 object categories of industrially manufactured products. The task is to train a model on anomaly-free samples and to localize anomalies that occur as defects on the manufactured products during inference.

4.1. Experiment Setup

We benchmark the performance of our 3D-ST method against existing methods for unsupervised 3D anomaly detection. In particular, we follow the initial benchmark on MVTec 3D-AD and compare 3D-ST against the Voxel f-AnoGAN, the Voxel Autoencoder, and the Voxel Variation

Model. The benchmark also includes their respective counterparts that process depth images instead of voxel grids by exchanging 3D with 2D convolutions. The GAN- and autoencoder-based methods derive anomaly scores by a per-pixel or per-voxel comparison of their reconstructions to the input samples. The Variation Model is a shallow machine learning model that computes the per-pixel or per-voxel means and standard deviations over the training set. During inference, anomaly scores are obtained by computing the per-pixel or per-voxel Mahalanobis distance from a test sample to the training distribution. We employ the same training and evaluation protocols and hyperparameters setting as listed in [9].

Teacher Pretraining. To pretrain the teacher network (cf. Section 3.1), we generate synthetic 3D scenes using objects of the ModelNet10 dataset [49]. It consists of over 5000 3D models divided into 10 different object categories.

We generate a scene of our pretraining dataset by randomly selecting 10 samples from ModelNet10 and scaling the longest side of their bounding box to 1. The objects are rotated around each 3D axis with angles sampled uniformly from the interval $[0, 2\pi]$. Each object is placed at a random location sampled uniformly from $[-3, 3]^3$. Point clouds are created by selecting n points from the scene using farthest point sampling [35]. The training and validation datasets consist of 1000 and 50 point clouds, respectively. Our experiments show that using such a synthetic dataset for pretraining yields local descriptors that are well suited for 3D anomaly detection. In our ablation studies, we additionally investigate the use of real-world datasets from different domains for pretraining, namely Semantic KITTI [5, 24], MVTec ITODD [21], and 3DMatch [51].

The teacher network T consists of 4 residual blocks and processes $n = 64000$ input points. We perform experiments using two different feature dimensions $d \in \{64, 128\}$. The shared MLPs in all network blocks are implemented with a single dense layer, followed by a LeakyReLU activation with a negative slope of 0.2. The input and output dimensions of each shared MLP are given in Figure 4. For local feature aggregation, a nearest neighbor graph with $k = 32$ neighbors is constructed. The pretraining runs for 250 epochs using the Adam optimizer with an initial learning rate of 10^{-3} and a weight decay of 10^{-6}. At each training step, a single input sample is fed through the teacher network. To generate reconstructions of local receptive fields, 16 randomly selected descriptors from the output of T are passed through the decoder network D, which is implemented as an MLP with input dimension d, two hidden layers of dimension 128, and an output layer that reconstructs $m = 1024$ points. Each hidden layer is followed by a LeakyReLU activation with negative slope of 0.05. After the training, we select the model with the lowest validation error as the teacher network.

Anomaly Detection. The student network S in our 3D-ST method has the same network architecture as the teacher. It is trained for 100 epochs on the anomaly-free training split of the MVTec 3D-AD dataset. We train with a batch size of 1. This is equivalent to processing a large number of local patches per iteration due to the limited receptive field of the employed networks. We use Adam with an initial learning rate of 10^{-3} and weight decay 10^{-5}. Each point cloud is reduced to $n = 64000$ input points using farthest point sampling. For inference, we select the student network with the lowest validation error.

The evaluation on MVTec 3D-AD requires to predict an anomaly score for each pixel in the original (x, y, z) images. To do this, we apply harmonic interpolation [23] to the pixels that were not assigned anomaly scores by our method. We follow the standard evaluation protocol of MVTec 3D-AD and compute the per-region overlap (PRO) [7] and the corresponding false positive rate for successively increasing anomaly thresholds. We then report the area under the PRO curve (AU-PRO) integrated up to a false positive rate of 30%. We normalize the resulting values to the interval $[0, 1]$.

4.2. Experiment Results

Table 1 shows quantitative results of each evaluated method on every object category of MVTec 3D-AD. The top three rows list the performance of the voxel-based methods. The following three rows list the performance of the respective methods on 2D depth images. The bottom two rows show the performance of our 3D-ST method on 3D point cloud data, evaluated for two different descriptor dimensions $d \in \{64, 128\}$. Our method performs significantly better than all other methods on every dataset category. Increasing the descriptor dimension from 64 to 128 yields a slight overall improvement of 1.5 percentage points. The latter outperforms the previously leading method by 25.0 points.

Figure 1 shows qualitative results of our method. 3D-ST manages to localize anomalies over a range of different object categories, such as the crack in the *bagel*, the contamination on the *rope* and the *tire*, or the cut in the *foam*. Additional qualitative results are shown in the supplement.

The MVTec 3D-AD paper states that real-world anomaly detection applications require particularly low false positive rates. We therefore report the mean performance of all evaluated methods when varying the integration limit of the PRO curve in Figure 5. Our method outperforms all other evaluated methods for any chosen integration limit. The relative difference in performance is particularly large for lower integration limits. This makes our approach well-suited for practical applications. Exact values for several

		bagel	cable gland	carrot	cookie	dowel	foam	peach	potato	rope	tire	mean
Voxel	GAN	0.440	0.453	0.825	0.755	0.782	0.378	0.392	0.639	0.775	0.389	0.583
	AE	0.260	0.341	0.581	0.351	0.502	0.234	0.351	0.658	0.015	0.185	0.348
	VM	0.453	0.343	0.521	0.697	0.680	0.284	0.349	0.634	0.616	0.346	0.492
Depth	GAN	0.111	0.072	0.212	0.174	0.160	0.128	0.003	0.042	0.446	0.075	0.143
	AE	0.147	0.069	0.293	0.217	0.207	0.181	0.164	0.066	0.545	0.142	0.203
	VM	0.280	0.374	0.243	0.526	0.485	0.314	0.199	0.388	0.543	0.385	0.374
PCD	3D-ST$_{64}$	0.939	0.440	0.984	0.904	0.876	**0.633**	0.937	**0.989**	0.967	0.507	0.818
	3D-ST$_{128}$	**0.950**	**0.483**	**0.986**	**0.921**	**0.905**	0.632	**0.945**	0.988	**0.976**	**0.542**	**0.833**

Table 1: Anomaly detection results for each evaluated method and dataset category. The area under the PRO curve is reported for an integration limit of 0.3. The best performing method is highlighted in boldface.

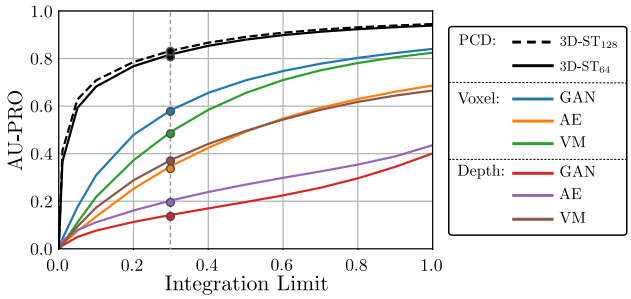

Figure 5: Performance of each method with respect to the integration limit. The AU-PRO at an integration limit of 0.3 is marked by a vertical line. In real-world scenarios, the performance at lower integration limits is of particular importance.

integration limits can be found in the supplement.

4.3. Ablation Studies

We additionally perform various ablation studies with respect to the key hyperparameters of our proposed method. Again, the exact values for each experiment can be found in the supplementary material. Figure 6 shows the dependency of the mean performance of our method on the number of input points n, the feature dimension d, or the number of nearest neighbor points k used for local feature aggregation. We additionally visualize the inference time and the memory consumption of each model during training and evaluation[1]. We find that our method is insensitive to the choice of each hyperparameter. In particular, the mean performance of each evaluated model outperforms the best performing competing model from the baseline experiments by a large margin. The mean performance of our model grows monotonically with respect to each considered hyperparameter. It eventually saturates, whereas the inference time and memory consumption continue to increase super-linearly.

[1]All models were implemented in PyTorch [37]. Inference times and memory consumption were measured on an NVIDIA Tesla V100 GPU.

Feature Space of the Teacher Network. We depict the effectiveness of our pretraining strategy in Figure 7. The left bar plot shows the mean performance with respect to changes in the training strategy of our method. The first bar indicates how the performance changes when we initialize the teacher's weights randomly and perform no pretraining. As expected, the performance drops significantly. The next bar shows the performance when concatenating the absolute point coordinates of each 3D point to the local feature aggregation function G, as proposed in [28]. This no longer makes our network translation invariant and decreases the performance. This indicates that translation invariance is indeed important for our network architecture and that our modification to the local feature aggregation module has a significant impact. The third bar shows the performance of our method when trying to additionally incorporate rotation invariance. We achieve this by augmentation of the training data with randomly sampled rotations such that locally rotated geometries are also considered as anomaly-free. The performance is still significantly below our method, which indicates that sensitivity to local rotations is beneficial for 3D anomaly detection.

Pretraining Dataset. In most of our experiments, we use synthetically generated scenes created from objects from the ModelNet10 dataset as described above. Our pretraining strategy does not require any human annotations and can operate on arbitrary input point clouds. We are thus interested in whether the performance varies when using different pretraining datasets. As first experiment, we randomly select 1000 training scenes from the Semantic KITTI autonomous driving dataset, which is captured with a LIDAR sensor. Secondly, we pretrain a teacher on all samples of 3DMatch, an indoor dataset originally designed for point cloud registration. Finally, we use all samples of the MVTec ITODD dataset, an industrial dataset originally designed for 3D pose estimation. The center bar chart in Figure 7 shows the mean performance of our method when using these three datasets for pretraining, compared to our baseline model. We find that our method does not strongly depend on the

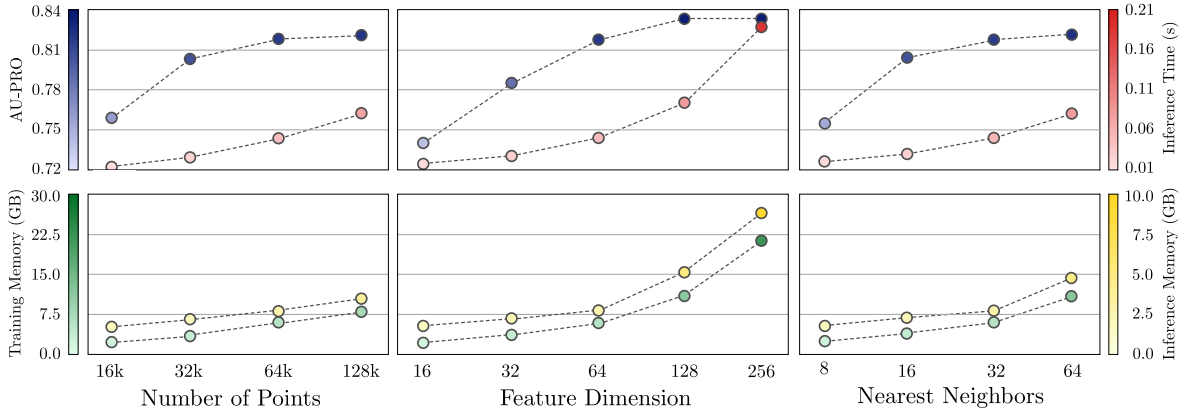

Figure 6: Performance of our method when changing various key hyperparameters, i.e., the number of input points, the feature dimension, and the number of nearest neighbors used for local feature aggregation.

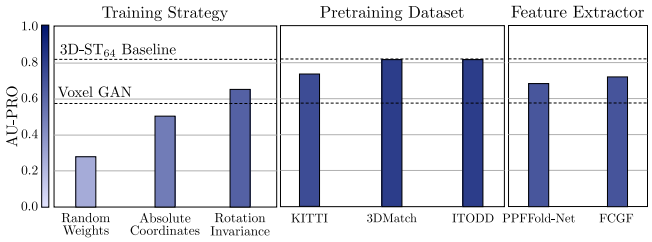

Figure 7: Sensitivity of our method to the pretraining strategy of the teacher network, different pretraining datasets, and different feature extractors.

specific dataset chosen for pretraining when using ITODD or 3DMatch. We observe a slight performance gap for the KITTI dataset, which is likely due to the large domain shift.

Feature Extractor. We additionally test the performance of our method when our pretrained teacher network is replaced by descriptors obtained from different feature extractors. We compare against features obtained from PPFFoldNet [19] and FCGF [16]. For both, we use publicly available pretrained models [1, 15]. PPF-FoldNet outputs a single 512-dimensional descriptor for patches cropped from a local neighborhood of 1024 points around each input point. Since it requires patch-based feature extraction, producing descriptors for a large number of input points becomes prohibitively slow. We therefore only extract 1000 descriptors for each point cloud with PPF-FoldNet. FCGF outputs 32-dimensionsal descriptors and was pretrained in a supervised fashion on the 3DMatch dataset by finding correspondences for 3D registration. Since it requires a prior voxelization of the input data, we select a voxel size of 3.5 mm and extract descriptors for 64000 points.

We train our student network to match the features extracted from these pretrained networks instead of our proposed teacher network. The feature dimension of the output layer of our student network is adapted to match the feature dimension of the descriptors. The results are shown in the right bar plot in Figure 7. Transferring the features of both networks yields better performance than the Voxel GAN, which is the previously best-performing method that was trained from scratch. This underlines the effectiveness of using pretrained geometric descriptors for 3D anomaly detection. Both extractors do not reach the performance of our proposed pretraining strategy that is specifically designed for the anomaly detection problem.

5. Conclusion

We propose 3D-ST, a new approach to the challenging problem of unsupervised anomaly detection in 3D point clouds. Our method is an adaption of student-teacher anomaly detection from 2D to 3D. While existing methods are trained from random weight initialization, our method leverages the descriptiveness of deep local geometric features extracted from a pretrained network. To address the lack of pretraining protocols for 3D anomaly detection, we introduce a self-supervised strategy based on the reconstruction of local receptive fields. This enables the training of teacher networks that produce dense local geometric descriptors for arbitrary 3D point clouds.

For anomaly detection, a student network matches the geometric descriptors of the teacher on anomaly-free data. During inference, anomalies are detected in regions where the student fails to reproduce the output of the teacher. Extensive experiments on the MVTec 3D Anomaly Detection dataset show that our method outperforms all existing methods by a large margin. Our ablation studies demonstrate that our method is computationally efficient and robust to the choice of hyperparameters and pretraining datasets.

References

[1] Xuyang Bai. PyTorch PPF-FoldNet. `https://github.com/XuyangBai/PPF-FoldNet`, 2020.

[2] Spyridon Bakas, Hamed Akbari, Aristeidis Sotiras, Michel Bilello, Martin Rozycki, Justin S. Kirby, et al. Advancing the Cancer Genome Atlas Glioma MRI Collections with Expert Segmentation Labels and Radiomic Features. *Scientific Data*, 4(1), 2017.

[3] Harry G. Barrow, Jay M. Tenenbaum, Robert C. Bolles, and Helen C. Wolf. Parametric Correspondence and Chamfer Matching: Two New Techniques for Image Matching. In *IJCAI*, pages 659–663, 1977.

[4] Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, and Nassir Navab. Deep Autoencoding Models for Unsupervised Anomaly Segmentation in Brain MR Images. In Alessandro Crimi, Spyridon Bakas, Hugo Kuijf, Farahani Keyvan, Mauricio Reyes, and Theo van Walsum, editors, *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pages 161–169, Cham, 2019. Springer International Publishing.

[5] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jürgen Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9296–9306, 2019.

[6] Marcel Bengs, Finn Behrendt, Julia Krüger, Roland Opfer, and Alexander Schlaefer. Three-dimensional deep learning with spatial erasing for unsupervised anomaly segmentation in brain MRI. *International Journal of Computer Assisted Radiology and Surgery*, 16, 2021.

[7] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. The MVTec Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. *International Journal of Computer Vision*, 129(4):1038–1059, 2021.

[8] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed Students: Student-Teacher Anomaly Detection With Discriminative Latent Embeddings. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4182–4191, 2020.

[9] 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 - Volume 5: VISAPP*, pages 202–213. INSTICC, SciTePress, 2022.

[10] Paul Bergmann, Sindy Löwe, Michael Fauser, David Sattlegger, and Carsten Steger. Improving Unsupervised Defect Segmentation by Applying Structural Similarity to Autoencoders. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP*, pages 372–380. INSTICC, SciTePress, 2019.

[11] Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar Cadena. Fishyscapes: A Benchmark for Safe Semantic Segmentation in Autonomous Driving. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 2403–2412, 2019.

[12] Philippe Burlina, Neil Joshi, and I-Jeng Wang. Where's wally now? deep generative and discriminative embeddings for novelty detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11499–11508, 2019.

[13] Fabio Carrara, Giuseppe Amato, Luca Brombin, Fabrizio Falchi, and Claudio Gennaro. Combining GANs and AutoEncoders for efficient anomaly detection. In *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021.

[14] Diego Carrera, Fabio Manganini, Giacomo Boracchi, and Ettore Lanzarone. Defect Detection in SEM Images of Nanofibrous Materials. *IEEE Transactions on Industrial Informatics*, 13(2):551–561, 2017.

[15] Chris Choy. FCGF. `https://github.com/chrischoy/FCGF`, 2021.

[16] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully Convolutional Geometric Features. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8957–8965, 2019.

[17] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences. arXiv preprint arXiv:2005.02357, 2020.

[18] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. PaDiM: A Patch Distribution Modeling Framework for Anomaly Detection and Localization. In Alberto Del Bimbo, Rita Cucchiara, Stan Sclaroff, Giovanni Maria Farinella, Tao Mei, Marco Bertini, Hugo Jair Escalante, and Roberto Vezzani, editors, *Pattern Recognition. ICPR International Workshops and Challenges*, pages 475–489. Springer International Publishing, 2021.

[19] Haowen Deng, Tolga Birdal, and Slobodan Ilic. PPF-FoldNet: Unsupervised Learning of Rotation Invariant 3D Local Descriptors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[20] Haowen Deng, Tolga Birdal, and Slobodan Ilic. PPFNet: Global Context Aware Local Features for Robust 3D Point Matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[21] Bertram Drost, Markus Ulrich, Paul Bergmann, Philipp Härtinger, and Carsten Steger. Introducing MVTec ITODD — A Dataset for 3D Object Recognition in Industry. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 2200–2208, 2017.

[22] Thibaud Ehret, Axel Davy, Jean-Michel Morel, and Mauricio Delbracio. Image Anomalies: A Review and Synthesis of Detection Methods. *Journal of Mathematical Imaging and Vision*, 61(5):710–743, 2019.

[23] Lawrence C. Evans. *Partial differential equations*. American Mathematical Society, Providence, R.I., 2010.

[24] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[25] Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. CFLOW-AD: Real-Time Unsupervised Anomaly Detection With Localization via Conditional Normalizing Flows. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 98–107, 2022.

[26] Dan Hendrycks, Steven Basart, Mantas Mazeika, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. A Benchmark for Anomaly Segmentation. *arXiv preprint arXiv:1911.11132*, 2019.

[27] Eungi Hong and Yoonsik Choe. Latent Feature Decentralization Loss for One-Class Anomaly Detection. *IEEE Access*, 8:165658–165669, 2020.

[28] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11105–11114, 2020.

[29] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Learning Semantic Segmentation of Large-Scale Point Clouds with Random Sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[30] Wadim Kehl, Fausto Milletari, Federico Tombari, Slobodan Ilic, and Nassir Navab. Deep Learning of Local RGB-D Patches for 3D Object Detection and 6D Pose Estimation. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 205–220. Springer International Publishing, 2016.

[31] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, pages 1097–1105, 2012.

[32] Wenqian Liu, Runze Li, Meng Zheng, Srikrishna Karanam, Ziyan Wu, Bir Bhanu, Richard J. Radke, and Octavia Camps. Towards Visually Explaining Variational Autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8639–8648, 2020.

[33] Bjoern H. Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, et al. The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). *IEEE Transactions on Medical Imaging*, 34(10):1993–2024, 2015.

[34] Pankaj Mishra, Claudio Piciarelli, and Gian Luca Foresti. A Neural Network for Image Anomaly Detection with Deep Pyramidal Representations and Dynamic Routing. *International Journal of Neural Systems*, 30(10):2050060, 2020.

[35] Carsten Moenning and Neil A. Dodgson. Fast Marching farthest point sampling. In *Eurographics 2003 - Posters*. Eurographics Association, 2003.

[36] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. Deep Learning for Anomaly Detection: A Review. *ACM Comput. Surv.*, 54(2), 2021.

[37] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[38] Kevin M. Potter, Brendan Donohoe, Benjamin Greene, Abigail Pribisova, and Emily Donahue. Automatic detection of defects in high reliability as-built parts using x-ray CT. In *Applications of Machine Learning 2020*, volume 11511, pages 120 – 136. International Society for Optics and Photonics, SPIE, 2020.

[39] Tal Reiss, Niv Cohen, Liron Bergman, and Yedid Hoshen. PANDA: Adapting Pretrained Features for Anomaly Detection and Segmentation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2805–2813, 2021.

[40] Oliver Rippel, Arnav Chavan, Chucai Lei, and Dorit Merhof. Transfer Learning Gaussian Anomaly Detection by Fine-Tuning Representations. arXiv preprint arXiv:2108.04116, 2021.

[41] Mohammadreza Salehi, Niousha Sadjadi, Soroosh Baselizadeh, Mohammad H. Rohban, and Hamid R. Rabiee. Multiresolution Knowledge Distillation for Anomaly Detection. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14897–14907, 2021.

[42] Samuele Salti, Federico Tombari, and Luigi Di Stefano. SHOT: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251–264, 2014.

[43] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Georg Langs, and Ursula Schmidt-Erfurth. f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks. *Medical Image Analysis*, 54:30–44, 2019.

[44] Jaime Simarro Viana, Ezequiel de la Rosa, Thijs Vande Vyvere, David Robben, Diana M. Sima, and 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.

[45] Kechen Song and Yunhui Yan. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Applied Surface Science*, 285:858–864, 2013.

[46] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique Signatures of Histograms for Local Surface Description. In *Proceedings of the 11th European Conference on Computer Vision: Part III*, page 356–369, Berlin, Heidelberg, 2010. Springer-Verlag.

[47] Shashanka Venkataramanan, Kuan-Chuan Peng, Rajat Vikram Singh, and Abhijit Mahalanobis. Attention Guided Anomaly Localization in Images. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 485–503. Springer International Publishing, 2020.

[48] Lu Wang, Dongkai Zhang, Jiahao Guo, and Yuexing Han. Image Anomaly Detection Using Normal Data Only by Latent Space Resampling. *Applied Sciences*, 10(23), 2020.

[49] 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.

[50] Saining Xie, Jiatao Gu, Demi Guo, Charles R. Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pretraining for 3d point cloud understanding. In *Computer Vision – ECCV 2020*, pages 574–591, Cham, 2020. Springer International Publishing.

[51] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 199–208, 2017.